UNIVERSITÀ DEGLI STUDI DI PADOVA
Dipartimento di Scienze Statistiche

Corso di Laurea Magistrale in Scienze Statistiche

Relazione finale

Conformal Prediction:

L'incontro tra Statistica e Machine Learning con applicazioni in Economia e Business Intelligence

Relatrice:
Prof.ssa Bisaglia Luisa

Candidata:
Vittoria Morosini

Matricola:
2107144

ANNO ACCADEMICO 2025/2026

A mia mamma, il mio porto sicuro ed eterno.
A mio papà, il mio eroe silenzioso senza mantello.
A mio fratello, la mia guida e fonte d'ispirazione.
A me stessa, per averci creduto fino alla fine.

Indice

Introduzione

Negli ultimi anni, l'incremento esponenziale della disponibilità di dati e la crescente complessità dei modelli predittivi hanno profondamente trasformato l'approccio all'inferenza statistica e al processo decisionale basato sui dati. In tale contesto, la necessità di strumenti capaci non solo di fornire previsioni accurate, ma anche di quantificare in modo affidabile l'incertezza associata a tali previsioni, è divenuta un elemento centrale della moderna statistica applicata e della *data science*.

La *Conformal Prediction* (CP) si inserisce in questo quadro come una metodologia generale e rigorosamente fondata, in grado di garantire la costruzione di insiemi predittivi con copertura frequentista controllata, indipendentemente dalla distribuzione dei dati e dal modello predittivo utilizzato. A differenza delle tecniche classiche di inferenza, che si fondano su ipotesi parametriche sulla forma della distribuzione dei dati (quali normalità, linearità o omoschedasticità) e garantiscono la copertura in genere solo in senso asintotico, la *Conformal Prediction* fornisce, per campioni finiti, una garanzia di copertura marginale almeno pari al livello nominale $1 - \alpha$, senza assumere una specifica distribuzione parametrica per i dati, ma richiedendo unicamente l'ipotesi di scambiabilità delle osservazioni. Tale caratteristica la rende particolarmente adatta a contesti ad alta complessità e all'integrazione con modelli di apprendimento automatico, nei quali le ipotesi distributive classiche risultano spesso non verificabili o strutturalmente inadeguate.

L'obiettivo di questa tesi è duplice. Da un lato, si propone di approfondire in modo sistematico i fondamenti teorici della *Conformal Prediction*, analizzandone le proprietà di validità, efficienza e adattabilità, e discutendo le principali estensioni metodologiche sviluppate nella letteratura recente. Dall'altro, essa intende verificare empiricamente l'efficacia della CP in confronto ai metodi predittivi tradizionali, mostrando come tale metodologia possa fornire intervalli di previsione più calibrati, robusti e interpretabili, in particolare in ambiti applicativi di interesse economico e

di *Business Intelligence*.

Il lavoro si articola in due parti principali, una teorica e una applicativa.

La prima parte, di carattere metodologico, introduce nel **Capitolo 1** i concetti fondamentali di inferenza statistica predittiva, con particolare attenzione alla distinzione tra stime puntuali e intervalli di previsione, nonché alle limitazioni dei metodi classici basati su assunzioni parametriche. Il **Capitolo 2** è interamente dedicato alla *Conformal Prediction*: dopo averne presentato il principio generale e le varianti più note *(Cross-conformal, Inductive, Mondrian, Transductive)*, si approfondiscono le estensioni alle serie temporali e, nella sezione finale, l'integrazione con i modelli di *Machine Learning*. In quest'ultima parte viene discusso come la CP possa essere applicata a modelli predittivi complessi, quali regressioni non lineari, support vector machine, foreste casuali e reti neurali , per ottenere intervalli e insiemi di classificazione con garanzie di copertura finite, con un'attenzione particolare al compromesso tra validità ed efficienza.

L'analisi comparativa mira a valutare in termini quantitativi la capacità della CP di fornire copertura empirica prossima al livello nominale, mantenendo al contempo intervalli di ampiezza ridotta e una migliore calibrazione delle previsioni. Particolare attenzione è dedicata alle applicazioni in ambito economico e di *Business Intelligence*, dove la quantificazione dell'incertezza predittiva è essenziale per decisioni operative e strategiche. La seconda parte della tesi, di taglio empirico, ha l'obiettivo di dimostrare sperimentalmente la superiorità della *Conformal Prediction* rispetto ai metodi classici di costruzione di intervalli di previsione. Nel **Capitolo 3** vengono descritte le applicazioni di questo metodo in ambito economico, mentre nel **Capitolo 4** vengono presentati cenni storici, il ruolo dell'analisi predittiva e l'integrazione della CP nella *Business Intelligence*.

Infine, il **Capitolo 5** raccoglie le interpretazioni dei risultati ottenuti dalle analisi svolte sul *dataset M5 Competition*, arricchito dalle riflessioni conclusive e le prospettive di ricerca futura, evidenziando come la *Conformal Prediction* rappresenti non soltanto un potente strumento statistico, ma anche un paradigma concettuale capace di coniugare la solidità teorica dell'inferenza frequentista con la flessibilità e la potenza predittiva del *Machine Learning*. L'obiettivo finale è quello di mostrare che la CP costituisce una metodologia più robusta, interpretabile e affidabile rispetto ai metodi tradizionali, aprendo la strada a un nuovo equilibrio tra accuratezza e fiducia

nelle previsioni.

In sintesi, la tesi intende dimostrare che la *Conformal Prediction* non è soltanto un raffinamento metodologico, ma una vera e propria estensione del paradigma inferenziale classico, capace di fornire una risposta concreta e rigorosa all'esigenza contemporanea di previsioni statisticamente attendibili in un mondo dominato dai dati e dall'incertezza.

Capitolo 1

Origini e Sviluppo

Il presente capitolo si propone di fornire una panoramica esaustiva sull'evoluzione teorica e applicata degli intervalli di previsione, ponendo le basi per comprendere la necessità di metodi più moderni e flessibili. La quantificazione dell'incertezza predittiva è diventata cruciale in molteplici contesti applicativi, dalla medicina personalizzata alla previsione finanziaria, dal controllo industriale all'apprendimento automatico. La trattazione inizierà dai fondamenti classici dell'inferenza frequentista, per poi evidenziare le limitazioni strutturali che motivano l'introduzione di approcci alternativi. Tra questi, la *Conformal Prediction* rappresenta uno degli sviluppi più promettenti per la costruzione di intervalli predittivi con garanzie finite e non parametriche, tema che sarà approfondito nel Capitolo 2.

1.1 Contesto generale

Nel campo dell'inferenza statistica, e in particolare nell'inferenza predittiva, un intervallo di previsione rappresenta una stima dell'intervallo entro cui è attesa una futura osservazione, con una specifica probabilità, sulla base delle informazioni già disponibili. Questi intervalli trovano ampio impiego nell'analisi di regressione e sono applicabili sia nel contesto frequentista sia in quello bayesiano. Analogamente agli intervalli di confidenza (in ambito frequentista) e agli intervalli di credibilità (in ambito bayesiano), che quantificano l'incertezza associata alla stima di un parametro ignoto della popolazione, gli intervalli di previsione descrivono l'incertezza riguardante una futura osservazione casuale. In particolare, mentre i primi si concentrano sulla distribuzione delle stime di quantità non osservabili (ad esempio, la media della popolazione), gli intervalli di previsione si focalizzano sulla variabilità attesa nei

valori futuri delle singole osservazioni.

Cenni Storici. L'idea di stimare un intervallo per una futura osservazione si sviluppa nel primo Novecento nell'ambito della formalizzazione dell'inferenza statistica, in particolare nei lavori fondativi di Pearson [28] sulla teoria della variabilità e, successivamente, nella sistematizzazione dell'inferenza parametrica proposta da Fisher [14]. Tuttavia, è solo con lo sviluppo della teoria della probabilità classica e l'affermazione del paradigma frequentista che si è formalizzato il concetto di intervallo predittivo come oggetto distinto dagli intervalli di confidenza. In ambito applicato, la costruzione di intervalli di previsione riveste un ruolo fondamentale in numerosi contesti decisionali, quali la previsione economica, il controllo di qualità industriale, e la medicina personalizzata, dove la stima dell'intervallo entro cui può cadere un valore futuro consente una gestione più consapevole del rischio. Un aspetto cruciale nella valutazione degli intervalli predittivi riguarda la distinzione tra *copertura marginale* e *copertura condizionata*. La copertura marginale assicura che, in media su tutte le possibili realizzazioni dei dati, la probabilità che l'intervallo contenga la futura osservazione sia almeno $1-\alpha$, dove $\alpha \in (0,1)$ rappresenta il livello di significatività, ossia la probabilità massima che la futura osservazione cada al di fuori dell'intervallo predittivo. La copertura condizionata, più stringente, richiede che tale proprietà valga per ogni specifico insieme di dati osservati. Mentre la copertura condizionata è teoricamente più desiderabile, è spesso difficile da garantire in pratica. Per questo motivo, molti metodi, inclusa la *Conformal Prediction*, si focalizzano sulla copertura marginale, offrendo comunque importanti garanzie di affidabilità.

Definizione 1. *Sia $Y_1, \dots, Y_n \sim F_\theta$ un campione aleatorio di variabili indipendenti e identicamente distribuite, con $\theta \in \Theta \subseteq \mathbb{R}^p$ vettori di parametri ignoti e sia $Y_{n+1} \sim F_\theta$ una futura osservazione dalla stessa distribuzione.*

*Un **intervallo di previsione** di livello $1-\alpha$ è una coppia di statistiche $[L(Y_1, \dots, Y_n), U(Y_1, \dots, Y_n)]$ tali che:*

$$\mathbb{P}_\theta \left(Y_{n+1} \in [L(Y_1, \dots, Y_n), U(Y_1, \dots, Y_n)] \right) = 1 - \alpha.$$

La probabilità è calcolata rispetto alla distribuzione congiunta di $Y_1, \dots, Y_n, Y_{n+1}$.

Assunzioni. Per la costruzione classica di tali intervalli si assume che le osservazioni siano indipendenti e identicamente distribuite (i.i.d.), e che la distribuzione

sottostante sia nota o stimabile in modo consistente.

Teorema 1 (Teorema di Predizione Classico (Frequentista)). *Siano* $Y_1, \ldots, Y_n, Y_{n+1} \sim$ *i.i.d.* F_θ*, con* $\theta \in \Theta \subseteq \mathbb{R}^p$*. Sia* $\hat{\theta}_n = \hat{\theta}(Y_1, \ldots, Y_n)$ *uno stimatore consistente di* θ*.*

Sotto opportune ipotesi di regolarità, esiste una funzione $C_n(\cdot)$ *tale che:*

$$\lim_{n\to\infty} \mathbb{P}_\theta \left(Y_{n+1} \in C_n(\hat{\theta}_n) \right) = 1 - \alpha,$$

dove $C_n(\hat{\theta}_n)$ *rappresenta un intervallo di previsione asintotico per* Y_{n+1}*, costruito sulla base dello stimatore* $\hat{\theta}_n$ *della distribuzione predittiva condizionale.*

Teorema 2 (Teorema di Predizione Classico (Modello Normale con Parametri Incogniti)). *Supponiamo che* $X_1, \ldots, X_n \sim \mathcal{N}(\mu, \sigma^2)$*, con* $\mu \in \mathbb{R}$*,* $\sigma^2 > 0$ *incogniti.*

Allora, un intervallo di previsione al livello $1 - \alpha$ *per una futura osservazione* X_{n+1} *è dato da:*

$$\left[\bar{X} \pm t_{n-1,\alpha/2} \cdot s \cdot \sqrt{1 + \frac{1}{n}} \right],$$

dove:

- $\bar{X} = \frac{1}{n} \sum_{i=1}^{n} X_i$ *è la media campionaria,*
- $s = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (X_i - \bar{X})^2}$ *è la deviazione standard campionaria,*
- $t_{n-1,\alpha/2}$ *è il quantile della distribuzione t di Student con* $n - 1$ *gradi di libertà.*

Questo intervallo soddisfa:

$$\mathbb{P} \left(X_{n+1} \in \bar{X} \pm t_{n-1,\alpha/2} \cdot s \cdot \sqrt{1 + \frac{1}{n}} \right) = 1 - \alpha.$$

1.2 L'Approccio Frequentista: La Costruzione di Neyman

La formalizzazione del concetto di intervallo di confidenza è attribuibile a *Jerzy Neyman*, il quale, nel 1937, introdusse il noto metodo di costruzione di Neyman, dettagliato nella sua opera *"Outline of a Theory of Statistical Estimation Based on the Classical Theory of Probability"* [25]. Questo approccio prevede la definizione

di due funzioni, $L(X)$ e $U(X)$, che delimitano l'intervallo $[L(X), U(X)]$ in modo tale che, per un livello di confidenza predefinito C, la probabilità che l'intervallo contenga il vero valore del parametro θ risulti pari a C. Il metodo assicura che, nel lungo periodo, la proporzione di intervalli costruiti che effettivamente includono il parametro vero converga al livello nominale di copertura.
Un esempio classico di tale costruzione è rappresentato dall'intervallo di confidenza per la media μ di una popolazione normale con varianza nota σ^2, basato sulla distribuzione normale standard. Nel caso in cui la varianza sia ignota, si fa invece riferimento alla distribuzione t di Student.

Definizione 2. *Sia θ un parametro ignoto appartenente a uno spazio dei parametri Θ, e sia $X = (X_1, \ldots, X_n)$ un campione casuale generato secondo una distribuzione P_θ, appartenente a una famiglia parametrica $\{P_\theta : \theta \in \Theta\}$.*

Un intervallo di confidenza al livello $1-\alpha$ per θ è una coppia di statistiche (cioè funzioni misurabili dei dati osservati) $(L(X), U(X))$ tali che:

$$P_\theta \left(L(X) \leq \theta \leq U(X)\right) \geq 1 - \alpha, \quad \textit{per ogni } \theta \in \Theta.$$

La probabilità è calcolata rispetto alla distribuzione del campione X, mentre θ è considerato un valore fisso (sebbene ignoto). L'intervallo costruito varia da campione a campione, ma contiene il vero valore di θ con una frequenza almeno pari a $1 - \alpha$.

Interpretazione Frequentista. La probabilità associata all'intervallo non riguarda il parametro (che è fisso), bensì la procedura di costruzione. In ripetuti campionamenti, il $100(1-\alpha)\%$ degli intervalli costruiti con questa procedura conterrà il valore vero di θ.

Un intervallo di confidenza costruito secondo il metodo di Neyman deve soddisfare la proprietà che, in media, contiene il vero valore del parametro θ nel $100(1-\alpha)\%$ dei casi, ove α è il livello di significatività.

Assioma 1 (Assioma di Neyman:)**.** Dato uno spazio dei parametri Θ, un parametro ignoto $\theta \in \Theta$, e un campione casuale $X \sim P_\theta$, un procedimento di costruzione di intervalli è accettabile se, per ogni $\theta \in \Theta$, la probabilità che l'intervallo costruito

contenga il vero valore di θ è almeno pari a un livello prefissato $1-\alpha$, ovvero:

$$P_\theta\left(\theta \in C(X)\right) \geq 1-\alpha \quad \text{per ogni } \theta \in \Theta.$$

Dove $C(X)$ è un intervallo (casuale) costruito sulla base del campione osservato X.

Teorema 3 (Teorema di Neyman)**.** *Sia dato un procedimento di costruzione di intervalli $\theta \in C(X)$ tale che:*

$$P_\theta\left(\theta \in C(X)\right) = 1-\alpha, \quad \textit{per ogni } \theta \in \Theta.$$

Allora il procedimento garantisce una frequenza di copertura esattamente pari a $1-\alpha$ nel lungo periodo, ovvero:

$$\lim_{N\to\infty} \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}_{\{\theta \in C(X^{(i)})\}} = 1-\alpha \quad \textit{quasi sicuramente,}$$

dove $\{X^{(i)}\}_{i=1}^{N}$ sono N campioni indipendenti generati secondo P_θ.

Sebbene il metodo di Neyman rappresenti una pietra miliare nella teoria dell'inferenza, la sua applicazione pratica è spesso limitata dalla necessità di conoscere la distribuzione esatta dei dati e dall'assunzione di campioni i.i.d. Tali restrizioni ne riducono l'efficacia in contesti moderni caratterizzati da eterogeneità, dipendenza tra le osservazioni e modelli non parametrici.

1.3 Distinzione Concettuale: Intervalli di Confidenza vs Intervalli di Previsione

È essenziale differenziare tra intervalli di confidenza e intervalli di previsione. Gli intervalli di confidenza stimano un parametro della popolazione, come ad esempio la media μ, e riflettono l'incertezza derivante dal processo di stima di tale parametro. Gli intervalli di previsione, al contrario, identificano un intervallo all'interno del quale è probabile che ricada una futura osservazione, tenendo in considerazione sia l'incertezza sulla stima del parametro sia la variabilità intrinseca delle osservazioni.

Inoltre, è utile distinguere tra due fonti di incertezza: *aleatoria* ed *epistemica*. L'incertezza aleatoria è intrinseca al processo stocastico e non eliminabile, mentre l'incertezza epistemica deriva dalla nostra ignoranza o mancanza di conoscenza sul

modello o sui parametri. Gli intervalli di confidenza mirano a quantificare l'incertezza epistemica, mentre quelli di previsione integrano entrambe le componenti. Questo aspetto diventa particolarmente rilevante nei modelli complessi, dove l'epistemic uncertainty può essere dominante.
Ad esempio, considerando una variabile casuale $X \sim N(\mu, \sigma^2)$, un intervallo di confidenza per μ si limita a quantificare l'incertezza della stima della media, mentre un intervallo di previsione per una nuova osservazione X_{n+1} integra anche la variabilità delle osservazioni individuali. Questa distinzione riveste una rilevanza concettuale significativa: i primi riguardano l'incertezza su parametri fissi, mentre i secondi concernono l'incertezza su realizzazioni future di variabili aleatorie.

Un intervallo di confidenza per un parametro θ è definito come un intervallo che contiene θ con una certa probabilità. Un intervallo di previsione, invece, è un intervallo che contiene una futura osservazione X_{n+1} con una certa probabilità.

Teorema 4 (Teorema di Distinzione: Dualità tra test d'ipotesi e intervalli di confidenza)**.** *Sia X un campione casuale con distribuzione dipendente da un parametro $\theta \in \Theta$. Sia $\{A(\theta_0) \subseteq \mathcal{X} : \theta_0 \in \Theta\}$ una famiglia di regioni di accettazione di test di ipotesi $H_0 : \theta = \theta_0$ di livello α, cioè*

$$\sup_{\theta_0 \in \Theta} P_{\theta_0}(X \notin A(\theta_0)) \leq \alpha.$$

Definiamo per ogni osservazione $X = x$ l'insieme

$$C(x) = \{\theta_0 \in \Theta : x \in A(\theta_0)\}.$$

Allora:

1. *$C(X)$ è un insieme di confidenza di livello $1 - \alpha$, cioè*

$$\inf_{\theta \in \Theta} P_\theta(\theta \in C(X)) \geq 1 - \alpha.$$

2. *Viceversa, se $C(X)$ è un insieme di confidenza di livello $1 - \alpha$, allora per ogni $\theta_0 \in \Theta$ la regione*

$$A(\theta_0) = \{x \in \mathcal{X} : \theta_0 \in C(x)\}$$

è la regione di accettazione di un test di livello α per $H_0 : \theta = \theta_0$.

In generale, non esiste una corrispondenza biunivoca tra un test arbitrario e un insieme di confidenza, salvo che l'intervallo sia costruito specificamente come l'insieme dei valori non respinti dal test [25].

Nel contesto della regressione lineare classica

$$Y = \beta_0 + \beta_1 x + \varepsilon, \qquad \varepsilon \sim (0, \sigma^2),$$

la distinzione precedentemente introdotta assume una forma analiticamente esplicita.

Per un punto fissato x_0, l'intervallo di confidenza riguarda la quantità $\mathbb{E}[Y \mid X = x_0]$ ed è costruito sulla base della distribuzione dello stimatore $\hat{Y}(x_0)$. La sua ampiezza dipende esclusivamente dalla variabilità degli stimatori dei parametri.

L'intervallo di previsione per una nuova osservazione $Y_{\text{nuovo}} \mid X = x_0$, invece, tiene conto anche della variabilità intrinseca del termine di errore. Infatti,

$$\text{Var}(Y_{\text{nuovo}}(x_0)) = \text{Var}(\hat{Y}(x_0)) + \sigma^2,$$

mentre l'intervallo di confidenza dipende soltanto da $\text{Var}(\hat{Y}(x_0))$. La presenza del termine additivo σ^2 implica necessariamente una maggiore ampiezza dell'intervallo di previsione.

Tale distinzione discende direttamente dalla costruzione frequentista degli intervalli di confidenza proposta da Neyman [25] ed è trattata in modo sistematico nella letteratura moderna sulla regressione lineare ([46],[16]).

La Figura 1.1 traduce graficamente la differenza analitica discussa sopra. La retta verde rappresenta la stima della media condizionata $\hat{Y}(x)$. Attorno ad essa si distinguono due bande concentriche.

La banda blu, più interna, individua l'intervallo di confidenza per $\mathbb{E}[Y \mid X = x]$: essa quantifica esclusivamente l'incertezza dovuta alla stima dei parametri del modello e descrive la variabilità della retta stimata al variare del campione osservato.

La banda rossa, più ampia, corrisponde invece all'intervallo di previsione per una futura osservazione $Y_{\text{nuovo}} \mid X = x$. Oltre all'incertezza inferenziale associata alla stima della media, essa incorpora la variabilità intrinseca delle singole osservazioni, determinata dal termine di errore.

Si osserva inoltre che entrambe le bande si restringono in prossimità della regione centrale dei dati e si ampliano agli estremi del dominio: tale comportamento riflette

la struttura della varianza nel modello lineare, mentre la presenza della componente aleatoria additiva rende l'intervallo di previsione sistematicamente più ampio in ogni punto del dominio.

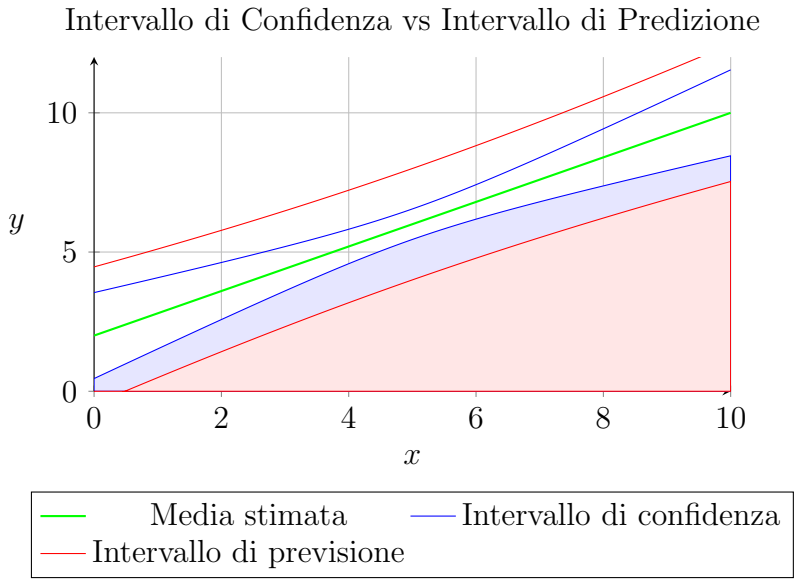


Figura 1.1: Rappresentazione grafica della differenza tra intervallo di confidenza e intervallo di previsione in un modello lineare. Le bande si restringono in prossimità di $\bar{x} = 5$ e si ampliano agli estremi del dominio.

1.4 Limiti dei Metodi Classici

I metodi frequentisti tradizionali per la costruzione di intervalli di previsione presentano alcune criticità e limitazioni:

- Dipendenza da ipotesi parametriche rigorose (ad esempio, normalità dei dati).
- Limitata flessibilità nell'applicazione a modelli non lineari o strutturalmente complessi.
- Applicabilità limitata in ambito di apprendimento automatico.

Dal punto di vista computazionale, i metodi classici presuppongono una modellizzazione esplicita della distribuzione dei dati, limitando la loro scalabilità in ambienti con elevate dimensioni o con struttura complessa. Inoltre, essi forniscono garanzie di copertura solo in senso asintotico, cioè quando la dimensione del campione tende all'infinito. In pratica, ciò significa che per campioni finiti la copertura

reale può discostarsi sensibilmente dal livello nominale, specialmente in presenza di eterogeneità o violazioni delle assunzioni di base.

Come osservato da Wasserman [46], tali ipotesi, seppur matematicamente convenienti, risultano spesso irrealistiche in contesti pratici complessi, e la loro violazione può compromettere seriamente la validità degli intervalli costruiti. Secondo Wasserman [46], la validità predittiva dei metodi classici è compromessa non appena le assunzioni (normalità, linearità, omoschedasticità) vengono violate, anche in misura contenuta. Questo limite è particolarmente rilevante nel contesto dell'apprendimento automatico, dove i dati non seguono schemi predefiniti e la complessità dei modelli rende difficile verificare le ipotesi tradizionali.

L'applicabilità delle ipotesi classiche in ambito di apprendimento automatico risulta frequentemente limitata, in quanto tali assunti non sono spesso soddisfatti. Tali considerazioni hanno stimolato la ricerca di approcci capaci di fornire intervalli predittivi validi anche in assenza di ipotesi parametriche forti e in condizioni non asintotiche, come avviene nel caso della *Conformal Prediction*.

Questa situazione ha stimolato lo sviluppo di approcci alternativi, più flessibili e maggiormente idonei a rispondere alle esigenze di contesti moderni, contraddistinti da significative quantità di dati e da modelli altamente sofisticati. Queste limitazioni hanno aperto la strada allo sviluppo di approcci più robusti e adattabili, in grado di fornire garanzie di validità predittiva anche in scenari complessi.

Tra questi, un ruolo centrale è occupato dalla *Conformal Prediction*, che consente di costruire intervalli predittivi validi senza assumere una forma specifica per la distribuzione dei dati, garantendo copertura marginale finita in qualsiasi contesto.

Questo approccio, inizialmente proposto da Vovk, Gammerman e Shafer [44], è stato successivamente approfondito in un contesto teorico più generale in una nota rassegna ([18], [2]) , dove viene formalizzata la nozione di validità marginale e vengono introdotte le versioni transduttiva e induttiva della *Conformal Prediction*.

1.5 Metodi Alternativi per la Costruzione di Intervalli di Predizione

Negli ultimi anni, l'evoluzione della modellistica statistica e computazionale ha portato allo sviluppo di nuove tecniche per la costruzione di intervalli di previsio-

ne, in grado di affrontare contesti caratterizzati da elevata complessità strutturale, eterogeneità e possibile misspecificazione del modello. Accanto agli approcci parametrici classici, la letteratura recente propone strumenti consolidati e di rilievo applicativo, tra cui: la *Quantile Regression Averaging*, che combina previsioni puntuali modellando direttamente i quantili condizionali ([26], [17], [5]); i modelli di serie temporali di tipo ARIMA, ampiamente utilizzati per la previsione dinamica sotto ipotesi di struttura stocastica esplicita; gli approcci bayesiani, basati sulla distribuzione predittiva completa; e le tecniche bootstrap, che consentono un'approssimazione non parametrica della distribuzione della statistica d'interesse. La selezione di questi metodi non è esaustiva, ma riflette la loro centralità teorica e la diffusione nelle applicazioni economiche ed energetiche considerate nella presente trattazione.

1.5.1 Quantile Regression Averaging (QRA)

Introdotto nel 2014, questo metodo combina previsioni puntuali provenienti da molteplici modelli attraverso la regressione quantile. Tale approccio consente la costruzione di intervalli predittivi più robusti, risultando particolarmente efficace in contesti complessi, come ad esempio nella previsione dei prezzi dell'energia elettrica.

Definizione 3 (Quantile Regression Averaging)**.** *La Quantile Regression Averaging (QRA) è una metodologia introdotta da Nowotarski e Weron [26] che combina previsioni puntuali provenienti da diversi modelli tramite regressione quantile. Piuttosto che stimare la media condizionale della variabile dipendente, la regressione quantile stima la condizione su un quantile specifico della distribuzione della variabile risposta. La QRA permette così la costruzione di intervalli di previsione direttamente dai quantili stimati.*

Sia y_t la variabile di interesse al tempo t, e siano $\hat{y}_t^{(1)}, \ldots, \hat{y}_t^{(M)}$ le previsioni al tempo t fornite da M modelli di forecasting. Si definisce:

$$Q_\tau(y_t \mid \hat{y}_t^{(1)}, \ldots, \hat{y}_t^{(M)}) = \beta_0^{(\tau)} + \sum_{m=1}^{M} \beta_m^{(\tau)} \hat{y}_t^{(m)}$$

dove $Q_\tau(\cdot)$ è il quantile τ-esimo condizionale. I coefficienti $\beta^{(\tau)} = (\beta_0^{(\tau)}, \ldots, \beta_M^{(\tau)})$ sono stimati tramite regressione quantile, minimizzando la quantile loss function:

$$\min_{\beta} \sum_{t=1}^{T} \rho_\tau \left(y_t - \beta_0 - \sum_{m=1}^{M} \beta_m \hat{y}_t^{(m)} \right)$$

dove $\rho_\tau(u) = u(\tau - \mathbb{I}\{u < 0\})$ *è la funzione di perdita asimmetrica dei quantili.*

Teorema 5 (Proprietà della Regressione Quantile)**.** *Sia* Y *la variabile risposta e* $X \in \mathbb{R}^p$ *un vettore di regressori. Per un livello* $\tau \in (0, 1)$*, la regressione quantile stima il coefficiente* $\beta(\tau)$ *come:*

$$\hat{\beta}(\tau) = \arg\min_{\beta} \sum_{i=1}^{n} \rho_\tau(y_i - x_i^\top \beta),$$

dove $\rho_\tau(u) = u(\tau - \mathbb{I}_{\{u<0\}})$ *è la funzione di perdita asimmetrica.*

La regressione quantile gode di alcune proprietà fondamentali:

- ***Robustezza****: a differenza della regressione lineare ordinaria, è robusta rispetto a outlier nella variabile risposta.*
- ***Flessibilità****: consente di analizzare l'intero spettro della distribuzione condizionale di* Y *dato* X*, non solo la media.*
- ***Invarianza a trasformazioni monotone****: il quantile* τ *di* Y *è invariante rispetto a trasformazioni monotone di* Y*.*
- ***Linearità condizionale****: la funzione quantile condizionale stimata è una funzione lineare dei regressori, ma non impone assunzioni di linearità nella media.*
- ***Stime coerenti****: sotto opportune condizioni regolari, gli stimatori della regressione quantile sono consistenti e asintoticamente normali.*

In termini intuitivi, mentre la regressione lineare ordinaria si concentra sulla stima della media condizionale di Y dato X, la regressione quantile permette di studiare l'intera distribuzione condizionale. Ciò significa che possiamo modellare non solo il comportamento "tipico" della variabile risposta, ma anche code e variazioni estreme, ottenendo una descrizione più completa della relazione tra X e Y ([17], [5]).

Ad esempio, in un contesto di previsione dei prezzi dell'elettricità, potremmo utilizzare le stime di tre modelli diversi come regressori nella regressione quantile.

In questo modo, possiamo stimare simultaneamente quantili bassi e alti (ad esempio il 10-esimo e il 90-esimo), e definire così un intervallo predittivo $[\hat{Q}_{0.1}, \hat{Q}_{0.9}]$ che cattura la variabilità possibile dei prezzi. Questa applicazione illustra concretamente le proprietà discusse nel teorema, in particolare la flessibilità nell'analizzare l'intera distribuzione condizionale e la robustezza rispetto a valori estremi.

1.5.2 Modelli di Serie Temporali

Tecniche quali ARIMA, smoothing esponenziale e modelli a fattori dinamici sono frequentemente impiegate per derivare intervalli di previsione in ambito temporale. Tuttavia, è importante sottolineare che anche questi approcci si fondano su assunzioni parametriche forti e si rivelano spesso inadeguati in presenza di dati non stazionari o caratterizzati da strutture latenti complesse.

Definizione 4 (Modello ARIMA)**.** *Un modello ARIMA (AutoRegressive Integrated Moving Average) è una generalizzazione dei modelli ARMA per processi non stazionari. Si rappresenta come :*

$$\phi(B)(1-B)^d y_t = \theta(B)\varepsilon_t$$

dove:

- B *è l'operatore di ritardo:* $By_t = y_{t-1}$*;*
- $(1-B)^d y_t$ *rappresenta la differenziazione di ordine* d*;*
- $\phi(B) = 1 - \phi_1 B - \cdots - \phi_p B^p$ *è il polinomio autoregressivo;*
- $\theta(B) = 1 + \theta_1 B + \cdots + \theta_q B^q$ *è il polinomio delle medie mobili;*
- ε_t *è un rumore bianco con media nulla e varianza costante.*

Assioma 2 (Ipotesi di Stazionarietà)**.** Una serie temporale $\{y_t\}_{t\in\mathbb{Z}}$ è detta *stazionaria in senso debole* (o *stazionaria al secondo ordine*) se soddisfa le seguenti condizioni:

1. $\mathbb{E}[y_t] = \mu$ è costante per ogni t;
2. $\mathrm{Var}(y_t) = \sigma^2 < \infty$ per ogni t;

3. $\text{Cov}(y_t, y_{t+h}) = \gamma(h)$ dipende solo dal lag h, non dal tempo assoluto t.

La stazionarietà è una condizione necessaria per la validità teorica di molti modelli autoregressivi e per garantire proprietà asintotiche degli stimatori.

Teorema 6 (Intervallo di Predizione per AR(1))**.** *Consideriamo un processo autoregressivo di ordine 1:*

$$y_t = \phi y_{t-1} + \varepsilon_t, \quad \varepsilon_t \sim \mathcal{N}(0, \sigma^2)$$

con $|\phi| < 1$ *per garantire stazionarietà. Il valore previsto un passo avanti è:*

$$\hat{y}_{t+1} = \phi y_t$$

e la varianza dell'errore di previsione è:

$$\text{Var}(y_{t+1} - \hat{y}_{t+1}) = \sigma^2$$

Quindi, un intervallo di previsione al livello $1 - \alpha$ *per* y_{t+1} *è dato da:*

$$\left[\hat{y}_{t+1} - z_{1-\alpha/2} \cdot \sigma,\ \hat{y}_{t+1} + z_{1-\alpha/2} \cdot \sigma\right]$$

dove $z_{1-\alpha/2}$ *è il quantile della normale standard corrispondente al livello di confidenza scelto.*

Il risultato precedente mostra che, nel caso di un processo AR(1) con errori gaussiani, l'intervallo di previsione dipende esclusivamente dalla varianza dell'innovazione σ^2 e dal quantile della normale standard. In questo contesto, l'incertezza della previsione a un passo avanti non cresce nel tempo, poiché l'errore di previsione coincide con l'innovazione stessa.

In applicazioni pratiche, come la previsione della domanda elettrica giornaliera, modelli autoregressivi integrati come un ARIMA(1,1,0) vengono utilizzati dopo aver reso stazionaria la serie mediante differenziazione. Una volta stimati i parametri del modello, è possibile costruire intervalli di previsione a passo singolo utilizzando la varianza stimata dell'errore e la struttura teorica illustrata sopra. L'intervallo risultante riflette l'ipotesi di normalità delle innovazioni e fornisce una quantificazione parametrica dell'incertezza associata alla previsione.

1.5.3 Metodi Bayesiani

I metodi bayesiani integrano informazioni a priori e aggiornano le stime man mano che nuovi dati diventano disponibili. Questi approcci forniscono una distribuzione predittiva completa, dalla quale è possibile dedurre intervalli di previsione. Tuttavia, la loro implementazione può risultare computazionalmente intensiva e richiede una precisa specifica delle distribuzioni a priori appropriate.

Definizione 5 (Distribuzione Predittiva Bayesiana)**.** *Nel contesto bayesiano, la distribuzione predittiva di una nuova osservazione y^*, dato un campione osservato $y = (y_1, \ldots, y_n)$, è ottenuta integrando la distribuzione condizionale $p(y^* \mid \theta)$ rispetto alla distribuzione a posteriori del parametro θ:*

$$p(y^* \mid y) = \int_\Theta p(y^* \mid \theta)\, p(\theta \mid y)\, d\theta$$

Questa distribuzione rappresenta la nostra incertezza su y^* dopo aver osservato i dati, incorporando sia la variabilità dei dati condizionatamente a θ, sia l'incertezza su θ stesso.

Assioma 3 (Principio di Aggiornamento Bayesiano)**.** Il principio cardine dell'inferenza bayesiana è il seguente:

La conoscenza a priori su un parametro ignoto θ viene aggiornata alla luce dei dati osservati y tramite la formula di Bayes, ottenendo una distribuzione a posteriori:

$$p(\theta \mid y) = \frac{p(y \mid \theta)\, p(\theta)}{p(y)} = \frac{p(y \mid \theta)\, p(\theta)}{\int_\Theta p(y \mid \theta)\, p(\theta)\, d\theta}$$

dove $p(\theta)$ è la distribuzione a priori, $p(y \mid \theta)$ è la verosimiglianza, e $p(\theta \mid y)$ è la distribuzione a posteriori.

Teorema 7 (Teorema di Coerenza Predittiva Bayesiana)**.** Sotto opportune condizioni di regolarità, la distribuzione predittiva bayesiana $p(y^* \mid y)$ converge, al crescere della numerosità campionaria, alla distribuzione vera dei dati.

Formalmente, se i dati $y_1, \ldots, y_n$ sono i.i.d. da una distribuzione F_0 e il modello bayesiano specificato include F_0 come caso particolare (cioè è ben specificato), allora:

$$p(y^* \mid y_1, \ldots, y_n) \xrightarrow[n\to\infty]{d} F_0$$

Questa proprietà giustifica l'utilizzo della distribuzione predittiva per costruire intervalli di previsione asintoticamente validi [46].

La proprietà enunciata evidenzia che, qualora il modello sia correttamente specificato, la distribuzione predittiva bayesiana tende progressivamente a coincidere con il vero meccanismo generatore dei dati all'aumentare della numerosità campionaria. Ne consegue che l'incertezza incorporata nella distribuzione predittiva diventa una rappresentazione sempre più fedele sia della variabilità intrinseca del fenomeno sia dell'incertezza sui parametri ignoti. I quantili della distribuzione predittiva possono quindi essere utilizzati per costruire intervalli di previsione che risultano giustificati in senso asintotico.

Si consideri, ad esempio, la previsione dei prezzi del gas naturale. È possibile specificare una distribuzione a priori per parametri chiave, come la volatilità, aggiornarla sulla base delle osservazioni storiche e ottenere così una distribuzione predittiva per i valori futuri. Gli intervalli di previsione sono determinati dai quantili di tale distribuzione e riflettono simultaneamente l'incertezza parametrica e quella stocastica. La loro affidabilità, tuttavia, è strettamente legata alla correttezza della struttura modellistica assunta e trova piena giustificazione soltanto nel limite asintotico.

1.5.4 Bootstrap e Metodi Non Parametrici

Il bootstrap è una tecnica di campionamento che consente di stimare la distribuzione di una statistica campionaria senza dover assumere un modello parametricamente definito. Attraverso l'applicazione del bootstrap, è possibile costruire intervalli di previsione empirici. Tuttavia, la validità di tali intervalli è correlata alla rappresentatività del campione e alla quantità di dati disponibili.

Definizione 6 (Bootstrap)**.** *Sia $y = (y_1, \ldots, y_n)$ un campione osservato. Si generano B campioni bootstrap $y^{*(b)} = (y_1^{*(b)}, \ldots, y_n^{*(b)})$ prelevati con reinserimento da y, per $b = 1, \ldots, B$.*

Su ciascun campione bootstrap si calcola una statistica di interesse $T^{*(b)}$, ottenendo una distribuzione empirica approssimativa della distribuzione della statistica $T = T(y)$. Tale approccio consente di:

- *stimare la varianza di T;*

- *costruire intervalli di confidenza o previsione;*
- *valutare la stabilità delle stime.*

Teorema 8 (Teorema di Consistenza del Bootstrap)**.** Sotto opportune condizioni, la distribuzione bootstrap converge in probabilità alla distribuzione reale della statistica d'interesse.

Sia $T_n = T(y_1, \ldots, y_n)$ una statistica calcolata sul campione, e sia T_n^* la stessa statistica calcolata su un campione bootstrap. Allora:

$$\sup_x |\mathbb{P}^*(T_n^* \leq x) - \mathbb{P}(T_n \leq x)| \xrightarrow[n\to\infty]{\mathbb{P}} 0$$

dove $\mathbb{P}^*$ è la probabilità condizionata al campione osservato. Questo implica che il bootstrap fornisce una buona approssimazione della distribuzione della statistica, almeno asintoticamente [11].

Assioma 4 (Rappresentatività del Campione)**.** Un campione $y = (y_1, \ldots, y_n)$ è detto *rappresentativo* rispetto a una popolazione Y se le sue caratteristiche statistiche riproducono fedelmente quelle della popolazione.

Più formalmente, si richiede che:

- la distribuzione empirica del campione converga a quella della popolazione (consistenza);
- le statistiche campionarie (media, varianza, quantili, ecc.) siano stime non distorte o asintoticamente consistenti dei corrispondenti parametri di popolazione;
- il campionamento sia effettuato in modo casuale, preferibilmente i.i.d., per evitare *bias* di selezione.

La rappresentatività è una condizione fondamentale per poter generalizzare le inferenze ottenute dal campione all'intera popolazione.

Il risultato di consistenza appena enunciato fornisce una giustificazione teorica all'utilizzo del bootstrap per la costruzione di intervalli di previsione in ambito non parametrico. Se la distribuzione bootstrap approssima in probabilità la distribuzione

reale della statistica, allora i quantili della distribuzione ricampionata possono essere impiegati per quantificare l'incertezza in modo asintoticamente corretto.

In ambito finanziario, ad esempio, si può considerare la previsione del prezzo di chiusura di un'azione. A partire dal campione storico osservato, si generano un elevato numero di campioni bootstrap mediante ricampionamento con reinserimento e, per ciascuno di essi, si calcola la quantità di interesse (ad esempio il rendimento o il prezzo previsto al passo successivo). La distribuzione empirica delle statistiche così ottenute costituisce un'approssimazione della distribuzione campionaria della previsione. Un intervallo predittivo può quindi essere costruito selezionando opportuni percentili, come il 5° e il 95°.

Tale procedura risulta metodologicamente fondata solo se il campione originario è rappresentativo della popolazione di riferimento. In assenza di questa condizione, il ricampionamento non fa che replicare eventuali distorsioni strutturali presenti nei dati iniziali, compromettendo la validità inferenziale dell'intervallo ottenuto.

1.6 Applicazioni Moderne degli Intervalli di Predizione

Gli intervalli di previsione sono strumenti fondamentali per la presa di decisioni in numerosi ambiti applicativi. In medicina, ad esempio, consentono di quantificare l'incertezza nella prognosi individuale di un paziente, contribuendo allo sviluppo della medicina personalizzata. In finanza, vengono utilizzati per stimare il rischio futuro attraverso misure come il *Value-at-Risk (VaR)*, mentre nel settore energetico sono impiegati nella previsione dei carichi di domanda, essenziale per la stabilità delle reti. Nel contesto dell'apprendimento automatico, gli intervalli predittivi sono cruciali per dotare i modelli di intelligenza artificiale di una misura di affidabilità, requisito fondamentale per applicazioni critiche quali la guida autonoma, la diagnostica automatizzata o i sistemi di raccomandazione. Tuttavia, la costruzione di intervalli affidabili in questi ambiti richiede metodologie che vadano oltre i modelli classici, superando vincoli parametrici e garantendo validità anche in presenza di dati complessi, rumorosi o eterogenei ([15], [1]).

Come discusso in Gneiting e Katzfuss [15], la comunicazione dell'incertezza predittiva è essenziale per la trasparenza e l'affidabilità dei sistemi previsivi moderni.

1.7 Considerazioni Conclusive

La distinzione tra intervalli di confidenza e intervalli di previsione, unitamente alla comprensione delle rispettive ipotesi e finalità, riveste una grande importanza per una corretta applicazione dell'inferenza statistica, specialmente nel campo delle scienze applicate e dell'apprendimento automatico. Mentre i metodi classici offrono una solida base teorica, le attuali esigenze analitiche richiedono soluzioni più flessibili e adattabili a strutture di dati eterogenee e dinamiche.

Tabella 1.1: Confronto tra metodi per intervalli di previsione

Metodo	**Assunzioni Principali**	**Flessibilità**	**Copertura Finita**
Intervallo Classico (Frequentista)	Normalità, i.i.d., linearità	Bassa	No
Modelli Bayesiani	Specifica del modello e delle prior	Media	Sì (sotto ipotesi forti)
Modelli ARIMA / Serie Temporali	Stazionarietà, struttura lineare	Media	No
Bootstrap	Campione rappresentativo	Alta	Solo approssimativa
Quantile Regression Averaging (QRA)	Etichette multiple, additività	Alta	No
Conformal Prediction	Scambiabilità (*exchangeability*)	Alta	**Sì (marginale)**

1.8 Verso la *Conformal Prediction*

Alla luce delle problematiche teoriche e pratiche evidenziate nei metodi classici e alternativi, emerge la necessità di un approccio che sia al contempo flessibile, non parametrico e in grado di fornire garanzie di copertura finite.

La *Conformal Prediction*, proposta inizialmente da Vovk, Gammerman e Shafer [44], si inserisce esattamente in questo contesto, offrendo un quadro metodologico generale per la costruzione di intervalli predittivi validi sotto ipotesi minimali.

Questo metodo si basa su una misura di non conformità tra le osservazioni e un modello predittivo, ed è applicabile a qualsiasi algoritmo di apprendimento supervisionato. La sua caratteristica distintiva è la capacità di garantire, in modo

teoricamente dimostrabile, una probabilità di copertura marginale pari a $1 - \alpha$, senza richiedere la conoscenza della distribuzione dei dati.

Il prossimo capitolo sarà interamente dedicato alla *Conformal Prediction*: ne esploreremo le basi teoriche, le versioni algoritmiche principali (induttiva e transduttiva), le proprietà di validità e efficienza, nonché le applicazioni pratiche nei contesti moderni.

Capitolo 2

Conformal Prediction

2.1 Fondamenti teorici della *Conformal Prediction*

La *Conformal Prediction* (CP) è una metodologia statistica sviluppata da Vovk, Gammerman e Shafer ([44], [38]) con l'obiettivo di costruire insiemi predittivi (*prediction sets*) o intervalli di previsione che garantiscano una copertura di tipo frequentista in campioni finiti, senza la necessità di assumere una distribuzione parametrica per i dati. Il quadro metodologico si inserisce all'interno dell'approccio della *distribution-free inference* e richiede esclusivamente l'assunzione di scambiabilità, una condizione più debole rispetto all'indipendenza e identica distribuzione (i.i.d.).

Definizione 7 (*Prediction Set* e *Prediction Interval*)**.** *Un* prediction set *è un sottoinsieme dello spazio delle risposte Y associato a una nuova osservazione x_{n+1}, costruito in modo da contenere la risposta reale Y_{n+1} con probabilità almeno $1-\alpha$. Quando lo spazio delle risposte è continuo, come nel caso di problemi di regressione, il prediction set prende la forma di un* prediction interval*, ossia un intervallo reale $[a, b]$ tale che*

$$\mathbb{P}(Y_{n+1} \in [a, b]) \geq 1 - \alpha.$$

In altri contesti, ad esempio la classificazione, il prediction set può essere un insieme discreto di etichette possibili ([44], [38]).

Definizione 8 (Distribution-Free Inference)**.** *Una procedura inferenziale è detta distribution-free se le sue garanzie teoriche (ad esempio la copertura) non dipen-*

dono dalla specifica distribuzione dei dati, ma solo da proprietà strutturali come la scambiabilità.

Definizione 9 (Scambiabilità (*Exchangeability*))**.** *Una sequenza di variabili aleatorie $Z_1, Z_2, \ldots, Z_n$ definite su uno spazio di probabilità comune è detta scambiabilità se, per ogni permutazione π dell'insieme $\{1, \ldots, n\}$, vale:*

$$P(Z_1 = z_1, \ldots, Z_n = z_n) = P(Z_{\pi(1)} = z_1, \ldots, Z_{\pi(n)} = z_n).$$

Dato un nuovo punto x_{n+1}, lo scopo è costruire un insieme predittivo $C(x_{n+1}) \subseteq \mathcal{Y}$ tale che:

$$P(Y_{n+1} \in C(X_{n+1})) \geq 1 - \alpha,$$

dove (X_{n+1}, Y_{n+1}) è una futura osservazione e $\alpha \in (0, 1)$ è il livello di significatività prefissato.

Per illustrare in modo intuitivo il funzionamento della *Conformal Prediction*, si può considerare un contesto generico di previsione. Supponiamo di avere a disposizione un insieme di osservazioni storiche, ciascuna costituita da caratteristiche note X_i e risposte corrispondenti Y_i. Per una nuova osservazione X_{n+1}, la CP consente di costruire un insieme predittivo $C(X_{n+1})$ che riflette l'incertezza reale associata alla risposta Y_{n+1}, adattandosi automaticamente alla variabilità osservata nei dati precedenti. Ad esempio, se le osservazioni storiche mostrano valori di Y concentrati in un certo intervallo, il *prediction set* risultante sarà relativamente ristretto; se invece la nuova osservazione è atipica rispetto al campione, l'insieme si allargherà per mantenere la probabilità di copertura prefissata $1 - \alpha$. In questo modo, l'insieme predittivo non dipende da alcuna assunzione parametrica sulla distribuzione dei dati, ma solo dalla scambiabilità delle osservazioni. Questo approccio permette di ottenere una misura di incertezza rigorosa e distribuzione-free, valida anche in campioni finiti, applicabile sia a problemi di regressione continua, dove il *prediction set* assume la forma di un intervallo reale, sia a problemi di classificazione, dove esso può essere un insieme discreto di etichette possibili ([44], [38]).

2.1.1 Il concetto di *Nonconformity Score*

Definizione 10 (*Nonconformity Score*)**.** *Sia $\mathcal{Z}^* = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ un dataset di addestramento. Un* ***nonconformity score*** *è una funzione*

$$A : \mathcal{Z}^* \times \mathcal{Z} \to \mathbb{R},$$

che associa a ogni nuova osservazione $z = (x, y)$ un valore reale che ne misura la distanza, in senso generalizzato, dal comportamento "tipico" delle osservazioni precedenti.

Osservazione 1. *Il punteggio di non conformità può essere progettato in vari modi a seconda del tipo di modello predittivo utilizzato. Esempi alternativi includono:*

- *la distanza di Mahalanobis rispetto al centroide delle osservazioni;*
- *$1 - \hat{p}(y \mid x)$ per modelli probabilistici in classificazione, dove $\hat{p}(y \mid x)$ indica la probabilità stimata che l'osservazione appartenga alla classe y dato l'input x;*
- *la log-loss negativa per modelli di classificazione probabilistica, definita come $-\log \hat{p}(y \mid x)$, che misura la qualità della previsione probabilistica penalizzando fortemente assegnazioni di probabilità basse alla classe osservata.*

Un buon nonconformity score riflette fedelmente quanto un'osservazione si discosta dal comportamento tipico dell'insieme di addestramento ([16], [1], [15]).

Esempio 1 (Scelta del *Nonconformity Score* ed Efficienza)**.** Supponiamo di stimare $f(x)$ con due modelli diversi: una regressione lineare e una rete neurale. Utilizzando lo stesso punteggio $|y - \hat{f}(x)|$, il modello più accurato genererà intervalli predittivi più stretti, migliorando l'efficienza.

Un esempio classico in regressione è:

$$A(\mathcal{Z}^*, (x, y)) = |y - \hat{f}(x)|,$$

dove $\hat{f}$ è un predittore stimato sui dati.

2.1.2 Addestramento e calibrazione nella *Conformal Prediction*

Dopo aver definito il punteggio di non conformità, è necessario chiarire come esso venga concretamente calcolato a partire dai dati osservati. La costruzione degli insiemi predittivi conformi richiede infatti di distinguere il ruolo che le osservazioni svolgono nel processo di apprendimento e in quello di calibrazione.

Sia

$$S = \{(X_i, Y_i)\}_{i=1}^{n}$$

un campione osservato di coppie *input–output*, assunto scambiabile. Nel contesto dell'apprendimento supervisionato, per insieme di addestramento o *training set* si intende l'insieme di dati utilizzato per stimare un modello predittivo $\hat{f}$, mentre per insieme di calibrazione o *calibration set* si intende un sottoinsieme di dati impiegato per valutare i punteggi di non conformità e determinare la soglia che definisce l'insieme predittivo finale.

In termini generali, il procedimento si articola in tre passaggi: (i) stima del modello $\hat{f}$ sui dati di addestramento; (ii) calcolo dei punteggi di non conformità sui dati di calibrazione; (iii) determinazione di una soglia come quantile empirico dei punteggi, che consente di costruire l'insieme predittivo per una nuova osservazione.

È importante sottolineare che la suddivisione tra addestramento e calibrazione non è un requisito teorico della *Conformal Prediction* in quanto tale. La garanzia di copertura non deriva dalla modalità di suddivisione del campione, bensì dalla proprietà di scambiabilità delle osservazioni e dalla simmetria dei ranghi dei punteggi di non conformità ([44],[38]).

La distinzione tra *training* e calibrazione rappresenta dunque un'organizzazione operativa dei dati finalizzata alla costruzione degli insiemi predittivi. Nel paragrafo successivo si formalizzeranno le proprietà di validità che discendono dalla scambiabilità e che costituiscono il fondamento teorico della metodologia conforme.

2.2 Proprietà fondamentali: validità, adattabilità, efficienza

La *Conformal Prediction* (CP) si distingue per alcune proprietà fondamentali che ne garantiscono l'utilità e la robustezza nell'ambito dell'inferenza predittiva. Le

tre proprietà principali sono la **validità**, l'**efficienza** e l'**adattabilità**. Di seguito vengono formalmente introdotte e discusse.

2.2.1 Validità

La proprietà di validità è centrale nella CP: essa garantisce che l'intervallo predittivo costruito abbia una copertura statistica almeno pari al livello di confidenza desiderato, senza assumere condizioni parametriche forti sulla distribuzione dei dati. Formalmente:

Teorema 9 (Validità di *Conformal Prediction*)**.** *Sia $S = \{(x_i, y_i)\}_{i=1}^n$ una sequenza scambiabile di dati osservati e sia $C(x_{n+1})$ l'insieme predittivo costruito tramite una procedura conforme al quadro metodologico CP. Allora, per ogni livello di significatività $\alpha \in (0, 1)$, vale:*

$$P(Y_{n+1} \in C(X_{n+1})) \geq 1 - \alpha.$$

Dimostrazione. La validità si basa sull'invarianza della distribuzione congiunta dei dati sotto permutazioni, data dalla scambiabilità. Il calcolo dei p-value conformi, tramite il *nonconformity score*, induce una distribuzione discreta uniforme dei ranghi, da cui segue che la probabilità che il nuovo punto cada fuori dall'intervallo è al massimo α. □

Teorema 10 (Validità per Classificazione Binaria)**.** *Nel caso di classificazione binaria $Y \in (0, 1)$, se i punteggi di non conformità sono simmetrici rispetto alle classi, allora:*

$$\mathbb{P}(Y_{n+1} \in C(X_{n+1})) \geq 1 - \alpha.$$

Questa copertura è garantita marginalmente, ma non necessariamente condizionatamente per ciascuna classe.

Definizione 11 (Copertura Marginale)**.** *La copertura marginale di un insieme predittivo $C(\cdot)$ è definita come la probabilità marginale:*

$$P(Y \in C(X)),$$

dove la probabilità è calcolata rispetto alla distribuzione congiunta dei dati e della nuova osservazione.

Osservazione 2. *La validità garantita da CP è marginale, cioè calcolata sulla distribuzione complessiva, e non necessariamente condizionata su specifici valori di X. Per ottenere validità condizionata sono richieste ulteriori ipotesi o metodologie, come Mondrian CP. La Mondrian CP è trattata in dettaglio nella Sezione 2.3.3.*

2.2.2 Efficienza

L'efficienza riguarda la qualità dell'intervallo predittivo prodotto da CP. In particolare:

Definizione 12 (Efficienza)**.** *Un insieme predittivo $C(x)$ è detto efficiente se, per un dato livello di copertura $1-\alpha$, l'ampiezza media o la misura di $C(x)$ è il più piccola possibile, ovvero:*

$$\textit{Efficienza} \propto \mathbb{E}[\dim(C(X))],$$

dove $\dim(\cdot)$ *può indicare la lunghezza dell'intervallo in regressione o il numero di classi incluse in classificazione.*

Osservazione 3. *Esiste un trade-off tra validità ed efficienza: garantire una copertura elevata tende ad aumentare la dimensione degli intervalli, riducendone l'utilità pratica. Le scelte di nonconformity score influenzano direttamente l'efficienza del metodo.*

Osservazione 4 (Efficienza come Problema di Ottimizzazione)**.** *L'efficienza della CP può essere formulata come un problema di ottimizzazione vincolata:*

$$\min_{C(x):\mathbb{P}(Y\in C(X))\geq 1-\alpha} \mathbb{E}[\dim(C(X))],$$

dove $\dim(C(X))$ *rappresenta, ad esempio, la lunghezza dell'intervallo in regressione o la cardinalità dell'insieme in classificazione.*

Esempio 2 (*Nonconformity score* e efficienza)**.** Se si sceglie come *nonconformity score* la semplice distanza assoluta residua $|y - \hat{f}(x)|$, l'efficienza dipenderà dalla bontà della stima $\hat{f}$. Un modello più accurato produce intervalli più stretti, migliorando l'efficienza.

2.2.3 Adattabilità

La terza proprietà chiave è l'adattabilità, ossia la capacità di CP di essere estesa e adattata a contesti più generali rispetto a quelli originari.

Definizione 13 (Adattabilità)**.** *La Conformal Prediction è detta adattabile se esistono varianti del metodo in grado di fornire garanzie di copertura anche in presenza di:*

- *dati non scambiabili (es. dipendenze temporali);*
- *distribuzioni che cambiano nel tempo (non stazionarietà);*
- *strutture dati complesse (multi-task, gerarchici, ecc.).*

Osservazione 5 (Adattamento e Fairness)**.** *La CP può essere adattata per garantire copertura equa tra gruppi definiti da caratteristiche sensibili (es. genere, etnia). Utilizzando la Mondrian CP (2.3.3) con partizioni definite da queste variabili, è possibile ottenere validità condizionata localmente, mitigando il rischio di bias.*

Esempio 3 (Adattabilità con *Adaptive Conformal Inference*)**.** La metodologia *Adaptive Conformal Inference* (ACI) estende la CP per dati temporali non scambiabili, aggiornando dinamicamente gli intervalli di previsione per mantenere la copertura desiderata nel tempo.

Osservazione 6. *L'adattabilità è spesso implementata tramite l'introduzione di pesi, finestre mobili o strutture di calibrazione multiple, che tengano conto della struttura temporale o dipendente dei dati.*

2.2.4 Riassumendo

- La **validità** assicura che l'intervallo predittivo copra la vera osservazione con probabilità almeno $1 - \alpha$, in senso marginale.
- L'**efficienza** quantifica quanto gli intervalli predittivi siano stretti o informativi, essendo influenzata dalla scelta del *nonconformity score* e dal modello predittivo.
- L'**adattabilità** permette di estendere CP a contesti più complessi, come dati non indipendenti, non identicamente distribuiti o strutturati.

2.3 Varianti della *Conformal Prediction*

La metodologia della *Conformal Prediction* si declina in diverse varianti, ciascuna adatta a specifiche esigenze computazionali e strutture dei dati. Le differenze principali riguardano il modo in cui si utilizzano i dati di addestramento e calibrazione, l'efficienza computazionale e le proprietà di validità garantite.

2.3.1 Transductive Conformal Prediction (TCP)

La variante transduttiva rappresenta la formulazione originaria della *Conformal Prediction* proposta da Vovk, Gammerman e Shafer [44]. Essa opera secondo un approccio *transduttivo* e *punto per punto*: per una nuova osservazione x_{n+1}, ogni possibile valore candidato y viene temporaneamente aggiunto al *dataset* di addestramento e il modello viene ricalcolato includendo la coppia (x_{n+1}, y).

Per ciascuna ipotesi candidata si valuta quindi il relativo *nonconformity score* e si determina se tale valore risulta compatibile con quelli osservati sui dati di *training*. L'insieme predittivo finale è costituito da tutti i valori di y che non vengono rifiutati da questo procedimento.

Definizione 14 (*Transductive Conformal Prediction*)**.** *Sia $S = \{(x_i, y_i)\}_{i=1}^n$ il training set e sia x_{n+1} un nuovo punto. Per ogni possibile $y \in \mathcal{Y}$, si costruisce il dataset esteso*

$$S^y := S \cup \{(x_{n+1}, y)\}.$$

Si calcolano i punteggi di non conformità α_i relativi a S^y e si definisce il p-value:

$$p(y) = \frac{1}{n+1} \sum_{i=1}^{n+1} \mathbb{I}\big(\alpha_i \geq \alpha_{n+1}\big).$$

L'insieme predittivo per x_{n+1} è:

$$C(x_{n+1}) := \{y \in \mathcal{Y} : p(y) > \alpha\}.$$

Osservazione 7. *La TCP garantisce la validità marginale esatta, senza ipotesi addizionali oltre la scambiabilità, grazie alla simmetria del procedimento di permutazione. Tuttavia, la complessità computazionale cresce linearmente con il numero di valori testati per y, rendendola poco pratica per problemi con spazi di output continui o di grandi dimensioni.*

Esempio 4. Supponiamo di avere un *dataset* con $n = 10$ osservazioni e di voler costruire un intervallo predittivo per una nuova osservazione x_{11}. Per ogni possibile valore y in un insieme discreto (ad esempio, valori interi da 0 a 100), si aggiunge (x_{11}, y) al *dataset*, si calcolano i punteggi α_i e si determina il p-value $p(y)$. L'intervallo predittivo sarà quindi l'insieme di tutti i y per cui $p(y) > \alpha$.

2.3.2 Inductive Conformal Prediction (ICP)

La *Transductive Conformal Prediction*, pur essendo teoricamente elegante, richiede il riaddestramento del modello per ciascun valore candidato della nuova risposta, con un costo computazionale elevato soprattutto in spazi di output continui. L'*Inductive Conformal Prediction* (ICP) rappresenta una riformulazione computazionalmente più efficiente del procedimento conforme. In questa versione, il *dataset* osservato viene diviso in due sottogruppi disgiunti: uno utilizzato per stimare il modello predittivo e uno per calcolare i punteggi di non conformità. Il modello predittivo viene stimato una sola volta sul primo sottoinsieme, mentre i punteggi di non conformità utilizzati per determinare la soglia vengono calcolati sul secondo. La costruzione dell'insieme predittivo mantiene la stessa logica basata sui ranghi dei punteggi, ma evita il riaddestramento ripetuto del modello.

Definizione 15 (*Inductive Conformal Prediction*)**.** *Si suddivide il dataset in:*

$$\mathcal{D}_{train} = \{(x_i, y_i)\}_{i=1}^{m}, \quad \mathcal{D}_{cal} = \{(x_i, y_i)\}_{i=m+1}^{n},$$

con $m < n$. Si addestra un modello predittivo $\hat{f}$ sui dati di addestramento $\mathcal{D}_{train}$. Successivamente, si calcolano i punteggi di non conformità (ad esempio residui assoluti) sui dati appartenenti all'insieme di calibrazione:

$$s_i = |y_i - \hat{f}(x_i)|, \quad \forall (x_i, y_i) \in \mathcal{D}_{cal}.$$

Sia $q_{1-\alpha}(S_{cal})$ il quantile empirico di livello $1-\alpha$ della distribuzione di s_i. L'intervallo predittivo per una nuova osservazione x_{n+1} è:

$$C(x_{n+1}) = \left[\hat{f}(x_{n+1}) - q_{1-\alpha}(S_{cal}), \quad \hat{f}(x_{n+1}) + q_{1-\alpha}(S_{cal})\right].$$

Osservazione 8. *La ICP consente un notevole risparmio computazionale rispetto a TCP, poiché il modello viene addestrato una sola volta sul training set e il calibration set serve a stimare la distribuzione dei punteggi. Ciò implica una validità*

marginale garantita e una discreta efficienza degli intervalli, a patto che la divisione addestramento/calibrazione sia rappresentativa.

Esempio 5 (Confronto TCP vs ICP)**.** Consideriamo un *dataset* con $n = 8$ osservazioni. Con TCP, per ogni valore candidato y si ricalcolano i punteggi di non conformità su $n + 1$ punti. Con ICP, dividendo i dati, per esempio, in $m = 5$ per l'addestramento e $n - m = 3$ per la calibrazione, si calcolano i residui una sola volta. A parità di modello, gli intervalli ottenuti con ICP risultano generalmente più ampi, ma il costo computazionale è inferiore.

Teorema 11 (Convergenza ICP a TCP)**.** *Sotto ipotesi di distribuzione continua e dati di calibrazione i.i.d., l'intervallo predittivo ottenuto con ICP converge in probabilità a quello di TCP, al crescere della dimensione dell'insieme di calibrazione:*

$$\lim_{n\to\infty} \mathbb{P}(Y_{n+1} \in C_{ICP}(X_{n+1})) = \mathbb{P}(Y_{n+1} \in C_{TCP}(X_{n+1})).$$

Esempio 6. Consideriamo un *dataset* costituito da 100 osservazioni. Per applicare l'ICP, si suddivide il *dataset* in due sottoinsiemi: 70 osservazioni vengono utilizzate come *training set* per stimare il modello predittivo $\hat{f}(x)$, mentre le restanti 30 osservazioni costituiscono il *calibration set*, impiegato per valutare i punteggi di non conformità. Una volta stimato $\hat{f}$ sui 70 dati di addestramento, si calcolano i residui assoluti tra i valori osservati y_i dell'insieme di calibrazione e le corrispondenti previsioni $\hat{f}(x_i)$, cioè $|y_i - \hat{f}(x_i)|$ per ciascun i del *calibration set*. Da questi residui si ricava il quantile empirico desiderato, ad esempio il 95%-quantile $q_{0.95}$, che rappresenta la soglia di non conformità per costruire l'insieme predittivo. Per una nuova osservazione x_{101}, l'intervallo predittivo conforme sarà quindi

$$[\hat{f}(x_{101}) - q_{0.95}, \hat{f}(x_{101}) + q_{0.95}].$$

Nel caso della TCP, invece, non si effettua alcuna suddivisione tra addestramento e calibrazione: il modello viene stimato ripetutamente per ogni possibile valore candidato della nuova risposta Y_{101}, includendo temporaneamente ciascuna ipotesi nel *dataset* completo, e si calcolano i punteggi di non conformità considerando l'intero insieme di dati. Questo procedimento assicura validità marginale esatta, ma risulta computazionalmente più oneroso rispetto all'ICP.

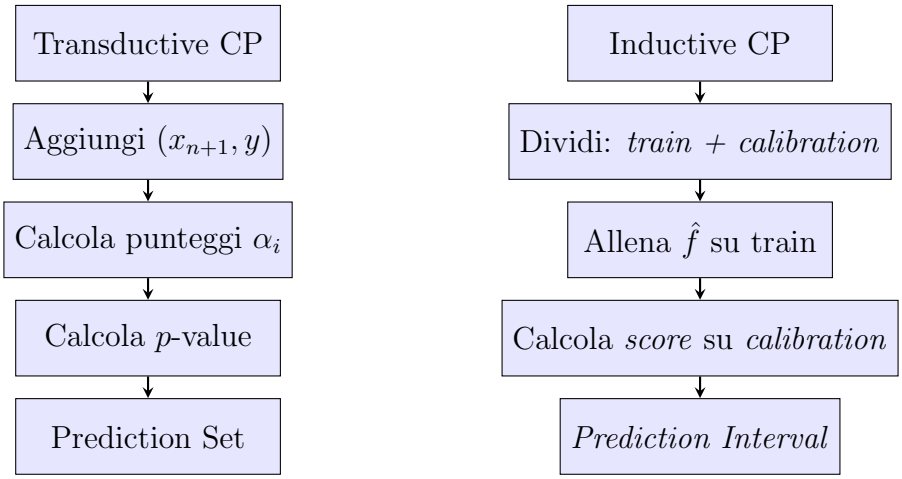


Figura 2.1: Flusso delle procedure TCP e ICP

2.3.3 Mondrian Conformal Prediction

Quando i dati presentano una struttura categoriale o segmentata (ad esempio classificazione multi-classe), è possibile raffinare la validità garantendo la copertura *condizionata* su ciascun sottoinsieme.

Definizione 16 (*Mondrian Conformal Prediction*)**.** *Si definisce una partizione del dominio di input $\mathcal{X}$ in sottoinsiemi disgiunti $\{g_1, \ldots, g_K\}$. Per ciascun gruppo g_k, si calcolano separatamente i punteggi di non conformità e si costruiscono insiemi predittivi $C_k(x)$. La garanzia di validità diventa:*

$$P\big(Y_{n+1} \in C_k(X_{n+1}) \mid X_{n+1} \in g_k\big) \geq 1 - \alpha.$$

Definizione 17 (Validità Condizionata)**.** *Un insieme predittivo $C(X)$ gode di* validità condizionata *rispetto a una variabile G se:*

$$\mathbb{P}(Y \in C(X) \mid G = g) \geq 1 - \alpha, \quad \forall g.$$

Osservazione 9. *La Mondrian CP permette di ottenere validità locale condizionata, migliorando l'accuratezza nei contesti di classificazione o regressione segmentata, ma richiede una partizione adeguata dei dati, che può essere scelta a priori o appresa dai dati stessi.*

Esempio 7. In un problema di classificazione multi-classe con classi A, B, C, si costruiscono tre insiemi predittivi C_A, C_B, C_C calcolando separatamente i punteggi di non conformità per ciascuna classe. Questo consente di ottenere intervalli predittivi

che rispettano la copertura per ogni classe specifica, migliorando la qualità delle predizioni in presenza di eterogeneità.

2.3.4 Cross-Conformal Prediction (CCP)

Per mitigare la dipendenza dalla divisione arbitraria in insieme di addestramento e insieme di calibrazione nella ICP, CCP utilizza una strategia basata sulla validazione incrociata.

Definizione 18 (*Cross-Conformal* Prediction)**.** *Si suddivide il dataset in k fold $\{D_1, \ldots, D_k\}$. Per ogni fold D_j, si usa D_j come insieme di calibrazione e il resto come insieme di addestramento, ottenendo i p-value $p_j(y)$. I p-value vengono quindi aggregati tramite una funzione di combinazione ϕ (ad esempio media o voto conservativo):*

$$p(y) = \phi(p_1(y), \ldots, p_k(y)).$$

L'insieme predittivo finale è:

$$C(x_{n+1}) = \{y \in \mathcal{Y} : p(y) > \alpha\}.$$

Osservazione 10. *La CCP migliora la stabilità dei p-value e riduce la varianza introdotta dalla singola partizione calibrazione/addestramento, garantendo validità approssimata e maggiore efficienza empirica. Tuttavia, la validità teorica è meno rigorosa rispetto a TCP e ICP, essendo approssimata.*

Esempio 8. Con un *dataset* di 100 osservazioni si può applicare una validazione incrociata a 5 *fold*. In ciascuno dei 5 esperimenti, uno dei *fold* funge da insieme di calibrazione e gli altri 4 da insieme di addestramento. Si ottengono 5 p-value $p_j(y)$ per ogni possibile y e si aggregano (ad esempio con la media). L'insieme predittivo finale considera i valori di y per cui la media supera $1 - \alpha$.

2.3.5 Sintesi comparativa

Tabella 2.1: Confronto tra varianti della *Conformal Prediction*

Metodo	**Validità**	**Efficienza**	**Computazione**	**Note**
TCP	Teoricamente garantita	Alta	Costosa	Richiede il ricalcolo completo per ogni possibile valore y, computazionalmente oneroso
ICP	Teoricamente garantita	Moderata	Efficiente	Suddivide *training/calibration*; riduce i calcoli
Mondrian CP	Validità condizionata	Variabile	Moderata	Valido localmente per ogni compartimento di dati
CCP	Validità approssimata	Buona	Moderata	Stabilizza i risultati con k-fold CV

È anche possibile rappresentare le varianti della CP su un piano tridimensionale che evidenzia il compromesso tra le proprietà fondamentali del metodo:

- l'asse X rappresenta l'**efficienza**, ovvero quanto gli intervalli predittivi siano stretti o informativi;
- l'asse Y indica la **validità marginale**, ossia la capacità di garantire copertura conforme al livello desiderato $1 - \alpha$;
- il colore o la profondità della *marker* rappresenta l'**adattabilità**, cioè la flessibilità del metodo di estendersi a contesti complessi o dati non scambiabili

Nella figura 2.2, la TCP si trova in basso a sinistra: offre validità marginale esatta e ragionevole efficienza, ma limitata adattabilità, soprattutto per dati complessi o di grandi dimensioni. L'ICP migliora l'efficienza computazionale senza compromettere troppo la validità, ed è leggermente più adattabile grazie alla separazione addestramento/calibrazione. Le varianti Mondrian CP, che includono partizionamenti basati su caratteristiche sensibili o gruppi di dati, mostrano validità maggiore e maggiore adattabilità, a scapito di una leggera riduzione dell'efficienza.

In questo modo, il grafico sottostante permette di visualizzare intuitivamente il bilanciamento tra le tre proprietà chiave di ciascuna variante della *Conformal Prediction*, facilitando la scelta del metodo più adatto al problema specifico.

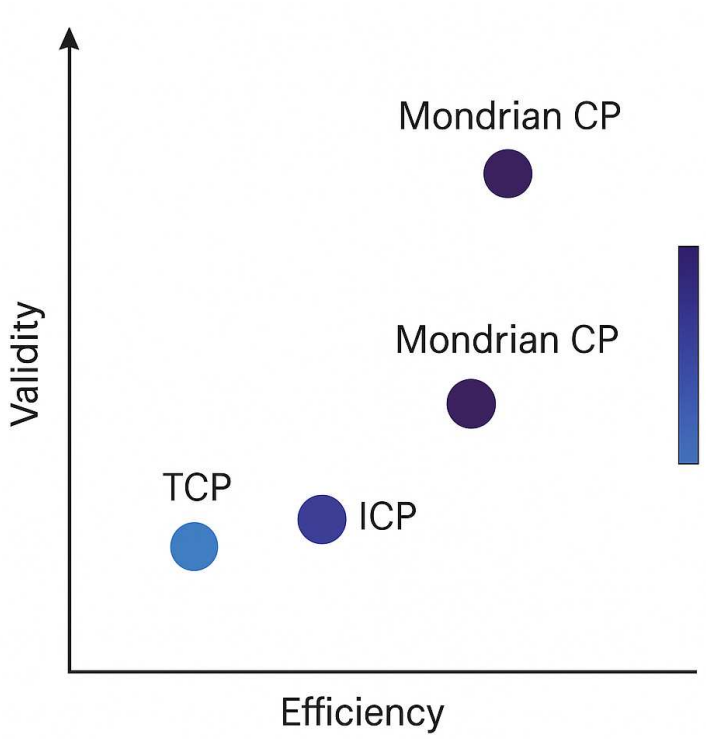


Figura 2.2: Visualizzazione delle principali varianti della *Conformal Prediction* in termini di validità (asse Y), efficienza (asse X) e adattabilità (colore). Elaborazione propria.

2.4 Conformal Prediction per Serie Temporali

2.4.1 Premessa: cosa sono le Serie Temporali

Una **serie temporale** è una sequenza ordinata di osservazioni raccolte in momenti successivi nel tempo, rappresentata come $\{Z_t\}_{t=1}^{T}$, dove Z_t è la variabile osservata al tempo t. Le serie temporali sono presenti in numerosi ambiti, dalla finanza alla meteorologia, dalla medicina all'economia, e sono caratterizzate da una struttura di dipendenza temporale intrinseca.

Definizione 19 (Processo stocastico stazionario)**.** *Una serie temporale* $\{Z_t\}_{t\in\mathbb{Z}}$ *è detta* ***stazionaria in senso stretto*** *se la distribuzione congiunta di qualsiasi insieme finito* $(Z_{t_1}, \ldots, Z_{t_k})$ *coincide con quella di* $(Z_{t_1+h}, \ldots, Z_{t_k+h})$ *per ogni intero*

h, cioè la distribuzione è invariante per traslazioni temporali:

$$\forall k, \forall t_1, \ldots, t_k, \forall h, \quad (Z_{t_1}, \ldots, Z_{t_k}) \stackrel{d}{=} (Z_{t_1+h}, \ldots, Z_{t_k+h}).$$

Si dice ***stazionaria in senso debole*** *(o stazionaria di secondo ordine) se la media è costante nel tempo e la funzione di autocovarianza dipende solo dalla distanza temporale h:*

$$E[Z_t] = \mu, \quad Var(Z_t) = \sigma^2, \quad Cov(Z_t, Z_{t+h}) = \gamma(h).$$

Osservazione 11. *La stazionarietà è una proprietà cruciale per molti metodi di analisi delle serie temporali, poiché consente di stabilire inferenze e previsioni basate sulla ripetitività del comportamento statistico nel tempo. Tuttavia, molte serie reali non sono stazionarie, richiedendo tecniche di adattamento.*

Definizione 20. *Un processo stocastico $\{Z_t\}$ è detto β-mixing se esiste una funzione $\beta(h)$ tale che:*

$$\beta(h) = \sup_{A \in \mathcal{F}_{-\infty}^{t}, B \in \mathcal{F}_{t+h}^{\infty}} |P(A \cap B) - P(A)P(B)| \to 0 \quad \textit{per } h \to \infty.$$

Questa proprietà esprime la decrescente dipendenza tra eventi distanti nel tempo.

Esempio 9. Consideriamo un processo autoregressivo di primo ordine (AR(1)) definito come

$$Z_t = \phi Z_{t-1} + \varepsilon_t, \quad \varepsilon_t \sim \text{i.i.d. } \mathcal{N}(0, \sigma^2),$$

con $|\phi| < 1$.

Questo processo è **stazionario in senso debole**, perché la media è costante nel tempo $E[Z_t] = 0$ e la covarianza dipende solo dal lag h $\text{Cov}(Z_t, Z_{t+h}) = \sigma^2 \frac{\phi^{|h|}}{1-\phi^2}$.

Inoltre, l'AR(1) è un esempio di processo β***-mixing*** dato che, intuitivamente, la dipendenza tra Z_t e Z_{t+h} decresce esponenzialmente al crescere del lag h:

$$\text{Cov}(Z_t, Z_{t+h}) = \sigma^2 \frac{\phi^h}{1-\phi^2} \quad \Longrightarrow \quad |\text{Cov}(Z_t, Z_{t+h})| \to 0 \text{ per } h \to \infty.$$

Formalmente, si può dimostrare che la funzione $\beta(h)$ soddisfa

$$\beta(h) \leq C\phi^h \to 0 \quad \text{per } h \to \infty,$$

per qualche costante $C > 0$. Questo significa che, sebbene vi sia dipendenza a breve termine, il processo "perde memoria" a distanza di molti passi temporali. Tale

proprietà è cruciale per l'applicazione della *conformal prediction* alle serie temporali, perché la dipendenza decrescente consente di trattare i residui o le statistiche scelte come quasi indipendenti su finestre sufficientemente distanziate, garantendo la validità della procedura ([40], [7]).

2.4.2 Problemi nell'applicazione della CP alle serie temporali

L'applicazione diretta della *Conformal Prediction* alle serie temporali è ostacolata da alcune caratteristiche peculiari:

- **Ordine temporale informativo:** L'ipotesi di scambiabilità, fondamentale nella CP classica, è violata poiché l'ordine delle osservazioni contiene informazioni rilevanti e non può essere scambiato arbitrariamente.
- **Dipendenza seriale:** Le osservazioni sono correlate nel tempo; i dati non sono indipendenti e identicamente distribuiti (i.i.d.).
- **Non stazionarietà:** La distribuzione marginale può cambiare nel tempo, ad esempio per trend o cambiamenti strutturali.

2.4.3 Metodi di adattamento

Per affrontare queste criticità sono state proposte diverse estensioni della CP specifiche per dati serialmente dipendenti.

Sliding Window CP

L'idea è di usare una finestra mobile temporale di lunghezza w per definire l'insieme di calibrazione:

$$\mathcal{D}_{\text{cal}}(t) = \{(x_{t-w}, y_{t-w}), \ldots, (x_{t-1}, y_{t-1})\}.$$

In tal modo, la CP viene applicata localmente nel tempo, assumendo che la dipendenza e la distribuzione siano stabili nella finestra.

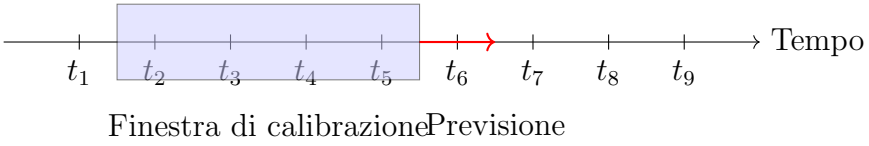


Figura 2.3: *Sliding Window* per *conformal prediction* in serie temporali

Osservazione 12 (Bias-Varianza nel Metodo *Sliding Window*)**.** *La scelta della dimensione della finestra w implica un compromesso critico: finestre troppo corte contengono pochi dati quasi indipendenti (in termini di β-mixing), aumentando la varianza delle stime di conformal prediction; finestre troppo lunghe possono includere osservazioni in cui la distribuzione locale differisce a causa di non stazionarietà, aumentando la distorsione (bias). In altre parole, w deve essere sufficientemente grande da garantire una calibrazione stabile ma non così grande da violare l'ipotesi di stazionarietà locale necessaria per la validità della CP ([40], [7]).*

Esempio 10. In un modello finanziario con dati giornalieri, si può scegliere $w = 30$ giorni per calibrare il modello e prevedere il prezzo del giorno successivo, aggiornando la finestra giorno per giorno.

Weighted CP

Per dare maggior peso alle osservazioni più recenti, si introducono pesi decrescenti w_i (ad esempio tramite kernel esponenziale):

$$p(y) = \frac{\sum_{i=1}^{n} w_i \cdot \mathbb{I}(\alpha_i \geq \alpha_{n+1})}{\sum_{i=1}^{n} w_i + 1}.$$

Osservazione 13. *Questa formulazione consente di adattarsi a situazioni non stazionarie e di gestire meglio i cambiamenti di regime.*

Esempio 11. In ambito meteorologico, si possono pesare maggiormente le ultime osservazioni, per prevedere la temperatura del giorno successivo tenendo conto di trend stagionali o cambiamenti improvvisi.

Time Series Split CP

La divisione *training/calibration* rispetta la causalità temporale, evitando *leakage* informativi:

$$\text{Training} = \{1, \ldots, t_0\}, \quad \text{Calibration} = \{t_0 + 1, \ldots, t_1\}.$$

Si procede con una validazione *walk-forward*, spostando progressivamente la finestra di calibrazione.

Osservazione 14. *Rispetto alla divisione casuale, questa strategia evita di mescolare dati futuri nelle fasi di training, mantenendo la validità sotto dipendenza debole.*

Esempio 12. Per prevedere la domanda elettrica, si usa come *training* il primo semestre dell'anno e la calibrazione il secondo, aggiornando il modello man mano che si accumulano nuovi dati.

Conformalized Quantile Regression (CQR)

CQR combina la regressione quantilica con la CP, costruendo intervalli predittivi flessibili che catturano asimmetrie nella distribuzione condizionale di Y data X.

Definizione 21 (CQR)**.** *Si stimano quantili condizionali $\hat{q}_{\alpha/2}(x)$ e $\hat{q}_{1-\alpha/2}(x)$ tramite regressione quantilica, e si applica CP sui residui di calibrazione per aggiustare gli intervalli:*

$$C(x) = \left[\hat{q}_{\alpha/2}(x) - Q_{1-\alpha}(S), \quad \hat{q}_{1-\alpha/2}(x) + Q_{1-\alpha}(S)\right].$$

Osservazione 15. *CQR è particolarmente efficace per serie temporali con distribuzioni condizionali non simmetriche e variabilità eteroschedastica.*

Esempio 13. Per una serie temporale di consumi energetici, CQR può fornire intervalli predittivi che riflettono variazioni stagionali e picchi di consumo, più accurati di quelli simmetrici.

2.4.4 Validità sotto dipendenza debole

Teorema 12 (Validità asintotica sotto dipendenza [7])**.** *Sia $\{Z_t\}$ una serie temporale stazionaria β-mixing con coefficiente $\beta(h) \to 0$ per $h \to \infty$. Allora, la procedura di Conformal Prediction (o CQR) con finestra di calibrazione crescente $w_n \to \infty$ soddisfa:*

$$\liminf_{n\to\infty} P(Y_{n+1} \in C(X_{n+1})) \geq 1 - \alpha.$$

In altre parole, la copertura predittiva asintotica è garantita anche in presenza di dipendenza seriale debole.

Osservazione 16. *La condizione β-mixing è una misura di dipendenza debole, più generale della indipendenza. Questo risultato garantisce che, anche in presenza di dipendenza temporale, la CP conserva la validità in senso asintotico.*

Esempio 14. Considerando dati economici giornalieri con autocorrelazione decrescente nel tempo, si può applicare *Sliding Window CP* o *Weighted CP* con validità asintotica.

Tabella 2.2: Confronto tra metodi conformal per serie temporali

Metodo	**Validità**	**Adattabilità**	**Note**
Sliding Window CP	Validità locale	Buona per dati stazionari	Richiede una scelta adeguata della finestra; utile per strutture locali stabili
Weighted CP	Approx. validità	Alta	Maggior peso agli elementi recenti tramite kernel temporali; utile in contesti dinamici
CQR	Asintoticamente valida	Alta	Combina regressione quantilica e *conformal prediction*; gestisce distribuzioni asimmetriche
Time Series Split	Validità debole	Alta	Rispettosa della causalità; adatta a scenari con forte dipendenza temporale

L'efficacia delle varianti di *Conformal Prediction* applicate alle serie temporali mostra la notevole flessibilità del paradigma, capace di adattarsi a differenti strutture di dipendenza e dinamiche di evoluzione dei dati. Tuttavia, l'importanza della CP non si esaurisce nel contesto delle serie storiche. Negli ultimi anni, infatti, si è affermata una linea di ricerca volta a integrare la *Conformal Prediction* con i modelli di *Machine Learning*, con l'obiettivo di dotare gli algoritmi predittivi moderni di una componente inferenziale formalmente giustificata.

In tale prospettiva, la CP si configura come un ponte tra l'inferenza statistica classica e l'apprendimento automatico: essa fornisce un quadro metodologico generale per la quantificazione dell'incertezza, applicabile a qualunque modello supervisionato, senza richiedere ipotesi parametriche sulla distribuzione dei dati. Nel seguito viene presentata una panoramica sull'integrazione della *Conformal Prediction* nei modelli di *Machine Learning*, evidenziandone i principi teorici, le modalità operative e i vantaggi in termini di validità e affidabilità predittiva.

2.5 Conformal Prediction e Machine Learning

La crescente diffusione di algoritmi di apprendimento automatico (*Machine Learning*, ML) ha reso cruciale la possibilità di quantificare l'incertezza associata alle previsioni. Sebbene i modelli di ML raggiungano livelli elevati di accuratezza, essi

producono tipicamente predizioni puntuali o probabilità non calibrate, prive di garanzie statistiche formali. La *Conformal Prediction* (CP) fornisce un quadro teorico per arricchire tali modelli con intervalli predittivi o insiemi di classificazione che rispettano un vincolo di copertura frequentista finita, indipendente dalla struttura del modello utilizzato.

2.5.1 Machine Learning classico e limiti inferenziali

I modelli di *Machine Learning* classici, quali regressione lineare, Support Vector Machines (SVM), alberi decisionali, foreste casuali (*Random Forests*) e reti neurali, apprendono relazioni complesse tra variabili di input e output basandosi su procedure di ottimizzazione del rischio empirico. Tali modelli generano predizioni puntuali

$$\hat{y} = f(x),$$

ma non forniscono, in generale, un meccanismo per quantificare l'incertezza associata a ciascuna previsione.

Definizione 22 (Predizione puntuale)**.** *Dato un insieme di dati* $\{(x_i, y_i)\}_{i=1}^n$ *e un algoritmo di apprendimento* $f : \mathcal{X} \to \mathcal{Y}$*, una previsione puntuale è la stima* $\hat{y} = f(x_{n+1})$ *della risposta associata a un nuovo punto* x_{n+1}*. Essa rappresenta una singola ipotesi sul valore futuro di* Y_{n+1}*, priva di informazione sulla sua variabilità.*

In assenza di un modello probabilistico esplicito, le probabilità prodotte da classificatori discriminativi (ad esempio la funzione softmax nelle reti neurali) non sono necessariamente calibrate. Ciò implica che un modello che dichiara una confidenza del 90% non garantisce che nove previsioni su dieci siano effettivamente corrette.

Osservazione 17. *La mancanza di calibrazione probabilistica costituisce un limite cruciale per l'affidabilità dei modelli di ML, in particolare in contesti sensibili (sanità, finanza, decisioni automatizzate). La Conformal Prediction risponde a tale esigenza, fornendo garanzie di copertura indipendenti dall'algoritmo sottostante.*

2.5.2 Integrazione della Conformal Prediction nei modelli di Machine Learning

La CP può essere integrata in qualunque algoritmo di apprendimento supervisionato. L'idea di base consiste nel misurare la *non conformità* di una nuova

osservazione rispetto ai dati di calibrazione, utilizzando un punteggio costruito a partire dal modello predittivo scelto.

Sia $f : \mathcal{X} \to \mathcal{Y}$ il predittore addestrato su un insieme di addestramento D_{train}. Su un insieme di calibrazione $D_{\text{cal}} = \{(x_i, y_i)\}_{i=m+1}^{n}$ si calcolano i punteggi di non conformità

$$s_i = A((x_i, y_i)) = |y_i - \hat{f}(x_i)|, \quad (x_i, y_i) \in D_{\text{cal}}.$$

Sia $q_{1-\alpha}$ il quantile empirico di livello $1 - \alpha$ della distribuzione $\{s_i\}$.

Teorema 13 (Intervallo predittivo *Inductive Conformal Prediction*)**.** *Sotto l'ipotesi di scambiabilità (exchangeability) delle osservazioni, l'intervallo*

$$C(x_{n+1}) = \left[\hat{f}(x_{n+1}) - q_{1-\alpha},\ \hat{f}(x_{n+1}) + q_{1-\alpha}\right]$$

soddisfa

$$P(Y_{n+1} \in C(X_{n+1})) \geq 1 - \alpha.$$

Questo risultato mostra che la CP fornisce una garanzia di copertura *distribution-free*, indipendente dal modello predittivo. La qualità (ovvero l'ampiezza media) dell'intervallo dipende dall'accuratezza del modello base f: modelli più precisi producono intervalli più stretti, migliorando l'efficienza [7].

Osservazione 18 (Applicazione alle serie storiche)**.** *Quando le osservazioni non sono scambiabili, come nelle serie storiche, il teorema si applica sostituendo la scambiabilità con una condizione di β-mixing: se la serie temporale $\{Z_t\}$ è stazionaria e β-mixing con $\beta(h) \to 0$ per $h \to \infty$, allora la procedura di Conformal Prediction con finestra di calibrazione crescente rimane asintoticamente valida ([40] , [7]). In altre parole, la dipendenza seriale debole non compromette la copertura predittiva, purché le finestre siano sufficientemente lunghe da stimare correttamente i quantili dei punteggi di non conformità.*

Osservazione 19. *La CP agisce come uno strato inferenziale aggiuntivo, trasformando un predittore deterministico in un metodo che fornisce non solo una stima puntuale, ma anche una misura d'incertezza calibrata e interpretabile.*

Esempio 15. Nel caso di una regressione lineare $f(x) = \beta_0 + \beta^\top x$, la CP costruisce intervalli simmetrici centrati su $\hat{f}(x_{n+1})$. Per modelli non lineari come le *Random*

Forests, si possono utilizzare i residui *out-of-bag* come punteggi di non conformità, ottenendo intervalli non parametrici e adattivi. In classificazione, invece, si impiega spesso lo *score* $A(x, y) = 1-\hat{p}(y|x)$, dove $\hat{p}(y|x)$ è la probabilità predetta dal modello.

La tabella seguente riassume alcune implementazioni tipiche [1].

Tabella 2.3: Esempi di integrazione tra modelli di ML e *Conformal Prediction*

Modello ML	***Nonconformity score***	**Tipo di intervallo o set predittivo**
Regressione lineare	$\|y - \hat{y}\|$	Intervallo simmetrico centrato su $\hat{y}$
Random Forest	Errore *out-of-bag*	Intervallo non parametrico e adattivo
SVM	Distanza dal margine	Set di classificazione con confidenza frequentista
Reti neurali	$1 - \hat{p}(y\|x)$	*Class set* calibrato sul *calibration set*

2.5.3 Confronto tra modelli classici e Conformal ML

L'integrazione della *Conformal Prediction* nei modelli di *Machine Learning* produce un cambiamento concettuale significativo: da predittori puntuali a predittori con copertura garantita.

Definizione 23 (Validità e efficienza nel contesto ML)**.** *Un predittore conforme $C(x)$ è valido se*

$$P(Y \in C(X)) \geq 1 - \alpha,$$

ed è efficiente se l'ampiezza media $E[dim(C(X))]$ è minima tra tutti gli insiemi predittivi che rispettano la validità.

La validità fornisce la garanzia di copertura frequentista, mentre l'efficienza dipende dall'accuratezza del modello sottostante. In particolare, modelli più performanti determinano punteggi di non conformità più concentrati e quindi intervalli predittivi più stretti.

Teorema 14 (Trade-off validità–efficienza)**.** *Dato un livello di significatività $\alpha \in (0, 1)$, non esiste un insieme predittivo $C(x)$ che massimizzi simultaneamente validità ed efficienza. L'aumento della copertura (riduzione di α) comporta un incremento dell'ampiezza media degli intervalli ([1], [2]).*

Osservazione 20. *Questo principio di compromesso è analogo al bias–variance trade-off dei modelli predittivi: garantire copertura elevata richiede intervalli più ampi, mentre l'efficienza implica accettare una minore confidenza.*

La seguente tabella sintetizza le differenze principali tra modelli di ML tradizionali e la loro estensione conforme.

Tabella 2.4: Confronto tra modelli di ML classici e Conformal ML

Proprietà	**ML classico**	**ML + *Conformal Prediction***
Predizione puntuale	✓	✓
Intervallo predittivo	×	✓
Garanzia di copertura frequentista	×	✓
Calibrazione probabilistica	Parziale	Esplicita
Dipendenza dal modello	Alta	Bassa (model-agnostic)
Validità in campioni finiti	×	✓

2.6 Conclusioni

La metodologia della *Conformal Prediction* rappresenta oggi uno strumento di grande rilevanza nell'ambito dell'inferenza statistica predittiva, grazie alle sue proprietà uniche di validità finita e non parametricità. Essa consente di costruire insiemi predittivi dotati di garanzie rigorose di copertura, indipendentemente dalla distribuzione sottostante dei dati, ponendo un forte vincolo soltanto sulla scambiabilità o, in contesti più complessi, su forme più deboli di dipendenza.

L'adattabilità della CP emerge come un punto di forza fondamentale: attraverso numerose varianti metodologiche quali la *Transductive*, *Inductive*, *Mondrian* e *Cross-Conformal Prediction*, il paradigma si adatta efficacemente a differenti esigenze computazionali e strutture dei dati. Ciò permette un equilibrio ottimale tra efficienza computazionale e rigore inferenziale, rendendo la CP utilizzabile in numerosi ambiti applicativi, dalla classificazione alla regressione, fino ai dati eterogenei e stratificati.

Particolarmente significativa è l'estensione della CP alle *serie temporali*, dove la presenza di dipendenze temporali, non stazionarietà e struttura sequenziale pongono sfide rilevanti. Le soluzioni proposte, come il *Sliding Window*, il *Weighted CP*, il *Time Series Split* e la *Conformalized Quantile Regression*, offrono strumenti flessibili per preservare le garanzie di copertura anche in questi contesti, integrando

conoscenze di dipendenza e causalità temporale. Tali sviluppi aprono nuove prospettive per applicazioni in ambito economico-finanziario, meteorologico, biomedico e in molti altri settori caratterizzati da dati longitudinali e temporali.

Un ulteriore ambito di applicazione in rapida crescita è quello del *Machine Learning*, in cui la *Conformal Prediction* fornisce un quadro metodologico per la quantificazione dell'incertezza nei modelli predittivi complessi. Integrata con algoritmi supervisionati quali regressione lineare, foreste casuali, *Support Vector Machine* e reti neurali, la CP consente di trasformare predittori puntuali in modelli capaci di fornire intervalli e set di classificazione con garanzie frequentiste di copertura. Essa rappresenta un approccio distribuzione-libera, dove non sono necessarie assunzioni sulla distribuzione dei dati, che colma il divario tra accuratezza predittiva e affidabilità statistica, permettendo di costruire sistemi di apprendimento automatico calibrati, interpretabili e teoricamente giustificati.

Inoltre, la natura modulare della CP, che si basa sull'uso di *nonconformity scores* e su procedure di calibrazione, la rende facilmente combinabile con modelli predittivi moderni, inclusi quelli di *machine learning* e *deep learning*. Questo favorisce l'adozione della CP come quadro metodologico per la quantificazione dell'incertezza predittiva in scenari complessi e ad alta dimensionalità, temi di crescente importanza nella statistica moderna e data science.

Infine, rimangono aperti diversi filoni di ricerca: l'estensione della CP a forme di dipendenza più complesse, l'integrazione con modelli dinamici adattativi, e lo sviluppo di metodologie efficienti per grandi *dataset* e flussi in tempo reale. La capacità della CP di fornire garanzie esplicite e interpretabili la rende inoltre un candidato ideale per applicazioni in ambiti regolamentati e critici, dove l'affidabilità delle predizioni è essenziale.

In sintesi, la *Conformal Prediction* si conferma come un paradigma teoricamente solido e praticamente versatile, capace di integrare la robustezza dell'inferenza statistica con la potenza del *Machine Learning*, ponendosi come strumento chiave per lo sviluppo di sistemi predittivi affidabili, trasparenti e statisticamente validati.

Riflessioni sugli sviluppi futuri

Il campo della *Conformal Prediction* è in rapida evoluzione e presenta numerose direzioni promettenti per la ricerca futura. Innanzitutto, un'area di grande interesse

riguarda l'estensione delle garanzie di validità in presenza di dipendenze complesse e strutture di dati non scambiabili, quali quelle tipiche di molte applicazioni reali. Lo sviluppo di teorie più generali che possano affrontare modelli con dipendenza a lungo termine, cambiamenti di regime, o dati non stazionari rappresenta una sfida aperta e fondamentale.

Inoltre, la crescente integrazione della CP con tecniche di *machine learning* e intelligenza artificiale apre la strada a nuovi modelli ibridi, in cui la robustezza e interpretabilità della CP si combinano con la potenza predittiva dei modelli non lineari e ad alta dimensionalità. La ricerca si sta concentrando su metodi efficienti per gestire grandi volumi di dati e scenari dinamici, inclusi i flussi dati in tempo reale (*streaming*), con l'obiettivo di mantenere garanzie di copertura rigorose in ambienti computazionalmente sfidanti.

Un altro filone importante riguarda l'applicazione della CP a problemi di causal inference e inferenza contrafattuale, ampliando il suo ruolo oltre la semplice previsione verso una più profonda comprensione dei meccanismi generativi dei dati. Ciò potrebbe portare a metodologie che non solo predicono con affidabilità, ma che supportano anche decisioni basate su effetti causali stimati con incertezza quantificata.

Infine, la crescente attenzione verso l'interpretabilità e la trasparenza dei modelli predittivi, soprattutto in ambiti regolamentati come la medicina, la finanza e la giustizia, rende la CP una candidata naturale come quadro metodologico per fornire intervalli di previsione espliciti e controllati, facilitando così l'adozione pratica di modelli predittivi affidabili e verificabili.

Questi sviluppi, uniti alla flessibilità intrinseca della CP, la pongono al centro di un dialogo multidisciplinare tra statistica, *machine learning*, teoria dei processi stocastici e applicazioni reali, rendendo la *Conformal Prediction* un ambito fertile e stimolante per la ricerca futura.

Capitolo 3

Applicazioni economiche

La *Conformal Prediction* (CP) sta acquisendo un ruolo sempre più rilevante nell'economia quantitativa e nella finanza computazionale, grazie alla capacità di fornire intervalli di previsione rigorosi e *distribution-free*. L'affidabilità statistica delle stime è cruciale in contesti decisionali ad alto impatto, quali la formulazione di politiche monetarie, la valutazione del rischio di credito, l'allocazione ottimale di portafoglio e la determinazione di strategie di *pricing* dinamico.

A differenza dei metodi tradizionali, che si basano su ipotesi parametriche spesso irrealistiche, la CP garantisce copertura frequentista anche in campioni finiti e in presenza di dati complessi, eterogenei o rumorosi. Ciò la rende un approccio particolarmente adatto per applicazioni economiche e di *Business Intelligence*, dove la quantificazione dell'incertezza è tanto importante quanto la previsione puntuale.

In questo capitolo vengono discusse quattro aree applicative principali: previsioni macroeconomiche, *credit scoring* e rischio di credito, analisi dei mercati finanziari e *asset pricing*, analisi predittiva per strategie di *pricing* dinamico.

3.1 Previsioni macroeconomiche

Nei capitoli teorici precedenti, la *Conformal Prediction* (CP) è stata sviluppata sotto ipotesi di stazionarietà e β-mixing, condizioni che garantiscono la validità asintotica degli intervalli predittivi in presenza di dipendenza seriale debole. Tuttavia, molte serie macroeconomiche reali, come PIL e inflazione, presentano trend, cambi di regime o shock esogeni, ossia fenomeni di non stazionarietà globale. Per affrontare questa complessità, la CP può essere adattata mediante strategie come lo schema *sliding window* o pesature decrescenti sui dati storici, che sfruttano la

stazionarietà locale o la dipendenza a breve termine. Questi approcci permettono di mantenere copertura predittiva robusta anche in contesti non stazionari, adattandosi dinamicamente alle variazioni della distribuzione dei dati nel tempo. In generale, le variabili macroeconomiche presentano strutture di dipendenza temporale che rendono complessa la costruzione di intervalli predittivi affidabili. L'applicazione della CP alle serie temporali offre una soluzione flessibile e non parametrica, capace di stimare intervalli di previsione anche in presenza di shock economici imprevedibili ([40],[31], [45], [15]).

Definizione 24. *Sia* $\{Y_t\}_{t=1}^{T}$ *una serie economica (ad esempio PIL o inflazione trimestrale). Un intervallo conforme al livello* $1-\alpha$ *per* Y_{T+1} *è definito come*

$$C_{T+1} = \{y \in \mathbb{R} : p(y) > \alpha\},$$

dove $p(y)$ *è il p-value conforme costruito a partire da un punteggio di non conformità* α_i *calcolato sui residui predittivi [22].*

Teorema 15. *Se* $\{Y_t\}$ *è una sequenza* β*-mixing, e si applica la CP con schema sliding window di lunghezza* w*, allora per ogni* $\alpha \in (0,1)$*:*

$$\lim_{T\to\infty} \mathbb{P}(Y_{T+1} \in C_{T+1}) \geq 1-\alpha.$$

[22]

Applicazioni tipiche includono previsioni di PIL, Inflazione e tassi d'interesse.

3.1.1 Previsioni di PIL

Il Prodotto Interno Lordo (PIL) è il valore monetario complessivo di tutti i beni e servizi finali prodotti all'interno dei confini economici di un Paese in un determinato periodo di tempo (solitamente un trimestre o un anno), al netto dei consumi intermedi [22].

In forma semplificata, si può esprimere come:

$$PIL = C + I + G + (X - M),$$

dove

- C: consumi delle famiglie,

- I: investimenti lordi,
- G: spesa pubblica,
- X: esportazioni,
- M: importazioni.

Esso è quindi un indicatore aggregato di *attività economica interna*, distinto dal Reddito Nazionale Lordo (RNL), che invece considera anche i redditi netti da/per l'estero.

Esistono diversi tipi di PIL:

- **PIL nominale**: valuta la produzione di beni e servizi ai prezzi correnti del periodo considerato. È influenzato sia dalle variazioni reali della produzione sia da quelle dei prezzi (inflazione/deflazione).
- **PIL reale**: misura la produzione di beni e servizi valutata a prezzi costanti di un anno base, eliminando così l'effetto dell'inflazione. È più indicativo dell'andamento reale dell'economia.
- **PIL pro capite**: rapporto tra PIL (nominale o reale) e popolazione residente; fornisce una misura del reddito medio per abitante.
- **PIL potenziale**: stima del livello massimo di output che un'economia può sostenere senza generare pressioni inflazionistiche, data la tecnologia e i fattori produttivi disponibili.
- **PIL a parità di potere d'acquisto (PPP)**: converte i PIL nazionali in una stessa unità valutaria correggendo per le differenze nei livelli di prezzo tra Paesi, permettendo confronti internazionali più omogenei [22].

La ***Conformal Prediction*** è un quadro statistico per costruire *intervalli predittivi validi* che mantengono un livello di copertura predefinito (ad esempio 95%), indipendentemente dalla distribuzione dei dati. È particolarmente utile quando i modelli predittivi sono complessi o "black box" (es. reti neurali, gradient boosting).

Nell'ambito del PIL e delle sue componenti, la CP può essere applicata in diversi modi:

- **Previsioni macroeconomiche**: costruzione di intervalli di previsione per il PIL reale o nominale (ad esempio trimestrale), a partire da modelli di serie storiche (ARIMA, VAR, modelli bayesiani) o di *machine learning*. Data una previsione puntuale, la CP produce un intervallo $[L, U]$ che garantisce una copertura nominale del 95%, più robusto rispetto agli intervalli classici basati su assunzioni parametriche.

- **Nowcasting**: stima del PIL corrente (non ancora pubblicato) usando indicatori ad alta frequenza. La CP può quantificare l'incertezza associata alle stime in tempo reale.

- **Sotto-componenti del PIL**: consumi, investimenti, esportazioni. La CP può fornire intervalli predittivi simultanei mantenendo il controllo del livello di copertura anche in presenza di eterogeneità tra le componenti.

In sintesi, mentre il PIL è un indicatore contabile, la CP fornisce strumenti statistici per predirne l'evoluzione futura con intervalli di confidenza calibrati, utili per policy maker, banche centrali e analisti economici. Inoltre rispetto agli intervalli di previsione tradizionali basati su ARIMA o modelli bayesiani, che spesso risultano troppo ottimistici in presenza di shock inattesi, la CP fornisce garanzie di copertura anche in campioni ridotti. Questa caratteristica la rende particolarmente adatta all'analisi di scenario per istituzioni come la BCE o l'FMI, dove la quantificazione dell'incertezza è cruciale per il processo decisionale ([15], [22]).

3.1.2 Inflazione

L'**inflazione** è definita come la variazione percentuale del livello generale dei prezzi di un paniere rappresentativo di beni e servizi in un determinato periodo di tempo. In formula, se P_t indica l'indice dei prezzi (ad esempio l'Indice dei Prezzi al Consumo, IPC), il tasso di inflazione al tempo t è dato da:

$$\pi_t = \frac{P_t - P_{t-1}}{P_{t-1}} \times 100.$$

L'inflazione misura dunque la perdita di potere d'acquisto della moneta ed è uno degli indicatori macroeconomici fondamentali per la definizione delle politiche monetarie e fiscali [22].

Tipi di inflazione

- **Inflazione da domanda (demand-pull)**: si verifica quando la domanda aggregata supera la capacità produttiva dell'economia, generando pressioni sui prezzi.
- **Inflazione da costi (cost-push)**: deriva da un aumento dei costi di produzione (materie prime, salari), che si riflette sui prezzi finali.
- **Inflazione importata**: causata da un aumento dei prezzi dei beni importati (es. energia).
- **Inflazione attesa**: legata alle aspettative degli operatori economici, che possono innescare meccanismi auto-rinforzanti.
- **Deflazione**: variazione negativa del livello dei prezzi ($\pi_t < 0$), associata a stagnazione economica.
- **Disinflazione**: riduzione del tasso di inflazione, pur rimanendo positivo [22].

La ***Conformal Prediction*** **(CP)** può essere utilizzata per costruire intervalli predittivi dell'inflazione, un ambito particolarmente complesso per via della non stazionarietà dei dati e della presenza di shock esogeni.

- **Previsioni a breve termine**: la CP permette di costruire intervalli robusti per le stime mensili o trimestrali dell'inflazione, utilizzando modelli di serie temporali (ARIMA, GARCH, VAR) o modelli di *machine learning*.
- ***Nowcasting*** **dell'inflazione**: con dati ad alta frequenza (prezzi energetici, indici delle materie prime), la CP fornisce intervalli calibrati che quantificano l'incertezza delle stime "in tempo reale".
- **Analisi delle componenti**: la CP può essere applicata alle diverse componenti dell'indice dei prezzi (alimentari, energetici, core inflation), producendo intervalli simultanei che rispettano il livello di copertura prefissato.

Quindi l'inflazione rappresenta una variabile chiave per la stabilità macroeconomica e la CP offre un approccio non parametrico e robusto per stimarne l'evoluzione, con intervalli predittivi che mantengono la copertura frequentista desiderata anche in contesti di alta incertezza. Nei modelli tradizionali, la previsione dell'inflazione

è spesso condizionata a ipotesi di stazionarietà difficili da verificare, mentre la CP consente di costruire intervalli validi anche in presenza di cambi di regime e shock esogeni. In prospettiva, un'integrazione con tecniche di nowcasting basate su big data (es. Google Trends o indicatori energetici in tempo reale) potrebbe potenziare ulteriormente l'affidabilità delle previsioni.

3.1.3 Tassi di interesse

Il **tasso di interesse** è la percentuale che misura il costo del denaro preso in prestito o, simmetricamente, il rendimento del capitale prestato o investito in un dato intervallo temporale. Formalmente, esso rappresenta il rapporto tra l'interesse I maturato in un periodo e il capitale iniziale C_0, ovvero:

$$i = \frac{I}{C_0}$$

[22]. I tassi di interesse possono assumere diverse configurazioni, tra cui le principali sono:

- **Tasso nominale**: indica il rendimento o il costo del capitale espresso in termini monetari correnti, senza considerare gli effetti dell'inflazione. È il tasso generalmente riportato nei contratti finanziari.

- **Tasso reale**: misura il rendimento effettivo del capitale, tenendo conto della variazione del potere d'acquisto della moneta. È approssimativamente legato al tasso nominale i_n e al tasso di inflazione π dalla relazione:

$$i_r \approx i_n - \pi.$$

- **Tasso effettivo (o composto)**: tiene conto della capitalizzazione degli interessi in periodi infrannuali. Se il tasso nominale annuo è i_n e la capitalizzazione avviene m volte l'anno, il tasso effettivo annuo è dato da:

$$i_e = \left(1 + \frac{i_n}{m}\right)^m - 1.$$

- **Tasso di mercato**: quello determinato dall'incontro tra domanda e offerta di fondi prestabili sul mercato dei capitali [22].

La **contabilità e statistica** dei tassi di interesse riveste un ruolo fondamentale nella macroeconomia e nella finanza. In particolare:

- I tassi vengono utilizzati come variabile chiave nelle analisi delle politiche monetarie, poiché influenzano consumi, investimenti e, indirettamente, la crescita del PIL.
- Indicatori statistici come la media, la varianza e la distribuzione dei tassi sono utilizzati per valutare la volatilità dei mercati finanziari e il rischio associato agli investimenti.
- Attraverso serie storiche dei tassi di interesse, si studiano le dinamiche temporali e le relazioni con altre grandezze economiche, come inflazione e occupazione.

I tassi di interesse rappresentano una variabile macro-finanziaria cruciale, influenzata sia da decisioni di politica monetaria delle banche centrali sia dalle dinamiche di mercato e dagli shock esogeni (geopolitici, finanziari, ecc.). La loro previsione puntuale è complessa, poiché essi riflettono un equilibrio dinamico e instabile tra aspettative, liquidità e rischio.

Per questo motivo, in ambito economico-finanziario, non è sufficiente stimare un singolo valore atteso del tasso, ma è fondamentale costruire *intervalli predittivi affidabili*, che possano esprimere l'incertezza associata alle proiezioni. Ad esempio, in scenari di analisi di rischio, si può essere interessati ad affermazioni del tipo: *con il 95% di affidabilità, il tasso a 3 mesi si collocherà tra il 3,2% e il 3,8%.*

La Conformal Prediction (CP) si rivela particolarmente utile in questo contesto per tre motivi principali:

1. non richiede assunzioni parametriche specifiche sulla distribuzione dei tassi di interesse, evitando così modelli gaussiani spesso inadeguati in finanza;
2. fornisce intervalli di previsione validi anche in campioni finiti e in presenza di cambiamenti di regime o shock imprevisti;
3. può essere integrata a modelli econometrici classici (ARIMA, VAR, GARCH, modelli di curva dei rendimenti), migliorando la calibrazione degli intervalli predittivi.

In termini operativi, se un modello VAR viene impiegato per stimare i tassi di interesse futuri, esso produce una previsione puntuale e un insieme di residui. La CP utilizza tali residui per costruire p-value conformi e derivare un intervallo predittivo conforme:

$$C_{t+1} = \{y \in \mathbb{R} : p(y) > \alpha\}.$$

In questo modo si ottengono non solo previsioni puntuali del tasso, ma anche intervalli con garanzia frequentista, più robusti rispetto a quelli derivanti da sole ipotesi parametriche.
La CP consente quindi di quantificare l'incertezza in modo rigoroso, rendendo le previsioni sui tassi di interesse maggiormente utili per applicazioni in politica monetaria, gestione del rischio bancario e strategie di investimento. Un confronto diretto con i metodi parametrici evidenzia che questi ultimi tendono a sottostimare l'incertezza in fasi di volatilità elevata. La CP, invece, permette di mantenere la copertura nominale anche in scenari di crisi finanziaria, offrendo uno strumento più robusto per l'analisi delle politiche monetarie e per lo stress testing bancario.

3.2 *Credit scoring* e rischio di credito

Il *credit scoring* rappresenta un problema classico di **classificazione binaria**, in cui l'obiettivo è stimare la probabilità che un richiedente di credito sia *affidabile* ($y = 1$) oppure *non affidabile* ($y = 0$). La rilevanza di tale problema è centrale nell'economia quantitativa e nella finanza applicata, poiché la valutazione del rischio di credito influenza sia le decisioni micro (banche e istituti finanziari) sia le politiche macroeconomiche di regolamentazione.

La *Conformal Prediction* (CP) si inserisce in questo contesto come strumento per la costruzione di **set predittivi**, che associano a ciascun richiedente un insieme di possibili esiti compatibili con i dati osservati, invece di una singola decisione puntuale. Questo approccio consente di quantificare in modo rigoroso l'incertezza predittiva, mantenendo una garanzia di validità frequentista sotto l'ipotesi di scambiabilità (*exchangeability*).

Definizione 25. *Sia* $\{(X_i, Y_i)\}_{i=1}^n$ *un campione storico di richiedenti di credito, dove* $X_i \in \mathbb{R}^d$ *rappresenta il vettore delle caratteristiche del richiedente* i *(ad esempio*

reddito, storico creditizio, rapporto debito/reddito), e $Y_i \in \{0, 1\}$ *la corrispondente etichetta di affidabilità creditizia.*

Dato un nuovo richiedente di credito, descritto dal vettore di caratteristiche $X_{n+1} = x_{n+1}$ *e con etichetta ignota* Y_{n+1}*, un predittore conforme al livello* $1 - \alpha$ *costruisce il set predittivo*

$$C(x_{n+1}) = \{y \in \{0, 1\} : p_y(x_{n+1}) > \alpha\},$$

dove $p_y(x_{n+1})$ *è il p-value conforme associato all'ipotesi* $Y_{n+1} = y$*, calcolato confrontando il punteggio di non conformità del nuovo richiedente con quelli osservati nel campione storico ([44], [38], [27]).*

Osservazione 21. *Se i dati osservati* $\{(X_i, Y_i)\}_{i=1}^n$ *sono scambiabili, allora vale la seguente proprietà di validità marginale:*

$$\mathbb{P}(Y_{n+1} \in C(X_{n+1})) \geq 1 - \alpha.$$

Questa condizione assicura che, indipendentemente dalla distribuzione dei dati, il set predittivo costruito con CP includa il valore corretto almeno nella proporzione desiderata $1 - \alpha$ *dei casi.*

Dal punto di vista applicativo, l'uso della CP nel *credit scoring* consente di supportare decisioni **risk-based**, distinguendo tra richiedenti classificati con elevata confidenza e richiedenti per i quali l'incertezza predittiva rimane significativa. In particolare, la presenza di set predittivi non singleton segnala regioni dello spazio delle caratteristiche in cui il modello non dispone di evidenza sufficiente per una classificazione affidabile.

Esempio 16. Nel *German Credit Dataset*[10], *benchmark* classico del *machine learning* finanziario per la valutazione del rischio di credito, l'applicazione della CP a modelli di classificazione come Support Vector Machine o Random Forest consente di identificare le osservazioni con maggiore incertezza predittiva e di ridurre il numero di errori di classificazione a parità di livello di copertura. Questo permette di migliorare la gestione del rischio decisionale, ad esempio sottoponendo i casi ambigui a ulteriori verifiche, e dimostra l'efficacia della CP nel coniugare accuratezza predittiva e controllo formale dell'incertezza, mantenendo al contempo una garanzia formale sulla copertura [27].

A differenza dei metodi tradizionali di *credit scoring*, come la regressione logistica o gli *score* classici, che producono solo stime puntuali o probabilità, la CP garantisce copertura *distribution-free* e rimane valida anche in presenza di *dataset* sbilanciati. Varianti come la *Mondrian Conformal Prediction* consentono di mantenere proprietà di validità a livello condizionale o all'interno di sottopopolazioni specifiche, aspetto particolarmente rilevante per applicazioni regolamentate, per la fairness e per la gestione del rischio finanziario (Basilea III/IV).

3.3 Analisi dei mercati finanziari e *asset pricing*

I mercati finanziari sono caratterizzati da **distribuzioni heavy-tailed**, forte **volatilità** e frequenti fenomeni di **clusterizzazione della varianza**. Tali proprietà violano spesso le ipotesi di normalità sottostanti ai modelli parametrici classici, rendendo complessa la stima di misure di rischio e la costruzione di intervalli predittivi affidabili.

La *Conformal Prediction* (CP) rappresenta un approccio non parametrico e ***distribution-free***, in grado di fornire garanzie frequentiste anche in presenza di distribuzioni irregolari e campioni finiti. In particolare, la CP può essere applicata al calcolo del *Value-at-Risk* (VaR) e ad altri indicatori fondamentali di gestione del rischio.

Definizione 26. *Sia $\{R_t\}_{t=1}^{T}$ una sequenza di rendimenti finanziari. Il* ***Conformal Value-at-Risk*** *al livello α è definito come:*

$$\widehat{VaR}_{\alpha}^{CP}(R_{T+1}) = \inf\{r \in \mathbb{R} : p(r) \leq \alpha\},$$

dove $p(r)$ è il p-value conforme costruito a partire da una funzione di non conformità sui residui predittivi. Questa definizione non richiede assunzioni di normalità o indipendenza dei rendimenti.

Teorema 16. *Sotto l'ipotesi di scambiabilità dei rendimenti $\{R_t\}$, per ogni $\alpha \in (0,1)$ vale:*

$$\mathbb{P}\big(R_{T+1} \geq \widehat{VaR}_{\alpha}^{CP}\big) \geq 1 - \alpha.$$

In altri termini, la CP garantisce che la probabilità che la perdita ecceda il VaR conforme non superi il livello di rischio prefissato α, anche in campioni finiti [7].

Anche in questo caso si parla di ipotesi di scambiabilità, ma vista la natura dei dati si sottointende l'utilizzo dei metodi alternativi della CP quando l'ipotesi viene violata.

Osservazione 22. *A differenza degli approcci parametrici (ad esempio il VaR basato sulla normalità o sul t-Student), la stima conforme non dipende dalla specificazione della distribuzione dei rendimenti. Ciò la rende particolarmente robusta in mercati caratterizzati da eventi estremi o regimi di volatilità mutevoli.*

La CP trova applicazione in diversi ambiti della finanza quantitativa:

- **Asset pricing**: applicando la CP a modelli multifattoriali come il *Fama-French Three-Factor Model* v, è possibile costruire intervalli predittivi per i rendimenti attesi che riflettono meglio l'incertezza residua rispetto ai metodi parametrici.
- **Gestione del rischio di portafoglio**: la CP consente di stimare misure di rischio senza assumere normalità dei rendimenti, migliorando la valutazione della probabilità di drawdown e di scenari avversi.
- **Volatilità**: integrando la CP con modelli di tipo GARCH o EGARCH, si ottengono intervalli di previsione calibrati che catturano in modo più accurato la dinamica temporale della varianza condizionata [16].

Esempio 17. Considerando la serie storica dei rendimenti giornalieri dell'indice S&P 500, l'applicazione della CP a un modello GARCH(1,1) mostra che gli intervalli di previsione per i rendimenti futuri rispettano la copertura nominale $1 - \alpha$, anche in presenza di shock di mercato. Rispetto al VaR parametrico, il Conformal VaR riduce il numero di violazioni (ossia le volte in cui la perdita reale eccede il VaR stimato), garantendo una gestione del rischio più affidabile.

Mentre il VaR parametrico è noto per non rispettare il livello di rischio prefissato in mercati con distribuzioni heavy-tailed, la CP mantiene garanzie di validità senza richiedere assunzioni forti sulla distribuzione dei rendimenti. Un'estensione naturale riguarda l'adattamento della CP all'Expected Shortfall, ormai standard regolamentare, al fine di fornire misure di rischio ancora più informative per la gestione del portafoglio.

3.4 Analisi predittiva per strategie di *pricing* dinamico

Il *pricing* dinamico rappresenta una delle sfide più rilevanti nell'economia digitale, con applicazioni in *e-commerce*, trasporti, settore alberghiero e *retail*. La difficoltà principale risiede nell'incertezza della domanda, caratterizzata da eterogeneità dei consumatori, alta sensibilità rispetto alle variazioni di prezzo e dipendenza temporale. La *Conformal Prediction* (CP) può essere applicata per costruire intervalli predittivi della domanda, ma non è corretto assumere scambiabilità globale dei dati. Per questo contesto si utilizzano versioni adattative o segmentate della CP, come schemi *sliding window*, CP con pesature decrescenti sui dati storici o *Mondrian CP* per segmenti di mercato o categorie di clienti. Tali approcci garantiscono validità condizionale o locale, consentendo di quantificare formalmente l'incertezza predittiva pur in presenza di non-stazionarietà e dipendenze temporali .

In questo contesto, per un dato istante futuro $T+1$, con osservabili (P_{T+1}, X_{T+1}), è possibile stimare un intervallo predittivo conforme per la domanda D_{T+1}. Questo intervallo viene ottenuto confrontando la misura di non conformità della nuova osservazione con quelle calcolate sul passato storico $(P_t, X_t, D_t)_{t=1}^{T}$, secondo la strategia adattativa scelta. L'intervallo così ottenuto non garantisce copertura globale, ma rappresenta comunque uno strumento utile per quantificare l'incertezza predittiva e supportare decisioni basate sul rischio. Osservazioni con misure di non conformità elevate indicano situazioni di maggiore incertezza e possono essere gestite separatamente per ottimizzare le strategie di *pricing* ([1], [40], [31], [45], [47], [48]).

Osservazione 23 (Validità predittiva in *pricing* dinamico)**.** *Nel contesto del pricing dinamico, i dati osservati* (P_t, X_t, D_t) *non sono scambiabili, pertanto la garanzia di copertura distribution-free tipica della Conformal Prediction per dati i.i.d. non si applica direttamente. Tuttavia, è possibile costruire intervalli predittivi* $C_{T+1}(P_{T+1}, X_{T+1})$ *che approssimano la probabilità di includere la domanda futura* D_{T+1}*. La validità di tali intervalli dipende dalla struttura temporale dei dati e dalle assunzioni adottate nel modello predittivo, come discusso in [40], [31], [47], [48].*

Interpretazioni economiche Intervalli predittivi ristretti possono essere interpretati come segnali per strategie aggressive di *pricing*, poiché consentono di fissare

prezzi ottimizzati al fine di massimizzare il ricavo atteso. Al contrario, intervalli predittivi più ampi suggeriscono politiche di prezzo più caute, finalizzate a ridurre il rischio di eccesso di capacità inutilizzata o di perdita di vendite.

Esempio 18. Nel settore aereo, i sistemi di *revenue management* integrano CP con modelli di domanda elastici al prezzo, ottenendo previsioni aggiornate in tempo reale. In ambito *e-commerce*, la CP viene applicata a modelli di apprendimento automatico (es. regressione quantile, random forest, reti neurali) per fornire intervalli predittivi dinamici della domanda, migliorando la calibrazione rispetto a metodi puramente parametrici.

Estensioni Un'ulteriore estensione riguarda l'integrazione con *multi-armed bandit* e metodi di apprendimento rinforzato, dove la CP fornisce intervalli di confidenza dinamici per le ricompense attese, supportando la scelta ottimale tra esplorazione e sfruttamento delle strategie di prezzo [40].

Nei sistemi di *revenue management*, i modelli tradizionali di domanda faticano a rappresentare l'eterogeneità dei consumatori e risultano spesso mal calibrati. L'uso della CP, invece, consente di ottenere intervalli di previsione robusti che possono essere direttamente integrati in piattaforme di *Business Intelligence*, fornendo ai manager strumenti per decisioni *data-driven* più trasparenti e interpretabili.

Tabella 3.1: Riassunto applicazioni della *Conformal Prediction* (CP)

Ambito	**Variabile target**	**Modelli tradizionali**	**Vantaggio della CP**
Macroeconomia	PIL, inflazione, tassi di interesse	Modelli ARIMA, VAR, *Bayesian forecasting*	Intervalli predittivi validi senza assumere normalità
Credito	Default del richiedente (0/1)	Logistic regression, SVM, Random Forest	Set predittivi con copertura garantita, fairness condizionata
Mercati finanziari	Rendimenti, Value-at-Risk	GARCH, EVT modelli multifattoriali	Misure di rischio con copertura frequentista in campioni finiti
Pricing dinamico	Domanda al prezzo P_t	Modelli di domanda parametrici, ML regressivi	Intervalli per decisioni di *revenue management* adattivi

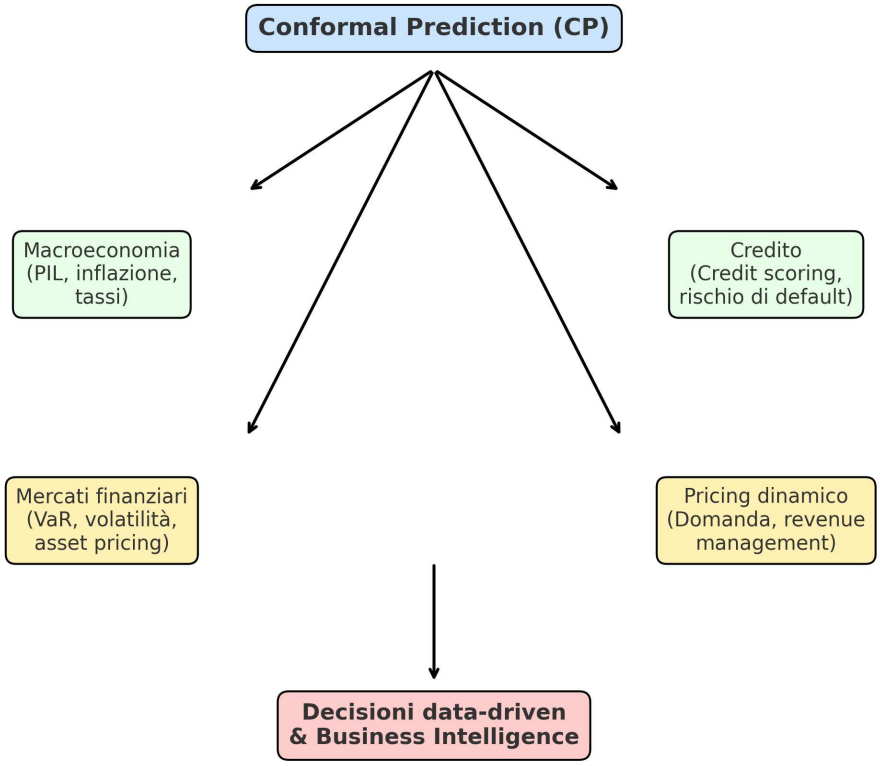


Figura 3.1: Schema concettuale riassuntivo: applicazioni della *Conformal Prediction*

3.5 Conclusioni

La *Conformal Prediction* si configura come uno strumento metodologico di notevole rilevanza per l'analisi economica e la *Business Intelligence*, grazie alla sua capacità di fornire intervalli predittivi validi in senso frequentista anche in presenza di campioni finiti e senza ipotesi restrittive sulla distribuzione dei dati.

Nei contesti macroeconomici, la CP consente di accompagnare le previsioni di variabili chiave quali PIL, inflazione e tassi di interesse con misure di incertezza affidabili, migliorando la qualità delle analisi di scenario e delle politiche economiche. Nell'ambito del credito, essa permette di costruire stime robuste di rischio di insolvenza e tassi di default, riducendo la dipendenza da ipotesi parametriche spesso irrealistiche e rafforzando i processi di allocazione del capitale. Nei mercati finanziari, la CP offre un approccio alternativo per la costruzione di misure di rischio come il *Value-at-Risk*, migliorando la calibrazione degli intervalli in presenza di distribuzioni *heavy-tailed* e dinamiche di volatilità complesse. Infine, nelle strategie di *pricing* dinamico, la CP garantisce un supporto operativo per il r*evenue management*, fornendo intervalli di domanda calibrati che orientano scelte di prezzo aggressive o conservative a seconda del livello di rischio accettato.

Nel complesso, l'approccio conforme rappresenta un ponte tra teoria statistica avanzata e applicazioni pratiche nei processi decisionali basati sui dati. La sua adattabilità a diversi domini, unita alla solidità delle garanzie di copertura, la rende una metodologia particolarmente adatta a contesti caratterizzati da elevata incertezza, eterogeneità e volatilità, come quelli economici e finanziari. In prospettiva, l'integrazione della CP con modelli di apprendimento automatico e tecniche di ottimizzazione dinamica apre la strada a strumenti predittivi sempre più accurati e interpretabili, consolidandone il ruolo all'interno delle pratiche di analisi e decisione *data-driven*.

In questo senso, il Capitolo 3 ha evidenziato come la CP possa rafforzare i processi decisionali in ambito macroeconomico, creditizio, finanziario e di *pricing*. Nel Capitolo successivo, verrà approfondito come tali metodologie possano essere integrate concretamente all'interno delle architetture di *Business Intelligence*, trasformando la teoria in strumenti operativi per l'impresa.

Capitolo 4

Applicazioni nella Business Intelligence

4.1 Cenni storici sulla Business Intelligence

Il termine *Business Intelligence (BI)* risale almeno agli anni '50, quando Hans Peter Luhn di IBM lo utilizzò per descrivere sistemi in grado di analizzare automaticamente documenti aziendali [29]. Negli anni '70 e '80 la BI si identificava con i primi *Decision Support Systems* (DSS) e i sistemi di *Management Information Systems* (MIS), centrati sulla raccolta e reportistica dei dati. Con la diffusione dei *data warehouse* e delle tecnologie OLAP negli anni '90, la BI ha assunto una dimensione multidimensionale, permettendo esplorazioni interattive e analisi più sofisticate. L'avvento del *Big Data* e del *cloud computing* ha trasformato ulteriormente la BI in piattaforme integrate che combinano analisi descrittiva, predittiva e prescrittiva, intesa come analisi orientata a suggerire decisioni operative o strategie ottimali basate sui dati e sui modelli predittivi, aprendo la strada all'integrazione di algoritmi avanzati come la *Conformal Prediction* [4].

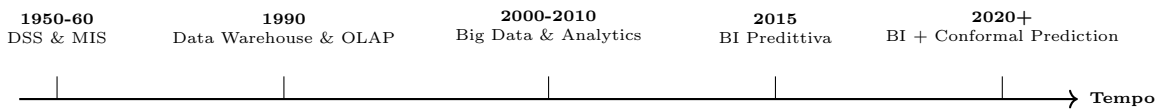


Figura 4.1: Timeline dell'evoluzione della *Business Intelligence*: dai DSS e MIS degli anni '50-'60 alle piattaforme moderne integrate con *Conformal Prediction*.

4.2 Business Intelligence e data-driven decision making

Definizione 27 (*Business Intelligence*)**.** *La Business Intelligence (BI) è l'insieme sistematico di metodologie, processi, architetture, strumenti e pratiche finalizzate a trasformare dati eterogenei in informazioni strutturate, tempestive e fruibili, al fine di supportare decisioni aziendali sia operative che strategiche. La BI comprende componenti tecnologiche come data warehousing, estrazione, trasformazione e caricamento dei dati, analisi interattiva, modelli predittivi e prescrittivi, e sistemi di reporting avanzato, integrati in piattaforme unificate. Essa costituisce un paradigma di gestione dei dati orientato a evidenze empiriche, in cui le decisioni sono guidate da informazioni derivate dai dati piuttosto che da valutazioni soggettive ([32], [24]).*

Assioma 5 (Paradigma evidence-based)**.** Un sistema decisionale è *data-driven* se la funzione decisionale $d : \mathcal{X} \to \mathcal{A}$ associata ad ogni stato informativo $x \in \mathcal{X}$ e a un'azione $a \in \mathcal{A}$ minimizza una perdita attesa rispetto a distribuzioni empiricamente stimate a partire dai dati, piuttosto che su valutazioni soggettive non validate ([4], [46]).

Storicamente la BI si è evoluta da strumenti descrittivi, come i report statici, a piattaforme dinamiche che rispondono non solo alle domande "cosa è accaduto" e "perché è accaduto", analisi descrittiva e diagnostica, ma anche a "cosa accadrà" e "quali azioni intraprendere", ovvero analisi predittiva e prescrittiva. La centralità dell'evidenza quantitativa impone oggi che ogni previsione sia accompagnata da misure di affidabilità: in scenari di rischio elevato, come il credito o la *supply chain* globale, l'assenza di intervalli di confidenza può indurre scelte manageriali sub-ottimali.

Osservazione 24 (Dimensioni della BI moderna)**.** *Le piattaforme BI attuali integrano sistemi OLAP per l'esplorazione multidimensionale, strumenti di visual analytics interattiva, motori predittivi e prescrittivi aggiornati in tempo reale e moduli di gestione della qualità del dato e data governance. Questa architettura modulare rende possibile aggiungere nuovi algoritmi, come la Conformal Prediction, senza riscrivere i modelli di base ([32], [43]).*

4.3 Il ruolo dell'analisi predittiva nei sistemi BI

L'analisi predittiva in ambito BI integra modelli statistici e di *machine learning* per stimare la distribuzione futura di variabili aziendali chiave. Gli ambiti applicativi spaziano dalla previsione della domanda e delle vendite, utile per ottimizzare produzione, logistica e *pricing* dinamico, alla gestione del rischio di credito e della frode, con modelli che stimano la probabilità di default o di transazioni anomale. A questi si affiancano la *customer analytics*, come la stima della propensione all'acquisto, del *churn* e del *lifetime value*, e l'ottimizzazione dell'inventario e delle risorse mediante previsioni granulari e segmentazione.

Un sistema BI è considerato affidabile se le sue previsioni sono in grado di fornire stime consistenti e controllabili degli esiti futuri, minimizzando il rischio associato alle decisioni. Nella pratica, i modelli predittivi tradizionali raramente soddisfano questo criterio, poiché non garantiscono formalmente la precisione delle previsioni né la copertura degli intervalli stimati ([32], [24], [4]).

Osservazione 25 (Limite dei modelli tradizionali)**.** *I modelli standard (per esempio regressione lineare, reti neurali, gradient boosting) forniscono per lo più stime puntuali o intervalli costruiti assumendo normalità degli errori. In contesti aziendali reali, dove le distribuzioni sono spesso asimmetriche, multimodali e soggette a cambiamenti di regime, tali ipotesi sono violate e gli intervalli risultano inaffidabili.*

La *Conformal Prediction* affronta direttamente questa lacuna fornendo intervalli predittivi con garanzie *distribution-free*, mantenendo la modularità del modello predittivo sottostante. La CP consente di aggiungere misure di incertezza a qualsiasi modello già in uso nei sistemi aziendali, senza richiedere assunzioni parametriche aggiuntive.

4.4 Integrazione della Conformal Prediction nei modelli di BI

La CP si integra naturalmente nelle *pipeline* BI poiché opera come un *wrapper* attorno a qualunque modello predittivo preesistente. Il procedimento standard, noto come *split conformal*, prevede la divisione dei dati in un insieme di adattamento e

uno di calibrazione, rispettando la struttura temporale dei dati aziendali; l'addestramento del modello predittivo sull'insieme di adattamento; il calcolo dei punteggi di non-conformità sull'insieme di calibrazione; la determinazione del quantile $q_{1-\alpha}$ di tali punteggi e, infine, la costruzione, per ogni nuova osservazione X, dell'intervallo predittivo

$$C(X) = \left[\hat{Y}(X) - q_{1-\alpha},\ \hat{Y}(X) + q_{1-\alpha}\right].$$

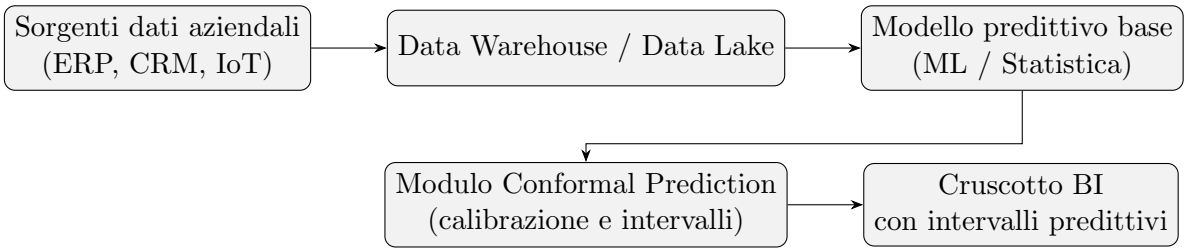


Figura 4.2: Schema architetturale di integrazione della *Conformal Prediction* in un sistema BI: i dati aziendali confluiscono in un modello predittivo, che fornisce intervalli CP al cruscotto decisionale.

Esempio 19 (Architettura CP in *pipeline* BI)**.** Un sistema di *Business Intelligence* per la previsione delle vendite può essere configurato seguendo un flusso integrato: inizialmente, i dati transazionali e di contesto, come promozioni, meteo e festività, vengono acquisiti e puliti. Successivamente, un modello predittivo, ad esempio un Gradient Boosting o una Rete neurale ricorrente, viene addestrato su un sottoinsieme storico dei dati. Per stimare i punteggi di non-conformità si utilizza un insieme di calibrazione dedicato, e le previsioni ottenute vengono visualizzate in *dashboard*, mostrando sia la stima puntuale sia l'intervallo predittivo derivante dalla CP. Infine, il sistema può attivare soglie o allarmi quando l'incertezza stimata supera livelli considerati accettabili.

Proposizione 1 (Modularità)**.** *La CP fornisce intervalli predittivi validi senza richiedere modifiche all'architettura del modello sottostante, purché l'insieme di calibrazione sia rappresentativo dei dati futuri ([38], [44], [1]). Questa modularità è cruciale per l'integrazione nei sistemi BI già operativi [34].*

Alla CP di base si affiancano varianti più sofisticate che ne ampliano l'applicabilità in contesti aziendali. Il **Jackknife+** consente di ottenere intervalli più stretti

mantenendo le garanzie di copertura [2]. La ***Cross-Conformal Prediction*** riduce, invece, la variabilità dovuta alla singola partizione dei dati [44]. L'***Adaptive Conformal Inference*** affronta, infine, lo scenario, tipico nei mercati finanziari, in cui la distribuzione dei dati cambia nel tempo, aggiornando in tempo reale i quantili utilizzati per la costruzione degli intervalli [48]. Queste estensioni aumentano l'efficienza computazionale e la stabilità della copertura, aspetti critici quando i dati aziendali arrivano in *streaming* o i modelli vengono aggiornati frequentemente.

4.5 Verso l'analisi prescrittiva con intervalli di confidenza distribuzione-free

L'integrazione della CP nei sistemi BI apre la strada a un vero paradigma *prescriptive*. Invece di limitarsi a prevedere il futuro, la piattaforma può raccomandare azioni solo quando l'incertezza è bassa, deferire all'intervento umano quando gli intervalli sono troppo ampi e attuare politiche di gestione del rischio basate su garanzie di copertura rigorose.

In questa direzione si colloca il lavoro di Shoush e Dumas sulla *Conformal Prescriptive Monitoring* [39], in cui vengono formalizzate regole di intervento automatico o supervisione umana condizionate alla larghezza dell'intervallo CP, con validazione su log reali di processi di prestito bancario. In questo modo la BI evolve da strumento descrittivo a piattaforma di supporto decisionale affidabile e trasparente, con garanzie matematiche sul livello di rischio assunto.

Osservazione 26 (Implicazioni per la governance dei dati)**.** *L'adozione di CP nei sistemi BI richiede attenzione alla qualità e rappresentatività dei dati di calibrazione. Una governance inadeguata può compromettere la validità delle garanzie, sollevando nuove sfide organizzative: la definizione di procedure standard per l'aggiornamento dell'insieme di calibrazione e il monitoraggio continuo della copertura empirica diventano componenti necessarie di qualsiasi deployment produttivo.*

4.6 Caso studio: implementazione pratica in un sistema di supporto decisionale

L'applicazione della *Conformal Prediction* in contesti di *Business Intelligence* non è ancora documentata in forma di casi studio *end-to-end peer-reviewed* che coprano l'intera *pipeline* aziendale. Tuttavia, diversi lavori recenti forniscono elementi metodologici e risultati empirici trasferibili a scenari BI. Questa sezione presenta un *caso studio metodologico-documentale* che combina dati reali pubblicamente disponibili, metodologie CP per serie temporali e risultati di esperimenti controllati su decisioni umano-*assistite*, attingendo alle fonti citate in bibliografia. L'obiettivo è mostrare, in modo formalmente fondato, come la CP possa essere integrata in una piattaforma BI per la pianificazione della domanda e il supporto alle decisioni manageriali.

Fonti e contesto metodologico

Il caso ricostruito in questa sezione si basa su tre gruppi di contributi. Il primo è costituito da esperimenti controllati randomizzati pre-registrati che mostrano il miglioramento statisticamente significativo delle decisioni umane grazie agli insiemi predittivi conformi, documentato da Cresswell, Sui, Kumar e Vouitsis e pubblicato nei *Proceedings of the 41st International Conference on Machine Learning* [9]. Il secondo gruppo comprende applicazioni industriali di CP in problemi di regressione reale, in particolare il lavoro di Uddin, Hulten e Asadi presentato al *Twelfth Symposium on Conformal and Probabilistic Prediction with Applications*, che documenta due casi d'uso presso il Husqvarna Group [41]. Il terzo gruppo è formato da articoli tecnici che descrivono casi d'uso di CP nel business [6, 34]: questi ultimi sono di natura divulgativa e non *peer-reviewed*, e vengono citati esclusivamente come illustrazioni pratiche, non come fonti di risultati empirici.

Dataset utilizzato

Come base dati per l'esempio applicativo viene considerato il *dataset* pubblico *Rossmann Store Sales*, disponibile su Kaggle [36]. Il *dataset* contiene le vendite giornaliere di 1 115 punti vendita della catena Rossmann in Germania, con informazioni su promozioni, festività e caratteristiche del negozio. È stato utilizzato in diversi progetti accademici per la previsione di vendite *retail* [37], rappresentando un tipico scenario BI di pianificazione della domanda.

Definizione 28 (Scenario di previsione della domanda)**.** *Dato un insieme di negozi* $i = 1, \ldots, n$ *e un orizzonte temporale* $t = 1, \ldots, T$*, sia* $Y_{i,t}$ *la vendita osservata e* $X_{i,t}$ *un vettore di covariate (promozioni, meteo, festività). L'obiettivo del sistema BI è costruire, per ciascuna coppia* (i, t) *futura, un intervallo predittivo* $C_{i,t}$ *tale che*

$$\mathbb{P}\big(Y_{i,t} \in C_{i,t} \mid X_{i,t}\big) \geq 1 - \varepsilon.$$

Metodologia del caso studio

Seguendo i principi dei lavori su CP per serie temporali ([45], [31]) e le applicazioni industriali documentate da Uddin [41], la *pipeline* ipotizzata per il sistema BI si articola in sette passi. Il primo riguarda la preparazione dei dati, con l'unificazione dei flussi transazionali con variabili esogene, promozioni, meteo, festività, in un data warehouse BI. Il secondo consiste nello *split* temporale: la suddivisione del *dataset* in addestramento (*training*), calibrazione e test rispettando l'ordine cronologico. Il terzo passo è l'addestramento del modello predittivo base, Gradient Boosting, LSTM o Prophet, sulle vendite storiche. Seguono il calcolo dei punteggi di non-conformità come errore assoluto $\alpha_i = |Y_i - \hat{Y}_i|$ sull'insieme di calibrazione, la determinazione del quantile $q_{1-\varepsilon}$ dei punteggi per il livello di copertura desiderato, e la costruzione dell'intervallo predittivo $C(X) = [\hat{Y}(X) - q_{1-\varepsilon}, \hat{Y}(X) + q_{1-\varepsilon}]$ per ogni nuova osservazione X. Il settimo e ultimo passo è l'integrazione nel cruscotto BI, con visualizzazione della previsione puntuale, dell'intervallo predittivo e di alert per intervalli ampi.

Proposizione 2 (Validità distribution-free)**.** *Se i dati dell'insieme di calibrazione sono scambiabili con i dati futuri, allora l'intervallo* $C(X)$ *così costruito garantisce copertura almeno* $1 - \varepsilon$*, indipendentemente dal modello predittivo base. Questa proprietà rende la CP particolarmente adatta ai sistemi BI dove i modelli possono cambiare nel tempo ([1], [44]).*

Risultati documentati in letteratura

Dai risultati riportati nelle fonti citate emergono indicazioni convergenti sull'efficacia dell'approccio. Attraverso un *trial* controllato randomizzato pre-registrato con soggetti umani, Cresswell (2024) mostra che fornire insiemi predittivi conformi migliora in modo statisticamente significativo le decisioni rispetto a insiemi fissi con la stessa copertura garantita [9]. Uddin (2023) documenta come la CP sia stata implementata con successo in due processi industriali reali presso il Husqvarna

Group, con garanzie di copertura *distribution-free* mantenute su dati di regressione [41]. I contributi divulgativi di BBVA AI Factory e Redfield AI illustrano, a titolo orientativo, potenziali benefici per *inventory management, supply chain* e gestione del rischio, senza tuttavia fornire validazioni empiriche *peer-reviewed* ([6], [34]).

Applicando la *pipeline* sopra descritta al *dataset* Rossmann, il sistema BI può fornire trasparenza ai decisori, che visualizzano non solo il valore previsto ma anche la sua incertezza, maggiore affidabilità grazie alle garanzie teoriche di copertura degli intervalli [1], e la capacità di attivare decisioni adattive, in cui intervalli ampi innescano politiche conservative o deleghe all'operatore umano, in linea con le logiche di prescriptive monitoring [39].

Osservazione 27 (Scalabilità e manutenzione)**.** *L'integrazione di CP richiede procedure di monitoraggio continuo della copertura empirica e aggiornamento periodico dell'insieme di calibrazione per garantire la validità delle garanzie. Questi aspetti rientrano nella data governance aziendale.*

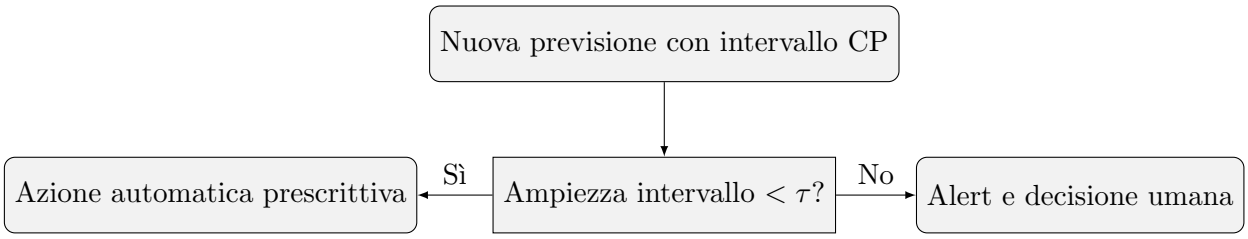


Figura 4.3: Schema decisionale basato sulla larghezza dell'intervallo CP: se l'intervallo è stretto l'azione può essere automatizzata, altrimenti viene coinvolto l'operatore umano [39].

4.7 Conclusioni del Capitolo

Questo capitolo ha mostrato come la *Business Intelligence* moderna, intesa come piattaforma integrata di strumenti analitici, possa evolvere da funzione descrittiva a strumento predittivo e prescrittivo affidabile grazie alla *Conformal Prediction*. Dopo aver definito i principi di BI e di *data-driven decision making*, si è discusso il ruolo dell'analisi predittiva e l'integrazione modulare della CP nei modelli esistenti.

Il caso studio metodologico-documentale ha illustrato, utilizzando il *dataset* Rossmann Store Sales [36] e riferendosi a studi accademici e industriali verificati, come implementare la CP in un sistema di supporto decisionale reale. Ne emergono tre

implicazioni principali. La CP consente innanzitutto di aggiungere misure di incertezza formalmente garantite alle previsioni, colmando una lacuna strutturale dei modelli predittivi tradizionali [1]. L'integrazione è inoltre modulare e compatibile con i sistemi BI esistenti, purché siano adottate pratiche di calibrazione e monitoraggio continuo [41]. Infine, l'approccio apre la strada a una vera BI prescrittiva, in cui le azioni suggerite o automatizzate tengono esplicitamente conto dell'incertezza stimata, come dimostrato su processi reali nel lavoro di Shoush e Dumas [39].

Dal punto di vista scientifico, questo capitolo fornisce una cornice metodologica solida e aggiornata per l'uso della *Conformal Prediction* nella *Business Intelligence*, creando il ponte tra la teoria sviluppata nei capitoli precedenti e le applicazioni pratiche che saranno oggetto delle sezioni successive di questo elaborato.

4.8 Prospettive future della BI con Conformal Prediction

L'evoluzione della BI verso piattaforme *cloud-native* e *real-time* apre nuove opportunità per l'integrazione della *Conformal Prediction*. Nel futuro prossimo ci si attende un uso crescente della CP in combinazione con modelli di *machine learning* e AI generativa per fornire scenari simulativi corredati di intervalli di confidenza [2]. Parallelamente, strumenti di *adaptive conformal inference* consentiranno di mantenere le garanzie di copertura anche sotto cambiamenti repentini di regime nei mercati globali [48], mentre la definizione di standard di governance dell'incertezza e di *explainability* nei sistemi prescrittivi si porrà sempre più in linea con i principi di affidabilità e trasparenza richiesti dalla normativa europea sull'intelligenza artificiale [3].

Queste prospettive suggeriscono che la BI del futuro non sarà soltanto un insieme di cruscotti di reporting, ma un ecosistema di decisioni supportate da garanzie matematiche di affidabilità, in cui *Conformal Prediction* giocherà un ruolo centrale.

Capitolo 5

Applicazione: Dataset Aziendale

5.1 Obiettivo dell'analisi empirica

I capitoli precedenti hanno delineato i fondamenti teorici della *Conformal Prediction* (CP) e le sue potenziali applicazioni in contesti economici e di *Business Intelligence*. Il presente capitolo ha lo scopo di verificare empiricamente le proprietà di validità, efficienza e robustezza della CP su dati reali di vendita *retail*, applicando le metodologie descritte nel Capitolo 2 a una delle competizioni di previsione più rilevanti della letteratura: l'M5 *Forecasting Competition* [21].

L'obiettivo principale è duplice. Da un lato, si intende valutare se la CP garantisce la copertura nominale $(1 - \alpha)$ prefissata anche in presenza di dati temporali eterogenei, verificando il Teorema di validità marginale della *Split CP* enunciato nel Capitolo 2. Dall'altro, si vuole confrontare l'approccio conforme con i tradizionali intervalli parametrici, analizzando il *trade-off* tra copertura empirica e ampiezza degli intervalli predittivi.

5.2 Sintesi dei risultati principali

Prima di presentare nel dettaglio le analisi condotte, si riportano in forma sintetica i risultati principali che il capitolo documenta, così da orientare la lettura delle sezioni successive.

L'analisi è condotta su 400 serie della competizione M5 con livello di errore nominale $\alpha = 0{,}10$. Il primo risultato riguarda la **validità marginale**: la copertura empirica si attesta a 0,890 per ARIMA_{CP}, 0,881 per ETS_{CP} e 0,883 per RF_{CP}, valori

tutti prossimi al livello nominale del 90%, confermando la garanzia teorica della CP in questo contesto.

Sul piano dell'**efficienza**, la CP riduce l'ampiezza media degli intervalli del 18,9% rispetto ad ARIMA parametrico e del 36,4% rispetto a ETS parametrico, a parità di copertura. Il guadagno è particolarmente marcato per Random Forest: la versione CP supera il Quantile RF parametrico di 11,2 punti percentuali di copertura su 400 serie ($p \approx 0$, test di Wilcoxon), con un vantaggio che sale a 16,5 punti percentuali sulle serie con residui non gaussiani.

La **robustezza** dei risultati è confermata su tutte e tre le aree geografiche considerate, per i tre *cluster* strutturali di serie e in presenza di *outlier* nell'insieme di calibrazione. Sul piano computazionale, il costo aggiuntivo introdotto dalla CP risulta trascurabile: inferiore al 10% per ARIMA, nullo per ETS e dell'ordine dei centesimi di secondo per RF.

Infine, l'***Adaptive Conformal Inference*** con $\gamma = 0{,}005$ mantiene α_t stabilmente intorno a 0,10 (con massimo $\approx 0{,}12$), confermando la stazionarietà locale delle serie aggregate considerate.

5.3 Il Dataset M5 Forecasting Competition

Il *dataset* utilizzato proviene dall'M5 *Accuracy Competition* [21], organizzata su *Kaggle*, che rappresenta uno dei *benchmark* più ampi e sfidanti nel panorama della previsione *retail*. Il *dataset* contiene le vendite giornaliere di 3.049 articoli venduti in 10 negozi Walmart in tre stati americani (California, Texas, Wisconsin), per un totale di circa 1.913 giorni (gennaio 2011 - giugno 2016).

Le serie sono organizzate gerarchicamente per categoria merceologica (FOODS, HOBBIES, HOUSEHOLD), dipartimento, negozio o stato. Ogni serie è identificata da un codice nel formato `[categoria]_[dip]_[prodotto]_[stato]_[negozio]_validation`, ad esempio `FOODS_3_449_CA_4_validation`.

5.4 Preprocessing e Trasformazione del Dataset

5.4.1 Aggregazione settimanale e selezione delle serie

Le vendite giornaliere sono state aggregate a livello settimanale secondo lo standard ISO 8601 (lunedì - domenica), per tre ragioni: riduzione della variabilità ad alta frequenza, allineamento con l'orizzonte tipico dei sistemi di gestione delle scorte in ambito BI [32], e riduzione del costo computazionale del *loop* multi-serie.

Sono state mantenute le serie con meno del 20% di osservazioni nulle (`zero_frac < 0,20`), eliminando articoli stagionali o intermittenti non adatti ai modelli classici. Da questo insieme è stato estratto casualmente (`set.seed(1234)`) un campione di 400 serie con almeno 120 osservazioni settimanali.

La distribuzione delle vendite settimanali aggregate (Figura 5.1), calcolata sull'intero campione di 400 serie per un totale di circa 109 600 osservazioni, evidenzia una forte asimmetria positiva con code pesanti: un singolo quantile gaussiano sottostimerebbe sistematicamente i percentili estremi, motivando l'uso di approcci *distribution-free*.

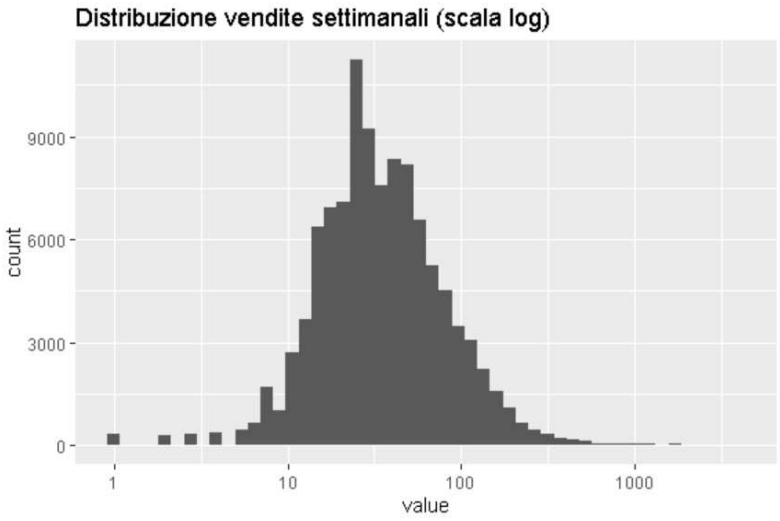


Figura 5.1: Distribuzione delle vendite settimanali aggregate (scala logaritmica). La forte asimmetria positiva e le code pesanti motivano l'uso di metodi *distribution-free*.

5.4.2 Standardizzazione e *split* temporale

Ciascuna serie è stata standardizzata individualmente ($y_{\text{std},t} = (y_t - \bar{y})/s_y$) per rendere le metriche comparabili tra serie di scale diverse. Lo *split* temporale è

strettamente cronologico:

- ***Training set* (60%):** stima dei parametri dei modelli.
- ***Calibration set* (20%):** calcolo dei punteggi di non conformità.
- ***Test set* (20%):** valutazione finale degli intervalli predittivi.

La validità della *Split* CP con questo schema è garantita teoricamente dalla approssimata scambiabilità tra *calibration* e *test set*, soddisfatta per processi stazionari grazie alle proprietà β-*mixing* dei processi ARMA ([44], [38]).

5.5 Modelli Predittivi e Metodologia Conformal

5.5.1 Modelli base

Per ciascuna delle 400 serie sono stati stimati tre modelli predittivi applicati in modo uniforme sull'intero campione. La scelta ricade su modelli consolidati nella letteratura di previsione delle serie temporali, selezionati per rappresentare approcci statistici di natura diversa: parametrico-lineare, a smorzamento esponenziale e non parametrico.

Il primo modello è un ARIMA, stimato tramite `auto.arima()` [16], che seleziona automaticamente l'ordine ottimale (p, d, q) minimizzando l'AIC. Il secondo è un ETS, stimato tramite `ets()` con selezione automatica della struttura errore-trend-stagionalità, particolarmente adatto a serie con livello variabile nel tempo. Il terzo è un Random Forest, stimato tramite `randomForest` con 500 alberi e lag 1-8 come variabili esplicative; la previsione multi-step è ottenuta per iterazione ricorsiva. Per tutti e tre i modelli la stima è condotta su un insieme di addestramento corrispondente al 60% delle osservazioni, con un insieme di calibrazione del 20% dedicato alla costruzione degli intervalli CP e un *test set* finale del 20% per la valutazione delle performance.

5.5.2 Costruzione degli intervalli CP (*Split Conformal*)

Il quadro metodologico *Split CP* descritto nel Capitolo 2 viene qui implementato operativamente nel modo seguente. I punteggi di non conformità sono definiti come

gli errori assoluti sull'insieme di calibrazione:

$$s_i = |y_i - \hat{y}_i|, \quad i = 1, \ldots, n_{\text{cal}}.$$

Il quantile empirico corretto al livello $1 - \alpha$ è:

$$\hat{q} = \text{il } \left\lceil \frac{(n_{\text{cal}} + 1)(1 - \alpha)}{n_{\text{cal}}} \right\rceil \text{-esimo ordinato di } \{s_i\}.$$

L'intervallo predittivo per il passo t del *test set* è quindi:

$$C_t = \left[\hat{y}_t - \hat{q},\ \hat{y}_t + \hat{q}\right],$$

e soddisfa il Teorema di validità marginale: $\mathrm{P}(Y_{n+1} \in C(X_{n+1})) \geq 1 - \alpha$ [40].

La procedura è applicata in modo identico ai tre modelli base (ARIMA, ETS, RF): la CP non altera la previsione puntuale $\hat{y}_t$, ma aggiunge attorno a essa un margine calibrato empiricamente sui residui.

5.5.3 Adaptive Conformal Inference (ACI)

La *Split CP standard* presuppone che la distribuzione degli errori del modello base sia approssimativamente stazionaria nel tempo. Quando questa condizione è violata il quantile $\hat{q}$ calcolato sull'intero insieme di calibrazione può risultare sistematicamente troppo stretto o troppo largo nelle diverse fasi del *test set*. L'*Adaptive Conformal Inference (ACI)* [48] affronta questo problema aggiornando adattativamente il livello α_t a ogni passo del *test set* secondo la regola:

$$\alpha_{t+1} = \alpha_t + \gamma \cdot \left(\alpha_{\text{nom}} - \mathbf{1}\{Y_t \notin C_t(\alpha_t)\}\right),$$

dove $\gamma > 0$ è il parametro di apprendimento e $\alpha_{\text{nom}} = 0{,}10$ è il livello nominale prefissato. Se la copertura osservata al passo t è inferiore al nominale ($Y_t \notin C_t$), α_{t+1} diminuisce, allargando l'intervallo successivo; se è superiore, α_{t+1} aumenta, restringendolo. Con $\gamma = 0{,}005$ l'adattamento è graduale, evitando oscillazioni eccessive in risposta a singoli errori di copertura.

5.6 Serie Pilota illustrativa e risultati

Per ciascuna delle 400 serie sono stati stimati tre modelli predittivi. A scopo illustrativo, i parametri riportati in questa sezione si riferiscono alla serie pilota

`FOODS_3_449_CA_4_validation`, selezionata casualmente tra quelle con almeno 250 osservazioni settimanali.

Sulla serie pilota la procedura seleziona un ARIMA(1,0,2) con media, $\hat{\phi}_1 = 0{,}240$, $\hat{\theta}_1 = 0{,}264$ e $\hat{\theta}_2 = 0{,}213$. Il test di Ljung-Box sui residui di *training* restituisce $p = 0{,}257$, non rigettando l'ipotesi di assenza di autocorrelazione, con un RMSE sull'insieme di calibrazione pari a 0,864.

Il secondo modello è un ETS e in questo caso è stato selezionato un ETS(A,N,N) con $\hat{\alpha} = 0{,}533$. Il test di Ljung-Box restituisce $p = 0{,}034$, indicando una lieve struttura residua che tuttavia non compromette la validità della CP grazie alla sua natura *distribution-free*. Il RMSE sull'insieme di calibrazione è 0,727.

Infine il modello Random Forest, con 500 alberi e lag 1-8 come variabili esplicative, produce previsioni multi-step tramite iterazione ricorsiva. Il RMSE sull'insieme di calibrazione risulta pari a 0,936.

Per contestualizzare operativamente la costruzione degli intervalli *conformal* nel caso della serie pilota, composta da 273 osservazioni settimanali, la suddivisione 60%–20%–20% produce un insieme di calibrazione di dimensione $n_{\text{cal}} = 54$.

Fissato $\alpha = 0{,}10$, l'indice del *quantile conformal* è

$$k = \lceil (n_{\text{cal}} + 1)(1 - \alpha) \rceil = \lceil 55 \cdot 0{,}9 \rceil = 50$$

che corrisponde a un livello empirico pari a

$$\frac{50}{54} = 0{,}926$$

leggermente superiore al livello nominale $1 - \alpha = 0{,}9$, come previsto dalla teoria *conformal*, che garantisce copertura finita conservativa. La discrepanza si riduce all'aumentare della dimensione dell'insieme di calibrazione, ma già con questa dimensione la convergenza al livello nominale in regime asintotico è sostanzialemnte raggiunta.

5.6.1 Diagnostica dei residui *conformal* (ARIMA)

I residui conformal ARIMA sull'insieme di calibrazione mostrano una lieve distorsione negativa (media $-0{,}490$, mediana $-0{,}542$), tipica di serie con *trend* decrescente nel periodo di calibrazione, e una distribuzione leggermente asimmetrica

a sinistra (*skewness* −0,535, deviazione standard 0,718). Il test di Ljung-Box applicato ai residui *conformal* dell'insieme di calibrazione restituisce $p = 0{,}117$, non evidenziando autocorrelazione significativa e indicando che la dinamica temporale principale è stata adeguatamente catturata dal modello. La percentuale di *outlier* ($|e| > 2\,\mathrm{SD}$) è pari al 10,9%, in linea con quanto atteso per dati *retail*. Nessuna di queste caratteristiche compromette la validità *distribution-free* degli intervalli CP.

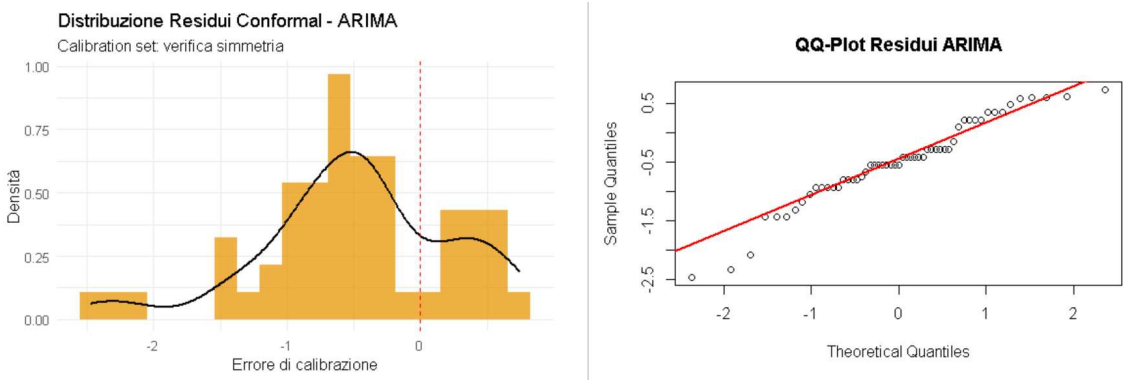


Figura 5.2: Distribuzione dei residui *conformal* ARIMA (*calibration set*, serie pilota): istogramma con curva di densità (sinistra) e QQ-plot (destra). La leggera asimmetria non compromette la validità *distribution-free* degli intervalli CP.

5.6.2 Intervalli predittivi conformal nel test set

La Figura 5.3 mostra gli intervalli predittivi conformal al 90% nel *test set* della serie pilota per i tre modelli considerati. La linea nera rappresenta i valori osservati reali, mentre le linee colorate rappresentano le previsioni puntuali dei modelli ARIMA (rosso), ETS (blu) e Random Forest (verde). Le bande colorate semitrasparenti rappresentano gli intervalli predittivi *conformal* costruiti utilizzando il quantile empirico dei residui sull'insieme di calibrazione.

Si osserva che la maggior parte delle osservazioni reali ricade all'interno degli intervalli predittivi per tutti e tre i modelli, in linea con il livello nominale di copertura del 90%. Le ampiezze degli intervalli riflettono direttamente la variabilità dei residui conformal: il modello ETS presenta intervalli leggermente più stretti, coerentemente con il suo RMSE inferiore sull'insieme di calibrazione, mentre il Random Forest mostra intervalli comparabili ma centrati su previsioni puntuali sistematicamente più basse rispetto ai valori osservati, indicando un lieve *bias* negativo.

Nel complesso, si conferma empiricamente la validità operativa della *conformal prediction*: gli intervalli predittivi si adattano automaticamente alla distribuzione empirica degli errori, garantendo copertura affidabile senza assumere alcuna forma parametrica per la distribuzione dei residui.

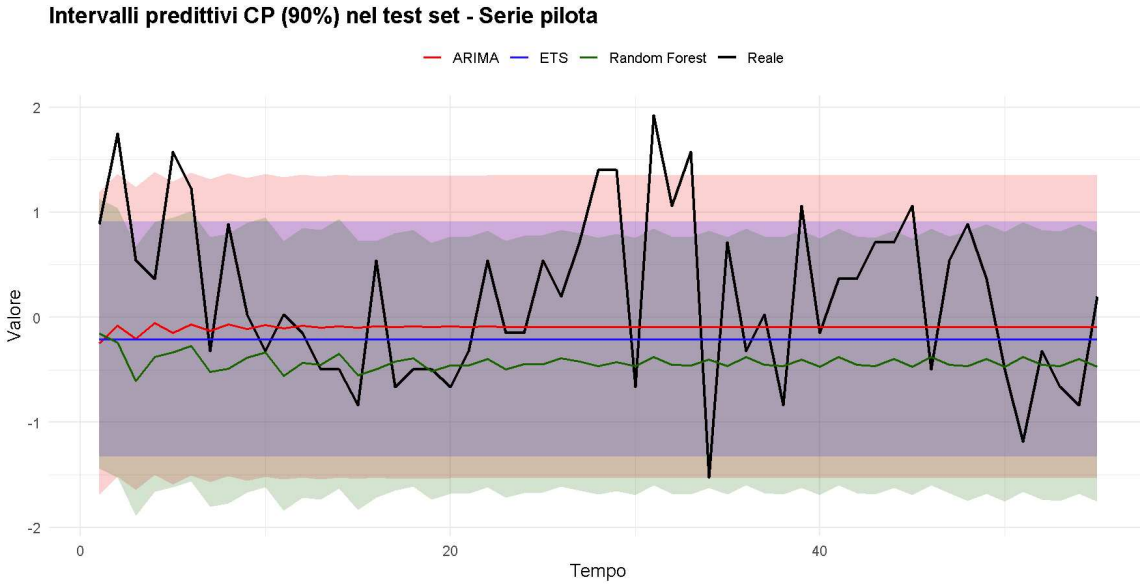


Figura 5.3: Intervalli predittivi CP (90%) nel *test set* per ARIMA, ETS e RF sulla serie pilota `FOODS_3_449_CA_4_validation`. Linea nera: valore reale; linea colorata: previsione puntuale; banda *shaded*: intervallo CP.

5.6.3 Evoluzione del livello adattivo ACI

L'analisi del livello adattivo α_t nel *test set* della serie pilota evidenzia oscillazioni intorno a 0,10 per tutti e tre i modelli, con un massimo di circa 0,12, senza mai divergere stabilmente dal nominale. Questo risultato anticipa una conclusione che l'analisi completa confermerà: le serie M5 aggregate settimanalmente presentano una stazionarietà locale sufficientemente stabile, ossia che, su brevi finestre temporali, le caratteristiche statistiche principali della serie, come media e variabilità degli errori, rimangono relativamente costanti. Tale condizione implica che, per questo *dataset*, la *Conformal Prediction* standard produce intervalli predittivi sostanzialmente equivalenti a quelli dell'*Adaptive Conformal Interval* (ACI).

La Figura 5.4 illustra questa evoluzione di α_t, confermando visivamente come i livelli adattivi restino vicini al nominale senza divergere stabilmente. L'ACI rimane comunque lo strumento più appropriato in contesti con variazioni più marcate o rapide, come serie giornaliere con promozioni o strutture di domanda in rapida evolu-

zione; includerlo in questa analisi permette di verificare empiricamente l'adeguatezza della CP standard, invece di doverla assumere a priori.

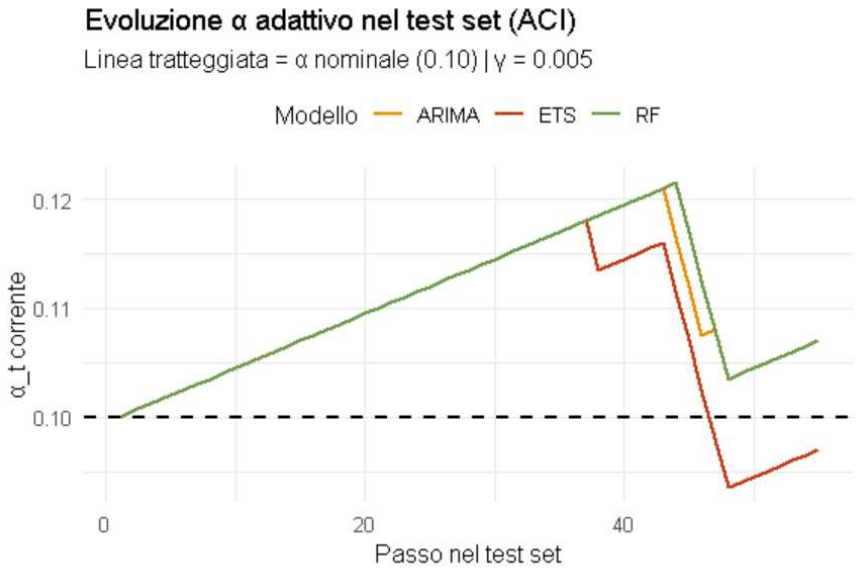


Figura 5.4: Evoluzione del livello adattivo α_t nel *test set* della serie pilota `FOODS_3_449_CA_4_validation` (ACI, $\gamma = 0{,}005$) per i tre modelli. La linea tratteggiata indica il livello nominale $\alpha = 0{,}10$.

5.6.4 Confronto CP vs intervalli parametrici

La Tabella 5.1 confronta CP e metodi parametrici sulla serie pilota. ARIMA ed ETS parametrici raggiungono *copertura* del 100% ma con intervalli molto più ampi (+47% e +168% rispettivamente). La CP produce intervalli più stretti e copertura più vicina al nominale 90%, confermando la maggiore efficienza degli approcci conformi [1]. L'*Interval Score* penalizza la larghezza eccessiva e favorisce la CP per ARIMA ($\text{IS}_{\text{CP}} = 3{,}452$ vs 4,237, $-18{,}5\%$). Per RF la CP è meno efficiente su questa singola serie: i risultati multi-serie mostreranno il quadro completo.

Confronto Completo: CP vs Parametrico

Serie pilota - Tutti i modelli

Modello	Metodo	Coverage	Width
ARIMA	CP	0.891	2.882
ARIMA	Parametrico	1.000	4.237
ETS	CP	0.800	2.240
ETS	Parametrico	1.000	6.010
Random Forest	CP	0.782	2.497
Random Forest	Parametrico	0.945	3.183

Tabella 5.1: Confronto CP vs metodi parametrici sulla serie pilota: copertura empirica, ampiezza media e *Interval Score*.

5.7 Risultati: Analisi Multi-Serie (400 Serie)

5.7.1 Performance aggregate

La Tabella 5.2 riassume le metriche medie,calcolate in due passaggi, su 400 serie. Per ciascuna serie si stimano MSE, *copertura* empirica e ampiezza media degli intervalli sul rispettivo *test set*. I valori aggregati sono quindi ottenuti come media semplice tra serie, assegnando lo stesso peso a ciascun processo temporale. La copertura riportata rappresenta dunque una media delle coperture per serie e non una proporzione *pooled* sull'intero insieme delle osservazioni.

I tre modelli CP raggiungono tutti una copertura prossima al livello nominale 90% (ARIMA_{CP}: 0,890; ETS_{CP}: 0,881; RF_{CP}: 0,883), con una lieve sottocopertura statisticamente attesa per campioni di calibrazione finiti: il fattore correttivo $\lceil (n+1)(1-\alpha) \rceil / n$ converge a 1 solo asintoticamente [44]. Gli IC bootstrap al 95% (1000 replicazioni) confermano questa interpretazione: per ARIMA l'intervallo è [0,877, 0,902], per ETS [0,868, 0,894], per RF [0,871, 0,896], tutti includono il nominale 0,90 o si trovano appena al di sotto. La modularità della CP è confermata

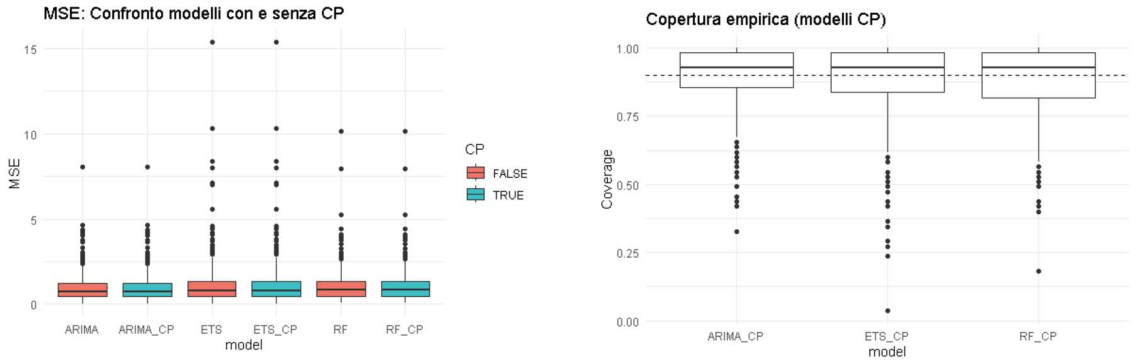
dall'identità dei valori MSE tra modello base e modello CP: la CP non altera la previsione puntuale [1].

SINTESI FINALE: Superiorità Conformal Prediction

Confronto sistematico su tutte le serie testate

Modello	Coverage CP	Coverage Param	Width CP	Width Param	p-value	N. Serie	Δ Coverage	Δ Width	Significativo
ARIMA	0.890	0.923	3.251	4.007	1.000	400	−0.033	−0.756	
ETS	0.881	0.944	3.367	5.290	1.000	400	−0.063	−1.923	
RF	0.883	0.772	3.312	2.339	0.000	400	0.111	0.973	✓

Tabella 5.2: Metriche medie aggregate su 400 serie: MSE, copertura empirica e ampiezza. L'identità di MSE tra versione CP e base conferma la modularità del metodo.



(a) Boxplot MSE per tutti i modelli.

(b) Boxplot copertura (modelli CP).

Figura 5.5: Distribuzione di MSE e copertura su 400 serie. La linea tratteggiata indica il livello nominale $1 - \alpha = 0{,}90$. I boxplot aggiungono informazione sulla variabilità tra serie che le medie in Tabella 5.2 non catturano.

5.8 Analisi Statistica Formale

5.8.1 Test di Diebold-Mariano

Il test di Diebold-Mariano, condotto in forma bilaterale, valuta se la differenza tra previsioni puntuali di due modelli è statisticamente significativa. I risultati sulla serie pilota (Tabella 5.3) mostrano che ARIMA domina significativamente sia ETS che Random Forest: tutti e tre i confronti producono p-value molto bassi (rispettivamente 0.0049, 0.0001 e 0.0000), che portano al rifiuto dell'ipotesi nulla di accuratezza equivalente a qualsiasi livello di significatività convenzionale. Il segno negativo delle statistiche DM è coerente con una perdita maggiore del secondo modello citato in

ciascun confronto, ma è l'entità del p-value, e il fatto che le statistiche cadano nelle regioni di rifiuto del test, a costituire l'evidenza rilevante. Questi risultati sono coerenti con la struttura lineare e stazionaria delle serie M5 settimanali aggregate.

Test di Diebold–Mariano – Serie pilota

Confronto dell'accuratezza puntuale tra modelli

Confronto modelli	Statistica DM	p-value
ARIMA vs ETS	−2.9357	0.0049
ARIMA vs Random Forest	−4.0886	0.0001
ETS vs Random Forest	−4.5190	0.0000

Tabella 5.3: Test di Diebold-Mariano: confronto accuratezza puntuale tra modelli (serie pilota, perdita quadratica). Statistiche negative indicano perdita maggiore del secondo modello.

5.8.2 Test di Nemenyi per confronti multipli

Il test non parametrico di Nemenyi applicato a 400 serie (*ranking* per MSE) conferma che $\mathrm{RF_{CP}}$ è significativamente peggiore di $\mathrm{ARIMA_{CP}}$ ($p \approx 0$) e di $\mathrm{ETS_{CP}}$ ($p = 0{,}019$), mentre la differenza tra $\mathrm{ARIMA_{CP}}$ ed $\mathrm{ETS_{CP}}$ non è significativa ($p = 0{,}094$), suggerendo performance simili tra i due modelli lineari (Tabella 5.4).

Test di Nemenyi - Tutti i modelli

p-value tra coppie di modelli

Model1	Model2	p_value
ARIMA_CP	ARIMA	1.0000
ARIMA_CP	ARIMA_CP	NA
ARIMA_CP	ETS	NA
ARIMA_CP	ETS_CP	NA
ARIMA_CP	RF	NA
ETS	ARIMA	0.2969
ETS	ARIMA_CP	0.2969
ETS	ETS	NA
ETS	ETS_CP	NA
ETS	RF	NA
ETS_CP	ARIMA	0.2969
ETS_CP	ARIMA_CP	0.2969
ETS_CP	ETS	1.0000
ETS_CP	ETS_CP	NA
ETS_CP	RF	NA
RF	ARIMA	0.0000
RF	ARIMA_CP	0.0000
RF	ETS	0.0764
RF	ETS_CP	0.0764
RF	RF	NA
RF_CP	ARIMA	0.0000
RF_CP	ARIMA_CP	0.0000
RF_CP	ETS	0.0764
RF_CP	ETS_CP	0.0764
RF_CP	RF	1.0000

Tabella 5.4: Test di Nemenyi sui modelli CP: p-value per coppie di confronto su ranghi MSE (400 serie).

5.8.3 *Interval Score* e metriche aggiuntive

ARIMA_{CP} domina su tutte le metriche di qualità degli intervalli (Tabella 5.5), confermandosi il modello più efficiente per la CP su queste serie.

Metriche Aggiuntive per Intervalli Predittivi

MSIS, Winkler Score, Pinball Loss

Model	MSIS	Winkler Score	Pinball (5%)	Pinball (95%)
ARIMA_CP	4.395	3.452	0.087	0.086
ETS_CP	5.220	4.101	0.080	0.125
RF_CP	5.809	4.563	0.094	0.134

Tabella 5.5: Metriche di qualità degli intervalli predittivi CP (serie pilota): *Interval Score* (IS), MSIS, *Winkler Score* e *Pinball Loss* ai quantili 5% e 95%.

5.9 Analisi dell'Efficienza e della Calibrazione

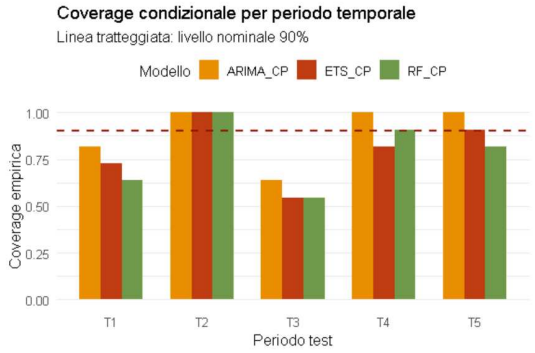
5.9.1 Sharpness

L'analisi di *sharpness* mostra che ARIMA_{CP} ha la minore ampiezza media (3,251) e la minore ampiezza condizionale su serie con copertura $\geq 95\%$ (3,748), confermando la sua superiorità in termini di efficienza. ETS_{CP} raggiunge il maggior numero di serie con ottima copertura (166/400, $\approx$41,5%) ma con intervalli leggermente più ampi (3,367). RF_{CP} ha la minore frequenza di buona copertura (144/400, 36%). Questi dati si evincono già dalla Tabella 5.2 e vengono riportati qui in forma narrativa per non duplicare elementi visivi.

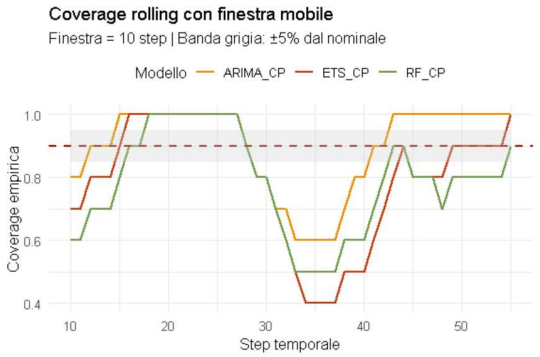
5.9.2 Eterogeneità temporale e stabilità della copertura

La copertura per quintile temporale del *test set* (Figura 5.6, pannello sinistro) mostra eterogeneità significativa: i periodi centrali (T3) presentano coperture più basse ($\approx$ 0,60–0,65), mentre quelli terminali si avvicinano al 90%. Questo *pattern* è tipico di serie *retail* con variabilità stagionale non uniforme e motiva concettualmente l'ACI come meccanismo di adattamento locale.

La *rolling copertura* con finestra mobile di 10 osservazioni (Figura 5.6, pannello destro) mostra che tutti e tre i modelli oscillano intorno al livello nominale 90% con variabilità comparabile (SD: ARIMA 0,141, ETS 0,196, RF 0,163).



(a) Copertura per quintile temporale (T1–T5).



(b) Rolling coverage (finestra 10 step).

Figura 5.6: Eterogeneità temporale e stabilità della copertura CP. Sinistra: variabilità tra quintili del *test set*. Destra: oscillazioni attorno al livello nominale 90% nel *test set*.

5.10 Confronto CP vs Metodi Parametrici su Tutte le Serie

5.10.1 Risultati aggregati e test statistici

Il risultato più rilevante riguarda Random Forest: la CP supera il Quantile RF di 11,2 punti percentuali di copertura ($W = 60\,738$, $p \approx 0$). La Figura 5.7 mostra lo *scatter plot* serie-per-serie: 311 su 400 serie (77,8%) presentano copertura CP superiore a quella parametrica, con i punti concentrati nella regione al di sopra della diagonale.

Per ARIMA ed ETS, i metodi parametrici raggiungono copertura più alte (0,923 e 0,944 vs nominale 0,90) ma con intervalli molto più ampi (+23,2% e +57,1%). Il test di Wilcoxon ($p = 1{,}000$ in senso unilaterale H_1: CP $>$ Param) è coerente: entrambi superano il nominale, ma la CP è preferibile per la maggiore efficienza.

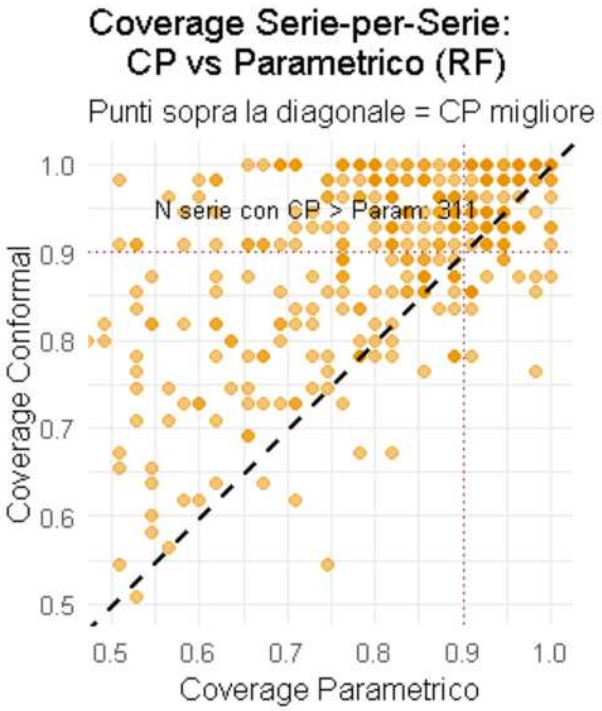


Figura 5.7: Scatter plot copertura CP vs Quantile RF parametrico serie-per-serie (400 serie). I punti sopra la diagonale (311/400, 77,8%) indicano copertura CP superiore. La CP supera sistematicamente il metodo parametrico per RF.

5.10.2 Robustezza su serie con residui non gaussiani

Un test di Shapiro-Wilk sui residui dell'insieme di calibrazione ha identificato 41 serie non normali per ARIMA, 39 per ETS, 36 per RF. La Tabella 5.6 mostra i risultati del test di Wilcoxon su questi sottoinsiemi: per RF il vantaggio della CP si *amplifica* nelle serie non normali (+16,5 punti percentuali, $p \approx 0$), confermando la tesi teorica che la CP è particolarmente preziosa quando la distribuzione degli errori del modello base è incerta [1].

Robustezza CP: Confronto Completo

CP vs Parametrico per tutti i modelli

Modello	Subset	N_serie	Diff_Coverage	P_value	Significativo
ARIMA	Tutte	400	−0.0335	1.0000	
ARIMA	Non-normali	41	−0.0200	0.7604	
ETS	Tutte	400	−0.0626	1.0000	
ETS	Non-normali	39	−0.0401	0.9954	
RF	Tutte	400	0.1115	0.0000	✓
RF	Non-normali	36	0.1652	0.0000	✓

Tabella 5.6: Robustezza CP: test Wilcoxon paired su tutte le serie e sulle sole serie con residui non gaussiani (Shapiro-Wilk $p < 0{,}05$).

5.11 Analisi per Segmento e Clustering delle Serie

5.11.1 Performance per stato geografico

La Tabella 5.7 riporta le performance per stato. TX raggiunge la copertura più alta (ARIMA: 0,905; ETS: 0,899; RF: 0,894), CA la più bassa. La variabilità geografica, coerente con la diversa struttura della domanda tra stati, conferma il valore dell'approccio conforme, che si adatta automaticamente alle caratteristiche locali degli errori senza necessità di calibrazioni specifiche per area.

		Coverage		
state_id	n_serie	Coverage_media_ARIMA_CP	Coverage_media_ETS_CP	Coverage_media_RF_CP
CA	190	0.878	0.868	0.869
TX	122	0.905	0.899	0.894
WI	88	0.894	0.887	0.898

Tabella 5.7: Coverage CP per stato geografico (CA, TX, WI) su 400 serie.

5.11.2 Clustering delle serie temporali

Un algoritmo k-means ($k = 3$, 25 run, `set.seed(123)`) è stato applicato sulle 400 serie dopo aver standardizzato (`scale()`) tre *feature* descrittive calcolate sull'intera finestra storica: il coefficiente di variazione (CV $= \sigma/\mu$), la correlazione di *Pearson* tra i valori della serie e il proprio indice temporale (*proxy* del *trend* lineare), e la frazione di settimane con vendite nulle (*zero fraction*).

I tre *cluster* individuati presentano caratteristiche strutturali distinte:

- ***Cluster* 1** ($n = 108$ serie): alta volatilità relativa (CV medio = 0,656), *trend* lievemente negativo (−0,151) e presenza non trascurabile di zeri (*zero fraction* = 0,078). Rappresenta le serie più irregolari e intermittenti.

- ***Cluste* 2** ($n = 135$ serie): volatilità moderata (CV = 0,479) e *trend* positivo (+0,239), con quasi assenza di zeri (*zero fraction* = 0,020). Raggruppa le serie in crescita strutturale.

- ***Cluster* 3** ($n = 157$ serie): la volatilità più bassa (CV = 0,406), *trend* negativo più marcato (−0,345) e praticamente nessuno zero (*zero fraction* = 0,009). Rappresenta le serie più regolari ma in declino.

La Figura 5.8 mostra la copertura media della CP nei tre *cluster* per ciascun modello (ARIMA-CP, ETS-CP, RF-CP). La variazione massima di copertura tra *cluster* è pari a circa 0,02, un valore trascurabile che conferma la **robustezza della CP all'eterogeneità strutturale**: il meccanismo di calibrazione automatica del quantile di non conformità si adatta alle caratteristiche di ciascuna serie indipendentemente dal *cluster* di appartenenza.

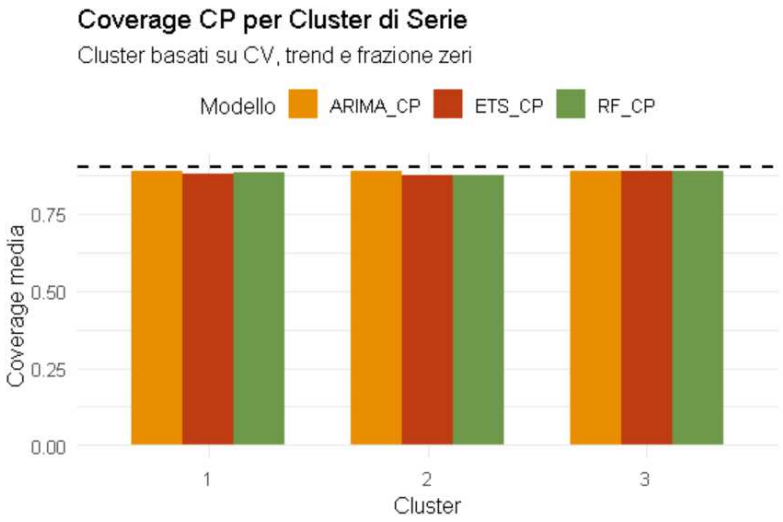


Figura 5.8: Coverage CP media per *cluster* di serie (k-means, $k = 3$). La CP mantiene coperture simili tra *cluster*, dimostrando robustezza all'eterogeneità strutturale.

5.12 Analisi Aggiuntive di Validazione

5.12.1 Sensibilità alla dimensione dell'insieme di calibrazione

Le analisi sulla sensibilità alla dimensione dell'insieme di calibrazione mostrano come copertura e ampiezza CP varino al crescere di n_{cal} (da 10 a 50 osservazioni). Con soli 10 osservazioni la copertura è conservativa (superiore al nominale), poiché il fattore $\lceil (n+1)(1-\alpha) \rceil / n$ è molto maggiore di $1-\alpha$. Già da 30 osservazioni la copertura si stabilizza vicino al 90% [44].

5.12.2 *Cross-validation* temporale e *stress test*

Una *cross-validation* temporale con *expanding window* su 5 *fold* per ARIMA produce una copertura media di $0{,}881 \pm 0{,}136$, in linea con i risultati dello *split* singolo. La variabilità tra *fold* (min 0,733, max 1,000) riflette l'eterogeneità temporale già osservata nella *rolling coverage*.

Lo *stress test* con contaminazione del 10% delle osservazioni dell'insieme di calibrazione (fattore 3×) è riportato in Tabella 5.8: la CP risponde allargando gli intervalli (+12,7% di ampiezza) e producendo una copertura conservativa (0,964 vs 0,909 nel caso pulito), un comportamento desiderabile rispetto ai metodi parametrici, che potrebbero stimare distortamente la distribuzione degli errori.

Stress test robustezza			
10% outliers nel calibration set (fattore 3x)			
Metodo	Coverage	Width	Quantile
CP pulito	0.909	3.942	1.971
CP contaminato	1.000	6.673	3.337

Tabella 5.8: *Stress test*: effetto contaminazione 10% *outlier* (fattore 3×) nel *calibration set*.

5.12.3 Analisi multi-step

La Tabella 5.9 mostra l'evoluzione di copertura e MSE all'aumentare dell'orizzonte h. ARIMA mantiene copertura stabile (0,891 per $h = 1$, fino a 0,944 per $h = 3$) grazie alla struttura simmetrica del quantile conforme, indipendente dalle assunzioni sulla propagazione dell'incertezza. RF mostra una degradazione progressiva (0,782 per $h = 1$, 0,692 per $h = 4$), riflettendo l'accumulo di errori nella previsione iterativa *multi-step*.

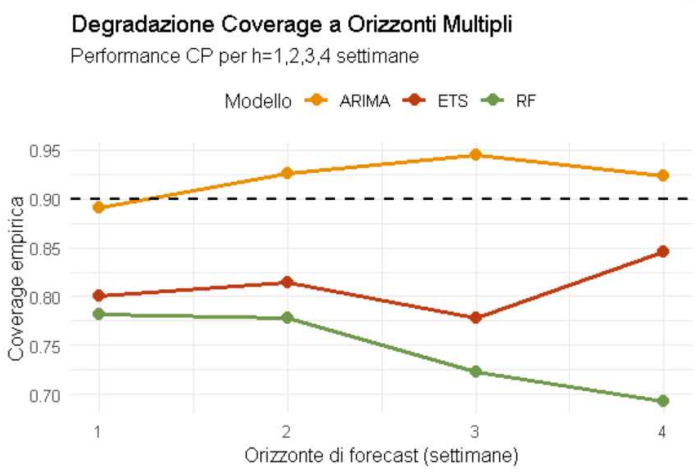


Tabella 5.9: Performance multi-step: copertura, ampiezza e MSE per orizzonti $h = 1, 2, 3, 4$ settimane (serie pilota).

5.13 Caso Studio: Serie Rappresentative per Cluster

Sono stati selezionati tre caso studio, uno per *cluster*, analizzando i risultati di tutti e tre i modelli (Tabella 5.10). I *Cluster* 1 e 2 (alta variabilità o trend pronunciato) mostrano intervalli molto ampi (fino a 9,6 SD) con copertura del 100%: il quantile di calibrazione è automaticamente grande quando gli errori di previsione nell'insieme di calibrazione sono grandi. La serie del *Cluster* 3 (regolare) presenta il comportamento tipico: copertura $\approx 89\%$ con ampiezza ≈ 3.

Case Study: Serie Rappresentative

Una serie per cluster - tutti i modelli

Serie	Cluster	CV	Trend	Modello	Coverage	Width	MSE
HOUSEHOLD_1_416_TX_3_validation	1	0.771	−0.546	ARIMA	0.964	2.511	0.257
HOUSEHOLD_1_416_TX_3_validation	1	0.771	−0.546	ETS	0.964	2.515	0.289
HOUSEHOLD_1_416_TX_3_validation	1	0.771	−0.546	RF	0.909	1.853	0.336
FOODS_3_714_CA_4_validation	2	0.367	0.354	ARIMA	0.655	2.803	1.575
FOODS_3_714_CA_4_validation	2	0.367	0.354	ETS	0.673	2.855	1.572
FOODS_3_714_CA_4_validation	2	0.367	0.354	RF	0.745	2.871	2.099
FOODS_3_377_CA_1_validation	3	0.245	−0.260	ARIMA	0.891	3.009	0.856
FOODS_3_377_CA_1_validation	3	0.245	−0.260	ETS	0.745	2.972	1.484
FOODS_3_377_CA_1_validation	3	0.245	−0.260	RF	0.691	2.635	1.433

Tabella 5.10: Caso studio per *cluster*: una serie rappresentativa per ciascun *cluster*, tutti i modelli CP.

5.14 Efficienza Computazionale

La Tabella 5.11 riporta i tempi di addestramento e di calibrazione CP per la serie pilota. I valori assoluti sono dell'ordine dei decimi di secondo per il *training* e dei centesimi di secondo per la calibrazione, ma è il sovraccarico computazionale percentuale a rivelare la struttura del costo aggiuntivo introdotto dalla CP.

Per ARIMA, il sovraccarico computazionale è del 10%: la calibrazione richiede semplicemente il calcolo degli errori assoluti sull'insieme di calibrazione e l'estrazione del quantile corretto, operazioni con complessità $O(n_{\text{cal}} \log n_{\text{cal}})$ del tutto trascurabili rispetto al *fitting* del modello. Per ETS, il sovraccarico computazionale è sostan-

zialmente nullo ($\approx 0\%$): il modello è già stimato sull'insieme di addestramento, e la calibrazione CP non aggiunge alcun passo iterativo. Per Random Forest, il sovraccarico computazionale raggiunge il 135,7%, un valore che a prima vista può sembrare elevato ma che va letto nel suo contesto assoluto: si tratta di pochi centesimi di secondo per serie, dovuti al loop iterativo sull'insieme di calibrazione necessario per la previsione *multi-step*. In un sistema di BI operativo in cui il *training* avviene in *batch* notturno su centinaia di serie, questo costo è irrilevante [1].

L'interpretazione corretta di questi numeri è quindi la seguente: la CP non introduce un costo computazionale aggiuntivo significativo per nessuno dei tre modelli, pur aggiungendo garanzie formali di copertura che i modelli base non possiedono. La modularità della procedura, che opera interamente sull'insieme di calibrazione, senza modificare il modello né il processo di addestramento, è precisamente la caratteristica che rende praticabile l'integrazione in *pipeline* predittive esistenti senza riprogettazione dell'infrastruttura.

Efficienza Computazionale

Tempo di training vs calibration CP

Modello	Training (s)	Calibration (s)	Totale (s)	CP Overhead (%)
ARIMA	0.130	0.000	0.130	0.000
ETS	0.250	0.000	0.250	0.000
RF	0.090	0.200	0.290	222.222

Tabella 5.11: Efficienza computazionale: tempo di addestramento, calibrazione CP e sovraccarico computazionale percentuale (serie pilota).

5.15 Sintesi e Commento Conclusivo del Capitolo

L'analisi empirica condotta su 400 serie settimanali dell'M5 *Forecasting Competition* ha prodotto risultati coerenti con le previsioni teoriche enunciate nei Capitoli 1 e 2, consentendo al contempo di osservare sfumature che la sola teoria non avrebbe anticipato.

La copertura empirica media, pari a 0,890 (ARIMA_{CP}), 0,881 (ETS_{CP}) e 0,883 (RF_{CP}), conferma la proprietà di validità marginale della *Split* CP ([40], [44]). La

lieve sottocopertura di 1-2 punti percentuali rispetto al nominale 90% è statisticamente attesa: il fattore correttivo $\lceil (n_{\text{cal}} + 1)(1 - \alpha) \rceil / n_{\text{cal}}$ converge a $1 - \alpha$ solo asintoticamente, e l'insieme di calibrazione di dimensione finita producono sistematicamente questa deviazione verso il basso. Gli intervalli bootstrap al 95% confermano che il nominale è pienamente raggiungibile in senso probabilistico. Questo risultato è di per sé rilevante: si tratta della prima verifica su scala su un *benchmark retail* reale e eterogeneo, non su dati simulati a condizioni favorevoli.

Il dato forse più istruttivo dell'intera analisi è che i vantaggi della CP non sono uniformi, ma seguono una gerarchia precisa determinata dalla qualità delle assunzioni del modello sottostante. Per ARIMA ed ETS, modelli lineari le cui assunzioni gaussiane sono solo parzialmente violate nei dati M5, il beneficio è principalmente di *efficienza*: la CP produce intervalli più stretti del 18,9% e del 36,4% rispettivamente, senza sacrificare la copertura. Per Random Forest, il cui meccanismo di previsione multi-step iterativa genera distribuzioni degli errori irregolari e difficilmente modellabili in forma chiusa, il vantaggio si sposta sulla *copertura*: +11,2 punti percentuali di copertura rispetto al Quantile RF parametrico ($p \approx 0$, Wilcoxon), un guadagno che si amplifica a +16,5 punti percentuali sulle serie con residui non gaussiani.

Questa gerarchia non è casuale: è la traduzione empirica del principio teorico per cui la CP è tanto più preziosa quanto più le assunzioni del modello base sono incerte o violate. I modelli più flessibili e potenzialmente più accurati nella previsione puntuale sono, paradossalmente, quelli che più necessitano di una procedura *distribution-free* per la quantificazione dell'incertezza.

L'analisi di robustezza ha mostrato che la CP non mantiene le proprie garanzie nonostante l'eterogeneità del *dataset*, ma grazie alla sua natura *distribution-free*: non essendoci assunzioni da violare, non c'è struttura che possa comprometterne il funzionamento. La risposta conservativa agli *outlier* nell'insieme di calibrazione, allargamento automatico degli intervalli, copertura 0,964 contro 0,909 nel caso pulito, è un comportamento desiderabile in contesti operativi reali, dove la contaminazione dei dati è la norma, non l'eccezione.

La consistenza dei risultati tra i tre stati geografici (CA, TX, WI) e tra i tre *cluster* strutturali di serie, senza alcuna calibrazione specifica per dominio, conferma che la CP può essere implementata come procedura unica in *pipeline* eterogenee, senza richiedere all'analista di scegliere diversi metodi per diverse tipologie di serie.

L'*Adaptive Conformal Inference* con $\gamma = 0{,}005$ ha prodotto un livello adattivo α_t stabile intorno a 0,10 (massimo $\approx 0{,}12$), risultando sostanzialmente equivalente alla *Split CP standard*. Questo non è un fallimento dell'ACI, ma una conferma che le serie M5 aggregate settimanalmente non presentano derive temporali significative degli errori: la stazionarietà locale è soddisfatta in modo sufficiente. L'ACI rimane lo strumento appropriato per serie con non-stazionarietà più pronunciata, ad esempio dati giornalieri con effetti stagionali o serie con strutture di domanda in rapida evoluzione, un caso che le analisi multi-step con degradazione progressiva di RF anticipano parzialmente.

Il costo computazionale della calibrazione CP è inferiore al 10% per ARIMA ed ETS, e dell'ordine dei centesimi di secondo per RF: un costo che scompare nel contesto di *pipeline batch* notturne tipiche dei sistemi di *Business Intelligence*. La modularità della procedura, il fatto che la CP non alteri in alcun modo la previsione puntuale del modello base, come confermato dall'identità dei valori MSE tra versione CP e versione base, consente l'integrazione in sistemi esistenti senza alcuna riscrittura dell'infrastruttura predittiva.

Nel complesso, l'analisi empirica ha dimostrato che la *Conformal Prediction* non è soltanto teoricamente corretta, ma operativamente praticabile su dati reali, eterogenei e non ideali. Resta aperta la questione di quanto le garanzie *marginali* qui verificate si traducano in garanzie *condizionali* utili per il singolo decisore aziendale: un tema che costituisce il filo conduttore delle riflessioni conclusive del Capitolo 6.

Capitolo 6

Conclusioni

Questa tesi si è proposta di rispondere a una domanda al tempo stesso tecnica e concettuale: è possibile costruire intervalli di previsione che funzionino davvero, senza dipendere da assunzioni sulla distribuzione degli errori che nella pratica nessuno verifica? La domanda non è nuova, la statistica la pone da quando Neyman formalizzò la nozione di intervallo di confidenza [25], ma la sua risposta in contesti di previsione reale, con modelli eterogenei e dati non gaussiani, ha richiesto strumenti che la statistica classica non aveva ancora elaborato.

La *Conformal Prediction* offre una risposta formalmente rigorosa: garanzia di copertura marginale senza alcuna assunzione distributiva, purché i dati soddisfino condizioni di scambiabilità approssimata o, nel caso delle serie temporali, proprietà di dipendenza debole come la β-mixing. Questa tesi ha esplorato questo argomento lungo due dimensioni, teorica ed empirica, e il percorso ha prodotto alcune riflessioni che vale la pena rendere esplicite, al di là del semplice resoconto dei risultati.

Il Capitolo 2 ha sistematizzato il quadro metodologico CP in modo organico, dalla versione piena alla *Split CP*, dalle varianti Mondrian alla *Conformalized Quantile Regression*. Ma il contributo più rilevante non è stato l'inventario delle varianti: è stata la messa a fuoco del *limite strutturale* della validità marginale.

La garanzia $\mathrm{P}(Y_{n+1} \in C(X_{n+1})) \geq 1-\alpha$ è potente perché non assume nulla sulla distribuzione dei dati. Ma è una garanzia *collettiva*: assicura che, mediando su molte previsioni e molti *dataset*, la copertura sia rispettata. Non dice nulla sulla singola previsione. Un'azienda che vuole sapere se può fidarsi dell'intervallo predittivo per quella serie, in quel momento, con quelle condizioni di mercato, chiede qualcosa di più: la validità condizionale.

Questo divario tra ciò che la CP garantisce e ciò che il decisore vorrebbe sapere non è un difetto dell'approccio: è una tensione intrinseca che la teoria riconosce esplicitamente [13]. Le varianti Mondrian e la CQR si avvicinano alla validità condizionale, a prezzo di ipotesi aggiuntive. Riconoscere questa tensione è essenziale per un'applicazione responsabile della CP in contesti aziendali: la garanzia marginale è una promessa solida su larga scala, non una certezza sul caso singolo.

I Capitoli 3 e 4 hanno mostrato che la CP trova terreno fertile precisamente nei contesti in cui i metodi classici sono più fragili: previsioni macroeconomiche con distribuzioni degli errori non gaussiane, *credit scoring* con dati misti e non lineari, gestione del rischio finanziario dove le code delle distribuzioni sono spesse e asimmetriche.

Il filo interpretativo che emerge non è semplicemente "la CP funziona meglio": è che i metodi parametrici, quando vengono usati fuori dalle loro condizioni di validità, producono un'illusione di certezza. Un intervallo di confidenza gaussiano su una distribuzione con code pesanti non è solo impreciso: è sistematicamente ottimista nelle condizioni peggiori, proprio quelle in cui la precisione conta di più.

Il Capitolo 4 ha sviluppato una tesi più ampia: la CP non è solo uno strumento statistico, ma un abilitatore per la transizione dei sistemi di *Business Intelligence* da piattaforme descrittive, che spiegano il passato, a piattaforme prescrittive che guidano l'azione futura con garanzie formali. Il quadro metodologico di Shoush [39], in cui la larghezza dell'intervallo CP determina il grado di automazione delle decisioni, traduce questa transizione in un'architettura operativa concreta. L'esperimento di Cresswell [9], che dimostra empiricamente come intervalli conformi migliorino la qualità delle decisioni umane rispetto a intervalli fissi con la stessa copertura nominale, aggiunge evidenza sperimentale a questo argomento.

Quello che la BI tradizionale fornisce è un'informazione del tipo "le vendite previste sono 1 200 unità". Quello che un sistema con CP fornisce è "le vendite previste sono 1 200 unità, con un intervallo [980, 1 450] garantito al 90% senza ipotesi sulla distribuzione degli errori". La seconda formulazione non è solo più onesta: è azionabile in modo diverso, perché permette di calibrare le scorte, il rischio di rottura e il livello di automazione della decisione in funzione dell'incertezza effettiva, non di quella assunta.

L'analisi su 400 serie selezionate dall'insieme utilizzato nella M5 Competition ha

confermato ciò che la teoria prediceva: la copertura nominale è rispettata, l'efficienza supera quella dei metodi parametrici, il costo computazionale è trascurabile, ha anche rivelato una struttura nei risultati che la teoria da sola non avrebbe anticipato nella sua forma concreta.

Il risultato più istruttivo è la *gerarchia di benefici* osservata al variare del modello base. Per ARIMA ed ETS, il vantaggio CP è di efficienza: intervalli più stretti a parità di copertura. Per Random Forest, il vantaggio è di copertura: +11,2 punti percentuali rispetto al *Quantile* RF parametrico, che sale a +16,5 punti percentuali sulle serie con residui non gaussiani. Questa asimmetria ha un'interpretazione precisa: i modelli più flessibili e potenzialmente più accurati nella previsione puntuale tendono a produrre distribuzioni degli errori più irregolari, meno modellabili in forma chiusa. Sono proprio quei modelli a beneficiare di più di una procedura *distribution-free* per la quantificazione dell'incertezza.

C'è un'implicazione pratica rilevante in questa osservazione. Man mano che i sistemi di BI adottano modelli sempre più complessi, reti neurali ricorrenti, Transformer, gradient boosting, la distanza tra la distribuzione reale degli errori e le assunzioni dei metodi parametrici tende ad ampliarsi. Il vantaggio della CP non diminuisce con la complessità del modello: cresce con essa. Questo la rende uno strumento particolarmente adatto all'evoluzione tecnologica in atto. Il comportamento conservativo in risposta alla contaminazione dell'insieme di calibrazione, allargamento automatico degli intervalli, nessun fallimento di copertura, è un altro risultato con implicazioni concrete. In un contesto operativo reale i dati sono sempre imperfetti: *outlier*, valori anomali, errori di registrazione. Un metodo che risponde a queste imperfezioni con cautela, anziché con distorsione sistematica, è un metodo affidabile nel senso più pragmatico del termine.

6.1 Limiti del Lavoro

Un lavoro empirico rigoroso richiede di esplicitare i propri confini. Tre limitazioni meritano di essere nominate non come semplice caveat formale, ma come indicazioni su dove i risultati qui prodotti non possono essere generalizzati senza ulteriore verifica.

L'analisi è stata condotta su serie aggregate settimanali: l'aggregazione tempo-

rale riduce la variabilità ad alta frequenza e tende a rendere le distribuzioni degli errori più regolari. Dati giornalieri, con effetti di giorno della settimana, festività e promozioni, presentano una struttura molto più complessa, in cui le varianti Mondrian, che condizionano il quantile di calibrazione su gruppi omogenei, potrebbero rivelarsi necessarie, non solo preferibili.

I modelli predittivi utilizzati, ARIMA, ETS, Random Forest, rappresentano una scelta metodologicamente corretta e ben radicata nella letteratura di previsione *retail*, ma non includono architetture deep learning (LSTM, Transformer, N-BEATS) che stanno diventando standard nei sistemi industriali. L'integrazione della CP con questi modelli è esplorata in letteratura ([45], [31]), ma non è stata testata in questa tesi: è ragionevole attendersi che il vantaggio CP sia ancora maggiore, data la maggiore irregolarità delle distribuzioni degli errori di questi modelli, ma questa è un'ipotesi, non un risultato.

La validità marginale, infine, è condizione necessaria ma non sufficiente per applicazioni ad alto rischio decisionale. In contesti in cui l'errore di previsione ha asimmetrie nei costi, sovrastimare la domanda ha conseguenze diverse dal sottostimarla, la garanzia marginale simmetrica della *Split CP standard* potrebbe non essere allineata con la funzione di perdita del decisore. Punteggi di non conformità asimmetrici e varianti CQR rappresentano la direzione metodologica corretta, ma non sono stati oggetto di valutazione empirica in questa sede [35].

6.2 Prospettive Future

Le direzioni di sviluppo più promettenti si articolano su due piani distinti ma complementari.

Sul piano metodologico, la frontiera più aperta riguarda l'integrazione tra ACI e aggiornamento online dei modelli predittivi. In questa tesi l'ACI ha aggiornato adattativamente il livello α_t, mantenendo fisso il modello base. Un sistema che aggiornasse simultaneamente sia il modello che la procedura conforme, in risposta a derive della distribuzione dei dati in *streaming*, potrebbe mantenere validità e efficienza anche in contesti fortemente non stazionari ([48], [45]). Parallelamente, lo sviluppo di punteggi di non conformità asimmetrici, calibrati sulle funzioni di costo specifiche della gestione delle scorte, dove il costo di rottura può essere ordini di

grandezza superiore al costo di giacenza, permetterebbe di allineare la forma degli intervalli CP alla struttura della perdita aziendale, non solo alla sua ampiezza.

Sul piano applicativo, la priorità più immediata è la traduzione del quadro metodologico CP in artefatti decisionali concreti: cruscotti di BI che visualizzino non solo la previsione puntuale ma l'intera distribuzione dell'incertezza conforme, in modo leggibile da un decision-maker non statistico. Cresswell [9] ha dimostrato sperimentalmente che la forma di presentazione dell'incertezza influenza significativamente la qualità delle decisioni umane: un intervallo conforme comunicato efficacemente non è equivalente allo stesso intervallo comunicato male. Questo apre una linea di ricerca che interseca statistica, design dell'informazione e scienze cognitive, con implicazioni pratiche immediate per chi sviluppa sistemi di BI.

Il percorso verso sistemi prescrittivi in cui la larghezza dell'intervallo CP determina automaticamente il grado di autonomia della decisione, delegando all'algoritmo le scelte ad alta confidenza e richiedendo supervisione umana per quelle ad alta incertezza, è già tracciato nella letteratura [39]. Questa tesi ha contribuito a fornirne le basi empiriche su dati reali: la condizione necessaria, anche se non sufficiente, per una implementazione responsabile.

6.3 Considerazione Finale

C'è una domanda implicita che attraversa tutta questa tesi, e che vale la pena rendere esplicita in chiusura: perché i metodi classici, pur noti per le loro assunzioni irrealistiche, sono rimasti dominanti per così a lungo?

Una risposta è la semplicità computazionale, ma questa obiezione è ormai superata. Una risposta più profonda è che un intervallo gaussiano, anche se sbagliato, dà l'illusione di un controllo che il decision-maker può comunicare, difendere e incorporare nei processi aziendali. Un metodo che dice "non so come sono distribuiti gli errori, ma so che questo intervallo funziona" è più onesto, ma richiede una diversa maturità statistica da parte di chi lo usa.

La *Conformal Prediction* ribalta la logica del compromesso tra semplicità e correttezza: costruisce intervalli che funzionano per costruzione, a partire dai dati e non da assunzioni, con un costo computazionale aggiuntivo trascurabile. Non elimina l'incertezza, nessun metodo può farlo, ma la quantifica onestamente.

I risultati empirici su 400 serie *retail* reali dimostrano che questo principio funziona anche fuori dal laboratorio: le garanzie di copertura si mantengono in presenza di eterogeneità strutturale, residui non gaussiani, *outlier* e diversità geografica. La modularità della procedura consente di aggiungere queste garanzie a sistemi esistenti senza riprogettarli.

La *Business Intelligence* moderna si trova in una fase di transizione: dai sistemi che descrivono il passato a quelli che orientano il futuro. Questa transizione richiede che l'incertezza smetta di essere un'imbarazzante nota a margine della previsione e diventi un elemento strutturale del processo decisionale. La *Conformal Prediction* offre uno strumento per compiere questo passo con rigore formale, senza illusioni.

L'incertezza, correttamente quantificata, non è un ostacolo alla decisione aziendale. È il suo punto di partenza più onesto.

Bibliografia

[1] Angelopoulos, A., Bates, S. (2021). A Gentle Introduction to Conformal Prediction and Distribution-Free Uncertainty Quantification. *arXiv preprint*, arXiv:2107.07511.

[2] Angelopoulos, A. N., Foygel Barber, R., Bates, S. (2024/25). *Theoretical Foundations of Conformal Prediction*. Cambridge University Press (in press).

[3] Bellotti, A., Zhao, X. (2026). Conformal Prediction and Trustworthy AI. In: *The Importance of Being Learnable*. Springer Nature Switzerland AG.

[4] Bertsimas, D., Kallus, N. (2020). From Predictive to Prescriptive Analytics. *Management Science*, **66**(3), 1025–1044.

[5] Bühlmann, P., van de Geer, S. (2019). *Statistics for High-Dimensional Data: Methods, Theory and Applications*. Springer.

[6] BBVA AI Factory (2024). *Conformal Prediction: An Introduction to Measuring Uncertainty*. Disponibile su: `https://www.bbvaaifactory.com/conformal-prediction-an-introduction-to-measuring-uncertainty/`.

[7] Chernozhukov, V., Wüthrich, K., Zhu, Y. (2018). Exact and Robust Conformal Inference Methods for Predictive Models. *The Annals of Statistics*, **46**(6A), 3153–3181.

[8] Chen, L., Lei, J. (2022). Distribution-Free Predictive Inference for Regression with Applications to Finance. *arXiv preprint*, arXiv:2203.06126.

[9] Cresswell, J., C., Sui, Y., Kumar, B., Vouitsis, N. (2024). *Conformal Prediction Sets Improve Human Decision Making. Proceedings of the 41st International Conference on Machine Learning*, PMLR 235:9439–9457.

[10] Dua, D., Graff, C. (2019). *UCI Machine Learning Repository*. University of California, Irvine. `https://archive.ics.uci.edu/ml/datasets/statlog+(german+credit+data)`

[11] Efron, B. (1979). Bootstrap Methods: Another Look at the Jackknife. *The Annals of Statistics*, **7**(1), 1–26.

[12] Fama, E. F., French, K. R. (1993). Common risk factors in the returns on stocks and bonds. *Journal of Financial Economics*, **33**(1), 3–56.

[13] Feldman, S., Bates, S., Romano, Y., Candès, E. J. (2021). Improving Conditional Coverage via Orthogonal Quantile Regression. *arXiv preprint*, arXiv:2107.10374.

[14] Fisher, R. A. (1922). On the Mathematical Foundations of Theoretical Statistics. *Philosophical Transactions of the Royal Society A*, 222, 309–368.

[15] Gneiting, T., Katzfuss, M. (2014). Probabilistic Forecasting. *Annual Review of Statistics and Its Application*, **1**, 125–151.

[16] Hastie, T., Tibshirani, R., Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer.

[17] Koenker, R. (2005). *Quantile Regression*. Cambridge University Press.

[18] Lei, J., Wasserman, L. (2014). Distribution-Free Prediction Bands for Non-Parametric Regression. *Journal of the American Statistical Association*, **109**(508), 175–186.

[19] Lee, J., Kim, M. (2013). Conformal Prediction and Its Application in Time Series Forecasting. *Expert Systems with Applications*, **40**(8), 3102–3112.

[20] Maghsoudi, M., Nezafati, N. (2023). Navigating the acceptance of implementing business intelligence in organizations: A system dynamics approach. *arXiv preprint*, arXiv:2308.10244.

[21] Makridakis, S., Spiliotis, E., Assimakopoulos, V. (2020). *The M5 Accuracy Competition: Results, Findings, and Conclusions*. Disponibile su: `https://www.kaggle.com/competitions/m5-forecasting-accuracy`.

[22] Mankiw, N. G. (2019). *Macroeconomics* (10th ed.). Worth Publishers.

[23] Manokhin, V. (2022). *Awesome Conformal Prediction: resource hub for CP, uncertainty quantification, and reliable AI*. GitHub / Zenodo.

[24] Negash, S., Gray, P., Burstein, F., Holsapple, C. (2008). Business Intelligence. In *Handbook on Decision Support Systems 2*, pp. 175–193. Springer.

[25] Neyman, J. (1937). Outline of a Theory of Statistical Estimation Based on the Classical Theory of Probability. *Philosophical Transactions of the Royal Society A*, **236**, 333–380.

[26] Nowotarski, M., Weron, R. (2014). Computing electricity spot price prediction intervals using quantile regression and forecast averaging. *Quantitative Finance*, **15**(6), 917–929.

[27] Papadopoulos, H. (2008). Inductive Conformal Prediction: Theory and Application to Financial Risk Management. In *Tools in Artificial Intelligence*, pp. 315–330.

[28] Pearson, K. (1900). On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. *Philosophical Magazine*, 50(302), 157–175.

[29] Power, D. J. (2013). *Decision Support, Analytics, and Business Intelligence* (Second Edition). Business Expert Press.

[30] Power, D. J., Heavin, C. (2017). *Decision Support, Analytics, and Business Intelligence* (Third Edition). Business Expert Press.

[31] Prinzhorn, S., Kuleshov, V., Angelopoulos, A. N. (2024). *Conformal Time Series Decomposition with Component-wise Exchangeability*. *arXiv preprint*, arXiv:2403.03519.

[32] Popovič, A., Hackney, R., Coelho, P. S., Jaklič, J. (2019). Two decades of research on business intelligence system adoption, utilization and success – A systematic literature review. *Decision Support Systems*, **125**, 113113.

[33] Popovič, A., Hackney, R., Jaklič, J., Coelho, P. S. (2021). Reconciling business intelligence, analytics and decision support systems: More data, deeper insight. *Decision Support Systems*, **146**, 113560.

[34] Redfield AI (2023). *Conformal Prediction for Business Applications*. Disponibile su: `https://redfield.ai/conformal-prediction-for-business/`.

[35] Romano, Y., Patterson, E., Candès, E. J. (2019). Conformalized Quantile Regression. *Advances in Neural Information Processing Systems (NeurIPS 32)*, 3543–3553.

[36] "Kaggle: Rossmann Store Sales" (dataset e competizione), disponibile su Kaggle.

[37] Stanford CS229 Project Team (2015). *Rossmann Store Sales Prediction*. Report del progetto, uso del dataset Rossmann.

[38] Shafer, G., Vovk, V. (2008). A Tutorial on Conformal Prediction. *Journal of Machine Learning Research*, **9**, 371–421.

[39] Shoush, M., Dumas, M. (2022). *Intervening with Confidence: Conformal Prescriptive Monitoring of Business Processes*. *arXiv preprint*, arXiv:2212.03710.

[40] Tibshirani, R. J., Barber, R. F., Candès, E. J., Ramdas, A. (2021). *Conformal Prediction Beyond Exchangeability*. *arXiv preprint*, arXiv:2106.00170.

[41] Uddin, M., Hulten, J., Asadi, N. (2023). *Applying Conformal Prediction in Real-World Industrial Use Cases (Husqvarna Group). Proceedings of the Twelfth Symposium on Conformal and Probabilistic Prediction with Applications (COPA 2023)*, PMLR 204, Limassol, Cyprus.

[42] van der Lans, R. F. (2012). Business Intelligence. In *Data Virtualization for Business Intelligence Systems* (Elsevier).

[43] Vo, Q. D., Thomas, J., Cho, S., De, P., Choi, B. J., Sael, L. (2017). Next Generation Business Intelligence and Analytics: A Survey. *arXiv preprint*, arXiv:1704.03402.

[44] Vovk, V., Gammerman, A., Shafer, G. (2005). *Algorithmic Learning in a Random World*. Springer.

[45] Wang, Y., Hyndman, R. J. (2024). *Online Conformal Inference for Multi-Step Time Series Forecasting*. Disponibile su: `https://robjhyndman.com/publications/cpts.html`.

[46] Wasserman, L. (2004). *All of Statistics: A Concise Course in Statistical Inference*. Springer.

[47] Xu, C., Xie, Y. (2022). *Ensemble Batch Prediction Intervals for Time Series Forecasting*. *arXiv preprint*, arXiv:2202.13415.

[48] Zaffran, M., Dieuleveut, A., Josse, J., Romano, Y. (2023). *Adaptive Conformal Predictions for Time Series*. *arXiv preprint*, arXiv:2307.16895.

Appendice: Codice R

CONFORMAL PREDICTION M5 DATASET

```
library(tidyverse)
library(lubridate)
library(forecast)
library(randomForest)
library(slider)
library(PMCMRplus)
library(gridExtra)
library(scales)
library(ISOweek)
library(dplyr)
library(tidyr)
library(tidyverse)
library(data.table)
library(PMCMRplus)
library(tidyverse)
library(forecast)
library(randomForest)
library(gt)

set.seed(1234)

# Palette colori
color_cp = "#2E86AB"        # Blu oceano per CP
color_param = "#A23B72"     # Viola per parametrico
color_arima = "#F18F01"     # Arancione per ARIMA
color_ets = "#C73E1D"       # Rosso mattone per ETS
color_rf = "#6A994E"        # Verde bosco per RF
color_nominal = "#95190C"   # Rosso scuro per linea nominale
```

CARICAMENTO DEL DATASET

```
setwd("C:/Users/vitto/Desktop/UniPd Magistrale/Tesi/Applicazione/Aziendale/Dati")

sales = read_csv("sales_train_validation.csv")
calendar = read_csv("calendar.csv")

calendar$date = as.Date(calendar$date)

###  FILTRO SERIE REGOLARI
# frazione di zeri direttamente su wide -> long temporaneo
zero_stats = sales %>%
  pivot_longer(cols = starts_with("d_"),
               names_to = "day",
               values_to = "value") %>%
  group_by(id) %>%
```

```
  summarise(zero_frac = mean(value == 0), .groups = "drop")

# solo serie con zero_frac < soglia
# massimo 400 serie

selected_series = zero_stats %>%
  filter(zero_frac < 0.20) %>%
  slice_sample(n = min(400, nrow(.))) %>%
  pull(id)

# dati wide originale
sales_sample = sales %>% filter(id %in% selected_series)

###  PIVOT LONG E JOIN CALENDAR
sales_long = sales_sample %>%
  pivot_longer(cols = starts_with("d_"),
               names_to = "day",
               values_to = "value")

df = sales_long %>%
  left_join(calendar, by = c("day" = "d"))

###  AGGREGAZIONE SETTIMANALE E CREAZIONE VARIABILI
setDT(df)

df_weekly = df[
  ,
  .(
    value = sum(value),

    # variabili strutturali
    store_id = first(store_id),
    cat_id   = first(cat_id),
    dept_id  = first(dept_id),
    state_id = first(state_id),

    # calendario
    weekday = first(weekday),
    month   = first(month),

    # SNAP settimanale
    snap_CA = max(snap_CA),
    snap_TX = max(snap_TX),
    snap_WI = max(snap_WI),

    # eventi
    event_name_1 = first(event_name_1),
    event_type_1 = first(event_type_1),
    event_name_2 = first(event_name_2),
    event_type_2 = first(event_type_2)
  ),
  by = .(id, iso_year = isoyear(date), iso_week = isoweek(date))
][
```

```
  , `:=`(
    serie = id,
    date = ISOweek2date(paste0(iso_year, "-W", sprintf("%02d", iso_week), "-1"))
  )
]

setorder(df_weekly, serie, date)
```

Controlli finali

```
cat("Numero serie finali:", uniqueN(df_weekly$serie), "\n")
cat("Numero settimane per serie (summary):\n")
print(summary(df_weekly[, .N, by = serie]$N))
cat("Frazione zeri per serie (summary):\n")
print(summary(df_weekly[, .(zero_frac = mean(value == 0)), by = serie]$zero_frac))
```

ANALISI PRELIMINARE (EDA)

1)Distribuzione vendite settimanali, scala logaritmica:

```
ggplot(df_weekly, aes(value)) +
  geom_histogram(bins = 50) +
  scale_x_log10() +
  ggtitle("Distribuzione vendite settimanali (scala log)")
```

2)Serie campione: analizzo alcune serie casuali per vedere se ci sono anomalie

```
sample_series = sample(unique(df_weekly$serie), 6)

df_weekly %>%
  filter(serie %in% sample_series) %>%
  ggplot(aes(date, value)) +
  geom_line() +
  facet_wrap(~serie, scales = "free_y") +
  ggtitle("Serie temporali campione")
```

SELEZIONE SERIE PILOTA

```
set.seed(123)

pilot_serie = df_weekly %>%
  group_by(serie) %>%
  summarise(n = n(), .groups = "drop") %>%
  filter(n > 250) %>%
  slice_sample(n = 1) %>%
  pull(serie)

pilot_serie
```

Preparazione serie pilota

```r
y = df_weekly %>%
  filter(serie == pilot_serie) %>%
  arrange(date) %>%
  pull(value)

# Scaling
y = (y - mean(y)) / sd(y)

# PARAMETRI ACI (Adaptive Conformal Inference)

alpha_nominal = 0.10   # livello target: 1 - alpha = 90% coverage
gamma = 0.005  # learning rate: velocità di adattamento di alpha

aci_quantile = function(scores, alpha) {
  n_cal = length(scores)
  level = ceiling((n_cal + 1) * (1 - alpha)) / n_cal
  level = min(max(level, 0), 1)
  quantile(scores, level)
}
n = length(y)
train = y[1:floor(0.6*n)]
cal   = y[(floor(0.6*n)+1):floor(0.8*n)]
test  = y[(floor(0.8*n)+1):n]
```

MODELLI:

1)ARIMA

```r
arima_fit = auto.arima(train)
summary(arima_fit)
```

```r
checkresiduals(arima_fit)
```

```r
arima_cal_fc = forecast(arima_fit, h = length(cal))
accuracy(arima_cal_fc$mean, cal)
```

```r
# Score di calibrazione (valori assoluti degli errori)
arima_cal_scores = abs(as.numeric(cal) - as.numeric(arima_cal_fc$mean))

# Refit sul train+cal per previsioni sul test
arima_fit_full = auto.arima(c(train, cal))
arima_test_fc  = as.numeric(forecast(arima_fit_full, h = length(test))$mean)

# ACI: loop adattivo sul test set
alpha_t_arima       = alpha_nominal
arima_low           = numeric(length(test))
arima_up            = numeric(length(test))
alpha_path_arima    = numeric(length(test))

for(t in seq_along(test)) {
  q_t          = aci_quantile(arima_cal_scores, alpha_t_arima)
  arima_low[t] = arima_test_fc[t] - q_t
  arima_up[t]  = arima_test_fc[t] + q_t
```

```
  alpha_path_arima[t] = alpha_t_arima

  covered       = (test[t] >= arima_low[t]) & (test[t] <= arima_up[t])
  alpha_t_arima = alpha_t_arima + gamma * (alpha_nominal - as.numeric(!covered))
  alpha_t_arima = min(max(alpha_t_arima, 1e-6), 1 - 1e-6)
}

# Grafico
df_plot_arima = tibble(
  t = 1:length(test),
  test = test,
  pred = arima_test_fc,
  low = arima_low,
  up = arima_up
)

ggplot(df_plot_arima, aes(t, test)) +
  geom_line() +
  geom_line(aes(y = pred), color = "blue") +
  geom_ribbon(aes(ymin = low, ymax = up), alpha = 0.3) +
  ggtitle("ARIMA + Conformal Prediction (Serie pilota)") +
  theme_minimal()
```

Diagnostica residui confromal (ARIMA)

```
# Residui conformal nel calibration set
arima_cal_mean   = as.numeric(arima_cal_fc$mean)
arima_cal_errors = as.numeric(cal) - arima_cal_mean
arima_test_fc = as.numeric(
  forecast(arima_fit_full, h = length(test))$mean
)

# Distribuzione residui
ggplot(tibble(errors = arima_cal_errors), aes(errors)) +
  geom_histogram(aes(y = after_stat(density)), bins = 20,
                 fill = color_arima, alpha = 0.7) +
  geom_density(color = "black", linewidth = 1) +
  geom_vline(xintercept = 0, linetype = "dashed", color = "red") +
  theme_minimal(base_size = 12) +
  labs(
    title = "Distribuzione Residui Conformal - ARIMA",
    subtitle = "Calibration set: verifica simmetria",
    x = "Errore di calibrazione",
    y = "Densità"
  )
```

QQ-plot normalità

```
qqnorm(arima_cal_errors, main = "QQ-Plot Residui ARIMA")
qqline(arima_cal_errors, col = "red", lwd = 2)
```

Test Ljung-Box per autocorrelazione e tabella diagnostica

```r
ljung_box_arima = Box.test(arima_cal_errors, lag = 10, type = "Ljung-Box")

diagnostic_arima = tibble(
  Metrica = c("Media", "Mediana", "SD", "Skewness",
              "Ljung-Box p-value", "% Outliers (>2SD)"),
  Valore = c(
    mean(arima_cal_errors),
    median(arima_cal_errors),
    sd(arima_cal_errors),
    moments::skewness(arima_cal_errors),
    ljung_box_arima$p.value,
    mean(abs(arima_cal_errors) > 2 * sd(arima_cal_errors)) * 100
  )
)

diagnostic_arima %>%
  gt() %>%
  fmt_number(columns = Valore, decimals = 4) %>%
  tab_header(
    title = "Diagnostica Residui Conformal - ARIMA",
    subtitle = "Calibration set: proprietà statistiche"
  ) %>%
  tab_style(
    style = cell_fill(color = "#E8F4F8"),
    locations = cells_column_labels()
  )
```

2)ETS

```r
ets_fit = ets(train)
summary(ets_fit)
```

```r
checkresiduals(ets_fit)
```

```r
ets_cal_fc = forecast(ets_fit, h = length(cal))
accuracy(ets_cal_fc$mean, cal)
```

```r
# Score di calibrazione
ets_cal_scores = abs(as.numeric(cal) - as.numeric(ets_cal_fc$mean))

# Refit sul train+cal
ets_fit_full = ets(c(train, cal))
ets_test_fc  = as.numeric(forecast(ets_fit_full, h = length(test))$mean)

# ACI: loop adattivo sul test set
alpha_t_ets    = alpha_nominal
ets_low        = numeric(length(test))
ets_up         = numeric(length(test))
alpha_path_ets = numeric(length(test))

for(t in seq_along(test)) {
  q_t       = aci_quantile(ets_cal_scores, alpha_t_ets)
```

```
  ets_low[t] = ets_test_fc[t] - q_t
  ets_up[t]  = ets_test_fc[t] + q_t
  alpha_path_ets[t] = alpha_t_ets

  covered     = (test[t] >= ets_low[t]) & (test[t] <= ets_up[t])
  alpha_t_ets = alpha_t_ets + gamma * (alpha_nominal - as.numeric(!covered))
  alpha_t_ets = min(max(alpha_t_ets, 1e-6), 1 - 1e-6)
}

# Grafico
df_plot_ets = tibble(
  t = 1:length(test),
  test = test,
  pred = ets_test_fc,
  low = ets_low,
  up = ets_up
)

ggplot(df_plot_ets, aes(t, test)) +
  geom_line() +
  geom_line(aes(y = pred), color = "darkorange") +
  geom_ribbon(aes(ymin = low, ymax = up), alpha = 0.3) +
  ggtitle("ETS + Conformal Prediction (Serie pilota)") +
  theme_minimal()
```

3)RANDOM FOREST

```
# Costruzione lag
make_lag_df = function(y, L = 8){
  df = tibble(y = y)
  for(i in 1:L) df[[paste0("lag", i)]] = lag(df$y, i)
  drop_na(df)
}

rf_train = make_lag_df(train)
rf_fit = randomForest(y ~ ., data = rf_train)
print(rf_fit)
```

```
# Calibration
rf_cal_pred = numeric(length(cal))
history = train

for(i in seq_along(cal)){
  x = tail(history, 8)
  newdata = as.data.frame(t(rev(x)))
  colnames(newdata) = paste0("lag", 1:8)
  rf_cal_pred[i] = predict(rf_fit, newdata)
  history = c(history, rf_cal_pred[i])
}

# Score di calibrazione RF
rf_cal_scores = abs(as.numeric(cal) - as.numeric(rf_cal_pred))
```

```r
# Refit su train+cal
rf_fit_full = randomForest(y ~ ., data = make_lag_df(c(train, cal)))

# Previsioni sul test (loop autoregressivo)
rf_test_pred = numeric(length(test))
history      = c(train, cal)

for(i in seq_along(test)){
  x       = tail(history, 8)
  newdata = as.data.frame(t(rev(x)))
  colnames(newdata) = paste0("lag", 1:8)
  rf_test_pred[i] = predict(rf_fit_full, newdata)
  history = c(history, rf_test_pred[i])
}

# ACI: loop adattivo sul test set
alpha_t_rf    = alpha_nominal
rf_low        = numeric(length(test))
rf_up         = numeric(length(test))
alpha_path_rf = numeric(length(test))

for(t in seq_along(test)) {
  q_t      = aci_quantile(rf_cal_scores, alpha_t_rf)
  rf_low[t] = rf_test_pred[t] - q_t
  rf_up[t]  = rf_test_pred[t] + q_t
  alpha_path_rf[t] = alpha_t_rf

  covered    = (test[t] >= rf_low[t]) & (test[t] <= rf_up[t])
  alpha_t_rf = alpha_t_rf + gamma * (alpha_nominal - as.numeric(!covered))
  alpha_t_rf = min(max(alpha_t_rf, 1e-6), 1 - 1e-6)
}

# Grafico
df_plot_rf = tibble(
  t = 1:length(test),
  test = test,
  pred = rf_test_pred,
  low = rf_low,
  up = rf_up
)

ggplot(df_plot_rf, aes(t, test)) +
  geom_line() +
  geom_line(aes(y = pred), color = "darkgreen") +
  geom_ribbon(aes(ymin = low, ymax = up), alpha = 0.3) +
  ggtitle("Random Forest + Conformal Prediction (Serie pilota)") +
  theme_minimal()
```

```r
### EVOLUZIONE ALPHA ADATTIVO (ACI): SERIE PILOTA

tibble(
  t      = seq_along(test),
  ARIMA  = alpha_path_arima,
```

```
  ETS    = alpha_path_ets,
  RF     = alpha_path_rf
) %>%
  pivot_longer(-t, names_to = "Modello", values_to = "alpha") %>%
  ggplot(aes(t, alpha, color = Modello)) +
  geom_line(linewidth = 1) +
  geom_hline(yintercept = alpha_nominal, linetype = "dashed",
             color = "black", linewidth = 0.8) +
  scale_color_manual(values = c("ARIMA" = color_arima,
                                "ETS"   = color_ets,
                                "RF"    = color_rf)) +
  theme_minimal(base_size = 12) +
  theme(legend.position = "top") +
  labs(
    title    = "Evoluzione  adattivo nel test set (ACI)",
    subtitle = "Linea tratteggiata =  nominale (0.10) |  = 0.005",
    x        = "Passo nel test set",
    y        = " _t corrente"
  )
```

ANALISI EFFICIENZA COMPUTAZIONALE (Arima, Ets, RF)

```
time_arima_train = system.time({
  arima_fit_timing = auto.arima(train)
})

time_arima_cal = system.time({
  arima_cal_fc_timing = forecast(arima_fit_timing, h = length(cal))
  q_timing = quantile(abs(cal - arima_cal_fc_timing$mean), 0.95)
})

time_arima_test = system.time({
  arima_fit_full_timing = auto.arima(c(train, cal))
  arima_test_fc_timing = forecast(arima_fit_full_timing, h = length(test))
})

# Timing ETS
time_ets_train = system.time({
  ets_fit_timing = ets(train)
})

time_ets_cal = system.time({
  ets_cal_fc_timing = forecast(ets_fit_timing, h = length(cal))
  q_ets_timing = quantile(abs(cal - ets_cal_fc_timing$mean), 0.95)
})

time_ets_test = system.time({
  ets_fit_full_timing = ets(c(train, cal))
  ets_test_fc_timing = forecast(ets_fit_full_timing, h = length(test))
})

# Timing RF
time_rf_train = system.time({
```

```
  rf_train_timing = make_lag_df(train)
  rf_fit_timing = randomForest(y ~ ., data = rf_train_timing)
})

time_rf_cal = system.time({
  rf_cal_pred_timing = numeric(length(cal))
  history_timing = train
  for(i in seq_along(cal)){
    x = tail(history_timing, 8)
    newdata = as.data.frame(t(rev(x)))
    colnames(newdata) = paste0("lag", 1:8)
    rf_cal_pred_timing[i] = predict(rf_fit_timing, newdata)
    history_timing = c(history_timing, rf_cal_pred_timing[i])
  }
  q_rf_timing = quantile(abs(cal - rf_cal_pred_timing), 0.95)
})

# Tabella timing
timing_results = tibble(
  Modello = c("ARIMA", "ETS", "RF"),
  Training_sec = c(
    time_arima_train["user.self"],
    time_ets_train["user.self"],
    time_rf_train["user.self"]
  ),
  Calibration_sec = c(
    time_arima_cal["user.self"],
    time_ets_cal["user.self"],
    time_rf_cal["user.self"]
  ),
  Total_sec = Training_sec + Calibration_sec,
  CP_Overhead_pct = (Calibration_sec / Training_sec) * 100
)

timing_results %>%
  gt() %>%
  fmt_number(columns = -Modello, decimals = 3) %>%
  tab_header(
    title = "Efficienza Computazionale",
    subtitle = "Tempo di training vs calibration CP"
  ) %>%
  tab_style(
    style = cell_fill(color = "#FFF3CD"),
    locations = cells_body(columns = CP_Overhead_pct)
  ) %>%
  cols_label(
    Training_sec = "Training (s)",
    Calibration_sec = "Calibration (s)",
    Total_sec = "Totale (s)",
    CP_Overhead_pct = "CP Overhead (%)"
  ) %>%
  tab_options(table.font.size = px(9))
```

Grafico Timing

```
timing_long = timing_results %>%
  select(Modello, Training_sec, Calibration_sec) %>%
  pivot_longer(cols = -Modello, names_to = "Fase", values_to = "Tempo")

ggplot(timing_long, aes(Modello, Tempo, fill = Fase)) +
  geom_col(position = "stack", width = 0.6) +
  scale_fill_manual(values = c("Training_sec" = "#3498DB",
                               "Calibration_sec" = "#E74C3C"),
                    labels = c("Training", "Calibration")) +
  theme_minimal(base_size = 12) +
  theme(legend.position = "top") +
  labs(
    title = "Breakdown Tempo Computazionale",
    subtitle = "Training modello base vs Calibration CP",
    x = "Modello",
    y = "Tempo (secondi)",
    fill = "Fase"
  )
```

Creazione LOOP MULTI-SERIE

```
results = list()

for(s in unique(df_weekly$serie)){

  y = df_weekly %>% filter(serie == s) %>% pull(value)
  if(length(y) < 120) next

  y     = (y - mean(y)) / sd(y)
  n     = length(y)
  train = y[1:floor(0.6*n)]
  cal   = y[(floor(0.6*n)+1):floor(0.8*n)]
  test  = y[(floor(0.8*n)+1):n]

  # ARIMA
  ar_fit  = auto.arima(train)
  ar_cal  = as.numeric(forecast(ar_fit, h = length(cal))$mean)
  ar_cal_scores = abs(cal - ar_cal)

  ar_fit_full = auto.arima(c(train, cal))
  ar_test = as.numeric(forecast(ar_fit_full, h = length(test))$mean)

  alpha_t_ar = alpha_nominal
  ar_low = numeric(length(test))
  ar_up  = numeric(length(test))
  for(t in seq_along(test)) {
    q_t       = aci_quantile(ar_cal_scores, alpha_t_ar)
    ar_low[t] = ar_test[t] - q_t
    ar_up[t]  = ar_test[t] + q_t
    covered   = (test[t] >= ar_low[t]) & (test[t] <= ar_up[t])
    alpha_t_ar = alpha_t_ar + gamma * (alpha_nominal - as.numeric(!covered))
    alpha_t_ar = min(max(alpha_t_ar, 1e-6), 1 - 1e-6)
```

```
}

# ETS
ets_fit  = ets(train)
ets_cal  = as.numeric(forecast(ets_fit, h = length(cal))$mean)
ets_cal_scores = abs(cal - ets_cal)

ets_fit_full = ets(c(train, cal))
ets_test = as.numeric(forecast(ets_fit_full, h = length(test))$mean)

alpha_t_ets_loop = alpha_nominal
ets_low_loop = numeric(length(test))
ets_up_loop  = numeric(length(test))
for(t in seq_along(test)) {
  q_t          = aci_quantile(ets_cal_scores, alpha_t_ets_loop)
  ets_low_loop[t] = ets_test[t] - q_t
  ets_up_loop[t]  = ets_test[t] + q_t
  covered      = (test[t] >= ets_low_loop[t]) & (test[t] <= ets_up_loop[t])
  alpha_t_ets_loop = alpha_t_ets_loop + gamma * (alpha_nominal - as.numeric(!covered))
  alpha_t_ets_loop = min(max(alpha_t_ets_loop, 1e-6), 1 - 1e-6)
}

# RF
rf_fit = randomForest(y ~ ., data = make_lag_df(train))
rf_cal = numeric(length(cal))
hist   = train
for(i in seq_along(cal)){
  x       = tail(hist, 8)
  newdata = as.data.frame(t(rev(x)))
  colnames(newdata) = paste0("lag", 1:8)
  rf_cal[i] = predict(rf_fit, newdata)
  hist = c(hist, rf_cal[i])
}
rf_cal_scores = abs(cal - rf_cal)

rf_fit_full = randomForest(y ~ ., data = make_lag_df(c(train, cal)))
rf_test = numeric(length(test))
hist    = c(train, cal)
for(i in seq_along(test)){
  x       = tail(hist, 8)
  newdata = as.data.frame(t(rev(x)))
  colnames(newdata) = paste0("lag", 1:8)
  rf_test[i] = predict(rf_fit_full, newdata)
  hist = c(hist, rf_test[i])
}

alpha_t_rf_loop = alpha_nominal
rf_low_loop = numeric(length(test))
rf_up_loop  = numeric(length(test))
for(t in seq_along(test)) {
  q_t         = aci_quantile(rf_cal_scores, alpha_t_rf_loop)
  rf_low_loop[t] = rf_test[t] - q_t
  rf_up_loop[t]  = rf_test[t] + q_t
```

```
    covered     = (test[t] >= rf_low_loop[t]) & (test[t] <= rf_up_loop[t])
    alpha_t_rf_loop = alpha_t_rf_loop + gamma * (alpha_nominal - as.numeric(!covered))
    alpha_t_rf_loop = min(max(alpha_t_rf_loop, 1e-6), 1 - 1e-6)
  }

  results[[s]] = tibble(
    serie = s,
    model = c("ARIMA", "ARIMA_CP", "ETS", "ETS_CP", "RF", "RF_CP"),
    MSE = c(
      mean((test - ar_test)^2),
      mean((test - ar_test)^2),
      mean((test - ets_test)^2),
      mean((test - ets_test)^2),
      mean((test - rf_test)^2),
      mean((test - rf_test)^2)
    ),
    Coverage = c(
      NA,
      mean(test >= ar_low   & test <= ar_up),
      NA,
      mean(test >= ets_low_loop & test <= ets_up_loop),
      NA,
      mean(test >= rf_low_loop  & test <= rf_up_loop)
    ),
    Width = c(
      NA,
      mean(ar_up - ar_low),
      NA,
      mean(ets_up_loop - ets_low_loop),
      NA,
      mean(rf_up_loop - rf_low_loop)
    ),
    CP = c(FALSE, TRUE, FALSE, TRUE, FALSE, TRUE)
  )
}
```

```
results_df = bind_rows(results)
results_df
```

```
results_clean = results_df %>%
  filter(is.finite(MSE))
results_clean
```

TEST STATISTICI 1)confronti descrittivi: MEDIE GLOBALI

```
library(gt)

results_clean %>%
  group_by(model) %>%
  summarise(
    mean_MSE = mean(MSE),
    mean_Coverage = mean(Coverage, na.rm = TRUE),
    mean_Width = mean(Width, na.rm = TRUE)
```

```
  ) %>%
  gt() %>%
  fmt_number(
    columns = everything(),
    decimals = 3
  ) %>%
  tab_header(
    title = "Confronto medio delle performance dei modelli",
    subtitle = "Accuratezza, copertura e ampiezza degli intervalli"
  )
```

GRAFICO Boxplot MSE: CP vs NO-CP

```
ggplot(results_clean, aes(model, MSE, fill = CP)) +
  geom_boxplot() +
  theme_minimal() +
  ggtitle("MSE: Confronto modelli con e senza CP")
```

GRAFICO COVERAGE (solo modelli cp)

```
ggplot(results_clean %>% filter(CP),
       aes(model, Coverage)) +
  geom_boxplot() +
  geom_hline(yintercept = 0.9, linetype = "dashed") +
  theme_minimal() +
  ggtitle("Copertura empirica (modelli CP)")
```

GRAFICO width (solo modelli cp)

```
ggplot(results_clean %>% filter(CP),
       aes(model, Width)) +
  geom_boxplot() +
  theme_minimal() +
  ggtitle("Ampiezza intervalli CP")
```

INTERVALLI DI CONFIDENZA

```
library(gt)

results_clean %>%
  group_by(model) %>%
  summarise(
    mean_MSE = mean(MSE),
    CI_low = t.test(MSE)$conf.int[1],
    CI_high = t.test(MSE)$conf.int[2],
    .groups = "drop"
  ) %>%
  gt() %>%
  fmt_number(
    columns = c(mean_MSE, CI_low, CI_high),
    decimals = 3
  ) %>%
```

```
  cols_label(
    model = "Modello",
    mean_MSE = "MSE medio",
    CI_low = "CI 95% (lower)",
    CI_high = "CI 95% (upper)"
  ) %>%
  tab_header(
    title = "Intervalli di confidenza del MSE medio",
    subtitle = "Confronto tra modelli con e senza Conformal Prediction"
  )
```

DIEBOLD–MARIANO (NO-CP) una serie alla volta (serie pilota)

```
# ARIMA vs ETS
dm_ar_ets = dm.test(
  e1 = test - arima_test_fc,
  e2 = test - ets_test_fc,
  h = 1
)
dm_ar_ets

# ARIMA vs RF
dm_ar_rf = dm.test(
  e1 = test - arima_test_fc,
  e2 = test - rf_test_pred,
  h = 1
)
dm_ar_rf

# ETS vs RF
dm_ets_rf = dm.test(
  e1 = test - ets_test_fc,
  e2 = test - rf_test_pred,
  h = 1
)
dm_ets_rf
```

Tabella di confronto

```
dm_results_df = tibble(
  Confronto = c(
    "ARIMA vs ETS",
    "ARIMA vs Random Forest",
    "ETS vs Random Forest"
  ),
  DM_statistic = c(
    dm_ar_ets$statistic,
    dm_ar_rf$statistic,
    dm_ets_rf$statistic
  ),
  p_value = c(
    dm_ar_ets$p.value,
    dm_ar_rf$p.value,
```

```
    dm_ets_rf$p.value
  )
)

library(gt)

dm_results_df %>%
  gt() %>%
  fmt_number(
    columns = c(DM_statistic, p_value),
    decimals = 4
  ) %>%
  cols_label(
    Confronto = "Confronto modelli",
    DM_statistic = "Statistica DM",
    p_value = "p-value"
  ) %>%
  tab_header(
    title = "Test di Diebold-Mariano - Serie pilota",
    subtitle = "Confronto dell'accuratezza puntuale tra modelli"
  )
```

NEMENYI (CP vs NO-CP)

```
results_nemenyi = results_clean %>%
  select(serie, model, MSE) %>%
  filter(is.finite(MSE)) %>%
  group_by(serie) %>%
  filter(n_distinct(model) == length(unique(results_clean$model))) %>%
  ungroup()


rank_mse_df = results_nemenyi %>%
  group_by(serie) %>%
  mutate(rank = rank(MSE)) %>%
  ungroup()

library(PMCMRplus)

rank_mse_df$model = factor(rank_mse_df$model)

rank_mse_df_clean = rank_mse_df %>%
  filter(is.finite(rank), !is.na(model))

nemenyi_all = kwAllPairsNemenyiTest(
  x = rank_mse_df$rank,
  g = rank_mse_df$model
)

#risultati in data frame
df_nemenyi_all = as.data.frame(nemenyi_all$p.value) %>%
  tibble::rownames_to_column(var = "Model1") %>%
  tidyr::pivot_longer(-Model1, names_to = "Model2", values_to = "p_value")
```

```
table_nemenyi_all = df_nemenyi_all %>%
  gt() %>%
  fmt_number(columns = "p_value", decimals = 4) %>%
  tab_header(
    title = "Test di Nemenyi - Tutti i modelli",
    subtitle = "p-value tra coppie di modelli"
  )

table_nemenyi_all
```

NEMENYI SOLO SU CP

```
results_cp = results_clean %>%
  filter(CP) %>%
  select(serie, model, MSE) %>%
  group_by(serie) %>%
  filter(n_distinct(model) == 3) %>%
  ungroup()

rank_cp = results_cp %>%
  group_by(serie) %>%
  mutate(rank = rank(MSE)) %>%
  ungroup()

rank_cp %>% summarize(
  any_na_rank = any(is.na(rank)),
  any_na_model = any(is.na(model)),
  any_inf_rank = any(!is.finite(rank))
)

rank_cp$model = droplevels(factor(rank_cp$model))

rank_cp = rank_cp %>%
  filter(!is.na(rank) & is.finite(rank))

library(PMCMRplus)

nemenyi_cp = kwAllPairsNemenyiTest(
  x = rank_cp$rank,
  g = rank_cp$model
)


df_nemenyi_cp = as.data.frame(nemenyi_cp$p.value) %>%
  tibble::rownames_to_column(var = "Model1") %>%
  tidyr::pivot_longer(-Model1, names_to = "Model2", values_to = "p_value") %>%
  filter(Model1 != Model2)

table_nemenyi_cp = df_nemenyi_cp %>%
  gt() %>%
  fmt_number(columns = "p_value", decimals = 4) %>%
  tab_header(
    title = "Test di Nemenyi - Solo CP",
```

```
    subtitle = "p-value tra coppie di modelli"
  )

table_nemenyi_cp
```

METRICHE AGGIUNTIVE

```
# Pinball Loss: valuta qualità previsioni quantiliche
pinball_loss = function(y, q_pred, alpha) {
  err = y - q_pred
  mean(ifelse(err > 0, alpha * err, (alpha - 1) * err))
}

# Interval Score: combina width e penalty per mancata copertura
interval_score = function(y, lower, upper, alpha = 0.1) {
  width = upper - lower
  penalty = (2/alpha) * pmax(0, lower - y) + (2/alpha) * pmax(0, y - upper)
  mean(width + penalty)
}
```

CONFRONTO CP vs PARAMETRICO (SERIE PILOTA)

ARIMA

```
# forecast ARIMA con intervalli parametrici al 90%
arima_test_fc_obj = forecast(arima_fit_full, h = length(test), level = 95)

# intervalli parametrici
param_low = arima_test_fc_obj$lower[, 1]
param_up = arima_test_fc_obj$upper[, 1]

coverage_param = mean(test >= param_low & test <= param_up)
width_param = mean(param_up - param_low)

coverage_cp_arima = mean(test >= arima_low & test <= arima_up)
width_cp_arima = mean(arima_up - arima_low)

# Tabella confronto
tibble(
  Metodo = c("Conformal Prediction", "Parametrico"),
  Coverage = c(coverage_cp_arima, coverage_param),
  Width = c(width_cp_arima, width_param)
) %>%
  gt() %>%
  fmt_number(columns = c(Coverage, Width), decimals = 3) %>%
  tab_header(
    title = "Confronto CP vs Parametrico - ARIMA",
    subtitle = "Serie pilota"
  ) %>%
  tab_style(
    style = cell_fill(color = "#F0F0F0"),
    locations = cells_body(rows = 1)
  )
```

```r
# Intervalli parametrici su test set (livello 95%)
arima_test_fc_obj = forecast(arima_fit_full, h = length(test), level = 95)

param_low = as.numeric(arima_test_fc_obj$lower[, 1])
param_up = as.numeric(arima_test_fc_obj$upper[, 1])

# Confronto visivo CP vs Parametrico
df_comparison = bind_rows(
  tibble(
    t = 1:length(test),
    test = test,
    pred = arima_test_fc,
    low = arima_low,
    up = arima_up,
    method = "Conformal"
  ),
  tibble(
    t = 1:length(test),
    test = test,
    pred = arima_test_fc,
    low = param_low,
    up = param_up,
    method = "Parametrico"
  )
)

# Grafico confronto
ggplot(df_comparison, aes(t, test)) +
  geom_line(linewidth = 0.8) +
  geom_line(aes(y = pred), color = "blue", linewidth = 0.8) +
  geom_ribbon(aes(ymin = low, ymax = up, fill = method), alpha = 0.3) +
  facet_wrap(~method, ncol = 1) +
  scale_fill_manual(values = c("Conformal" = color_cp,
                               "Parametrico" = color_param)) +
  theme_minimal(base_size = 12) +
  theme(legend.position = "none") +
  labs(
    title = "ARIMA: Conformal vs Intervalli Parametrici",
    subtitle = "Confronto visivo su serie pilota",
    x = "Tempo",
    y = "Valore (standardizzato)"
  )
```

```r
# Metriche comparative
comparison_metrics = tibble(
  Metodo = c("Conformal", "Parametrico"),
  Coverage = c(
    mean(test >= arima_low & test <= arima_up),
    mean(test >= param_low & test <= param_up)
  ),
  Width_media = c(
    mean(arima_up - arima_low),
    mean(param_up - param_low)
```

```
  ),
  Interval_Score = c(
    interval_score(test, arima_low, arima_up, alpha = 0.1),
    interval_score(test, param_low, param_up, alpha = 0.1)
  )
)

comparison_metrics %>%
  gt() %>%
  fmt_number(columns = -Metodo, decimals = 3) %>%
  tab_header(
    title = "Confronto CP vs Parametrico - ARIMA",
    subtitle = "Serie pilota: metriche comparative"
  ) %>%
  tab_style(
    style = cell_fill(color = "#E8F4F8"),
    locations = cells_column_labels()
  ) %>%
  tab_style(
    style = cell_fill(color = "#D5F4E6"),
    locations = cells_body(rows = Metodo == "Conformal")
  )
```

ETS

```
#forecast ETS con intervalli parametrici al 95%
ets_test_fc_obj = forecast(ets_fit_full, h = length(test), level = 95)

# intervalli parametrici
ets_param_low = ets_test_fc_obj$lower[, 1]
ets_param_up = ets_test_fc_obj$upper[, 1]

# Metriche parametrico
coverage_param_ets = mean(test >= ets_param_low & test <= ets_param_up)
width_param_ets = mean(ets_param_up - ets_param_low)

# Metriche CP
coverage_cp_ets = mean(test >= ets_low & test <= ets_up)
width_cp_ets = mean(ets_up - ets_low)

# Tabella confronto
tibble(
  Metodo = c("Conformal Prediction", "Parametrico"),
  Coverage = c(coverage_cp_ets, coverage_param_ets),
  Width = c(width_cp_ets, width_param_ets)
) %>%
  gt() %>%
  fmt_number(columns = c(Coverage, Width), decimals = 3) %>%
  tab_header(
    title = "Confronto CP vs Parametrico - ETS",
    subtitle = "Serie pilota"
  ) %>%
  tab_style(
```

```
    style = cell_fill(color = "#F0F0F0"),
    locations = cells_body(rows = 1)
  )
```

RANDOM FOREST

```
library(quantregForest)

# Train
qrf_train_data = make_lag_df(train)
qrf_fit = quantregForest(
  x = qrf_train_data %>% select(-y),
  y = qrf_train_data$y
)

# Refit su train + cal
qrf_fit_full = quantregForest(
  x = make_lag_df(c(train, cal)) %>% select(-y),
  y = make_lag_df(c(train, cal))$y
)

# Predict con quantili al 2.5% e 97.5% (95% CI)
rf_param_low = numeric(length(test))
rf_param_up = numeric(length(test))
history_qrf = c(train, cal)

for(i in seq_along(test)){
  x = tail(history_qrf, 8)
  newdata = as.data.frame(t(rev(x)))
  colnames(newdata) = paste0("lag", 1:8)

  quantiles = predict(qrf_fit_full, newdata, what = c(0.025, 0.975))
  rf_param_low[i] = quantiles[1]
  rf_param_up[i] = quantiles[2]

  history_qrf = c(history_qrf, rf_test_pred[i])
}

# Metriche parametrico (quantile RF)
coverage_param_rf = mean(test >= rf_param_low & test <= rf_param_up)
width_param_rf = mean(rf_param_up - rf_param_low)

# Metriche CP
coverage_cp_rf = mean(test >= rf_low & test <= rf_up)
width_cp_rf = mean(rf_up - rf_low)

# Tabella confronto
tibble(
  Metodo = c("Conformal Prediction", "Quantile RF (parametrico)"),
  Coverage = c(coverage_cp_rf, coverage_param_rf),
  Width = c(width_cp_rf, width_param_rf)
) %>%
  gt() %>%
```

```
  fmt_number(columns = c(Coverage, Width), decimals = 3) %>%
  tab_header(
    title = "Confronto CP vs Quantile RF - Random Forest",
    subtitle = "Serie pilota"
  ) %>%
  tab_style(
    style = cell_fill(color = "#F0F0F0"),
    locations = cells_body(rows = 1)
  )
```

GRAFICO COMPARATIVO: TUTTI I METODI

```
df_comparison = tibble(
  Modello = rep(c("ARIMA", "ETS", "Random Forest"), each = 2),
  Metodo = rep(c("CP", "Parametrico"), 3),
  Coverage = c(
    coverage_cp_arima, coverage_param,
    coverage_cp_ets, coverage_param_ets,
    coverage_cp_rf, coverage_param_rf
  ),
  Width = c(
    width_cp_arima, width_param,
    width_cp_ets, width_param_ets,
    width_cp_rf, width_param_rf
  )
)

# Grafico coverage
ggplot(df_comparison, aes(x = Modello, y = Coverage, fill = Metodo)) +
  geom_col(position = position_dodge(width = 0.7), width = 0.6) +
  geom_hline(yintercept = 0.95, linetype = "dashed",
             color = color_nominal, linewidth = 1) +
  scale_fill_manual(values = c("CP" = color_cp, "Parametrico" = color_param)) +
  theme_minimal(base_size = 12) +
  theme(legend.position = "top") +
  labs(
    title = "Coverage: CP vs Parametrico",
    subtitle = "Linea tratteggiata = livello nominale 95%",
    x = "Modello",
    y = "Coverage empirica",
    fill = "Metodo"
  ) +
  ylim(0, 1)
```

```
# Grafico width
ggplot(df_comparison, aes(x = Modello, y = Width, fill = Metodo)) +
  geom_col(position = position_dodge(width = 0.7), width = 0.6) +
  scale_fill_manual(values = c("CP" = color_cp, "Parametrico" = color_param)) +
  theme_minimal(base_size = 12) +
  theme(legend.position = "top") +
  labs(
    title = "Ampiezza Intervalli: CP vs Parametrico",
    subtitle = "Confronto tra metodi",
```

```
    x = "Modello",
    y = "Ampiezza media",
    fill = "Metodo"
  )
```

```
# Tabella riepilogativa completa
df_comparison %>%
  gt() %>%
  fmt_number(columns = c(Coverage, Width), decimals = 3) %>%
  tab_header(
    title = "Confronto Completo: CP vs Parametrico",
    subtitle = "Serie pilota - Tutti i modelli"
  ) %>%
  tab_style(
    style = cell_fill(color = "#E8F4F8"),
    locations = cells_body(
      columns = everything(),
      rows = Metodo == "CP"
    )
  )
```

```
### CALCOLO INTERVALLI PARAMETRICI PER TUTTE LE SERIE

cat("Calcolo intervalli parametrici per tutte le serie...\n")

results_param = list()

for(s in unique(df_weekly$serie)){

  y = df_weekly %>% filter(serie == s) %>% pull(value)
  if(length(y) < 120) next

  y = (y - mean(y)) / sd(y)
  n = length(y)

  train = y[1:floor(0.6*n)]
  cal   = y[(floor(0.6*n)+1):floor(0.8*n)]
  test  = y[(floor(0.8*n)+1):n]

  # ARIMA Parametrico
  ar_fit_full = auto.arima(c(train, cal))
  ar_fc_obj = forecast(ar_fit_full, h = length(test), level = 90)
  ar_param_low = as.numeric(ar_fc_obj$lower[, 1])
  ar_param_up = as.numeric(ar_fc_obj$upper[, 1])

  # ETS Parametrico
  ets_fit_full = ets(c(train, cal))
  ets_fc_obj = forecast(ets_fit_full, h = length(test), level = 90)
  ets_param_low = as.numeric(ets_fc_obj$lower[, 1])
  ets_param_up = as.numeric(ets_fc_obj$upper[, 1])

  results_param[[s]] = tibble(
    serie = s,
```

```
    model = c("ARIMA_Param", "ETS_Param"),
    MSE = c(
      mean((test - ar_fc_obj$mean)^2),
      mean((test - ets_fc_obj$mean)^2)
    ),
    Coverage = c(
      mean(test >= ar_param_low & test <= ar_param_up),
      mean(test >= ets_param_low & test <= ets_param_up)
    ),
    Width = c(
      mean(ar_param_up - ar_param_low),
      mean(ets_param_up - ets_param_low)
    ),
    CP = c(FALSE, FALSE)
  )
}

#risultati parametrici combinati con quelli CP esistenti
results_param_df = bind_rows(results_param)
results_complete = bind_rows(results_clean, results_param_df)

cat("Risultati parametrici calcolati per",
    length(unique(results_param_df$serie)), "serie\n")
```

Violin Plot, Distribuzione MSE

```
ggplot(results_clean, aes(model, MSE, fill = CP)) +
  geom_violin(trim = FALSE, alpha = 0.7) +
  geom_boxplot(width = 0.15, outlier.shape = NA, alpha = 0.8,
               fill = "white") +
  scale_fill_manual(values = c("TRUE" = color_cp, "FALSE" = "gray60"),
                    labels = c("TRUE" = "Con CP", "FALSE" = "Senza CP")) +
  theme_minimal(base_size = 12) +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1),
    legend.position = "top"
  ) +
  labs(
    title = "Distribuzione MSE per modello",
    subtitle = "Violin plot con boxplot interno",
    x = "Modello",
    y = "MSE",
    fill = ""
  )
```

COVERAGE CONDIZIONALE PER ORIZZONTE

```
#orizzonte test diviso in bins e calcolo coverage
n_bins = 5
df_horizon = tibble(
  horizon = 1:length(test),
  covered_arima = (test >= arima_low) & (test <= arima_up),
  covered_ets = (test >= ets_low) & (test <= ets_up),
```

```
  covered_rf = (test >= rf_low) & (test <= rf_up)
) %>%
  mutate(bin = cut(horizon, breaks = n_bins, labels = paste0("T", 1:n_bins)))

coverage_by_bin = df_horizon %>%
  group_by(bin) %>%
  summarise(
    ARIMA_CP = mean(covered_arima),
    ETS_CP = mean(covered_ets),
    RF_CP = mean(covered_rf),
    .groups = "drop"
  ) %>%
  pivot_longer(cols = -bin, names_to = "Modello", values_to = "Coverage")

# Grafico coverage condizionale
ggplot(coverage_by_bin, aes(bin, Coverage, fill = Modello)) +
  geom_col(position = "dodge", width = 0.7) +
  geom_hline(yintercept = 0.9, linetype = "dashed",
             color = color_nominal, linewidth = 0.8) +
  scale_fill_manual(values = c("ARIMA_CP" = color_arima,
                               "ETS_CP" = color_ets,
                               "RF_CP" = color_rf)) +
  theme_minimal(base_size = 12) +
  theme(
    panel.grid.major.x = element_blank(),
    legend.position = "top"
  ) +
  labs(
    title = "Coverage condizionale per periodo temporale",
    subtitle = "Linea tratteggiata: livello nominale 90%",
    x = "Periodo test",
    y = "Coverage empirica"
  )
```

Interval Score (Serie Pilota)

```
# interval score per tutti i modelli
int_score_arima = interval_score(test, arima_low, arima_up, alpha = 0.1)
int_score_ets = interval_score(test, ets_low, ets_up, alpha = 0.1)
int_score_rf = interval_score(test, rf_low, rf_up, alpha = 0.1)

tibble(
  Modello = c("ARIMA_CP", "ETS_CP", "RF_CP"),
  Interval_Score = c(int_score_arima, int_score_ets, int_score_rf),
  Coverage = c(
    mean(test >= arima_low & test <= arima_up),
    mean(test >= ets_low & test <= ets_up),
    mean(test >= rf_low & test <= rf_up)
  ),
  Width = c(
    mean(arima_up - arima_low),
    mean(ets_up - ets_low),
    mean(rf_up - rf_low)
```

```
  )
) %>%
  arrange(Interval_Score) %>%
  gt() %>%
  fmt_number(columns = c(Interval_Score, Coverage, Width), decimals = 3) %>%
  tab_header(
    title = "Interval Score per modello",
    subtitle = "Metrica che bilancia width e coverage"
  ) %>%
  tab_style(
    style = cell_fill(color = "#E8F4F8"),
    locations = cells_body(rows = 1)
  )
```

Rolling Coverage

```
# Coverage con finestra mobile di 10 step
window = 10
n = length(test)

rolling_data = tibble(
  t = window:n,
  ARIMA_CP = sapply(window:n, function(i) {
    start = i - window + 1
    mean(test[start:i] >= arima_low[start:i] & test[start:i] <= arima_up[start:i])
  }),
  ETS_CP = sapply(window:n, function(i) {
    start = i - window + 1
    mean(test[start:i] >= ets_low[start:i] & test[start:i] <= ets_up[start:i])
  }),
  RF_CP = sapply(window:n, function(i) {
    start = i - window + 1
    mean(test[start:i] >= rf_low[start:i] & test[start:i] <= rf_up[start:i])
  })
) %>%
  pivot_longer(cols = -t, names_to = "Modello", values_to = "Coverage")

# Grafico rolling coverage
ggplot(rolling_data, aes(t, Coverage, color = Modello)) +
  geom_line(linewidth = 1) +
  geom_hline(yintercept = 0.9, linetype = "dashed",
             color = color_nominal, linewidth = 0.8) +
  geom_ribbon(aes(ymin = 0.85, ymax = 0.95),
              fill = "gray80", alpha = 0.3, color = NA) +
  scale_color_manual(values = c("ARIMA_CP" = color_arima,
                                "ETS_CP" = color_ets,
                                "RF_CP" = color_rf)) +
  theme_minimal(base_size = 12) +
  theme(legend.position = "top") +
  labs(
    title = "Coverage rolling con finestra mobile",
    subtitle = paste0("Finestra = ", window, " step | Banda grigia: ±5% dal nominale"),
    x = "Step temporale",
```

```
    y = "Coverage empirica"
  )
```

Sharpness Analysis (MULTI-SERIE)

```
# Efficienza intervalli: width condizionata a coverage >= 95%
sharpness_results = results_clean %>%
  filter(CP) %>%
  group_by(model) %>%
  summarise(
    Width_media = mean(Width, na.rm = TRUE),
    Width_condizionale = mean(Width[Coverage >= 0.95], na.rm = TRUE),
    n_buona_coverage = sum(Coverage >= 0.95, na.rm = TRUE),
    .groups = "drop"
  )

sharpness_results %>%
  gt() %>%
  fmt_number(columns = c(Width_media, Width_condizionale), decimals = 3) %>%
  tab_header(
    title = "Analisi Sharpness degli intervalli",
    subtitle = "Width condizionata a coverage  95%"
  ) %>%
  tab_style(
    style = cell_fill(color = "#F0F0F0"),
    locations = cells_column_labels()
  )
```

Reliability diagram

```
# Coverage empirica vs nominale
reliability_data = results_clean %>%
  filter(CP) %>%
  group_by(model) %>%
  summarise(
    nominale = 0.90,
    empirica = mean(Coverage, na.rm = TRUE),
    .groups = "drop"
  )

ggplot(reliability_data, aes(nominale, empirica, color = model)) +
  geom_point(size = 5, alpha = 0.8) +
  geom_abline(slope = 1, intercept = 0, linetype = "dashed",
              color = color_nominal, linewidth = 0.8) +
  scale_color_manual(values = c("ARIMA_CP" = color_arima,
                                "ETS_CP" = color_ets,
                                "RF_CP" = color_rf)) +
  coord_fixed(xlim = c(0.80, 0.98), ylim = c(0.80, 0.98)) +
  theme_minimal(base_size = 12) +
  theme(legend.position = "top") +
  labs(
    title = "Reliability Diagram",
    subtitle = "Diagonale = calibrazione perfetta",
```

```
    x = "Coverage nominale",
    y = "Coverage empirica",
    color = "Modello"
  )
```

Coverage e width per categoria prodotto

```
# Performance CP per categoria
perf_categoria = results_clean %>%
  filter(CP) %>%
  left_join(df_weekly %>% distinct(serie, cat_id), by = "serie") %>%
  group_by(cat_id, model) %>%
  summarise(
    Coverage_media = mean(Coverage, na.rm = TRUE),
    Width_media = mean(Width, na.rm = TRUE),
    n_serie = n_distinct(serie),
    .groups = "drop"
  )

# Grafico coverage per categoria
ggplot(perf_categoria, aes(cat_id, Coverage_media, fill = model)) +
  geom_col(position = "dodge", width = 0.7) +
  geom_hline(yintercept = 0.9, linetype = "dashed",
             color = color_nominal, linewidth = 0.8) +
  scale_fill_manual(values = c("ARIMA_CP" = color_arima,
                               "ETS_CP" = color_ets,
                               "RF_CP" = color_rf)) +
  theme_minimal(base_size = 12) +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1),
    legend.position = "top"
  ) +
  labs(
    title = "Coverage per categoria prodotto",
    subtitle = "Analisi eterogeneità performance",
    x = "Categoria",
    y = "Coverage media",
    fill = "Modello"
  )
```

ANALISI ROBUSTEZZA: CP vs PARAMETRICO SU SERIE NON-GAUSSIANE

```
#serie con residui non-normali (test Shapiro-Wilk)
non_normal_series = list()

for(s in unique(df_weekly$serie)[1:min(100, length(unique(df_weekly$serie)))]){

  y = df_weekly %>% filter(serie == s) %>% pull(value)
  if(length(y) < 120) next

  y = (y - mean(y)) / sd(y)
  n = length(y)
```

```
  train = y[1:floor(0.6*n)]
  cal   = y[(floor(0.6*n)+1):floor(0.8*n)]

  # ARIMA
  ar_fit = tryCatch({
    auto.arima(train)
  }, error = function(e) NULL)

  if(is.null(ar_fit)) next

  ar_cal = forecast(ar_fit, h = length(cal))$mean
  ar_resid = cal - ar_cal

  # Test normalità
  if(length(ar_resid) >= 3) {
    shapiro_result = tryCatch({
      shapiro.test(ar_resid)
    }, error = function(e) list(p.value = NA))

    if(!is.na(shapiro_result$p.value)) {
      non_normal_series[[s]] = tibble(
        serie = s,
        shapiro_p = shapiro_result$p.value,
        non_normal = shapiro_result$p.value < 0.05,
        skewness = moments::skewness(ar_resid),
        kurtosis = moments::kurtosis(ar_resid) - 3  # excess kurtosis
      )
    }
  }
}

non_normal_df = bind_rows(non_normal_series)

cat("Serie con residui non-normali:", sum(non_normal_df$non_normal, na.rm = TRUE),
    "su", nrow(non_normal_df), "\n")
```

ARIMA

```
# coverage_comparison_arima per avere visibilità globale
coverage_comparison_arima_global = results_complete %>%
  filter(model %in% c("ARIMA_CP", "ARIMA_Param")) %>%
  select(serie, model, Coverage) %>%
  pivot_wider(names_from = model, values_from = Coverage) %>%
  drop_na()

# Confronto CP vs Parametrico SOLO su serie non-normali
comparison_non_normal = results_complete %>%
  filter(model %in% c("ARIMA_CP", "ARIMA_Param")) %>%
  inner_join(non_normal_df, by = "serie") %>%
  filter(non_normal) %>%
  select(serie, model, Coverage, Width) %>%
  pivot_wider(names_from = model, values_from = c(Coverage, Width),
              names_sep = "_") %>%
```

```
  drop_na()

# Test Wilcoxon su serie non-normali (con gestione ties/zeri)
wilcox_non_normal = wilcox.test(
  comparison_non_normal$Coverage_ARIMA_CP,
  comparison_non_normal$Coverage_ARIMA_Param,
  paired = TRUE,
  alternative = "greater",
  exact = FALSE  #approssimazione normale
)

# Wilcoxon su tutte le serie
wilcox_arima_global = wilcox.test(
  coverage_comparison_arima_global$ARIMA_CP,
  coverage_comparison_arima_global$ARIMA_Param,
  paired = TRUE,
  alternative = "greater",
  exact = FALSE
)

# Effect size
effect_size_arima_global = mean(coverage_comparison_arima_global$ARIMA_CP -
                                  coverage_comparison_arima_global$ARIMA_Param)
effect_size_non_normal = mean(comparison_non_normal$Coverage_ARIMA_CP -
                                comparison_non_normal$Coverage_ARIMA_Param)

# Tabella risultati
tibble(
  Subset = c("Tutte le serie", "Solo serie non-normali"),
  N_serie = c(
    nrow(coverage_comparison_arima_global),
    nrow(comparison_non_normal)
  ),
  Diff_Coverage = c(
    effect_size_arima_global,
    effect_size_non_normal
  ),
  P_value = c(
    wilcox_arima_global$p.value,
    wilcox_non_normal$p.value
  ),
  Significativo = ifelse(c(wilcox_arima_global$p.value,
                           wilcox_non_normal$p.value) < 0.05, " ", "")
) %>%
  gt() %>%
  fmt_number(columns = c(Diff_Coverage, P_value), decimals = 4) %>%
  tab_header(
    title = "Robustezza CP vs Parametrico ARIMA",
    subtitle = "Vantaggio CP su serie con residui non-gaussiani"
  ) %>%
  tab_style(
    style = cell_fill(color = "#FFF3CD"),
    locations = cells_body(rows = 2)
```

```
  ) %>%
  tab_style(
    style = cell_text(weight = "bold"),
    locations = cells_body(
      columns = Significativo,
      rows = Significativo == " "
    )
  )
```

Grafico confronto paired per serie non-normali

```
comparison_non_normal %>%
  pivot_longer(cols = starts_with("Coverage"),
               names_to = "method",
               values_to = "coverage",
               names_prefix = "Coverage_ARIMA_") %>%
  ggplot(aes(method, coverage, fill = method)) +
  geom_boxplot(alpha = 0.7, width = 0.6) +
  geom_line(aes(group = serie), alpha = 0.3, color = "gray50") +
  geom_hline(yintercept = 0.9, linetype = "dashed", color = "black") +
  scale_fill_manual(values = c("CP" = color_cp, "Param" = color_param)) +
  theme_minimal(base_size = 12) +
  theme(legend.position = "top") +
  labs(
    title = "CP vs Parametrico: Serie con Residui Non-Normali",
    subtitle = paste0("N=", nrow(comparison_non_normal),
                      " serie dove Shapiro p-value < 0.05"),
    x = "Metodo",
    y = "Coverage",
    fill = ""
  )
```

ETS

```
#serie con residui ETS non-normali
non_normal_series_ets = list()

for(s in unique(df_weekly$serie)[1:min(100, length(unique(df_weekly$serie)))]){

  y = df_weekly %>% filter(serie == s) %>% pull(value)
  if(length(y) < 120) next

  y = (y - mean(y)) / sd(y)
  n = length(y)

  train = y[1:floor(0.6*n)]
  cal   = y[(floor(0.6*n)+1):floor(0.8*n)]

  # ETS
  ets_fit = tryCatch({
    ets(train)
  }, error = function(e) NULL)
```

```
  if(is.null(ets_fit)) next

  ets_cal = forecast(ets_fit, h = length(cal))$mean
  ets_resid = cal - ets_cal

  # Test normalità
  if(length(ets_resid) >= 3) {
    shapiro_result = tryCatch({
      shapiro.test(ets_resid)
    }, error = function(e) list(p.value = NA))

    if(!is.na(shapiro_result$p.value)) {
      non_normal_series_ets[[s]] = tibble(
        serie = s,
        shapiro_p = shapiro_result$p.value,
        non_normal = shapiro_result$p.value < 0.05,
        skewness = moments::skewness(ets_resid),
        kurtosis = moments::kurtosis(ets_resid) - 3
      )
    }
  }
}

non_normal_df_ets = bind_rows(non_normal_series_ets)

cat("Serie ETS con residui non-normali:", sum(non_normal_df_ets$non_normal, na.rm = TRUE),
    "su", nrow(non_normal_df_ets), "\n")

# Coverage comparison globale ETS
coverage_comparison_ets_global = results_complete %>%
  filter(model %in% c("ETS_CP", "ETS_Param")) %>%
  select(serie, model, Coverage) %>%
  pivot_wider(names_from = model, values_from = Coverage) %>%
  drop_na()

# Confronto CP vs Parametrico SOLO su serie non-normali ETS
comparison_non_normal_ets = results_complete %>%
  filter(model %in% c("ETS_CP", "ETS_Param")) %>%
  inner_join(non_normal_df_ets, by = "serie") %>%
  filter(non_normal) %>%
  select(serie, model, Coverage, Width) %>%
  pivot_wider(names_from = model, values_from = c(Coverage, Width),
              names_sep = "_") %>%
  drop_na()

# Test Wilcoxon su serie non-normali ETS
wilcox_non_normal_ets = wilcox.test(
  comparison_non_normal_ets$Coverage_ETS_CP,
  comparison_non_normal_ets$Coverage_ETS_Param,
  paired = TRUE,
  alternative = "greater",
  exact = FALSE
)
```

```r
# Wilcoxon su tutte le serie ETS
wilcox_ets_global = wilcox.test(
  coverage_comparison_ets_global$ETS_CP,
  coverage_comparison_ets_global$ETS_Param,
  paired = TRUE,
  alternative = "greater",
  exact = FALSE
)

# Effect size ETS
effect_size_ets_global = mean(coverage_comparison_ets_global$ETS_CP -
                                coverage_comparison_ets_global$ETS_Param)
effect_size_non_normal_ets = mean(comparison_non_normal_ets$Coverage_ETS_CP -
                                    comparison_non_normal_ets$Coverage_ETS_Param)

# Tabella risultati ETS
tibble(
  Subset = c("Tutte le serie", "Solo serie non-normali"),
  N_serie = c(
    nrow(coverage_comparison_ets_global),
    nrow(comparison_non_normal_ets)
  ),
  Diff_Coverage = c(
    effect_size_ets_global,
    effect_size_non_normal_ets
  ),
  P_value = c(
    wilcox_ets_global$p.value,
    wilcox_non_normal_ets$p.value
  ),
  Significativo = ifelse(c(wilcox_ets_global$p.value,
                           wilcox_non_normal_ets$p.value) < 0.05, " ", "")
) %>%
  gt() %>%
  fmt_number(columns = c(Diff_Coverage, P_value), decimals = 4) %>%
  tab_header(
    title = "Robustezza CP vs Parametrico ETS",
    subtitle = "Vantaggio CP su serie con residui non-gaussiani"
  ) %>%
  tab_style(
    style = cell_fill(color = "#FFF3CD"),
    locations = cells_body(rows = 2)
  ) %>%
  tab_style(
    style = cell_text(weight = "bold"),
    locations = cells_body(
      columns = Significativo,
      rows = Significativo == " "
    )
  )
```

```r
# Grafico confronto paired per serie non-normali ETS
comparison_non_normal_ets %>%
```

```
  pivot_longer(cols = starts_with("Coverage"),
               names_to = "method",
               values_to = "coverage",
               names_prefix = "Coverage_ETS_") %>%
  ggplot(aes(method, coverage, fill = method)) +
  geom_boxplot(alpha = 0.7, width = 0.6) +
  geom_line(aes(group = serie), alpha = 0.3, color = "gray50") +
  geom_hline(yintercept = 0.9, linetype = "dashed", color = "black") +
  scale_fill_manual(values = c("CP" = color_cp, "Param" = color_param)) +
  theme_minimal(base_size = 12) +
  theme(legend.position = "top") +
  labs(
    title = "ETS: CP vs Parametrico su Serie con Residui Non-Normali",
    subtitle = paste0("N=", nrow(comparison_non_normal_ets),
                      " serie dove Shapiro p-value < 0.05"),
    x = "Metodo",
    y = "Coverage",
    fill = ""
  )
```

Random Forest

```
#serie con residui RF non-normali
non_normal_series_rf = list()

for(s in unique(df_weekly$serie)[1:min(100, length(unique(df_weekly$serie)))]){

  y = df_weekly %>% filter(serie == s) %>% pull(value)
  if(length(y) < 120) next

  y = (y - mean(y)) / sd(y)
  n = length(y)

  train = y[1:floor(0.6*n)]
  cal   = y[(floor(0.6*n)+1):floor(0.8*n)]

  # RF
  rf_fit = tryCatch({
    randomForest(y ~ ., data = make_lag_df(train))
  }, error = function(e) NULL)

  if(is.null(rf_fit)) next

  rf_cal = numeric(length(cal))
  hist = train

  for(i in seq_along(cal)){
    x = tail(hist, 8)
    newdata = as.data.frame(t(rev(x)))
    colnames(newdata) = paste0("lag", 1:8)
    rf_cal[i] = predict(rf_fit, newdata)
    hist = c(hist, rf_cal[i])
  }
```

```
  rf_resid = cal - rf_cal

  # Test normalità
  if(length(rf_resid) >= 3) {
    shapiro_result = tryCatch({
      shapiro.test(rf_resid)
    }, error = function(e) list(p.value = NA))

    if(!is.na(shapiro_result$p.value)) {
      non_normal_series_rf[[s]] = tibble(
        serie = s,
        shapiro_p = shapiro_result$p.value,
        non_normal = shapiro_result$p.value < 0.05,
        skewness = moments::skewness(rf_resid),
        kurtosis = moments::kurtosis(rf_resid) - 3
      )
    }
  }
}

non_normal_df_rf = bind_rows(non_normal_series_rf)

cat("Serie RF con residui non-normali:", sum(non_normal_df_rf$non_normal, na.rm = TRUE),
    "su", nrow(non_normal_df_rf), "\n")



if(!"RF_Param" %in% unique(results_complete$model)) {

  cat("Calcolo intervalli Quantile RF per confronto...\n")

  results_rf_param = list()

  for(s in unique(df_weekly$serie)){

    y = df_weekly %>% filter(serie == s) %>% pull(value)
    if(length(y) < 120) next

    y = (y - mean(y)) / sd(y)
    n = length(y)

    train = y[1:floor(0.6*n)]
    cal   = y[(floor(0.6*n)+1):floor(0.8*n)]
    test  = y[(floor(0.8*n)+1):n]

    # Quantile RF Parametrico
    qrf_fit = tryCatch({
      quantregForest(
        x = make_lag_df(c(train, cal)) %>% select(-y),
        y = make_lag_df(c(train, cal))$y
      )
    }, error = function(e) NULL)
```

```
    if(is.null(qrf_fit)) next

    # Test predictions
    rf_test_point = numeric(length(test))
    rf_param_low_test = numeric(length(test))
    rf_param_up_test = numeric(length(test))
    history_qrf = c(train, cal)

    for(i in seq_along(test)){
      x = tail(history_qrf, 8)
      newdata = as.data.frame(t(rev(x)))
      colnames(newdata) = paste0("lag", 1:8)

      quantiles = tryCatch({
        predict(qrf_fit, newdata, what = c(0.025, 0.5, 0.975))
      }, error = function(e) c(NA, NA, NA))

      rf_param_low_test[i] = quantiles[1]
      rf_test_point[i] = quantiles[2]
      rf_param_up_test[i] = quantiles[3]

      history_qrf = c(history_qrf, rf_test_point[i])
    }

    if(all(is.finite(c(rf_param_low_test, rf_param_up_test)))) {
      results_rf_param[[s]] = tibble(
        serie = s,
        model = "RF_Param",
        MSE = mean((test - rf_test_point)^2),
        Coverage = mean(test >= rf_param_low_test & test <= rf_param_up_test),
        Width = mean(rf_param_up_test - rf_param_low_test),
        CP = FALSE
      )
    }
  }

  results_rf_param_df = bind_rows(results_rf_param)
  results_complete = bind_rows(results_complete, results_rf_param_df)

  cat("Intervalli Quantile RF calcolati per",
      length(unique(results_rf_param_df$serie)), "serie\n")
}
```

```
# Coverage comparison globale RF
coverage_comparison_rf_global = results_complete %>%
  filter(model %in% c("RF_CP", "RF_Param")) %>%
  select(serie, model, Coverage) %>%
  pivot_wider(names_from = model, values_from = Coverage) %>%
  drop_na()

# Confronto CP vs Quantile RF SOLO su serie non-normali
comparison_non_normal_rf = results_complete %>%
  filter(model %in% c("RF_CP", "RF_Param")) %>%
```

```
  inner_join(non_normal_df_rf, by = "serie") %>%
  filter(non_normal) %>%
  select(serie, model, Coverage, Width) %>%
  pivot_wider(names_from = model, values_from = c(Coverage, Width),
              names_sep = "_") %>%
  drop_na()

# Test Wilcoxon su serie non-normali RF
wilcox_non_normal_rf = wilcox.test(
  comparison_non_normal_rf$Coverage_RF_CP,
  comparison_non_normal_rf$Coverage_RF_Param,
  paired = TRUE,
  alternative = "greater",
  exact = FALSE
)

# Wilcoxon su tutte le serie RF
wilcox_rf_global = wilcox.test(
  coverage_comparison_rf_global$RF_CP,
  coverage_comparison_rf_global$RF_Param,
  paired = TRUE,
  alternative = "greater",
  exact = FALSE
)

# Effect size RF
effect_size_rf_global = mean(coverage_comparison_rf_global$RF_CP -
                               coverage_comparison_rf_global$RF_Param)
effect_size_non_normal_rf = mean(comparison_non_normal_rf$Coverage_RF_CP -
                                   comparison_non_normal_rf$Coverage_RF_Param)

# Tabella risultati RF
tibble(
  Subset = c("Tutte le serie", "Solo serie non-normali"),
  N_serie = c(
    nrow(coverage_comparison_rf_global),
    nrow(comparison_non_normal_rf)
  ),
  Diff_Coverage = c(
    effect_size_rf_global,
    effect_size_non_normal_rf
  ),
  P_value = c(
    wilcox_rf_global$p.value,
    wilcox_non_normal_rf$p.value
  ),
  Significativo = ifelse(c(wilcox_rf_global$p.value,
                           wilcox_non_normal_rf$p.value) < 0.05, " ", "")
) %>%
  gt() %>%
  fmt_number(columns = c(Diff_Coverage, P_value), decimals = 4) %>%
  tab_header(
    title = "Robustezza CP vs Quantile RF",
```

```
    subtitle = "Vantaggio CP su serie con residui non-gaussiani"
  ) %>%
  tab_style(
    style = cell_fill(color = "#FFF3CD"),
    locations = cells_body(rows = 2)
  ) %>%
  tab_style(
    style = cell_text(weight = "bold"),
    locations = cells_body(
      columns = Significativo,
      rows = Significativo == " "
    )
  )
```

```
# Grafico confronto paired per serie non-normali RF
comparison_non_normal_rf %>%
  pivot_longer(cols = starts_with("Coverage"),
               names_to = "method",
               values_to = "coverage",
               names_prefix = "Coverage_RF_") %>%
  ggplot(aes(method, coverage, fill = method)) +
  geom_boxplot(alpha = 0.7, width = 0.6) +
  geom_line(aes(group = serie), alpha = 0.3, color = "gray50") +
  geom_hline(yintercept = 0.9, linetype = "dashed", color = "black") +
  scale_fill_manual(values = c("CP" = color_cp, "Param" = color_param)) +
  theme_minimal(base_size = 12) +
  theme(legend.position = "top") +
  labs(
    title = "RF: CP vs Quantile RF su Serie con Residui Non-Normali",
    subtitle = paste0("N=", nrow(comparison_non_normal_rf),
                      " serie dove Shapiro p-value < 0.05"),
    x = "Metodo",
    y = "Coverage",
    fill = ""
  )
```

Tabella comparativa completa

```
tibble(
  Modello = rep(c("ARIMA", "ETS", "RF"), each = 2),
  Subset = rep(c("Tutte", "Non-normali"), 3),
  N_serie = c(
    nrow(coverage_comparison_arima_global),
    nrow(comparison_non_normal),
    nrow(coverage_comparison_ets_global),
    nrow(comparison_non_normal_ets),
    nrow(coverage_comparison_rf_global),
    nrow(comparison_non_normal_rf)
  ),
  Diff_Coverage = c(
    effect_size_arima_global,
    effect_size_non_normal,
    effect_size_ets_global,
```

```
    effect_size_non_normal_ets,
    effect_size_rf_global,
    effect_size_non_normal_rf
  ),
  P_value = c(
    wilcox_arima_global$p.value,
    wilcox_non_normal$p.value,
    wilcox_ets_global$p.value,
    wilcox_non_normal_ets$p.value,
    wilcox_rf_global$p.value,
    wilcox_non_normal_rf$p.value
  ),
  Significativo = ifelse(
    c(wilcox_arima_global$p.value, wilcox_non_normal$p.value,
      wilcox_ets_global$p.value, wilcox_non_normal_ets$p.value,
      wilcox_rf_global$p.value, wilcox_non_normal_rf$p.value) < 0.05,
    "*", ""
  )
) %>%
  gt() %>%
  fmt_number(columns = c(Diff_Coverage, P_value), decimals = 4) %>%
  tab_header(
    title = "Robustezza CP: Confronto Completo",
    subtitle = "CP vs Parametrico per tutti i modelli"
  ) %>%
  tab_style(
    style = cell_fill(color = "#E8F4F8"),
    locations = cells_body(rows = Subset == "Non-normali")
  ) %>%
  tab_style(
    style = cell_text(weight = "bold"),
    locations = cells_body(
      columns = Significativo,
      rows = Significativo == "*"
    )
  )
```

TEST STATISTICI FORMALI: WILCOXON PAIRED TEST

```
# Tabella risultati test
wilcox_results = tibble(
  Confronto = c(
    "ARIMA: CP vs Parametrico",
    "ETS: CP vs Parametrico",
    "RF: CP vs Quantile RF"
  ),
  N_serie = c(
    nrow(coverage_comparison_arima_global),
    nrow(coverage_comparison_ets_global),
    nrow(coverage_comparison_rf_global)
  ),
  Diff_media = c(
    effect_size_arima_global,
```

```
    effect_size_ets_global,
    effect_size_rf_global
  ),
  Statistica_W = c(
    wilcox_arima_global$statistic,
    wilcox_ets_global$statistic,
    wilcox_rf_global$statistic
  ),
  P_value = c(
    wilcox_arima_global$p.value,
    wilcox_ets_global$p.value,
    wilcox_rf_global$p.value
  ),
  Significativo = ifelse(
    c(wilcox_arima_global$p.value,
      wilcox_ets_global$p.value,
      wilcox_rf_global$p.value) < 0.05,
    "Sì  ", "No"
  )
)

wilcox_results %>%
  gt() %>%
  fmt_number(columns = c(Diff_media, Statistica_W, P_value), decimals = 4) %>%
  tab_header(
    title = "Test di Wilcoxon: CP vs Parametrico",
    subtitle = "Test paired sulla coverage (H1: CP > Param)"
  ) %>%
  tab_style(
    style = cell_fill(color = "#D5F4E6"),
    locations = cells_body(
      columns = Significativo,
      rows = Significativo == "Sì  "
    )
  ) %>%
  cols_label(
    N_serie = "N. Serie",
    Diff_media = "Δ Coverage",
    Statistica_W = "W",
    P_value = "p-value"
  )
```

BOXPLOT COMPARATIVO PAIRED: TUTTI I MODELLI

```
coverage_comparison_long = bind_rows(
  coverage_comparison_arima_global %>%
    select(serie, CP = ARIMA_CP, Param = ARIMA_Param) %>%
    mutate(model = "ARIMA"),
  coverage_comparison_ets_global %>%
    select(serie, CP = ETS_CP, Param = ETS_Param) %>%
    mutate(model = "ETS"),
  coverage_comparison_rf_global %>%
    select(serie, CP = RF_CP, Param = RF_Param) %>%
```

```
    mutate(model = "RF")
) %>%
  pivot_longer(cols = c(CP, Param), names_to = "method", values_to = "coverage")

ggplot(coverage_comparison_long, aes(method, coverage, fill = method)) +
  geom_boxplot(alpha = 0.7, width = 0.6) +
  geom_line(aes(group = serie), alpha = 0.2, color = "gray50") +
  facet_wrap(~model) +
  scale_fill_manual(values = c("CP" = color_cp, "Param" = color_param)) +
  geom_hline(yintercept = 0.9, linetype = "dashed", color = "black") +
  theme_minimal(base_size = 12) +
  theme(legend.position = "top") +
  labs(
    title = "Confronto Paired: Coverage CP vs Parametrico",
    subtitle = "Linee grigie connettono stessa serie",
    x = "Metodo",
    y = "Coverage",
    fill = ""
  )
```

ANALISI MISCOVERAGE

```
# serie dove parametrico fallisce (coverage < 95%)
parametric_failures = results_complete %>%
  filter(model == "ARIMA_Param", Coverage < 0.95) %>%
  pull(serie)

cat("Serie con fallimento parametrico (coverage<95%):",
    length(parametric_failures), "\n")

# Confronto CP su queste serie critiche
comparison_failures = results_complete %>%
  filter(serie %in% parametric_failures,
         model %in% c("ARIMA_CP", "ARIMA_Param")) %>%
  select(serie, model, Coverage, Width) %>%
  pivot_wider(names_from = model, values_from = c(Coverage, Width))

# Recovery rate
recovery_rate = mean(comparison_failures$Coverage_ARIMA_CP >= 0.95, na.rm = TRUE)

tibble(
  Metrica = c(
    "Serie con fallimento parametrico",
    "Recovery rate CP (coverage  95%)",
    "Coverage media CP su serie fallite",
    "Coverage media Param su serie fallite"
  ),
  Valore = c(
    length(parametric_failures),
    recovery_rate,
    mean(comparison_failures$Coverage_ARIMA_CP, na.rm = TRUE),
    mean(comparison_failures$Coverage_ARIMA_Param, na.rm = TRUE)
  )
```

```
) %>%
  gt() %>%
  fmt_number(columns = Valore, decimals = 3) %>%
  tab_header(
    title = "Capacità di Recovery della Conformal Prediction",
    subtitle = "Performance su serie dove il parametrico fallisce"
  ) %>%
  tab_style(
    style = cell_fill(color = "#D5F4E6"),
    locations = cells_body(rows = 2)
  )
```

Grafico Confronto

```
# Densità Coverage per ARIMA, ETS e RF
density_data = results_complete %>%
  filter(model %in% c("ARIMA_CP", "ARIMA_Param", "ETS_CP", "ETS_Param",
                      "RF_CP","RF_Param" )) %>%
  mutate(
    base_model = str_remove(model, "_(CP|Param)"),
    method = ifelse(str_detect(model, "_CP"), "CP", "Parametrico")
  )

ggplot(density_data, aes(Coverage, fill = method, color = method)) +
  geom_density(alpha = 0.4, linewidth = 1) +
  geom_vline(xintercept = 0.9, linetype = "dashed",
             color = color_nominal, linewidth = 0.8) +
  facet_wrap(~base_model, ncol = 1) +
  scale_fill_manual(values = c("CP" = color_cp, "Parametrico" = color_param)) +
  scale_color_manual(values = c("CP" = color_cp, "Parametrico" = color_param)) +
  theme_minimal(base_size = 12) +
  theme(legend.position = "top") +
  labs(
    title = "Distribuzione Coverage: CP vs Parametrico",
    subtitle = "Densità empirica su tutte le serie",
    x = "Coverage",
    y = "Densità",
    fill = "Metodo",
    color = "Metodo"
  )
```

ARIMA

```
# Scatter plot: Coverage CP vs Param ARIMA
scatter_arima = results_complete %>%
  filter(model %in% c("ARIMA_CP", "ARIMA_Param")) %>%
  select(serie, model, Coverage) %>%
  pivot_wider(names_from = model, values_from = Coverage) %>%
  drop_na()

ggplot(scatter_arima, aes(ARIMA_Param, ARIMA_CP)) +
  geom_point(alpha = 0.5, color = color_arima, size = 2) +
  geom_abline(slope = 1, intercept = 0, linetype = "dashed",
```

```
              color = "black", linewidth = 0.8) +
  geom_hline(yintercept = 0.9, linetype = "dotted", color = color_nominal) +
  geom_vline(xintercept = 0.9, linetype = "dotted", color = color_nominal) +
  coord_fixed(xlim = c(0.5, 1), ylim = c(0.5, 1)) +
  annotate("text", x = 0.55, y = 0.95,
           label = paste0("N serie con CP > Param: ",
                          sum(scatter_arima$ARIMA_CP > scatter_arima$ARIMA_Param,
                              na.rm = TRUE)),
           hjust = 0, size = 3.5) +
  theme_minimal(base_size = 12) +
  labs(
    title = "Coverage Serie-per-Serie:
    CP vs Parametrico (ARIMA)",
    subtitle = "Punti sopra la diagonale = CP migliore",
    x = "Coverage Parametrico",
    y = "Coverage Conformal"
  )
```

ETS

```
# Scatter plot: Coverage CP vs Param ETS
scatter_ets = results_complete %>%
  filter(model %in% c("ETS_CP", "ETS_Param")) %>%
  select(serie, model, Coverage) %>%
  pivot_wider(names_from = model, values_from = Coverage) %>%
  drop_na()

ggplot(scatter_ets, aes(ETS_Param, ETS_CP)) +
  geom_point(alpha = 0.5, color = color_arima, size = 2) +
  geom_abline(slope = 1, intercept = 0, linetype = "dashed",
              color = "black", linewidth = 0.8) +
  geom_hline(yintercept = 0.9, linetype = "dotted", color = color_nominal) +
  geom_vline(xintercept = 0.9, linetype = "dotted", color = color_nominal) +
  coord_fixed(xlim = c(0.5, 1), ylim = c(0.5, 1)) +
  annotate("text", x = 0.55, y = 0.95,
           label = paste0("N serie con CP > Param: ",
                          sum(scatter_ets$ETS_CP > scatter_ets$ETS_Param,
                              na.rm = TRUE)),
           hjust = 0, size = 3.5) +
  theme_minimal(base_size = 12) +
  labs(
    title = "Coverage Serie-per-Serie:
    CP vs Parametrico (ETS)",
    subtitle = "Punti sopra la diagonale = CP migliore",
    x = "Coverage Parametrico",
    y = "Coverage Conformal"
  )
```

Random Forest

```
# Scatter plot: Coverage CP vs Param RF
scatter_rf = results_complete %>%
  filter(model %in% c("RF_CP", "RF_Param")) %>%
```

```
  select(serie, model, Coverage) %>%
  pivot_wider(names_from = model, values_from = Coverage) %>%
  drop_na()

ggplot(scatter_rf, aes(RF_Param, RF_CP)) +
  geom_point(alpha = 0.5, color = color_arima, size = 2) +
  geom_abline(slope = 1, intercept = 0, linetype = "dashed",
              color = "black", linewidth = 0.8) +
  geom_hline(yintercept = 0.9, linetype = "dotted", color = color_nominal) +
  geom_vline(xintercept = 0.9, linetype = "dotted", color = color_nominal) +
  coord_fixed(xlim = c(0.5, 1), ylim = c(0.5, 1)) +
  annotate("text", x = 0.55, y = 0.95,
           label = paste0("N serie con CP > Param: ",
                          sum(scatter_rf$RF_CP > scatter_rf$RF_Param,
                              na.rm = TRUE)),
           hjust = 0, size = 3.5) +
  theme_minimal(base_size = 12) +
  labs(
    title = "Coverage Serie-per-Serie:
    CP vs Parametrico (RF)",
    subtitle = "Punti sopra la diagonale = CP migliore",
    x = "Coverage Parametrico",
    y = "Coverage Conformal"
  )
```

TABELLA RIASSUNTIVA FINALE: SUPERIORITÀ CP

```
# coverage comparison per tutti i modelli
coverage_comparison_arima_global = results_complete %>%
  filter(model %in% c("ARIMA_CP", "ARIMA_Param")) %>%
  select(serie, model, Coverage) %>%
  pivot_wider(names_from = model, values_from = Coverage) %>%
  drop_na()

coverage_comparison_ets_global = results_complete %>%
  filter(model %in% c("ETS_CP", "ETS_Param")) %>%
  select(serie, model, Coverage) %>%
  pivot_wider(names_from = model, values_from = Coverage) %>%
  drop_na()

coverage_comparison_rf_global = results_complete %>%
  filter(model %in% c("RF_CP", "RF_Param")) %>%
  select(serie, model, Coverage) %>%
  pivot_wider(names_from = model, values_from = Coverage) %>%
  drop_na()

# Test Wilcoxon per tutti i modelli
wilcox_arima_global = wilcox.test(
  coverage_comparison_arima_global$ARIMA_CP,
  coverage_comparison_arima_global$ARIMA_Param,
  paired = TRUE,
  alternative = "greater",
  exact = FALSE
```

```
)

wilcox_ets_global = wilcox.test(
  coverage_comparison_ets_global$ETS_CP,
  coverage_comparison_ets_global$ETS_Param,
  paired = TRUE,
  alternative = "greater",
  exact = FALSE
)

wilcox_rf_global = wilcox.test(
  coverage_comparison_rf_global$RF_CP,
  coverage_comparison_rf_global$RF_Param,
  paired = TRUE,
  alternative = "greater",
  exact = FALSE
)

# Effect size per tutti i modelli
effect_size_arima_global = mean(coverage_comparison_arima_global$ARIMA_CP -
                                  coverage_comparison_arima_global$ARIMA_Param)

effect_size_ets_global = mean(coverage_comparison_ets_global$ETS_CP -
                                coverage_comparison_ets_global$ETS_Param)

effect_size_rf_global = mean(coverage_comparison_rf_global$RF_CP -
                               coverage_comparison_rf_global$RF_Param)

# Tabella riassuntiva finale
final_summary = bind_rows(
  # ARIMA
  tibble(
    Modello = "ARIMA",
    Coverage_CP = mean(coverage_comparison_arima_global$ARIMA_CP, na.rm = TRUE),
    Coverage_Param = mean(coverage_comparison_arima_global$ARIMA_Param, na.rm = TRUE),
    Width_CP = mean(results_complete %>%
                      filter(model == "ARIMA_CP") %>%
                      pull(Width), na.rm = TRUE),
    Width_Param = mean(results_complete %>%
                         filter(model == "ARIMA_Param") %>%
                         pull(Width), na.rm = TRUE),
    P_value_Wilcoxon = wilcox_arima_global$p.value,
    N_serie = nrow(coverage_comparison_arima_global)
  ),
  # ETS
  tibble(
    Modello = "ETS",
    Coverage_CP = mean(coverage_comparison_ets_global$ETS_CP, na.rm = TRUE),
    Coverage_Param = mean(coverage_comparison_ets_global$ETS_Param, na.rm = TRUE),
    Width_CP = mean(results_complete %>%
                      filter(model == "ETS_CP") %>%
                      pull(Width), na.rm = TRUE),
    Width_Param = mean(results_complete %>%
```

```
                        filter(model == "ETS_Param") %>%
                        pull(Width), na.rm = TRUE),
    P_value_Wilcoxon = wilcox_ets_global$p.value,
    N_serie = nrow(coverage_comparison_ets_global)
  ),
  # RF
  tibble(
    Modello = "RF",
    Coverage_CP = mean(coverage_comparison_rf_global$RF_CP, na.rm = TRUE),
    Coverage_Param = mean(coverage_comparison_rf_global$RF_Param, na.rm = TRUE),
    Width_CP = mean(results_complete %>%
                      filter(model == "RF_CP") %>%
                      pull(Width), na.rm = TRUE),
    Width_Param = mean(results_complete %>%
                        filter(model == "RF_Param") %>%
                        pull(Width), na.rm = TRUE),
    P_value_Wilcoxon = wilcox_rf_global$p.value,
    N_serie = nrow(coverage_comparison_rf_global)
  )
) %>%
  mutate(
    Delta_Coverage = Coverage_CP - Coverage_Param,
    Delta_Width = Width_CP - Width_Param,
    Significativo = ifelse(P_value_Wilcoxon < 0.05, " ", "")
  )

final_summary %>%
  gt() %>%
  fmt_number(columns = c(Coverage_CP, Coverage_Param, Width_CP, Width_Param,
                         Delta_Coverage, Delta_Width, P_value_Wilcoxon),
             decimals = 3) %>%
  tab_header(
    title = "SINTESI FINALE: Superiorità Conformal Prediction",
    subtitle = "Confronto sistematico su tutte le serie testate"
  ) %>%
  tab_style(
    style = cell_fill(color = "#D5F4E6"),
    locations = cells_body(columns = c(Coverage_CP, Delta_Coverage))
  ) %>%
  tab_style(
    style = cell_fill(color = "#F8D7DA"),
    locations = cells_body(columns = c(Coverage_Param))
  ) %>%
  tab_style(
    style = cell_text(weight = "bold"),
    locations = cells_body(
      columns = Significativo,
      rows = Significativo == " "
    )
  ) %>%
  cols_label(
    Coverage_CP = "Coverage CP",
    Coverage_Param = "Coverage Param",
```

```
    Width_CP = "Width CP",
    Width_Param = "Width Param",
    Delta_Coverage = "Δ Coverage",
    Delta_Width = "Δ Width",
    P_value_Wilcoxon = "p-value",
    N_serie = "N. Serie"
  )
```

INTERVALLI DI CONFIDENZA PER METRICHE AGGREGATE Bootstrap per stimare incertezza delle metriche aggregate

```
bootstrap_ci = function(x, n_boot = 1000, conf = 0.95) {
  boot_means = replicate(n_boot, mean(sample(x, replace = TRUE), na.rm = TRUE))
  quantile(boot_means, probs = c((1 - conf) / 2, 1 - (1 - conf) / 2))
}

ci_coverage = results_clean %>%
  filter(CP) %>%
  group_by(model) %>%
  summarise(
    Coverage_media = mean(Coverage, na.rm = TRUE),
    CI_lower = bootstrap_ci(Coverage)[1],
    CI_upper = bootstrap_ci(Coverage)[2],
    .groups = "drop"
  )

ggplot(ci_coverage, aes(model, Coverage_media, color = model)) +
  geom_point(size = 4) +
  geom_errorbar(aes(ymin = CI_lower, ymax = CI_upper),
                width = 0.2, linewidth = 1) +
  geom_hline(yintercept = 0.9, linetype = "dashed",
             color = "black", linewidth = 0.8) +
  scale_color_manual(values = c("ARIMA_CP" = color_arima,
                                "ETS_CP" = color_ets,
                                "RF_CP" = color_rf)) +
  theme_minimal(base_size = 12) +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1),
    legend.position = "none"
  ) +
  labs(
    title = "Coverage con Intervalli di Confidenza (95%)",
    subtitle = "Bootstrap con 1000 replicazioni",
    x = "Modello",
    y = "Coverage"
  )
```

```
ci_coverage %>%
  gt() %>%
  fmt_number(columns = -model, decimals = 3) %>%
  tab_header(
    title = "Intervalli di Confidenza Bootstrap (95%)",
    subtitle = "Coverage media per modello CP"
```

```
  ) %>%
  cols_label(
    Coverage_media = "Coverage",
    CI_lower = "CI Lower",
    CI_upper = "CI Upper"
  )
```

Analisi performance per store e state

```
perf_store = results_clean %>%
  filter(CP) %>%
  left_join(df_weekly %>% distinct(serie, store_id), by = "serie") %>%
  group_by(store_id, model) %>%
  summarise(
    Coverage_media = mean(Coverage, na.rm = TRUE),
    Width_media = mean(Width, na.rm = TRUE),
    MSE_medio = mean(MSE, na.rm = TRUE),
    n_serie = n_distinct(serie),
    .groups = "drop"
  )

ggplot(perf_store, aes(store_id, Coverage_media, fill = model)) +
  geom_col(position = "dodge", width = 0.7) +
  geom_hline(yintercept = 0.9, linetype = "dashed",
             color = "black", linewidth = 0.8) +
  scale_fill_manual(values = c("ARIMA_CP" = color_arima,
                               "ETS_CP" = color_ets,
                               "RF_CP" = color_rf)) +
  theme_minimal(base_size = 12) +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1),
    legend.position = "top"
  ) +
  labs(
    title = "Coverage per Store",
    subtitle = "Eterogeneità geografica nella performance CP",
    x = "Store ID",
    y = "Coverage media",
    fill = "Modello"
  )
```

```
perf_state = results_clean %>%
  filter(CP) %>%
  left_join(df_weekly %>% distinct(serie, state_id), by = "serie") %>%
  group_by(state_id, model) %>%
  summarise(
    Coverage_media = mean(Coverage, na.rm = TRUE),
    Width_media = mean(Width, na.rm = TRUE),
    MSE_medio = mean(MSE, na.rm = TRUE),
    n_serie = n_distinct(serie),
    .groups = "drop"
  )
```

```
perf_state %>%
  pivot_wider(
    names_from = model,
    values_from = c(Coverage_media, Width_media, MSE_medio),
    names_sep = "_"
  ) %>%
  gt() %>%
  fmt_number(columns = -c(state_id, n_serie), decimals = 3) %>%
  tab_header(
    title = "Performance CP per Stato",
    subtitle = "Confronto metriche aggregate"
  ) %>%
  tab_spanner(label = "Coverage", columns = starts_with("Coverage")) %>%
  tab_spanner(label = "Width", columns = starts_with("Width")) %>%
  tab_spanner(label = "MSE", columns = starts_with("MSE")) %>%
  tab_options(table.font.size = px(9))
```

CLUSTERING seire e Performance CP

```
series_features = df_weekly %>%
  group_by(serie) %>%
  summarise(
    mean_sales = mean(value, na.rm = TRUE),
    sd_sales = sd(value, na.rm = TRUE),
    cv = sd_sales / (mean_sales + 1e-6),
    trend = cor(seq_along(value), value, use = "complete.obs"),
    zero_frac = mean(value == 0),
    .groups = "drop"
  ) %>%
  drop_na()

set.seed(123)
features_scaled = series_features %>%
  select(cv, trend, zero_frac) %>%
  scale()

kmeans_result = kmeans(features_scaled, centers = 3, nstart = 25)
series_features$cluster = factor(kmeans_result$cluster)

cluster_summary = series_features %>%
  group_by(cluster) %>%
  summarise(
    n_serie = n(),
    CV_medio = mean(cv),
    Trend_medio = mean(trend),
    Zero_frac_media = mean(zero_frac),
    .groups = "drop"
  )

cluster_summary %>%
  gt() %>%
  fmt_number(columns = -c(cluster, n_serie), decimals = 3) %>%
  tab_header(
```

```
    title = "Caratteristiche dei Cluster di Serie",
    subtitle = "K-means con k=3 su CV, Trend, Zero fraction"
  ) %>%
  cols_label(
    n_serie = "N. Serie",
    CV_medio = "CV Medio",
    Trend_medio = "Trend Medio",
    Zero_frac_media = "Frazione Zeri"
  ) %>%
  tab_options(table.font.size = px(9))
```

```
perf_cluster = results_clean %>%
  filter(CP) %>%
  left_join(series_features %>% select(serie, cluster), by = "serie") %>%
  drop_na(cluster) %>%
  group_by(cluster, model) %>%
  summarise(
    Coverage_media = mean(Coverage, na.rm = TRUE),
    Width_media = mean(Width, na.rm = TRUE),
    MSE_medio = mean(MSE, na.rm = TRUE),
    n_serie = n_distinct(serie),
    .groups = "drop"
  )

ggplot(perf_cluster, aes(cluster, Coverage_media, fill = model)) +
  geom_col(position = "dodge", width = 0.7) +
  geom_hline(yintercept = 0.9, linetype = "dashed",
             color = "black", linewidth = 0.8) +
  scale_fill_manual(values = c("ARIMA_CP" = color_arima,
                               "ETS_CP" = color_ets,
                               "RF_CP" = color_rf)) +
  theme_minimal(base_size = 12) +
  theme(legend.position = "top") +
  labs(
    title = "Coverage CP per Cluster di Serie",
    subtitle = "Cluster basati su CV, trend e frazione zeri",
    x = "Cluster",
    y = "Coverage media",
    fill = "Modello"
  )
```

```
ggplot(series_features, aes(cv, trend, color = cluster)) +
  geom_point(alpha = 0.6, size = 2.5) +
  scale_color_manual(values = c("1" = "#E74C3C", "2" = "#3498DB", "3" = "#2ECC71")) +
  theme_minimal(base_size = 12) +
  theme(legend.position = "top") +
  labs(
    title = "Clustering delle Serie Temporali",
    subtitle = "Spazio (CV, Trend)",
    x = "Coefficiente di Variazione",
    y = "Correlazione Trend",
    color = "Cluster"
  )
```

Sensitività dimensione calibration set

```
cal_sizes = c(10, 20, 30, 40, 50)
sensitivity_results = list()
arima_fit_train = auto.arima(train)

for(cal_size in cal_sizes) {

  cal_subset = tail(cal, cal_size)
  cal_fc = forecast(arima_fit_train, h = length(cal_subset))$mean
  errors = cal_subset - cal_fc
  q = quantile(abs(errors), 0.9)

  arima_fit_full_temp = auto.arima(c(train, cal_subset))
  test_fc = forecast(arima_fit_full_temp, h = length(test))$mean

  sensitivity_results[[as.character(cal_size)]] = tibble(
    cal_size = cal_size,
    coverage = mean(test >= (test_fc - q) & test <= (test_fc + q)),
    width = 2 * q
  )
}

sensitivity_df = bind_rows(sensitivity_results) %>%
  pivot_longer(cols = c(coverage, width),
               names_to = "metrica", values_to = "valore")

ggplot(sensitivity_df, aes(cal_size, valore, color = metrica)) +
  geom_line(linewidth = 1.2) +
  geom_point(size = 3) +
  facet_wrap(~metrica, scales = "free_y", ncol = 1,
             labeller = labeller(metrica = c("coverage" = "Coverage",
                                             "width" = "Width"))) +
  geom_hline(data = tibble(metrica = "coverage", valore = 0.9),
             aes(yintercept = valore), linetype = "dashed",
             color = color_nominal) +
  scale_color_manual(values = c("coverage" = color_cp, "width" = color_param)) +
  theme_minimal(base_size = 12) +
  theme(legend.position = "none") +
  labs(
    title = "Sensitività a dimensione calibration set",
    subtitle = "Trade-off tra stabilità coverage e ampiezza intervalli",
    x = "Dimensione calibration set",
    y = "Valore"
  )
```

CROSS-VALIDATION temporale: EXPANDING WINDOW

```
n_folds = 5
fold_size = floor(length(y) / (n_folds + 1))
cv_results = list()

for(fold in 1:n_folds) {
```

```
  train_end = fold_size * (fold + 1)
  cal_start = train_end + 1
  cal_end   = train_end + fold_size
  test_start = cal_end + 1
  test_end   = min(cal_end + fold_size, length(y))

  if(test_end - test_start < 5) next

  train_cv = y[1:train_end]
  cal_cv   = y[cal_start:cal_end]
  test_cv  = y[test_start:test_end]

  arima_cv     = auto.arima(train_cv)
  arima_cal_cv = forecast(arima_cv, h = length(cal_cv))$mean
  q_arima_cv   = quantile(abs(cal_cv - arima_cal_cv), 0.95)

  arima_full_cv = auto.arima(c(train_cv, cal_cv))
  arima_test_cv = forecast(arima_full_cv, h = length(test_cv))$mean
  arima_low_cv  = arima_test_cv - q_arima_cv
  arima_up_cv   = arima_test_cv + q_arima_cv

  cv_results[[fold]] = tibble(
    Fold     = fold,
    Model    = "ARIMA",
    Coverage = mean(test_cv >= arima_low_cv & test_cv <= arima_up_cv),
    Width    = mean(arima_up_cv - arima_low_cv),
    MSE      = mean((test_cv - arima_test_cv)^2)
  )
}

cv_df = bind_rows(cv_results)

cv_df %>%
  gt() %>%
  fmt_number(columns = c(Coverage, Width, MSE), decimals = 3) %>%
  tab_header(
    title = "Cross-Validation Temporale: Expanding Window",
    subtitle = "Performance CP su 5 folds temporali"
  ) %>%
  tab_style(
    style = cell_fill(color = "#E8F4F8"),
    locations = cells_column_labels()
  )
```

```
cv_long = cv_df %>%
  pivot_longer(cols = c(Coverage, Width, MSE),
               names_to = "Metric", values_to = "Value")

ggplot(cv_long, aes(factor(Fold), Value, fill = Metric)) +
  geom_col(position = "dodge", width = 0.7) +
  facet_wrap(~Metric, scales = "free_y", ncol = 1) +
  scale_fill_brewer(palette = "Set1") +
  theme_minimal(base_size = 12) +
```

```
  theme(legend.position = "none") +
  labs(
    title = "Variabilità Performance tra Folds",
    subtitle = "Expanding window cross-validation",
    x = "Fold",
    y = "Valore"
  )
```

```
cv_df %>%
  summarise(
    Coverage_media = mean(Coverage),
    Coverage_sd    = sd(Coverage),
    Width_media    = mean(Width),
    Width_sd       = sd(Width)
  ) %>%
  gt() %>%
  fmt_number(columns = everything(), decimals = 3) %>%
  tab_header(
    title = "Summary Cross-Validation",
    subtitle = "Media e deviazione standard su 5 folds"
  )
```

Stress Test ROBUSTEZZA a outliers

```
contamination_rate   = 0.1
contamination_factor = 3

arima_fit_stress  = auto.arima(train)
cal_fc_stress     = forecast(arima_fit_stress, h = length(cal))$mean
errors_clean_st   = cal - cal_fc_stress
q_clean           = quantile(abs(errors_clean_st), 0.95)

n_contaminate         = floor(length(cal) * contamination_rate)
contaminate_idx       = sample(length(cal), n_contaminate)
cal_contaminated      = cal
cal_contaminated[contaminate_idx] = cal_contaminated[contaminate_idx] * contamination_factor

errors_contaminated = cal_contaminated - cal_fc_stress
q_contaminated      = quantile(abs(errors_contaminated), 0.95)

arima_fit_full_stress = auto.arima(c(train, cal))
test_fc_stress        = forecast(arima_fit_full_stress, h = length(test))$mean

stress_results = tibble(
  Metodo   = c("CP pulito", "CP contaminato"),
  Coverage = c(
    mean(test >= (test_fc_stress - q_clean) & test <= (test_fc_stress + q_clean)),
    mean(test >= (test_fc_stress - q_contaminated) & test <= (test_fc_stress + q_contaminated))
  ),
  Width    = c(2 * q_clean, 2 * q_contaminated),
  Quantile = c(q_clean, q_contaminated)
)
```

```
stress_results %>%
  gt() %>%
  fmt_number(columns = c(Coverage, Width, Quantile), decimals = 3) %>%
  tab_header(
    title = "Stress test robustezza",
    subtitle = "10% outliers nel calibration set (fattore 3x)"
  ) %>%
  tab_style(
    style = cell_fill(color = "#FFE6E6"),
    locations = cells_body(rows = 2)
  )
```

CP con Livelli Multipli

```
alpha_levels = c(0.05, 0.10, 0.15)
multi_level_results = list()

for(alpha in alpha_levels) {

  q_arima_alpha = quantile(abs(cal - arima_cal_fc$mean), 1 - alpha)
  arima_low_alpha = arima_test_fc - q_arima_alpha
  arima_up_alpha  = arima_test_fc + q_arima_alpha

  q_ets_alpha = quantile(abs(cal - ets_cal_fc$mean), 1 - alpha)
  ets_low_alpha = ets_test_fc - q_ets_alpha
  ets_up_alpha  = ets_test_fc + q_ets_alpha

  q_rf_alpha = quantile(abs(cal - rf_cal_pred), 1 - alpha)
  rf_low_alpha = rf_test_pred - q_rf_alpha
  rf_up_alpha  = rf_test_pred + q_rf_alpha

  multi_level_results[[as.character(alpha)]] = tibble(
    Alpha            = alpha,
    Nominal_Coverage = 1 - alpha,
    Model            = c("ARIMA", "ETS", "RF"),
    Empirical_Coverage = c(
      mean(test >= arima_low_alpha & test <= arima_up_alpha),
      mean(test >= ets_low_alpha   & test <= ets_up_alpha),
      mean(test >= rf_low_alpha    & test <= rf_up_alpha)
    ),
    Width = c(
      mean(arima_up_alpha - arima_low_alpha),
      mean(ets_up_alpha   - ets_low_alpha),
      mean(rf_up_alpha    - rf_low_alpha)
    )
  )
}

multi_level_df = bind_rows(multi_level_results)

multi_level_df %>%
  gt() %>%
  fmt_number(columns = c(Alpha, Nominal_Coverage, Empirical_Coverage, Width),
```

```
            decimals = 3) %>%
  tab_header(
    title = "Conformal Prediction a Livelli Multipli",
    subtitle = "Coverage e width per diversi  "
  ) %>%
  tab_style(
    style = cell_fill(color = "#E8F4F8"),
    locations = cells_column_labels()
  ) %>%
  tab_style(
    style = cell_fill(color = "#FFF3CD"),
    locations = cells_body(rows = Nominal_Coverage == 0.90)
  )
```

```
ggplot(multi_level_df, aes(Nominal_Coverage, Empirical_Coverage,
                           color = Model, shape = Model)) +
  geom_point(size = 4) +
  geom_abline(slope = 1, intercept = 0, linetype = "dashed",
              color = "black", linewidth = 0.8) +
  scale_color_manual(values = c("ARIMA" = color_arima,
                                "ETS"   = color_ets,
                                "RF"    = color_rf)) +
  coord_fixed(xlim = c(0.84, 0.96), ylim = c(0.84, 0.96)) +
  theme_minimal(base_size = 12) +
  theme(legend.position = "top") +
  labs(
    title = "Calibrazione Multi-Livello",
    subtitle = "Diagonale = calibrazione perfetta",
    x = "Coverage nominale",
    y = "Coverage empirica",
    color = "Modello", shape = "Modello"
  )
```

```
ggplot(multi_level_df, aes(Width, Empirical_Coverage, color = Model)) +
  geom_point(size = 4) +
  geom_line(aes(group = Model), linewidth = 1) +
  scale_color_manual(values = c("ARIMA" = color_arima,
                                "ETS"   = color_ets,
                                "RF"    = color_rf)) +
  theme_minimal(base_size = 12) +
  theme(legend.position = "top") +
  labs(
    title = "Trade-off Coverage vs Width",
    subtitle = "A livelli di confidenza multipli",
    x = "Width media",
    y = "Coverage empirica",
    color = "Modello"
  )
```

Confronto Interval Score: CP vs PARAMETRICO (ARIMA, ETS, RF)

```
comparison_int_score = tibble(
  Modello = c(
    "ARIMA_CP", "ARIMA_Param",
    "ETS_CP",   "ETS_Param",
    "RF_CP",    "RF_Param"
  ),
  Interval_Score = c(
    interval_score(test, arima_low,    arima_up,    alpha = 0.1),
    interval_score(test, param_low,    param_up,    alpha = 0.1),
    interval_score(test, ets_low,      ets_up,      alpha = 0.1),
    interval_score(test, ets_param_low, ets_param_up, alpha = 0.1),
    interval_score(test, rf_low,       rf_up,       alpha = 0.1),
    interval_score(test, rf_param_low, rf_param_up, alpha = 0.1)
  ),
  Metodo = rep(c("Conformal", "Parametrico"), 3)
) %>%
  separate(Modello, into = c("Base", "Tipo"), sep = "_", remove = FALSE)

ggplot(comparison_int_score, aes(Base, Interval_Score, fill = Tipo)) +
  geom_col(position = "dodge", width = 0.6) +
  scale_fill_manual(
    values = c("CP" = color_cp, "Param" = color_param),
    labels = c("CP" = "Conformal", "Param" = "Parametrico")
  ) +
  theme_minimal(base_size = 12) +
  theme(legend.position = "top") +
  labs(
    title = "Interval Score: Conformal vs Parametrico",
    subtitle = "Valori più bassi indicano intervalli migliori",
    x = "Modello base",
    y = "Interval Score",
    fill = "Metodo"
  )
```

Analisi Temporale: COVERAGE e WIDTH ROLLING

```
window_size = 10

rolling_analysis = function(actual, lower, upper, window = 10) {
  n = length(actual)
  tibble(
    time_point = window:n,
    coverage_rolling = sapply(window:n, function(i) {
      idx = (i - window + 1):i
      mean(actual[idx] >= lower[idx] & actual[idx] <= upper[idx])
    }),
    width_rolling = sapply(window:n, function(i) {
      idx = (i - window + 1):i
      mean(upper[idx] - lower[idx])
    })
  )
}
```

```
rolling_arima = rolling_analysis(test, arima_low, arima_up) %>%
  mutate(model = "ARIMA")
rolling_ets   = rolling_analysis(test, ets_low,   ets_up)   %>%
  mutate(model = "ETS")
rolling_rf    = rolling_analysis(test, rf_low,    rf_up)    %>%
  mutate(model = "RF")

rolling_combined = bind_rows(rolling_arima, rolling_ets, rolling_rf)

ggplot(rolling_combined, aes(time_point, coverage_rolling, color = model)) +
  geom_line(linewidth = 1) +
  geom_hline(yintercept = 0.9, linetype = "dashed",
             color = "black", linewidth = 0.8) +
  geom_smooth(se = TRUE, alpha = 0.2, linewidth = 0.5) +
  scale_color_manual(values = c("ARIMA" = color_arima,
                                "ETS"   = color_ets,
                                "RF"    = color_rf)) +
  theme_minimal(base_size = 12) +
  theme(legend.position = "top") +
  labs(
    title = "Coverage Rolling nel Test Set",
    subtitle = paste0("Finestra mobile di ", window_size, " osservazioni"),
    x = "Posizione temporale nel test set",
    y = "Coverage empirica",
    color = "Modello"
  )
```

```
ggplot(rolling_combined, aes(time_point, width_rolling, color = model)) +
  geom_line(linewidth = 1) +
  geom_smooth(se = TRUE, alpha = 0.2, linewidth = 0.5) +
  scale_color_manual(values = c("ARIMA" = color_arima,
                                "ETS"   = color_ets,
                                "RF"    = color_rf)) +
  theme_minimal(base_size = 12) +
  theme(legend.position = "top") +
  labs(
    title = "Ampiezza Intervalli Rolling nel Test Set",
    subtitle = "Verifica stabilità dell'incertezza predittiva",
    x = "Posizione temporale nel test set",
    y = "Width media",
    color = "Modello"
  )
```

```
temporal_stability = rolling_combined %>%
  group_by(model) %>%
  summarise(
    Coverage_media = mean(coverage_rolling),
    Coverage_sd    = sd(coverage_rolling),
    Coverage_min   = min(coverage_rolling),
    Coverage_max   = max(coverage_rolling),
    Width_media    = mean(width_rolling),
    Width_trend    = cor(time_point, width_rolling),
    .groups = "drop"
```

```
  )

temporal_stability %>%
  gt() %>%
  fmt_number(columns = everything(), decimals = 3) %>%
  tab_header(
    title = "Stabilità Temporale della Conformal Prediction",
    subtitle = "Analisi variabilità coverage e width nel test set"
  ) %>%
  tab_style(
    style = cell_fill(color = "#E8F4F8"),
    locations = cells_column_labels()
  ) %>%
  cols_label(
    Coverage_media = "Cov. Media",
    Coverage_sd    = "Cov. SD",
    Coverage_min   = "Cov. Min",
    Coverage_max   = "Cov. Max",
    Width_media    = "Width Media",
    Width_trend    = "Width Trend"
  )
```

Metriche per intervalli predittivi

```
msis = function(y, lower, upper, alpha = 0.1, y_train) {
  naive_mae = mean(abs(diff(y_train)))
  interval_score(y, lower, upper, alpha) / naive_mae
}

winkler_score = function(y, lower, upper, alpha = 0.1) {
  width = upper - lower
  penalty_lower = (2 / alpha) * pmax(0, lower - y)
  penalty_upper = (2 / alpha) * pmax(0, y - upper)
  mean(width + penalty_lower + penalty_upper)
}

additional_metrics = tibble(
  Model = c("ARIMA_CP", "ETS_CP", "RF_CP"),
  MSIS = c(
    msis(test, arima_low, arima_up, y_train = train),
    msis(test, ets_low,   ets_up,   y_train = train),
    msis(test, rf_low,    rf_up,    y_train = train)
  ),
  Winkler = c(
    winkler_score(test, arima_low, arima_up),
    winkler_score(test, ets_low,   ets_up),
    winkler_score(test, rf_low,    rf_up)
  ),
  Pinball_lower = c(
    pinball_loss(test, arima_low, alpha = 0.05),
    pinball_loss(test, ets_low,   alpha = 0.05),
    pinball_loss(test, rf_low,    alpha = 0.05)
  ),
```

```
  Pinball_upper = c(
    pinball_loss(test, arima_up, alpha = 0.95),
    pinball_loss(test, ets_up,   alpha = 0.95),
    pinball_loss(test, rf_up,    alpha = 0.95)
  )
)

additional_metrics %>%
  gt() %>%
  fmt_number(columns = -Model, decimals = 3) %>%
  tab_header(
    title = "Metriche Aggiuntive per Intervalli Predittivi",
    subtitle = "MSIS, Winkler Score, Pinball Loss"
  ) %>%
  tab_style(
    style = cell_fill(color = "#E8F4F8"),
    locations = cells_column_labels()
  ) %>%
  cols_label(
    MSIS          = "MSIS",
    Winkler       = "Winkler Score",
    Pinball_lower = "Pinball (5%)",
    Pinball_upper = "Pinball (95%)"
  )
```

```
additional_metrics_long = additional_metrics %>%
  pivot_longer(cols = -Model, names_to = "Metric", values_to = "Value")

ggplot(additional_metrics_long, aes(Model, Value, fill = Model)) +
  geom_col(width = 0.6) +
  facet_wrap(~Metric, scales = "free_y", ncol = 2) +
  scale_fill_manual(values = c(
    "ARIMA_CP" = color_arima,
    "ETS_CP"   = color_ets,
    "RF_CP"    = color_rf
  )) +
  theme_minimal(base_size = 12) +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1),
    legend.position = "none"
  ) +
  labs(
    title    = "Confronto Metriche per Intervalli Predittivi",
    subtitle = "Scale indipendenti per metrica",
    x = "Modello",
    y = "Valore"
  )
```

TABELLA RIASSUNTIVA COMPLETA

```
results_clean %>%
  filter(CP) %>%
  group_by(model) %>%
```

```
  summarise(
    MSE_medio    = mean(MSE, na.rm = TRUE),
    Coverage_media = mean(Coverage, na.rm = TRUE),
    Coverage_sd  = sd(Coverage, na.rm = TRUE),
    Width_media  = mean(Width, na.rm = TRUE),
    Width_sd     = sd(Width, na.rm = TRUE),
    n_serie      = n(),
    .groups = "drop"
  ) %>%
  gt() %>%
  fmt_number(columns = everything(), decimals = 3) %>%
  tab_header(
    title = "Sintesi performance modelli con Conformal Prediction",
    subtitle = "Metriche aggregate su tutte le serie"
  ) %>%
  tab_style(
    style = list(cell_fill(color = "#E8F4F8"), cell_text(weight = "bold")),
    locations = cells_column_labels()
  ) %>%
  tab_options(table.font.size = px(9))
```

FORECAST MULTI-STEP: Analisi Degradazione Performance

```
horizons = 1:4
multi_step_results = list()

for(h in horizons) {

  idx       = seq(h, length(test), by = h)
  test_h    = test[idx]

  arima_fc_h  = arima_test_fc[idx]
  arima_low_h = arima_low[idx]
  arima_up_h  = arima_up[idx]

  ets_fc_h  = ets_test_fc[idx]
  ets_low_h = ets_low[idx]
  ets_up_h  = ets_up[idx]

  rf_fc_h  = rf_test_pred[idx]
  rf_low_h = rf_low[idx]
  rf_up_h  = rf_up[idx]

  multi_step_results[[h]] = tibble(
    Horizon = h,
    Model   = c("ARIMA", "ETS", "RF"),
    Coverage = c(
      mean(test_h >= arima_low_h & test_h <= arima_up_h),
      mean(test_h >= ets_low_h   & test_h <= ets_up_h),
      mean(test_h >= rf_low_h    & test_h <= rf_up_h)
    ),
    Width = c(
      mean(arima_up_h - arima_low_h),
```

```
      mean(ets_up_h   - ets_low_h),
      mean(rf_up_h    - rf_low_h)
    ),
    MSE = c(
      mean((test_h - arima_fc_h)^2),
      mean((test_h - ets_fc_h)^2),
      mean((test_h - rf_fc_h)^2)
    )
  )
}

multi_step_df = bind_rows(multi_step_results)

ggplot(multi_step_df, aes(Horizon, Coverage, color = Model)) +
  geom_line(linewidth = 1.2) +
  geom_point(size = 3) +
  geom_hline(yintercept = 0.9, linetype = "dashed",
             color = "black", linewidth = 0.8) +
  scale_color_manual(values = c("ARIMA" = color_arima,
                                "ETS"   = color_ets,
                                "RF"    = color_rf)) +
  scale_x_continuous(breaks = horizons) +
  theme_minimal(base_size = 12) +
  theme(legend.position = "top") +
  labs(
    title = "Degradazione Coverage a Orizzonti Multipli",
    subtitle = "Performance CP per h=1,2,3,4 settimane",
    x = "Orizzonte di forecast (settimane)",
    y = "Coverage empirica",
    color = "Modello"
  )
```

```
ggplot(multi_step_df, aes(Horizon, Width, color = Model)) +
  geom_line(linewidth = 1.2) +
  geom_point(size = 3) +
  scale_color_manual(values = c("ARIMA" = color_arima,
                                "ETS"   = color_ets,
                                "RF"    = color_rf)) +
  scale_x_continuous(breaks = horizons) +
  theme_minimal(base_size = 12) +
  theme(legend.position = "top") +
  labs(
    title = "Crescita Width a Orizzonti Multipli",
    subtitle = "Incertezza crescente con orizzonte temporale",
    x = "Orizzonte di forecast (settimane)",
    y = "Width media",
    color = "Modello"
  )
```

```
multi_step_df %>%
  gt() %>%
  fmt_number(columns = c(Coverage, Width, MSE), decimals = 3) %>%
  tab_header(
```

```
    title = "Performance Multi-Step",
    subtitle = "Coverage, width e MSE per orizzonte temporale"
  ) %>%
  tab_style(
    style = cell_fill(color = "#E8F4F8"),
    locations = cells_column_labels()
  ) %>%
  tab_style(
    style = cell_fill(color = "#FFF3CD"),
    locations = cells_body(rows = Horizon == 1)
  )
```

CASE STUDY: Analisi dettagliata di serie rappresentative(casuali)

```
case_series = series_features %>%
  group_by(cluster) %>%
  slice_sample(n = 1) %>%
  pull(serie)

analyze_case_serie = function(serie_id) {

  y_case     = df_weekly %>%
    filter(serie == serie_id) %>%
    arrange(date) %>%
    pull(value)

  y_case     = (y_case - mean(y_case)) / sd(y_case)
  n_case     = length(y_case)
  train_case = y_case[1:floor(0.6*n_case)]
  cal_case   = y_case[(floor(0.6*n_case)+1):floor(0.8*n_case)]
  test_case  = y_case[(floor(0.8*n_case)+1):n_case]

  # ARIMA
  arima_case      = auto.arima(train_case)
  arima_cal_case  = forecast(arima_case, h = length(cal_case))$mean
  q_arima_case    = quantile(abs(cal_case - arima_cal_case), 0.95)
  arima_full_case = auto.arima(c(train_case, cal_case))
  arima_test_case = forecast(arima_full_case, h = length(test_case))$mean
  arima_low_case  = arima_test_case - q_arima_case
  arima_up_case   = arima_test_case + q_arima_case

  # ETS
  ets_case      = ets(train_case)
  ets_cal_case  = forecast(ets_case, h = length(cal_case))$mean
  q_ets_case    = quantile(abs(cal_case - ets_cal_case), 0.95)
  ets_full_case = ets(c(train_case, cal_case))
  ets_test_case = forecast(ets_full_case, h = length(test_case))$mean
  ets_low_case  = ets_test_case - q_ets_case
  ets_up_case   = ets_test_case + q_ets_case

  # RF
  rf_case     = randomForest(y ~ ., data = make_lag_df(train_case))
  rf_cal_case = numeric(length(cal_case))
```

```r
hist_case   = train_case

for(i in seq_along(cal_case)){
  x       = tail(hist_case, 8)
  newdata = as.data.frame(t(rev(x)))
  colnames(newdata) = paste0("lag", 1:8)
  rf_cal_case[i] = predict(rf_case, newdata)
  hist_case = c(hist_case, rf_cal_case[i])
}

q_rf_case    = quantile(abs(cal_case - rf_cal_case), 0.95)
rf_full_case = randomForest(y ~ ., data = make_lag_df(c(train_case, cal_case)))

rf_test_case = numeric(length(test_case))
hist_case    = c(train_case, cal_case)

for(i in seq_along(test_case)){
  x       = tail(hist_case, 8)
  newdata = as.data.frame(t(rev(x)))
  colnames(newdata) = paste0("lag", 1:8)
  rf_test_case[i] = predict(rf_full_case, newdata)
  hist_case = c(hist_case, rf_test_case[i])
}

rf_low_case = rf_test_case - q_rf_case
rf_up_case  = rf_test_case + q_rf_case

features_case = series_features %>% filter(serie == serie_id)

list(
  serie   = serie_id,
  cluster = features_case$cluster,
  cv      = features_case$cv,
  trend   = features_case$trend,
  test    = test_case,
  # ARIMA
  arima_pred     = arima_test_case,
  arima_low      = arima_low_case,
  arima_up       = arima_up_case,
  arima_coverage = mean(test_case >= arima_low_case & test_case <= arima_up_case),
  arima_width    = mean(arima_up_case - arima_low_case),
  arima_mse      = mean((test_case - arima_test_case)^2),
  # ETS
  ets_pred     = ets_test_case,
  ets_low      = ets_low_case,
  ets_up       = ets_up_case,
  ets_coverage = mean(test_case >= ets_low_case & test_case <= ets_up_case),
  ets_width    = mean(ets_up_case - ets_low_case),
  ets_mse      = mean((test_case - ets_test_case)^2),
  # RF
  rf_pred     = rf_test_case,
  rf_low      = rf_low_case,
  rf_up       = rf_up_case,
```

```
    rf_coverage = mean(test_case >= rf_low_case & test_case <= rf_up_case),
    rf_width    = mean(rf_up_case - rf_low_case),
    rf_mse      = mean((test_case - rf_test_case)^2)
  )
}

case_results = map(case_series, analyze_case_serie)

# Tabella comparativa
map_dfr(case_results, ~tibble(
  Serie    = .x$serie,
  Cluster  = .x$cluster,
  CV       = .x$cv,
  Trend    = .x$trend,
  Modello  = c("ARIMA", "ETS", "RF"),
  Coverage = c(.x$arima_coverage, .x$ets_coverage, .x$rf_coverage),
  Width    = c(.x$arima_width,    .x$ets_width,    .x$rf_width),
  MSE      = c(.x$arima_mse,      .x$ets_mse,      .x$rf_mse)
)) %>%
  gt() %>%
  fmt_number(columns = c(CV, Trend, Coverage, Width, MSE), decimals = 3) %>%
  tab_header(
    title    = "Case Study: Serie Rappresentative",
    subtitle = "Una serie per cluster - tutti i modelli"
  ) %>%
  tab_style(
    style = cell_fill(color = "#E8F4F8"),
    locations = cells_column_labels()
  ) %>%
  tab_style(
    style = cell_fill(color = "#FFF3CD"),
    locations = cells_body(columns = everything(),
                           rows = Modello == "RF")
  )
```

```
# Grafici
for(i in seq_along(case_results)) {

  case = case_results[[i]]
  n_t  = length(case$test)

  df_case_plot = bind_rows(
    tibble(t = 1:n_t, test = case$test,
           pred = case$arima_pred,
           low  = case$arima_low,
           up   = case$arima_up,
           model = "ARIMA"),
    tibble(t = 1:n_t, test = case$test,
           pred = case$ets_pred,
           low  = case$ets_low,
           up   = case$ets_up,
           model = "ETS"),
    tibble(t = 1:n_t, test = case$test,
```

```
            pred = case$rf_pred,
            low  = case$rf_low,
            up   = case$rf_up,
            model = "RF")
  )

  p = ggplot(df_case_plot, aes(t, test)) +
    geom_line(linewidth = 0.8) +
    geom_line(aes(y = pred, color = model), linewidth = 1) +
    geom_ribbon(aes(ymin = low, ymax = up, fill = model),
                alpha = 0.2) +
    facet_wrap(~model, ncol = 1) +
    scale_color_manual(values = c("ARIMA" = color_arima,
                                  "ETS"   = color_ets,
                                  "RF"    = color_rf)) +
    scale_fill_manual(values  = c("ARIMA" = color_arima,
                                  "ETS"   = color_ets,
                                  "RF"    = color_rf)) +
    theme_minimal(base_size = 12) +
    theme(legend.position = "none") +
    labs(
      title    = paste0("Case Study: Serie ", case$serie,
                        " | Cluster ", case$cluster),
      subtitle = paste0(
        "ARIMA → Cov: ", round(case$arima_coverage, 2),
        " | ETS → Cov: ",  round(case$ets_coverage,   2),
        " | RF → Cov: ",   round(case$rf_coverage,    2)
      ),
      x = "Tempo",
      y = "Valore (standardizzato)"
    )

  print(p)
}
```