UNIVERSITÀ
DEGLI STUDI
DI PADOVA

DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

CORSO DI LAUREA MAGISTRALE IN
INGEGNERIA DELL'AUTOMAZIONE

Identificazione di Stili di Guida

modello statistico
basato su tecniche di machine learning

Relatore: Prof. Alessandro Chiuso

Laureando: Dario Bragadin

Correlatore: Dott. Matteo Vanin

ANNO ACCADEMICO 2021 – 2022

Data di laurea 11 luglio 2022

Interdipendenza pensiero azione

Bortoluzzi Adelino

Ringraziamenti

Ringrazio innanzitutto il professor Alessandro Chiuso che mi ha seguito in questo periodo per me molto importante, ringrazio Matteo Vanin che, assieme a tutto il team di Data Science, mi ha accompagnato all'esplorazione di tutto il mondo "Targa", e ringrazio profondamene Francesco De Bettin per avermelo fatto scoprire.
Ringrazio Adelino che mi ha accompagnato in tutti questi anni universitari, probabilmente non ce l'avrei fatta senza di lui.
Ringrazio mia sorella Ilaria, gli amici di SMR, ben più che compagni d'avventura e tutte le splendide persone in Contrà Leopardi.
Ringrazio i miei genitori per avermi supportato soprattutto economicamente in questi lunghi ma brevi anni. Ringrazio tutte le persone conosciute in quel di Padova, nel me di adesso ci sono parti cocostruite con tutti voi.
Un ringraziamento particolare ai compagni di corso con i quali ho condiviso tante fatiche dello studio e ai Bombers, amici da quando l'università è cominciata.

Indice

Sommario

L'obiettivo di questo elaborato è riportare la mia esperienza spiegando gli argomenti affrontati nel corso del mio percorso di tesi.
Lo scopo finale del lavoro svolto è di assegnare un punteggio ad un determinato stile di guida di un soggetto in uno stabilito intervallo di tempo, ricostruendo il profilo di accelerazione-velocità partendo dalle statistiche descrittive di un segnale campionato.
Questo risulta utile nel caso di applicazione su flotte aziendali per l'assegnazione di premi aziendali alle persone più prudenti oppure, nel caso di compagnie assicurative di utilizzarlo come parametro per classificare una persona in una categoria di rischio, categoria che farà variare il costo della RCA, aumentandola nel caso in cui ad un individuo sia associato un valore elevato in quanto una guida aggressiva può portare ad un numero maggiore di incidenti.

Capitolo 1

Introduzione

Nel mondo sono più di 50 milioni le automobili prodotte ogni anno, soprattutto nei paesi industrializzati (Nord America, Europa e Giappone) le famiglie ne possiedono almeno una sebbene sia il bene più costoso dopo la casa.
In questi paesi un lavoratore ogni 7 viene impiegato per la sua fabbricazione e commercializzazione. Questo è già sufficiente per dare un'idea della dimensione dell'industria automobilistica.
Nessuna grande economia nel mondo moderno è impostata senza un significativo settore automobilistico e nessun prodotto nella storia ha avuto un così profondo impatto contemporaneamente sugli stili di vita, sull'ambiente, sui luoghi di lavoro e di residenza.
Grazie a questa diffusione, non si sono sviluppate solo le aziende manifatturiere, ma anche quelle legate ai servizi come, nel mio caso, "Targa telematics", l'azienda presso cui ho svolto il lavoro di tesi.
Uno dei servizi che offre è la gestione di flotte aziendali, che attua installando nelle auto del parco veicoli dell'azienda cliente uno o più dispositivi, per tenere monitorato ogni singolo mezzo tramite l'invio periodico dei dati relativi a qualsiasi segnale disponibile in un veicolo.
Nell'era dell'IOT e dei Big Data è sempre più importante riuscire a gestire questi grandi flussi di informazione e ad estrapolarne elementi utili.
Grazie alle tecniche di Machine Learning e al continuo miglioramento delle capacità computazionali a disposizone, è possibile elaborare velocemente

enormi quantitativi di dati. Ed è qui che entra in gioco il ruolo del Data-analyst, persona molto richiesta nella cosiddetta industria 4.0. Questa figura è in grado di avere le competenze adatte per utilizzare al meglio quanto spiegato in precedenza.
La tesi si basa proprio sulla capacità di ottenere quante più informazioni possibili da quanti meno elementi possibili, il tutto moltiplicato per grandissime quantità di dati in ingresso.
Utilizzando i dati che il dispositivo, installato all'interno delle automobili, invia, l'obiettivo è analizzare l'andamento delle accelerazioni e delle velocità in modo da capire come il propretario del veicolo guidi, attribuendogli un punteggio e confrontandolo con gli altri membri del parco macchine.
Il punteggio dipende da come il guidatore si approccia alla strada, come accelera, come frena, in che modo affronta le curve, se sfora i limiti di velocità e per quanto tempo.
Nel Capitolo 2-*Introduzione pratica*- è possibile trovare un'introduzione tecnica e dettagliata del problema affrontato.
Il Capitolo 3-*Setup e prime analisi*- contiene una descrizione del dispositivo utilizzato con il relativo firmware e un'analisi di fattibilità.
Al capitolo 4-*Analisi e Fit*- viene descritto il procedimento per la creazione del modello e tutte le analisi fatte per capire le condizioni iniziali migliori per la creazione del modello e un'implementazione hardware creata per riuscire ad avere più dati a disposizione di quanti forniti dall'apparecchio ufficiale,in modo da poter svolgere in autonomia un'indagine più approfondita senza doversi rivolgere ad enti esterni.
Il Capitolo 5-*Analisi e stili di guida*- tratta l'assegnazione dei punteggi per l'identificazione del tipo di stile tenuto durante un percorso.
Successivamente al Capitolo 6-*Riconoscimento dossi*- è presente l'approfondimento sul riconoscimento dei dossi, i rallentatori stradali, ed infine al Capitolo 7-*Conclusioni*- le conclusioni con dei suggerimenti di miglioramento.

Capitolo 2

Introduzione pratica

2.1 Introduzione all'azienda

Targa Telematics è un'azienda nata nell'anno 2000 con sede legale a Treviso, una sede secondaria a Torino e altre sparse in tutta Europa, ha un numero di dipendenti, costantemente in aumento, di 140, fatturato 49 M/€ nel 2021.
Si occupa di gestione di flotte aziendali, collaborando con aziende come Poste Italiane o compagnie assicurative come ad esempio Allianz.
Più in generale è possibile dire che si occupa di smart mobility, con tutto ciò che ha a che fare con soluzioni ad alto contenuto tecnologico per permettere ai vari clienti di sfruttare al meglio le potenzialità di un ambiente completamente collegato, più protetto e con maggiore sicurezza.

2.2 Stato dell'arte

L'azienda presso cui sto svolgendo il tirocinio fornisce già una stima di stile di guida, però meno precisa ed approfondita di come l'argomento verrà trattato in questo lavoro.
È possibile trovare esempi di identificazione di stili di guida nei siti di alcune compagnie assicurative, che sempre più spesso cercano di far comprendere, nella polizza auto, anche la presenza di una scatola nera, in modo da poter tenere traccia degli spostamenti (utili nel caso di furto del veicolo), per gestire l'emergenza nel caso di incidente ed anche per valutare lo stile di

guida del conducente.
Per quanto riguarda quest'ultimo punto, ogni compagnia spiega in maniera generica il metodo che utilizza.
Ad esempio Generali e Cattolica assigurazioni dichiarano quanto segue.
Generali assicurazioni:
L'app registra per te i tipi di eventi durante il viaggio in macchina come per esempio le accelerazioni, le frenate brusche, il comportamento nei pressi degli incroci, gli eccessi di velocità e le distrazioni alla guida. In più visualizzi il percorso effettuato, i km percorsi e la durata di ogni tuo viaggio.
Cattolica Assicurazioni,
Stile di guida: Per visualizzare lo stile di guida, il numero di chilometri percorsi, le tipologie di strade percorse (urbana, extraurbana, autostrada) i giorni di guida (week end, festivi, feriali), e gli orari (giorno e notte).
Come si può vedere, le descrizioni sui siti delle compagnie sono molto generici.
In questa tesi è stato fatto uno studio più approfondito in merito.

2.3 Obiettivi raggiunti

Partendo da alcuni dispositivi che verranno descritti nella Sezione 3.2, installati sulla flotta di veicoli di un'azienda privata ed avendo a disposizione le seguenti informazioni:

1. Dati accelerometro
2. Dati velocità
3. Limiti di velocità nelle varie strade
4. Tempo trascorso alla guida
5. Orario di guida

ho cercato di capire qual'è il comportamento che un utente tiene mentre si trova al volante.
Nel corso della tesi utilizzando queste informazioni è stato possibile studiare diversi approcci alla caratterizzazione del guidatore, da una parte orientata

più alla classificazione in categorie e dall'altra all'attribuzione di un punteggio.
L'analisi sperimentale fatta in validazione dei due metodi ha mostrato come attraverso la descrizione dello stile di guida di una persona tramite un punteggio, costruendo un modello dell'andamento della guida, da assegnare ad ogni conducente confrontandolo con il resto dei guidatori appartenenti alla flotta, si riesca a raggiungere il risultato sperato con anche ulteriori possibilità di implementazioni. L'altro invece risulta non essere adeguato e non porta, con le metodologie da me applicate, a risultati apprezzabili.

Capitolo 3

Setup e prime analisi

3.1 Posizionamento

Tutti i dati utilizzati provengono da questo strumento posizionato all'interno dei veicoli, che chiameremo "scatola nera" in quanto registra i dati relativi alla velocità del mezzo, alla localizzazione, alle accelerazione sugli assi x, y e z, dove l'asse x rappresenta l'avanzamento del veicolo, l'asse y lo spostamento laterale e l'asse z la traslazione verticale, come rappresentato in Figura 3.1, così come l'attivazione dei sistemi di sicurezza e spie d'allarme.

Sempre più auto sono in possesso di questi dispositivi, in quanto, nal caso di auto aziendali il responsabile del parco veicoli, utilizzando questi dispositivi, è in grado di tenere monitorati tutti i mezzi con estrema semplicità. Nel caso di auto private, alcuni decidono di installarle a fronte di agevolazioni sulle tariffe delle polizze auto, avere la black box consente infatti di risparmiare sul premio assicurativo ed evitare aumenti di prezzo in caso di guida prudente.

Con questo apparecchio, in caso di incidente è quindi possibile capire se il conducente andava troppo veloce, se usava la cintura o da quanto tempo era alla guida, inoltre, nel caso di furto, è possibile intervenire in maniera tempestiva e, se il dispositivo non viene rimosso, riuscire a fare un tracking del veicolo, riuscendo così a recuperarlo rapidamente e con estrema facilità.

Il dispositivo viene posizionato all'interno del veicolo, saldamente ancorato

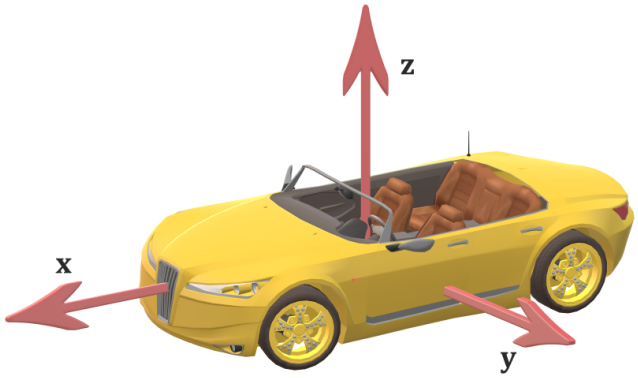


Figura 3.1: Posizione assi del dispositivo all'interno del veicolo

al telaio dietro al cruscotto e collegato al CAN-bus(Controller Area Network), uno standard seriale per bus di campo utilizzato per la comunicazione tra diverse unità di controllo elettronico, è stato espressamente progettato per funzionare senza problemi anche in ambienti fortemente disturbati dalla presenza di camoi elettromagnetici, temperature estreme o umidità.
In Figura 3.2 possimamo trovare un esempio di dispositivo utilizzato con relativo collegamento al CAN-bus.

Figura 3.2: Dispositivo e relativo collegamento al CAN-bus

3.2 Tipo di dispositivo

I congegni utilizzati sono i "GV300GP", un dispositivo creato dell'azienda "Queclink" in grado di tenere sotto controllo tutte le dinamiche del veicolo al quale è collegato e inviarle con una determinata periodicità.
Come mostrato in Figura 3.3, il dispositivo invia tramite rete GSM/GPRS i dati raccolti, che verranno acquisiti dal server aziendale ed interpretati tramite uno strumento di decodifica che permette di leggere le informazioni inviate in formato ASCII, per renderle fruibili agli utenti finali, convogliando il flusso di informazioni interpretate in un database interno.
L'utente potrà accedere a questo database o attraverso internet collegandosi alla VPN aziendale, oppure se si è già collegati alla rete aziendale, accederci direttamente.
In Figura 3.3 si possono notare due colori che rappresentano i due percorsi possibili, il rosso e il nero. Avendo sempre svolto il tirocinio in azienda, ho sempre potuto seguire il percorso rosso.
Il dispositivo sia spedisce che riceve valori in caratteri ASCII utilizzando

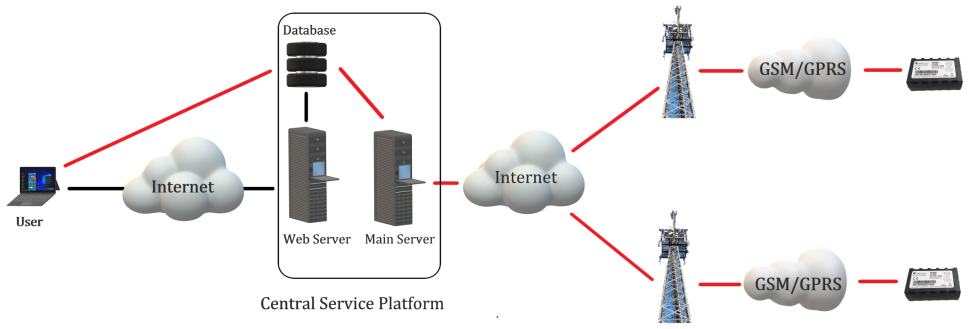


Figura 3.3: Architettura del sistema di comunicazione

la rete GSM (Sistema globale per comunicazioni mobili), uno standard di seconda generazione(2G), basato sulla comunicazione digitale, ampiamente utilizzato nell'ambito della telefonia mobile.
Analizzando l'apparecchio in maniera approfondita, è possibile osservare che al suo interno è composto da:

- accelerometro con frequenza di campionamento a 100 Hz.

- modulo GSM per l'invio dei dati.
- modulo GPS per ottenere tutte le informazioni sulla posizione del veicolo, con le seguenti caratteristiche:
  1. Precisione 2.5m.
  2. Partenza a frendo, cioè il primo avvio, dopo un lungo periodo di inutilizzo: 29 sec di media.
  3. partenza a caldo, cioè dopo un breve perido trascorso dall'ultima richiesta di posizione: < 1 sec.
- un componente elettronico programmato (Memoria flash) $MX25U12835FZNI\_10G$, 128Mbit flash, che è responsabile di tutto l'organico, è il luogo dove si trova il firmware, che da le indicazioni a tutti gli altri componenti su come comportarsi.
- interfaccia seriale RS232C per lo scambio di dati tra dispositivi digitali.
- batteria interna in grado di mantenere acceso il dispositivo per la rilevazione di uno stacco cavi (tentata manomissione del dispositivo) o altri eventi significativi

L'insieme di tutte le istruzioni sul tipo di dato da inviare, la frequenza di invio, le modalità etc., compongono il Firmware, che gestisce tutto il flusso di informazioni, istruzioni che l'utente non vede e non può modificare, ma fondamentali per l'invio e l'elaborazione dei dati.

3.3 Funzionamento del Firmware

In ogni apparecchiatura distribuita sul mercato dall'azienda, è presente un firmware che può variare in base alle esigenze e peculiarità del cliente, per esempio le necessità di un'azienda composta da autocarri per trasporto di materiali persanti sono diverse da una realtà composta da tricicli elettrici per il delivery.
In questo caso di studio è stato analizzato un parco veicoli aziendale con auto principalmente Toyota Yaris, è stato scelto in modo che fosse più uniforme

possibile per poter confrontare i risultati ottenuti con maggiore semplicità, infatti analizzando auto diverse, con motori molto diversi hanno comportamenti differenti tra loro, è difficile confrontare un'utilitaria con un'auto da corsa (es Fiat Panda con Ferrari F12).
Su queste auto è stato implementato un firmware apposito, in aggiunta a quello già presente in precedenza, che funziona in questo modo:
innanzitutto ci sono due diverse informazioni da tenere in considerazione, l'accelerazione e la velocità.
Relativamente all'accelerazione, dopo l'accensione del veicolo, viene generato un istogramma vuoto che verrà aggiornato con i valori ricavati dall'acceleromentro ad ogni campionamento, ogni campione verrà aggiunto all'istogramma preesistente arricchendolo, oppure nel caso in cui non non ci sia ancora l'istogramma, il campione darà il via alla routine per la sua creazione.
L'istogramma è diviso lungo l'asse x in un determinato numero di intervalli, chiamati bin, valore imponibile dall'utilizzatore, la sua analisi verrà trattata nella Sezione 4.2.
Trascorsi 2 minuti l'istogramma verrà trasmesso tramite rete GSM e il processo ricomincerà.
L'istogramma completo è inviato al server aziendale sotto forma di percentili, statistiche descrittive che illustrano come le accelerazioni sono distribuite sul'istogramma e che poi verranno utilizzati per ricostruire, a ritroso, il segnale originale.
Questo tipo di descrizione implica che alcuni andamenti vengano persi, in quanto due istogrammi uguali possono nascondere comportamenti differenti, ad esempio non viene evidenziato se in un certo momento (all'interno del periodo considerato) il comportamento cambia o è mediamente sempre rimasto uguale, oppure se c'è una costate oscillazione(cambiamento radicale di stile di guida avvenuto due o più volte all'interno di una stessa tratta di strada) tra un comportamento sportivo ed uno moderato.
Per il tipo di analisi considerato, queste informazioni sono irrilevanti, e tutte le informazioni utili sono ricavabili da processo in atto.
Quindi il compromesso della perdita di dati in cambio di un'invio di elementi limitato che occupi meno banda possibile e che riesca a contenere i costi è

un bilanciamento accettabile. Dopo un'attenta analisi spiegata nella Sezione 4.2 sono stati utilizzati diciassette percentili per descrivere l'istogramma ed inviati ogni due minuti, di seguito è possibile osservare i diciassette livelli dei percentili impiegati:

$$\begin{aligned}&[0, 0.01, 0.02, 0.04, 0.08, 0.15, 0.25, 0.37, 0.5,\\ &0.63, 0.75, 0.85, 0.92, 0.96, 0.98, 0.99, 1]\end{aligned} \tag{3.1}$$

In queste immagini, Figura 3.4, è possibile osservare 2 minuti del segnale originario, il corrispettivo istogramma e la descrizione in percentili di quest'ultimo. Il processo si interrompe anche nel caso in cui il veicolo venga spento con una conseguente perdita di informazione dell'ultimo periodo (periodo strettamente inferiore ai 2 minuti).
Nel caso della velocità, il processo è simile, mentre varia l'estrazione dei dati.
Il GPS raccoglie informazioni ogni secondo e calcola la differenza di spazio percorso in questo lasso di tempo, valore che rappresenta la velocità media in un secondo

$$v = (d_0 - d_1)\Delta t = \Delta d \tag{3.2}$$

dove d_0 è la posizione geografica iniziale, d_1 la posizione geografica finale, Δt il tempo uguale ad *1s* e Δd la differenza tra le due posizioni.
Questo valore di velocità è utilizzato per generare l'istogramma, che dopo due minuti, esattamente come nel caso delle accelerazioni,verrà inviato sotto forma di percentili.
È stato deciso di prendere una descrizione dei valori e non tutti gli elementi per il fatto che la trasmissione dei dati ha un costo che, per quanto modico, moltiplicato per un parco veicoli elevato, diventa significativo.

3.4 Considerazioni iniziali

La prima analisi fatta è stata quella per capire se, con i dati a disposizione, fosse possibile ricavare delle informazioni utili a descrivere lo stile di guida di una persona.
La valutazione è iniziata utilizzando un dispositivo nuovo, ancora da cablare in un veicolo.

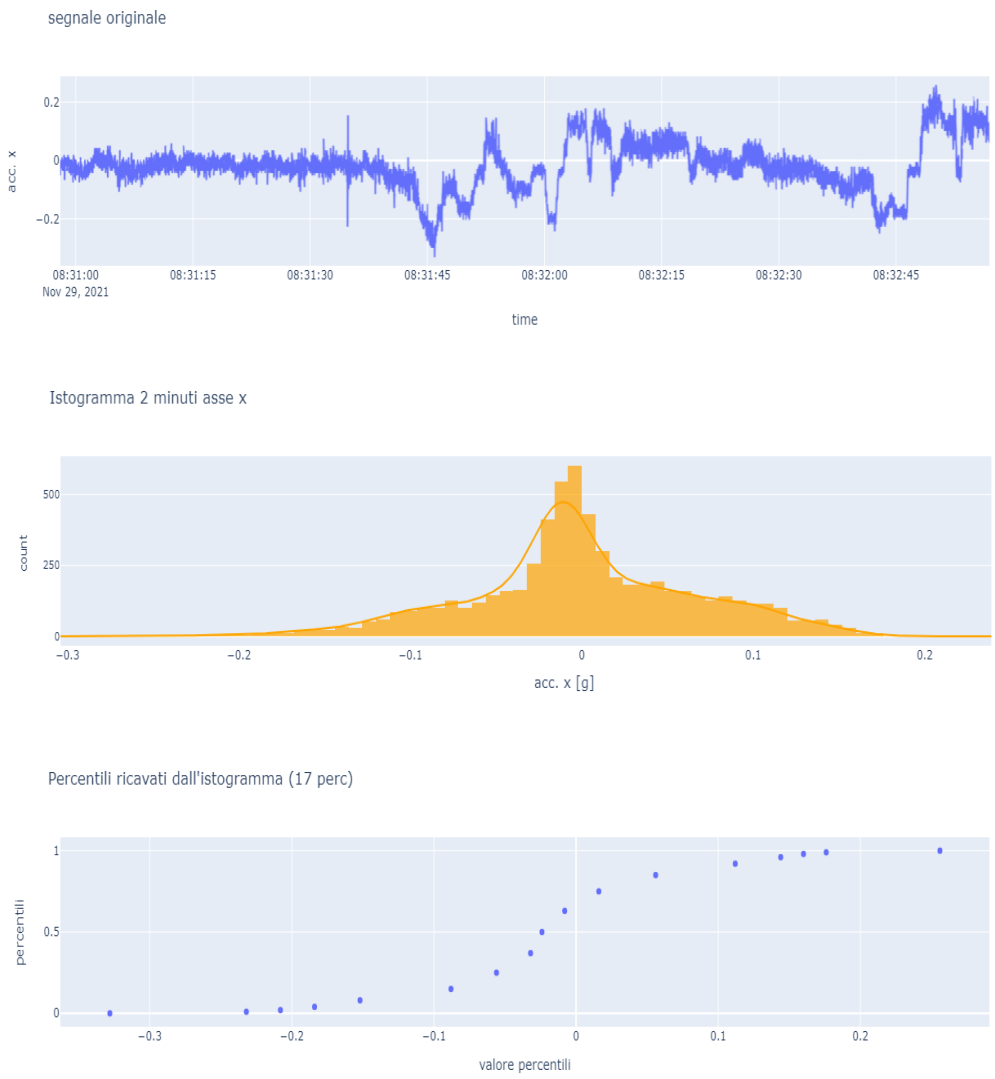


Figura 3.4: Elaborazione interna del dispositivo per invio dati

È stato utilizzato un alimentatore esterno per simulare la batteria del veicolo, in questo modo è stato possibile osservare i valori spediti dal dispositivo senza entrare fisicamente in un'auto.
L'obiettivo è stato quello di acquisire dati direttamente dall'apparecchio collegando il PC al dispositivo tramite l'attacco seriale che ha a disposizione, configurazione mostrata in Figura 3.5.
In tal modo è stato possibile osservare i dati grezzi che l'apparecchio for-

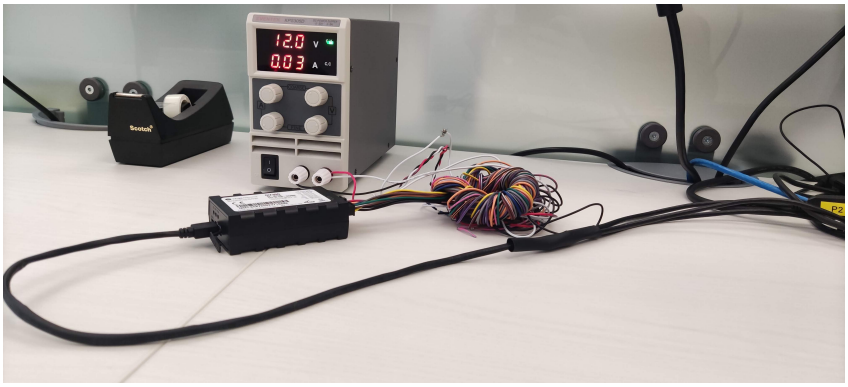


Figura 3.5: Collegamento dell'apparecchio a cavo seriale

nisce, dati che spedisce in codice ASCII.
Dopo aver scritto il programma per la decodifica, 3è stato controllato che tutte le specifiche dichiarate fossero corrette e se fosse possibile, con i dati a disposizione, riuscire a capire la dinamica del veicolo.
Perciò, utilizzando il medesimo programma sono stati acquisiti valori su un apparecchio realmente posizionato un un veicolo e, come si può vedere dal comportamento delle accelerazioni rappresentato in Figure 3.6, le due frenate rappresentate sono ben visibili e riconoscibili.

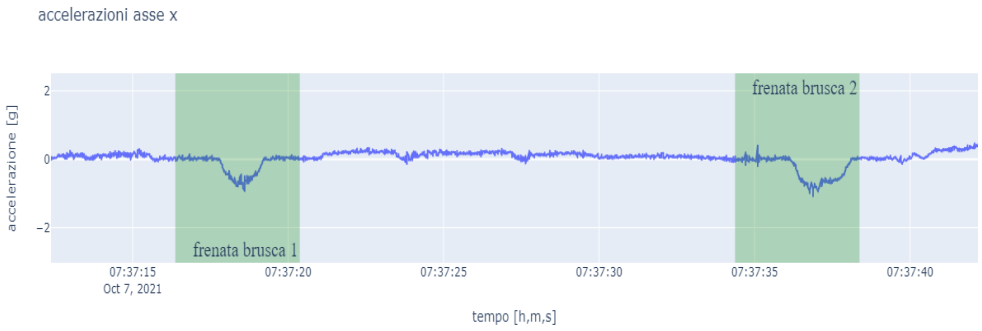


Figura 3.6: Due eventi di frenata brusca

Queste accelerazioni sono state acquisite utilizzando il pc portatile in auto mentre esegue il programma precedentemente scritto e collegato all'apparecchiatura tramite porta seriale. Gli eventi significativi (accelerate, frenate, rotonde etc.) sono stati registrati in maniera manuale e poi riportati sul grafico.

Quindi, avendo lo scopo di capire se sia possibile ricavare delle informazioni utili a descrivere la modalità di guida, è possibile concludere, dopo le valutazioni fatte sia sulla parte hardware che software, che le ci sono le condizioni per approfondire la ricerca.
Come spiegato nella Sezione 3.3 l'apparecchio invia i "percentili" come descrizione dell'istogramma rappresentante il segnale originale, questi valori, distribuiti come nella formula 3.1 rappresentano la funzione di ripartizione del grafico di partenza, cioè come le accelerazioni si distribuiscono nell'arco di tempo considerato. Questa funzione contiene tutte le informazioni necessarie per recuperare il segnale di partenza (nel modo spiegato nella Sezione 4.1.1, e così ottenere un punteggio per ogni tipo di guida.

Capitolo 4

Analisi e Fit

4.1 Stumenti e analisi

La descrizione in percentili del segnale analizzato rappresenta la funzione di ripartizione (cumulative function) che racchiude le informazioni della distribuzione di partenza, utilizzando quindi la relazione tra segnale originario e valori in nostro possesso, è possibile, attraverso una regressione non lineare, trovare il miglior fit della funzione di ripartizione e quindi la corrispettiva funzione di densità di probabilità.
Quest'ultima funzione rappresenta l'istogramma delle accelerazioni nell'arco dei due minuti considerati.
Per riuscire ad avere un fit affidabile, con poco errore, sono stati utilizzati 17 percentili, elencati nella formula 3.1 e una mistura di tre distribuzioni gaussiane, dove ad ogni singola gaussiana sono associati tre valori, quali media μ, varianza σ e peso α.
Per ottenere il risultato sono state utilizzate i seguenti strumenti:

- *curve_fit* della libreria *scipy.optimize* per generare il modello
- *Trust Region Reflective* per trovare i parametri ottimi per il modello
- *Cramer Von Mises test* utile a svolgere un'analisi comparativa tra i dati a disposizione

Nei capitoli successivi è possibile trovare le analisi a motivazione delle scelte fatte.

4.1.1 Creazione del modello

Per stimare la funzione di densità di probabilità partendo dai percentili è stata utilizzata la funzione *curve_fit* della libreria *scipy.optimize*.
Questa funzione consente di calcolare il modello migliore basandosi sul metodo non lineare dei minimi quadrati, con metodo per l'ottimizzazione dei parametri *Trust Region Reflective*, che permette di trovare una soluzione locale ottima di un sistema non lineare.
Non avendo a disposizione tutti i dati, la strada intrapresa è stata quella di utilizzare un procedimento che permettesse di tracciare un modello il più vicino possibile ai punti a disposizione, quindi un metodo che minimizzasse l'errore tra il fit e i percentili, la scelta più ovvia è ricaduta sul metodo dei minimi quadrati.
Quindi il problema da risolvere per trovare i parametri migliori è la minimizzazione della somma dei quadrati degli errori:

$$\min_{x \in S} ||E(x)||_2^2 = \min_{x \in S} \sum_i E_i^2(x) \tag{4.1}$$

dove E_i rappresenta la distanza del fit dal quantile i-esimo e S è l'insieme dei valori possibili.
Si può risolvere con varie modalità, però, sapendo che un'accelerazione e una frenata ordinaria non superano mai $\pm 2g$ di forza, e che la velocità è sempre maggiore di zero e minore di $300 km/h$, posso imporre questi limiti per velocizzare e concentrare la ricerca dei parametri ottimi in un sottoinsieme limitato.
Utilizzando questi limiti, facendo quindi variare i parametri all'interno di certe soglie, utilizzo il metodo *Trust Region Reflective* implementato all'interno del metodo *curve_fit*.
Il metodo funziona in questo modo:
Osserviamo il problema 4.1, chiamiamo *f(x)* la quantità da minimizzare e x il vettore n-dimensionale (nel nostro caso ha 9 dimensioni in quanto deve calcolare 9 valori, tre per ogni gaussiana, il peso α_i, la media μ_i e la varianza σ_i^2)

$$\sum_{i=1}^{3} \alpha_i \mathcal{N}(\mu_i, \sigma_i^2) \qquad where \tag{4.2}$$

$$\sum_i \alpha_i = 1, \quad \alpha_1 \geq 0, \quad \alpha_2 \geq 0, \quad \alpha_3 \geq 0$$

inizializzato in maniera randomica all'interno dei limiti imposti.
L'idea che sta alla base del procedimento è trovare una funzione q più semplice di f ma che ne rispecchi il comportamento in un intorno N al punto x, spostarsi all'interno di questo intorno minimizzando la funzione di costo e ripetere il procedimento fino a quando non si raggiunge una convergenza.
L'obiettivo è trovare uno step s tale che sommato al punto di partenza x risulti

$$f(x+s) < f(x). \tag{4.3}$$

Se questo non fosse verificato, il punto x rimarrebbe invariato ed N, la regione intorno ad x, viene diminuita secondo una determinata regola.
Risolvere il problema

$$\min_s \{q(s), s \in N\} \tag{4.4}$$

corrisponde, espandendo q con i primi due termini dell'approssimazione di Taylor, a trovare una soluzione a

$$\min_s \left\{ \frac{1}{2} s^T H s + s^T g \quad tale\ che \quad ||Ds||_2 \leq \Delta \right\} \tag{4.5}$$

dove g è il gradiente di f nel punto x, H è la matrice Hessiana, D è una matrice diagonale di riscalamento e Δ è uno scalare positivo.
Utilizzando questo algoritmo è quindi stato possibile ricavare la funzione di densità di probabilità, che rappresenta come la distribuzione delle accelerazioni nel tempo.
Qual'è la migliore funzione, o combinazione di funzioni, che permette dopo il fit di rappresentare al meglio l'istogramma di partenza?
La decisione è stata presa partendo con un'analisi visiva supervisionata.
Osservando i segnali originali è infatti possibile notare come i valori di accelerazione tendano a raggrupparsi su uno o più punti, degli esempi di distribuzione sono visibili in Figura 4.1.
Alcuni istogrammi sarebbero facilmente rappresentabili da una semplice gaussiana, altri per essere rappresentati con basso errore, necessitano di una mistura composta da più distribuzioni.
La scelta è ricaduta su un insieme di opzioni:

- Una mistura di 2 skew normal
- Una mistura di 3 skew normal

Figura 4.1: Istogrammi con diversi comportamenti

- Una mistura di 2 normali
- Una mistura di 3 normali
- Una mistura di 4 normali

La scelta si basa su un'analisi che mette in relazione l'errore tra i vari modelli, utilizzando la differenza tra la CDF di partenza e le CDF ottenute dai modelli ed anche il tempo di calcolo che ogni algoritmo ci impiega, in quanto, volendo applicare l'algoritmo su un gran numero di veicoli, sono stati cercati i parametri migliori anche per evitare di avere tempi di computazione elevati.
I risultati sono mostrati in Figura 4.2, divisi per una maggiore leggibilità, nelle prime due figure sono rappresentati gli errori e il tempo impiegato utilizzando due e tre skew normal, nelle seconde due l'errore e il tempo con due, tre e quattro distribuzioni normali.
Da una rapida occhiata si nota immediatamente come le distribuzioni *skew normal* impieghino un tempo molto alto per completare un fit ed è possibile riscontrare come nel caso di tre skew normal il risultato finale abbia un'errore non trascurabile, invece nel caso 2 skew normal, come visibile in Figura 4.3, se messa a confronto con la mistura di 2 distribuzioni gaussiane ha un'errore confrontabile, ma con un tempo di esecuzione molto superiore.
La scelta è stata presa in direzione delle distribuzioni gaussiane.
Concentrandosi esclusivamente sull'errore si osservi come aumentando il numero di gaussiane l'errore diminuisca, allo stesso tempo il tempo impiegato dall'algoritmo per arrivare a compimento incrementa vistosamente.
La decisione è stata ponderata tendendo conto dell'analisi visiva fatta precedentemente, in quanto si sono potuti notare figure con tre picchi, e perciò è stata esclusa la soluzione con due gaussiane, in quanto al massimo può rappresentare figure con sole due gobbe. La scelta si è concentrata tra la soluzione con tre o con quattro gaussiane.
Per poter scegliere tra queste due ipotesi, è stato applicato un test statistico, il *Cramer Von Mises test*[2], un criterio utilizzato per giudicare la bontà di adattamento di una funzione di distribuzione cumulativa F^* rispetto a una data funzione di distribuzione empirica F_n, ottenuta analizzando i dati acquisiti direttamente dal dispositivo, e si ottiene la Figura 4.4, dove in basso è rappresentato il *p value*, il valore che esprime la significatività del

Figura 4.2: Errore di fit rispetto ai quantili e tempo impiegato, id-campione preso all'interno di una guida con valori campionati ad alta frequenza, tra 2id e 2(id+1) minuti

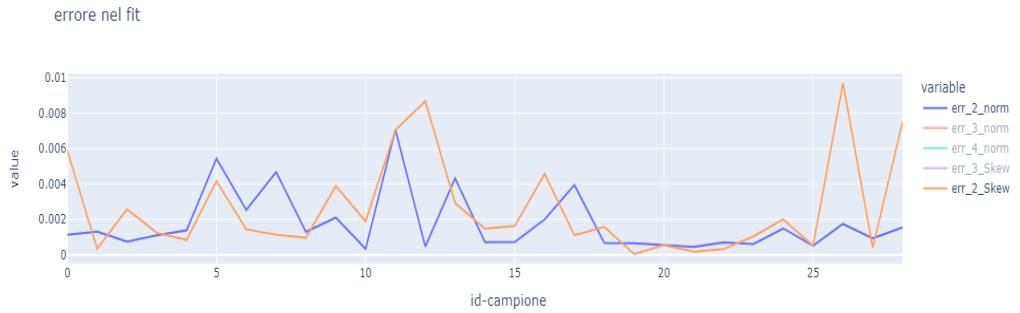


Figura 4.3: Errore di fit confrontando una mistura di due distribuzioni normali e due skew-normal, id-campione preso all'interno di una guida con valori campionati ad alta frequenza, tra 2id e 2(id+1) minuti

test, e il valore *statistic* che esprime l'errore rispetto a questa distribuzione.
Il test implementato dalla libreria di *scipy* funziona in questo modo:
Data l'ipotesi iniziale H_0: distribuzione empirica = distribuzione fittata, e l'ipotesi alternativa H_a: distribuzione empirica $\neq$ distribuzione fittata, verifico la bontà dell'affermazione H_0 calcolando il *p_value* che indica la significatività dell'affermazione cioè quanta probabilità c'è di ottenere l'errore ricavato o superiore assumendo H_0 vera.
Considerando $F(x)$ la distribuzione statistica empirica, e $x_1, x_2, ..., x_n$ i valori osservati ordinati per valore, la statistica T si calcola con questo criterio:

$$T = \frac{1}{12n} + \sum_{i=1}^{n} \left[\frac{2i-1}{2n} - F(x_i) \right]^2 . \tag{4.6}$$

In statistica è solitamente considerato 0.05 il valore del *p-value* che permette di avere un livello di significatività sufficiente per assertire che se ha un livello di significatività inferiore a 0.05 allora la supposizione H_a è verificata, cioè le due distribuzioni non porvengono dalla stessa fonte di dati, se è superiore, non è possibile scartare l'ipotesi H_0.
Nel nostro caso il p-value, sia nel caso di tre che quattro gaussiane, ha un valore sempre superiore a 0.151, quindi molto superiore a 0.05 e l'errore (statistic) tra le due distribuzioni è paragonabile.
Questa successione di affermazioni porta scegliere una mistura composta da tre gaussiane, infatti ha risultati molto simili alla distribuzione con quattro gaussiane, ma impiega un minor tempo per l'elaborazione.

4.2 Analisi Percentili

Per capire quale fosse quantità di percentili corretta per poter avere una buona descrizione dell'andamento delle accelerazioni nel tempo è stata fatta un'analisi, quest'analisi serve a trovare il migliore trade-off tra quantili e minuti in modo da descrivere i dati in ingresso e contenere il costo di invio limitando il flusso di dati.
Un valore di invio dei dati accettabile in riferimento dei costi da sostenere è di circa 9-11 dati al minuto.
È stata valutata l'opzione di avviare una comunicazione ogni 1, 2 o 3 minuti (diminuendo la frequenza di invio dei dati, si può inviare una quantità di

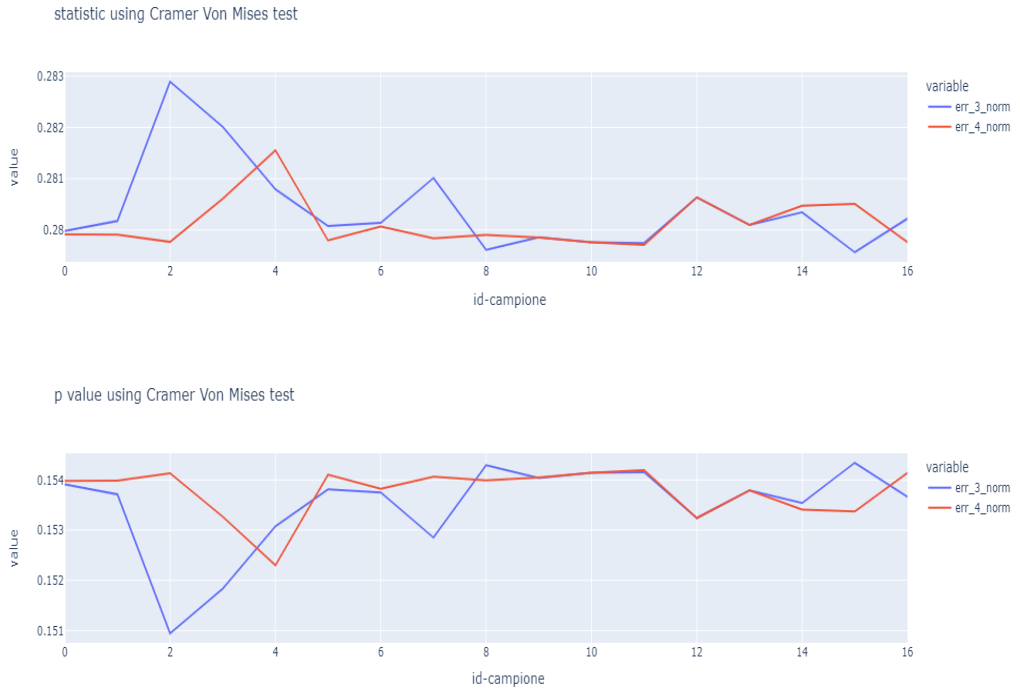


Figura 4.4: Cramer Von Mises test per analisi con mistura di 3 o 4 distribuzioni normali, id-campione preso all'interno di una guida con valori campionati ad alta frequenza, tra 2id e 2(id+1) minuti

valori maggiore).
è stata bocciata l'opzione di 1 minuto in quanto implica solamente 9 percentili. Infatti, utilizzando 3 distribuzioni normali in un solo minuto per fittare il modello, 3 distribuzioni * 3 variabili ciascuno = 9 variabili da calcolare con 9 percentili, l'errore di fit con questi quantili risulta essere rilevante.
Scegliendo tra due e tre minuti, l'analisi è stata fatta con 9, 11, 13, 15, 17, 19, 21, 23, 25 percentili, e per poter confrontare i risultati è stata scelta come metrica di errore il valore ricavato dal *Cramer Von Mises test*, test spiegato nella Sezione 4.1.1.
Per ogni gruppo di percentili sono stati ripetuti per 15 volte sia il fit che il test su un dataset di partenza di 30 minuti, successivamente è stata calcolata la media e la varianza di questo punteggio.
I risultati ricavati dall'analisi svolta con due e tre minuti sono rappresentati in Figura 4.5. La linea vedre superiore e la rossa inferiore indicano la deviazione standard dell'errore rispetto alla media, visibile in blu.
Osservando questo grafico è subito possibile notare come nel caso di tre

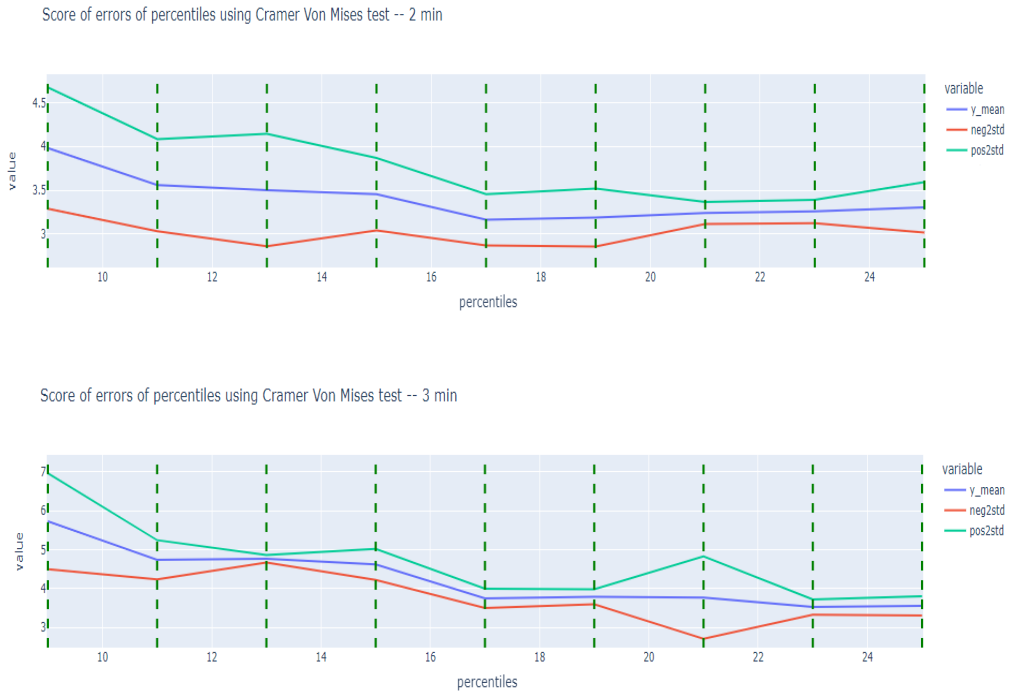


Figura 4.5: Analisi numero di percentili utilizzando Cramer Von Mises test

minuti l'errore calcolato sulla distribuzione fittata sia maggiore rispetto a quello dei due minuti, questo spiegabile dal fatto che una maggiore mole di dati possa portare ad una complicazione della funzione di arrivo, inoltre acquisire dati ogni 3 minuti potrebbe diventare un poblema per quanto riguarda la velocità, infatti il valore del limite della strada viene acquisito dal sistema solo allo scadere del periodo, e se il periodo risulta essere troppo lungo c'è il rischio che il veicolo attraversi zone con limiti diversi.
Osservando quindi il grafico relativo a due minuti, è possibile notare come dopo i 15 percentili l'errore rimanga stabile in media, quindi, la decisione è stata quella di utilizzare 17 percentili per la descrizione del segnale.
Altri parametri dell'apparecchio impostabili sono il valore massimo e il valore minimo che il dispositivo è in grado di acquisire, ed il binning, ovvero la divisione del dominio della variabile sulle ascisse dell'istogramma in intervalli, fondamentali per la costruzione dello stesso.
Sono stati impostati in modo che l'occorrenza da associare al singolo bin corrisponda esattamente con il valore fornito dallo strumento in base alla sua risuluzione.
La risoluzione del congegno, rilevata dall'acquisizione tramite seriale dei dati dell'accelerometro è di $0.008g$, pertanto il bound superiore è stato fissato a $1.6g$, quello inferiore a $-1.6g$, valori più che sufficenti per coprire l'intero spettro di accelerazioni anche di una guida più sportiva, come mostrato in Figura 4.6, e il $numero\ di\ bin = 400$, ottenendo così

$$risoluzione = \frac{upper_{bound} - lower_{bound}}{n_{bin}} = \frac{1.6 - (-1.6)}{400} = 0.008g \qquad (4.7)$$

dunque l'errore massimo calcolato sui dati corrisponde alla risoluzione del dispositivo.

Per verificare che le modifiche fatte fossero corrette sono state eseguite delle verifiche confrontando i dati in ingresso in seriale con la distribuzione risultante dopo il fit.
Oltre all'accelerazione sono stati analizzati anche i percentili relativi alla velocità, affrontando il problema nella stessa modalità, cambiando solamente i bounds entro i quali trovare i valori da assegnare ai parametri, in modo che fossero coerenti con i valori reali di velocità.

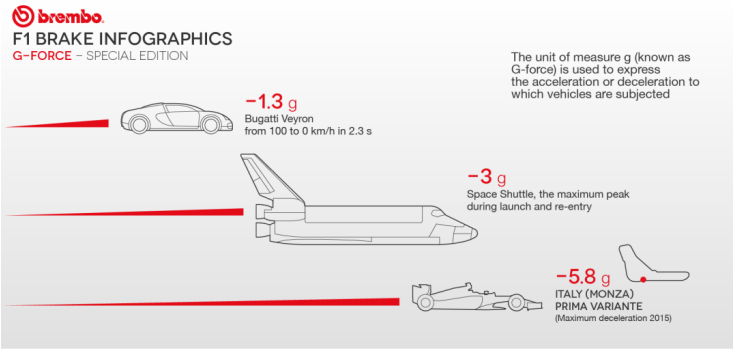


Figura 4.6: Accelerazione g su vari veicoli

4.3 Problemi riscontrati durante il fit

Durante la fase di fit si sono presentati vari problemi, dovuti sia dalla conformità della funzione usata che ai limiti del processo di fit.

4.3.1 Bounds non limitati

Cercando i parametri ottimi senza restrizioni, in tutto $\mathbb{R}^n$, dove n rappresenta il numero di parametri da calcolare, l'algoritmo rischia di andare in errore e segnalare *optimal parameters not found* perchè impiega troppo tempo per trovare i valori corretti.
Impostando dei bounds invece, visto che sappiamo che l'istogramma dei valori non ci fornisce valori superiori a $|1.6|g$ forniamo all'algoritmo un range di ricerca limitato. Il che permette all'algoritmo di concentrare le ricerche in un insieme ristretto, facilitando così il raggiungimento dell'obiettivo.

4.3.2 Inizializzazione non randomica

La funzione *curve_fit* per il calcolo dei valori ottimi locali viene inizializzata con una certa condizione iniziale, però può succedere, soprattutto se nel caso di pochi valori(percentili), che debba calcolare tanti parametri e che fallisca nella ricerca, anche con il miglioramento sopracitato, producendo l'errore *optimal parameters not found*, perciò necessita far ricominciare il calcolo. Tuttavia la funzione utilizzerebbe gli stessi parametri di prima per decidere la condizione iniziale, ed utilizzando la stessa procedura il risultato sarebbe

il medesimo.
La soluzione è stata quella di impostare dei valori randomici alla partenza, in questo modo al tentativo successivo, partendo da valori diversi, ha una probabilità maggiore di portare a termine con successo il fit, trovando un modello adeguato ai dati in input.

4.3.3 Errore CDF

La CDF per definizione dovrebbe partire dal valore 0 ed arrivare a 1, però può essere che fittando, l'algoritmo fitti molto bene sui valori centrali, e non riesca ad arrivare esattamente sul valore 1 in corrispondenza al percentile Max e nemmeno esattamente a 0 in corrispondenza del min, che per definizione hanno valore unitario e pari a zero.
Questo porta ad un errore rispetto alla distribuzione reale perchè si rispecchia sulla PDF con un valore molto alto anche dopo il massimo impostato. Inoltre essendo che per il fit sono state usate delle gaussiane, queste per definizione possiedono delle code infinite, che fanno discostare il fit dalla distribuzione reale.
È possibile risolvere qusto problema normalizzando la PDF, che andremo ad indicare con f in questo modo:

$$f_{X|min\leq X\leq Max}(x) = \begin{cases} 0 & se \quad x < minimo \\ 0 & se \quad x > massimo \\ \dfrac{f_X(x)}{P(min \leq X \leq Max)} = \dfrac{f_X(x)}{Area} & altrove \end{cases} \tag{4.8}$$

e la CDF indicata con F come segue,
sapendo che la CDF è definita

$$F_X(x) = \int_{-\infty}^{x} f_X(t)dt \tag{4.9}$$

rispetto alla PDF, allora la CDF considerata per x compresa tra min e Max

$$F_{X|min\leq X\leq Max}(x) = \int_{-\infty}^{x} f_{X|min\leq X\leq Max}(t)dt = \tag{4.10}$$

$$\int_{min}^{x} \frac{f_X(t)}{Area}dt = \frac{1}{Area}\int_{min}^{x} f_X(t)dt = \frac{F_X(x) - F_X(min)}{Area}$$

quindi

$$F_{X|min\leq X\leq Max}(x) = \begin{cases} 0 & x < minimo \\ 1 & x > massimo \\ \dfrac{F_X(x) - F_X(min)}{Area} & altrove \end{cases} \tag{4.11}$$

In questo modo le proprietà della CDF e della PDF rimangono invariate e rappresentano una situazione più plausibile.

4.3.4 Valori percentili uguali

Esaminando i quantili che verranno utilizzati per il calcolo del fit, è possibile notare che alcune volte ce ne siano due o più sovrapposti tra loro. La sovrapposizione dei quantili è causata dalla risoluzione dell'apparecchiatura, infatti, se su un bin dell'istogramma di partenza risultano accumulati molti valori rispetto ai bin precedenti, può essere che quel bin rappresenti il superamento di più valori percentuali, per esempio nel caso di un veicolo fermo al passaggio a livello, dove la larghezza dell'istogramma è dovuta alle vibrazioni dell'auto, uno stesso bin può indicare il superamento dello 0.5, 1 e 2%.
Osservando nel dettaglio il fit generato per ogni percentile, nel caso in cui ci siano almeno due percentili con lo stesso valore, l'algoritmo fa molta fatica a trovare la distribuzione corretta, in quanto cercherà di passare per il punto che minimizza l'errore tra i due(o più) percentili uguali facendolo pesare il doppio(o più), quindi cercando di creare il fit con il vincolo che passi il più vicino possibile a quel punto, anche scombinando gli altri vicino. Questo processo è errato, in quanto due percentili uguali significa che per un periodo di tempo relativamente lungo si è comportato in quel modo.
Quest'errore è dipendente dal limite della funzione che calcola l'errore sui percentili, ma è facilmente risolvibile traslando i percentili, anche di poco, l'importante è che non siano sulla stessa identica riga in modo da poter calcolare separatamente gli errori su ogni percentile.
In Figura 4.7 possiamo osservare i primi due grafici relativi alla CDF e alla PDF con due percentili uguali senza la "traslazione" di questi ultimi, è possibile notare come la PDF risultante risulti essere una distribuzione *skew normal*, poco diversa da una semplice gaussiana(distribuzione dalla quale

sono partito prima di sovrapporre il quinto percentile con il sesto),mentre le due immagini sottostanti rappresentano la distribuzione calcolata dopo aver traslato i percentili, ottenendo così due calcoli dell'errore separati. Quest'ultima distribuzione rappresenta con più fedeltà la distribuzione reale.

Quindi i percentili vengono "sistemati", cioè spostati un po', sempre all'interno dello stesso bin, come mostrato in Figura 4.8, altrimenti c'è il rischio di assegnare valori errati, l'algoritmo fa in modo che ricoprano l'intera area del bin di partenza stando il più distanziati possibile tra loro.

Figura 4.7: Fit con sovrapposizione di bin, i primi due grafici rappresentano la situazione senza traslazione, i secondi due con lo spostamento

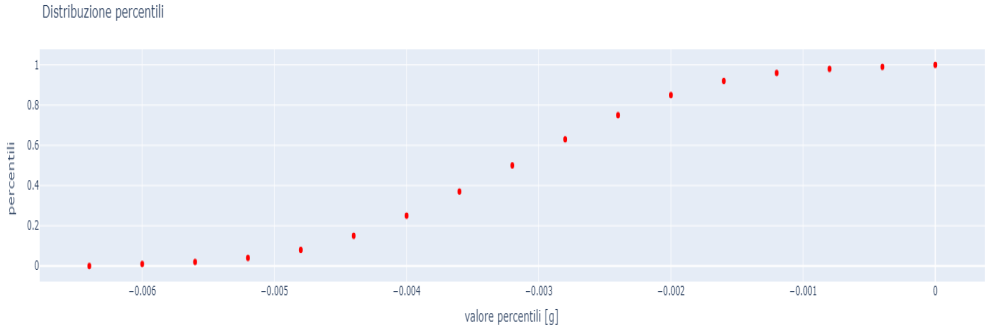


Figura 4.8: Diciasette percentili tutti con lo stesso valore, vengono traslati in modo da occupare tutto il bin(di valore 0.008) senza andare ad occupare il precedente

4.3.5 Spark

Come evidenziato nella Sezione 4.1.1 il tempo per eseguire l'algoritmo risulta essere relativamente lungo (arriva anche a mezzo secondo per singolo fit). Per questo motivo, nel momento di analizzare grandi moli di dati, sono state cercate altre vie rispetto all'esecuzione in serie di ogni elemento. Infatti l'assegnazione del punteggio, spiegato nella prossima Sezione 5.1, viene effettuato separatamente sia per la velocità che per ogni asse dell'accelerometro, questo per ogni raggruppamento di due minuti.
Perciò è facile calcolare il tempo impiegato per una settimana di dati, supponendo che tutta la flotta sia in movimento 4 ore al giorno e che sia composta da 100 veicoli attivi:

$$
\begin{aligned}
&tempo_{calcolo\_tot} = \\
&(Acc_{axis} + Vel_{axis}) N_{groups} * Tempo_{fit} * gg * N_{veicoli} = \\
&4 * 120 * 0.4 * 5 * 100 = 96000\ secondi \rightarrow \\
&\frac{96000}{60} = 1600\ minuti \ \rightarrow \ \frac{1600}{60} = \mathbf{26.7\ ore}
\end{aligned}
\tag{4.12}
$$

dove $Acc_{axis} = 3$ sono gli assi dell'accelerometro, $Vel_{axis} = 1$ è l'unica direzione in cui consideriamo la velocità, $N_{groups} = 120$ sono il # di gruppi di due minuti in 6 ore, $Tempo_{fit} = 0.4s$ è il tempo medio impiegato per un fit, $gg = 5$ è il numero di giorni lavorativi in una settimana e $N_{veicoli} = 100$ rappresentano il numero di veicoli nella flotta aziendale.
Il risultato è computazionalmente complicato, perciò si è presentata la necessità di utilizzare un'architettura più efficiente per raggiungere l'obiettivo. La soluzione è stata quella di utilizzare uno schema parallelo, facendo così eseguire tanti insiemi di dati simultaneamente.
La struttura utilizzata è quella di *Apache Spark*, un framework di elaborazione parallela open source che supporta l'elaborazione in memoria per migliorare le prestazioni delle applicazioni che analizzano i big data.
Utilizzando questo sistema di elaborazione distribuita è stato possibile elaborare una settimana di dati in soli 46 minuti rispetto alle 26.7 ore della Formula 4.12 utilizzando un semplice processo in serie.

4.4 Portable real-time data recorder

Volendo acquisire dati da seriale, avendo a disposizione tutta la dinamica nel periodo di guida, è possibile, come accennato nella Sezione 3.4, collegarsi al pc.
Però acquisire dati utilizzando il pc, anche se portatile, quando il dispositivo è in auto risulta difficoltoso, molto scomodo e non di facile utilizzo nel caso in cui il guidatore è anche colui che aziona il programma.
Perciò l'idea è stata quella di implementare un sistema hardware di facile impiego e che sostituisse il pc, un dispositivo che compisse molte operazioni in maniera automatica, un registratore portatile di dati in real-time.

4.5 Creazione dispositivo

Per questo registratore portatile di dati in real-time è stato utilizzato un raspberry pi[1], dispositivo portatile in grado di tenere al suo interno il sistema operativo Raspbian, sistema operativo ufficiale che utilizza Linux come kernel. Collegandolo ad uno schermo è stato possibile scrivere un programma che si azionasse ad ogni accensione, leggesse dati in input, sia dati da seriale che input generici come i pulsanti,impostasse uscite in output, come led e display e potesse scrivesse un file salvandolo in memoria, in modo da recuperare successivamente le informazioni accumulate per svolgere un'analisi.
Il dispositivo finale è rappresentato in Figura 4.9, ogni numero rappresentato sul display 7 segmenti rappresenta un differente stato, che verranno spiegati nella sottosezione 4.5.1.

In questa prima immagine (Figura 4.9) è possibile riconoscere:

- un display 7 segmenti che identifica lo stato in cui il programma si trova, un diverso stato porta l'algoritmo a comportarsi diversamente.
- 6 pulsanti di cui:
  - i tre sulla prima riga servono a far partire il programma, ricominciare il programma cominciando un nuovo file e mandare l'algorimo in uno stato di stop permanete

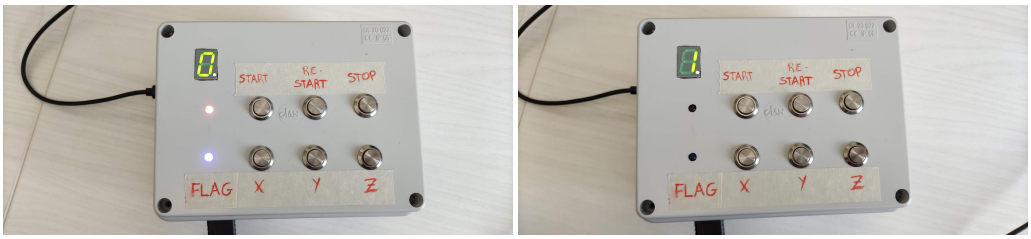


Figura 4.9: Configurazione finale dell'apparecchio

– i tre sulla seconda riga sono *flag*, utili a segnare eventi significativi durante la guida.
In questo modo, quando a posteriori si andrà a visionare il risultato prodotto dalla guida effettuata, se un flag(uno adibito ad ogni asse) è stato premuto, sulla linea temporale sarà evidenziato l'evento.

Il registratore portatile di dati in real-time al suo interno, come mostrato in Figura 4.10, oltre al raspberry pi è composto anche da una scheda millefori dove sono stati effettuati i collegamenti necessari per il corretto funzionamento della componentistica ed effettuate le stagnature delle resistenze per i pulsanti(utilizzando un collegamento pull-up), per i diodi e per il display, e i necessari per il funzionamento e da un RTC(realt time clock), utile in quanto il raspberry pi non ha una batteria interna, perciò nel momento di accensione, se non riesce a collegarsi ad una rete internet, non risce a recuperare il giorno e l'ora corretti.

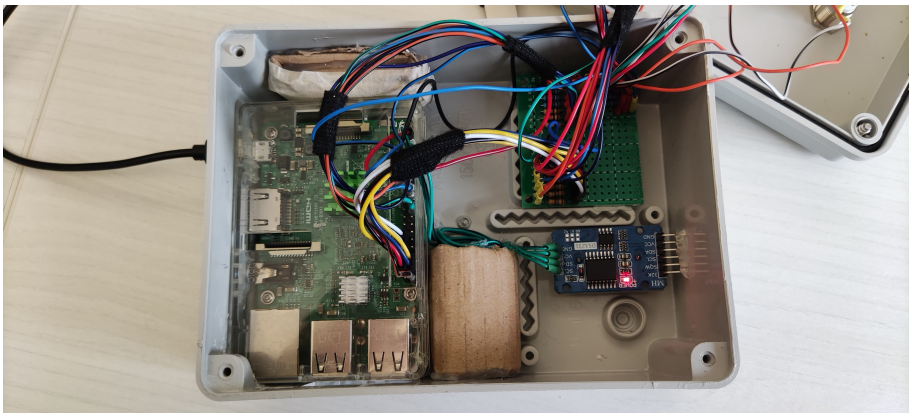


Figura 4.10: Cablaggio interno dell'apparecchio

Dove i componenti della scheda millefori sono stati saldati seguendo questo schema elettrico: Figura 4.11
L'algoritmo per funzionare è suddiviso in vari stati, dove si comporterà in maniere differenti, queste modalità sono spiegati nella sottosezione 4.5.1.

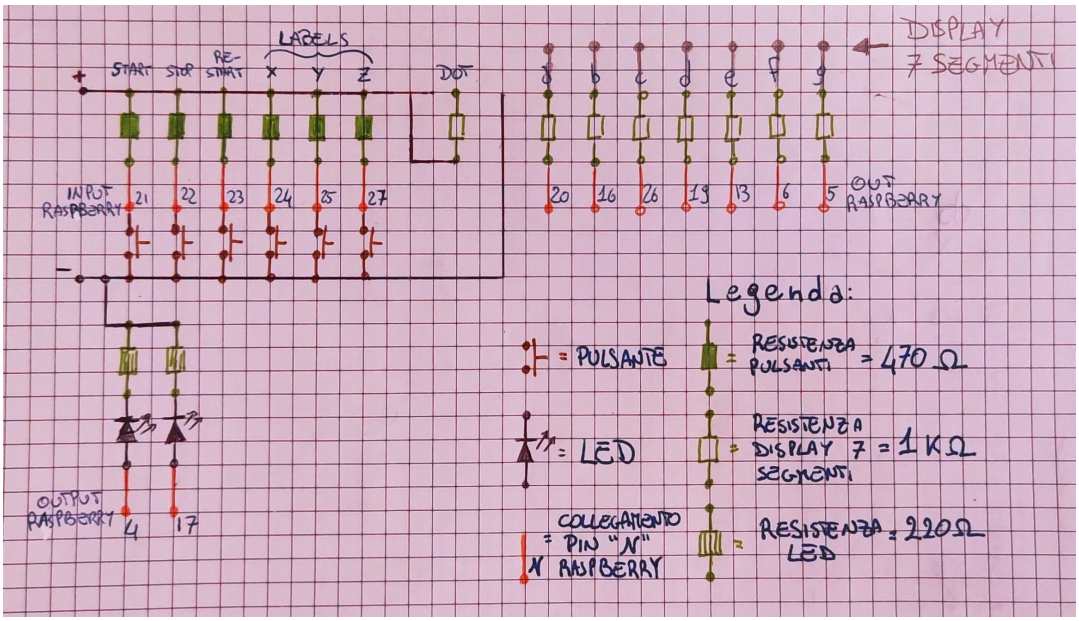


Figura 4.11: Schema elettrico

4.5.1 Comportamento algoritmo dispositivo

La Macchina a stati che descrive in maniera formale il comportamento del sistema che va dall'accensione del dispositivo al suo spegnimento è rappresentata in Figura 4.12.

Nello stato 0 avviene l'accensione del ruspberry pi con la sua routine di

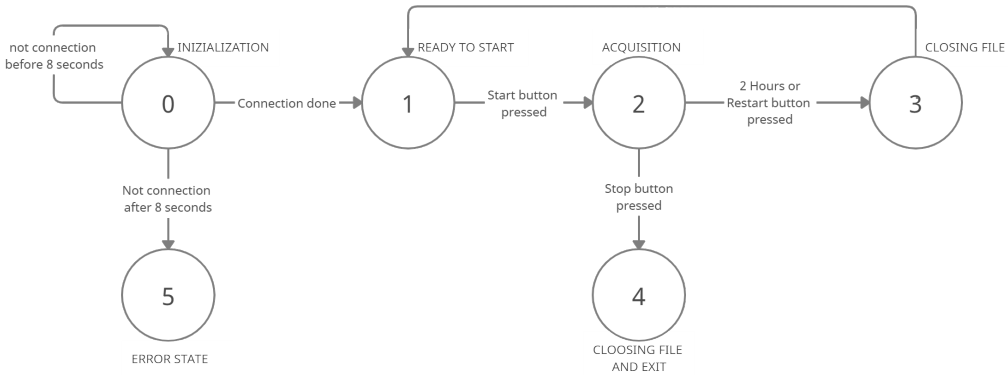


Figura 4.12: Diagramma a stati

inizializzazione, quando è pronto, dopo aver acquisito l'ora corretta dal modulo RTC ed essere predisposto a leggere un'ingresso seriale, tenta per 8 secondi di connettersi ad un ingresso tentando ogni volta una porta differente.

Se non ci riesce va nello stato di errore 5 dove, per uscirne, bisogna ricominciare il processo spegnendo ed accendendo il ruspberry pi.

Se la connessione va a buon fine, il display mostrerà per 2 secondi la porta su cui ha rilevato l'ingresso seriale, per poi mostrare nel display lo stato 1, stato utile ad avvisare l'utente che il processo per l'apertura di un canale di comunicazione è andato a buon fine(non basata che il cavo sia collegato, bisogna anche aprire la comunicazione) ed è tutto pronto per l'acquisizione.

Per uscire dallo stato 1 e andare allo stato 2 è necessario premere il pulsante di *Start*, che darà il via all'acquisizione, il primo dato che andrà ad acquisire creerà un nuovo file che avrà come titolo il giorno, l'ora, il minuto e il secondo in cui comincia ad acquisire. Tutti i campioni successivi andranno ad incrementare questo file, file che si salverà appena la scrittura del campione acquisito sarà ultimata con successo, in questo modo si evita di perdere dati nel caso di interruzione forzata dell'alimentazione.

Sul file verranno salvati l'accelerazione sull'asse x, y, z e un'eventuale notifica di labeling, cioè, se l'utente vuole mettere un tag di qualcosa di significativo potrà farlo premendo uno dei tre pulsanti *Flag*, uno per ogni asse, e in un momento successivo, andando ad analizzare i dati, potrà ritrovare l'istante di interesse.
Quando si vuole terminare l'acquisizione, ci sono due opzioni, due stati disponibili, lo stato 3 e lo stato 4.
Lo stato 3 si raggiunge premendo il pulsante di *Restart*, in quanto serve a chiudere il file che si sta scrivendo e far andare il processo sullo stato 1, *Ready to start*, pronto per una nuova acquisizione.
Lo stato 4 si innesca con il pulsante *Stop*, permette sempre la chiusura del file, ma anche annuncia la terminazione della comunicazione seriale (è buona norma chiudere le porte di comunicazione seriale al termine dell'utilizzo, in modo da evitare problemi di bufferizzazione all'avvio successivo).
Raggiunto questo stato si può procedere alla rimozione dell'alimentazione al raspberry pi.

4.5.2 Modifiche Portable real-time data recorder

Nella produzione di questo congegno ci sono state alcune difficoltà significative del tipo:

- Staccando l'alimentazione al raspberry pi la porta di comunicazione seriale non si chiude immediatamente, rimanendo così aperta per qualche secondo e continuando ad bufferizzare dati che invierà all'accensione successiva. Questo problema è stato risolto facendo un reset del buffer seriale ad ogni avvio di comunicazione.

- Utilizzando un sistema operativo basato su kernel linux, le varie porte a disposizione per inserire l'attacco seriale cambiano nome ad ogni accensione, rimanendo però invariato lingresso fisico. Per risolvere questo problema è stato implementato un ciclo che tenta di collegarsi alla porta corretta tentando vari nomi diversi, tenterà per 15 secondi prima di entrare nello stato di errore, oppure appena riuscirà a collegarsi andrà

allo stato di *ready*.

- Avendo implementato l'algoritmo in modo che scriva i dati acquisiti solo alla fine dell'acquisizione, se per errore l'alimentazione viene meno in un punto diverso dallo stato finale, tutti i dati relativi a quel viaggio vengono eliminati. È stato possibile correggere ques'evenienza scrivendo il file ad ogni acquisizione, senza aspettare la fine.

- Il raspberry, non avendo una batteria interna per tenere alimentato il bios, ha bisogno di collegarsi ad internet per acquisire la data, altrimenti segna un orario errato e il tempo proseguirà partendo da quella data errata. Nel caso in cui il dispositivo abbia accesso ad una rete, è possibile notare un salto temporale da una data/ora errata alla data/ora attuale. Risolto integrando un RTC(Real Time Clock), rappresentato in Figura 4.13, con una batteria esterna e disabilitato il "fake clock" interno del raspberry pi.

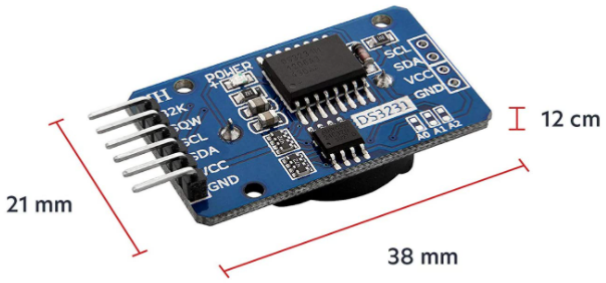


Figura 4.13: Real Time Clock

- Un ultimo problema più pratico è stato quello del dispositivo che si avvia ma dopo una decina di secondi si riavvia. Il problema sta nel cavo di alimentazione troppo sottile, che non fornisce abbastanza corrente per far funzionare tutti i singoli componenti a pieno regime, infatti il riavvio si verica nel momento in cui, dopo i vari controlli interni, deve

avviarsi il sistema operativo. Risolto cambiando il cavo.

Capitolo 5

Analisi stili di guida

5.1 Calcolo punteggio

Successivamente alla creazione del modello, è necessario decidere una modalità per fornire una metrica ed interpretare questo insieme di dati. È possibile definire due metriche diverse, uno per le accelerazioni e uno per le velocità.

- **Valutazione accelerazione:** dato il modello calcolato come spiegato nel Capitolo 4.1.1 la metrica più idonea a calcolare un punteggio è quella che rispondeva alla domanda, *quando il soggetto accelera, quanto accelera?*.
  Perciò il punteggio è determinato dal valore medio tenuto fuori da una certa soglia

$$\begin{aligned} &punteggio_{acc\_pos} = E[X|X > soglia_{pos}] = \\ &\int_{soglia_{pos}}^{Max} x f_{X|X>soglia_{pos}}(x)dx = \\ &\frac{1}{P(X > soglia_{pos})} \int_{soglia_{pos}}^{Max} x f(x)dx \end{aligned} \tag{5.1}$$

  nel caso di accelerazioni positive, nel caso di negative invece

$$\begin{aligned} &punteggio_{acc\_neg} = E[X|X < soglia_{neg}] = \\ &\int_{min}^{soglia_{neg}} x f_{X|X<soglia_{neg}}(x)dx = \\ &\frac{1}{P(X < soglia_{neg})} \int_{min}^{soglia_{neg}} x f(x)dx \end{aligned} \tag{5.2}$$

dove la $soglia_{pos}$ = *0.1g* e la $soglia_{neg}$ = *-0.1g*
Dopo questo calcolo viene calcolato il $punteggio_{acc\_asse}$ come la media pesata di ogni punteggio calcolato ogni due minuti, dove il peso è rappresentato dal tempo trascorso stando fuori dalla soglia. Il risultato finale $punteggio_{acc\_tot}$ è ottenuto dalla somma in valore assoluto dei punteggi sugli asse x e y, dove sull'asse y la domanda a cui risponde è, quanta forza viene impressa in fase di sterzata? *quando curva, quanto bruscamente curva?*

- **Valutazione velocità:** oltre ai dati statistici che il dispositivo invia, l'azienda fornisce al cliente anche un servizio di geolocalizzazione e, appoggiandosi alla piattaforma TomTom, è possibile ricavare il limite delle strade percorse nel momento in cui invia la posizione del veicolo, quindi l'algoritmo utilizza come soglia il limite di velocità di quel gruppo di minuti su quella strada, e calcola la media del valore di quando il guidatore è sopra i limiti.
  Una volta calcolato questo "superamento medio" per ogni gruppo di minuti calcolo un punteggio complessivo in modo ad dare più peso ad un superamento sui limiti più piccoli e normalizzando per il tempo cosicchè il punteggio non dipenda da quanto tempo una persona abbia guidato ma solo da come, in quel lasso di tempo, si sia comportato al volante.

$$punteggio_{vel} = \frac{1}{\sum_i(\Delta t_i)}\Big[\sum_i \Big(\frac{sup.medio}{L_i}\Big)^{\alpha}(\Delta t_i)\Big] \tag{5.3}$$

  dove Δt rappresenta la quantità di tempo trascorso fuori dal limite *i-esimo* (rappresentato con L_i), $\alpha > 0$ è un parametro arbitrario che si può utilizzare per penalizzare maggiormente una guida ad alta velocità con un limite basso e il superamento medio è:

$$\begin{aligned} &sup.medio = E[X|X > lim.velocità] = \\ &\int_{lim.velocità}^{Max} x f_{X|X>lim.velocità}(x)dx = \\ &\frac{1}{P(X > lim.velocità)} \int_{lim.velocità}^{Max} x f(x)dx \end{aligned} \tag{5.4}$$

- **Valutazione complessiva:** è dato dalla somma del punteggio dell'accelerazione e da quello della velocità con un termine β di riscalamento

per rendere questi valori confrontabili, in modo che abbiano entrambi la stessa significatività.

$$punteggio_{totale} = punteggio_{acc\_tot} + \beta punteggio_{vel} \tag{5.5}$$

5.1.1 Problema limiti

In Figura 5.1 è possibile trovare un esempio pratico di un problema incontrato in questa fase, se l'auto percorre per due minuti un limite di 50km/h, per poi entrare in un limite di 30 allo scadere dei due minuti, il limite che viene rilevato, 30 km/h verrebbe utilizzato per tutti i due minuti precedenti, e ciò risulterebbe sbagliato, appare, erroneamente, che per i 2 minuti precedenti ha sforato i limiti in maniera molto elevata.
Per evitare questa evenienza, nel caso di un passaggio da un limite superiore ad uno inferiore, il limite precedente viene esteso anche ai due minuti successivi, in questo modo non possono esserci errori, al massimo risulterà che il guidatore avrà avuto una guida migliore, ma mai peggiore. Questo

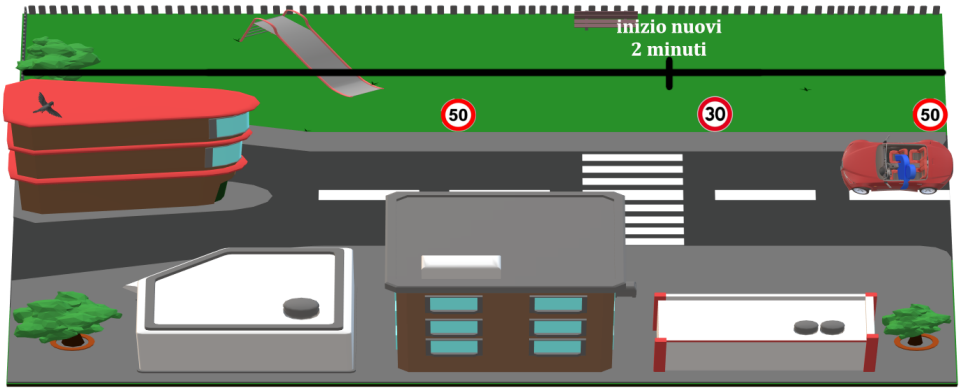


Figura 5.1: spiegazione pratica di uno specifico evento

serve ad evitare di assegnare un punteggio che andrebbe a penalizzare il guidatore non rispondente al vero.
Questa cosa non si verifica nel caso in cui, se l'auto esce da un limite di 30 e entra sui 50, ma la posizione la manda ancora quando è sui 50, in quanto l'algoritmo utilizza il limite confrontandolo con i due minuti precedenti.
Il caso peggiore è quello che non è possibile riconoscere, cioè quando partendo da un limite inferiore, il conducente si sposta su uno superiore e, prima

dello scadere dei due minuti, ritorna sul limite inferiore.
Questa informazione in questo caso non è possibile saperla e quindi gestirla, però analizzando un dataset composto di 266 029 veicoli in gestione all'azienda che inviano le posizioni più frequentemente, è risultato che nell'arco di un'intera giornata, cioè dalle ore 00.00 alle 24.00 hanno viaggiato per 40 607 020 minuti, che corrisponde a 28 199 giorni, e il periodo di tempo in cui si sono trovati nella situazione sopradescritta è il **0.92**% del tempo, che significa 260 giorni in totale, in pratica ogni dispositivo si è trovato in media per 1.4 minuti in questo caso, significa un campione da 2 minuti errato per giorno.
Perciò è evidente che questo dato sia trascurabile, non va ad influire in maniera significativa nel punteggio complessivo di guida.

5.2 Stile di guida

Grazie al dispositivo portatile spiegato nella sezione 4.4 è quindi stato possibile acquisire ed avere a disposizione un numero maggiore di dati, fondamentali per svolgere un'analisi dettagliata degli eventi, in questo modo è anche possibile avere un riscontro sul reale comportamento tenuto o segnalare qualche evento particolare.
Le prime guide effettuate sono state in funzione di identificare le differenze tra due stili di guida molto diversi, *sportivo* e *normale*. Utilizzando la stessa auto, lo stesso percorso e lo stesso autista, tramite il Portable real-time data recorder sono state acquisite le accelerazioni di due viaggi dove il primo è stato compiuto mantenendo una guida sobria, il secondo molto più spinta, come vediamo in Figura 5.2 Inoltre, per avere un confronto estremo di guida sportiva, sono stati registrati anche i dati provenienti da una guida su pista compiuta dalle *e-smart*, smart elettriche adibite a corsa su pista che partecipano allo *Smart E-Cup*, il primo campionato turismo al mondo dedicato ad una vettura elettrica, in Figura 5.3 è possibile osservare le accelerazioni in un segmento temporale significativo lungo l'asse y, confrontato con le accelerazioni dell'asse y della guida normale sopracitata, riportata sempre in Figura 5.3 in basso, è facile notare le differenze soprattutto in termini di ampiezza del segnale.

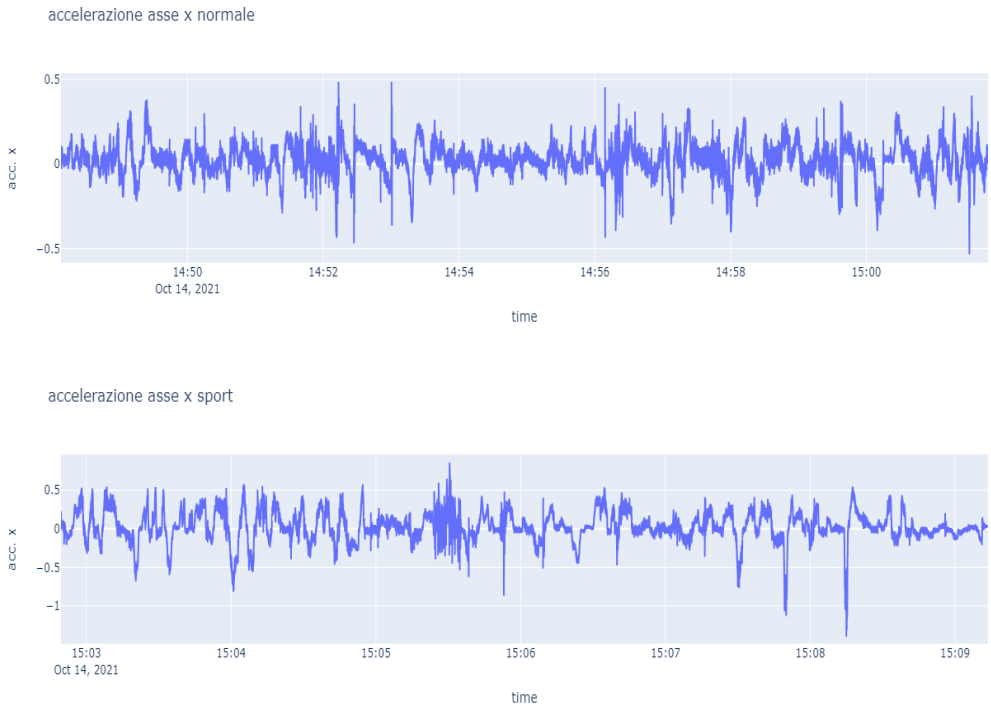


Figura 5.2: Grafici relativi a due stili di guida completamente opposti sul medesimo tratto di strada con le stesse condizioni iniziali

Figura 5.3: Grafico superiore: relativo all'accelerazione sull'asse y con prova su pista. Grafico inferiore: relatico all'accelerazione sull'asse y con guida moderata su strada

5.2.1 Generatore di guide

Avendo come unici dati utilizzabili quelli generati dalle guide svolte con il "Portable real-time data recorder", come spiegato nella Sezione 5, è nata l'esigenza di avere altre guide su cui testare gli algoritmi prodotti.
Perciò partendo da una guida con i dati di accelerazione grezzi, sono state ricavate varie guide prendendo i percentili dai dati grezzi con tempistiche diverse, facendo una traslazione dei minuti di acquisizione di un certo valore fissato, per esempio cominciare ad acquisire i dati appena dopo una rotonda porta ad una descrizione dell'istogramma dell'asse y molto diverso rispetto a cominciare l'acquisizione prima della rotonda.
Utilizzando questo algoritmo è stato possibile inoltre verificare la robustezza dell'algoritmo di calcolo delle valutazioni rispetto ad un campionamento differente.
Quindi il "generatore di guide" ha permesso di avere materiale maggiore per

la validazione dei fit ed avere la conferma di un punteggio molto simile su una stessa guida con i minuti traslati.

5.2.2 Analisi esplorativa dei dati

Per capire le relazioni tra i dati, come la correlazione, è stata utilizzata un'analisi esplorativa dei dati(EDA), analisi che ha permesso di visualizzare i dati utilizzando strumenti grafici.
La prima cosa analizzata è stata la correlazione tra le variabili, in particolare sono utilizati presi 9 percentili, quelli più significativi in modo che le rappresentazioni grafiche fossero più chiare e prendendo in considerazione una guida di 45 minuti. Il risultato, visibile in Figura 5.4, è in linea con i risultati aspettati, infatti possiamo osservare un'alta correlazione tra percentili simili, come ad esempio tra il minimo e il secondo percentile o tra il 92-esimo e il 98-esimo percentile. I percentili sugli assi sono distribuiti in modo che il grafico risulti più leggibile, è possibile infatti osservare i raggruppamenti viola scuro come quelli con una bassissima correlazione, mentre nei gialli e verdi chiaro la correlazione è molto alta(massima nel caso di confronto tra lo stesso percentile)
Un'analisi che ha codotto ad un risultato interessante è stata l'osservazione

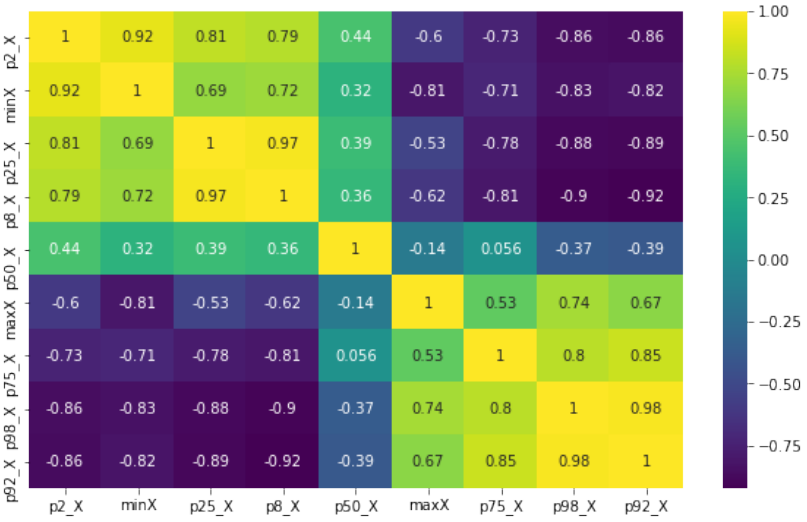


Figura 5.4: Correlazione tra percentili

della distribuzione dei singoli percentili nel tempo, infatti, guardando Figura 5.5 si possono identificare due stili di guida differenti, il blu moderato e l'arancione più agressivo.
È interessante notare come la distribuzione di ogni percentile della guida sportiva si distribuisca molto di più sull'asse dei valori, indice di una bassa stabilità e ripetibilità, mentre la guida tranquilla tende a concentrarsi in un intervallo molto più stretto.

Lo stesso vale per i grafici ottenuti dai percentili sull'asse y, quello che esprime la forza impressa al dispositivo durante le curve, dalla Figura 5.6 possiamo osservare una maggiore variabilità dei valori di accelerazione della guida sportiva, sia nel caso delle curve a destra che nel caso delle curve a sinistra.

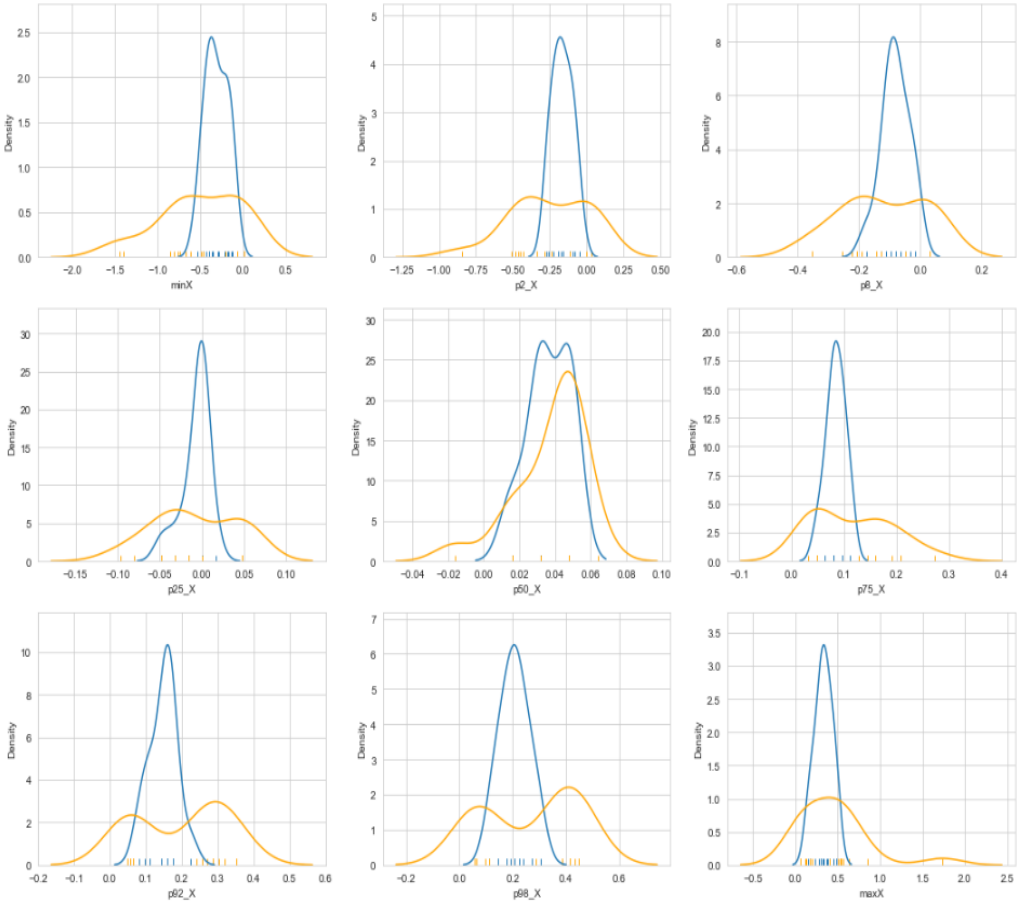


Figura 5.5: Distribuzioni dei percentili in fase di accelerazione e decelerazione

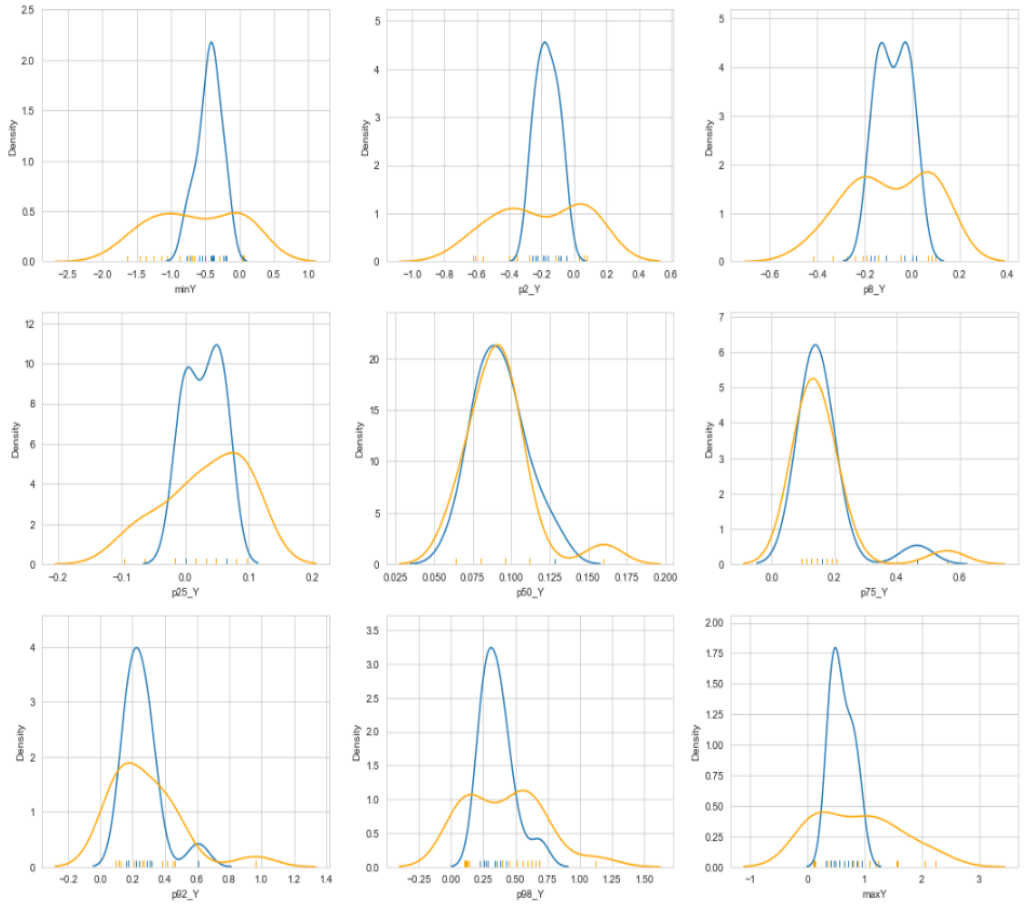


Figura 5.6: Distribuzioni dei percentili in fase di svolta a destra e a sinitra

5.2.3 K-means

Un ulteriore test effettuato è stato un *unsupervised test*, per cercare di capire come i dati si distribuissero su piani multidimensionali e per vedere se ci fosse qualche legame tra i valori o tra combinazioni di valori.
È stato utilizzato un algoritmo di clustering, in particolare il *k-means*, algoritmo che si basa sulla definizione di distanza dal centro di un cluster al punto preso in esame, che minimizza per trovare gli elementi apparteneti allo stesso cluster.
Per verificare la bontà delle rivelazioni dell'algoritmo sono stati utilizzati come dati i valori presi dalla guida moderata e dalla sportiva, l'obiettivo era osservare se si potesse utilizzare qualche particolare metrica per isolarle.
Sono stati provati due insiemi di dati in input:

- Percentili: utilizzando come input i percentili di una e l'altra guida, l'algoritmo non è riuscito ad identificare una regola univoca per la separazione desiderata.

- Percentili e punteggi: sono stati impiegati insiemi di valori composti dai percentili e dai punteggi associati ad ogni gruppo di minuti. Il risultato è stato soddisfacente, l'algoritmo riusciva ad identificare le due guide, però si basava quasi esclusivamente sui punteggi assegnati, mostrando una fortissima dipendenza da questi valori, un punteggio più alto lo assegnava ad una guida più sportiva, uno più basso ad una guida moderata. In Figura 5.7 è rappresentata la divisione in cluster utilizzando questo metodo relativamente al minimo, massimo e al cinqueantesimo percentile.

Quindi posso concludere che, utilizzando questo metodo di clustering, le valutazioni calcolate nella fase di fit del modello sono molto più significative dei percentili.

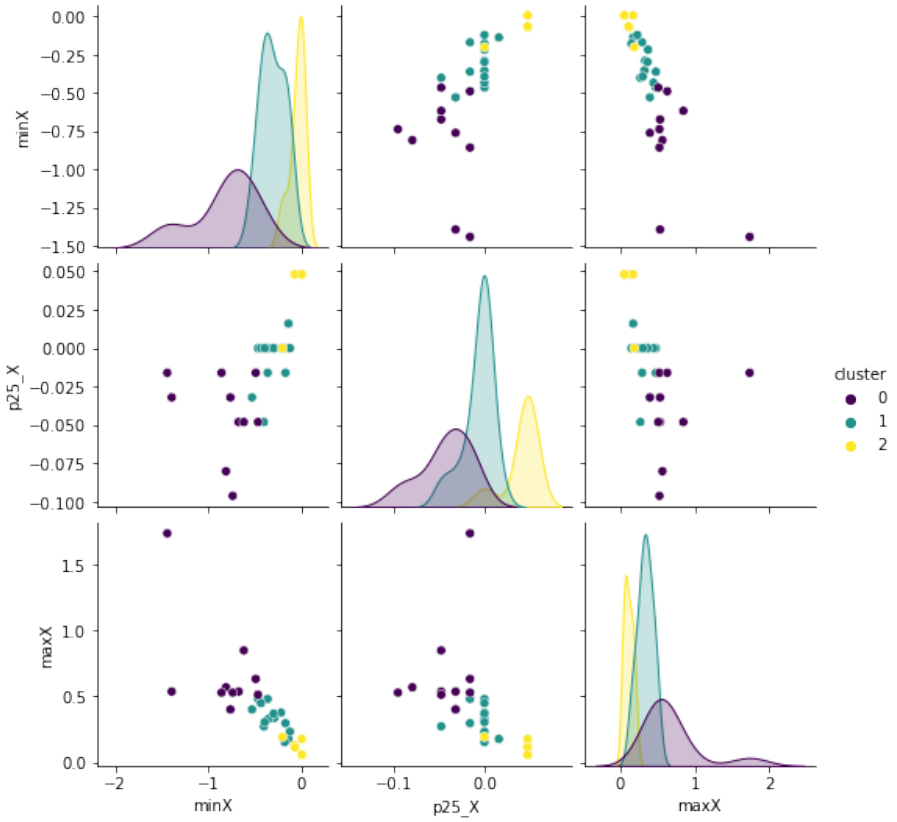


Figura 5.7: raggruppamento in cluster del minimo, massimo e cinquantesimo percentile, nella diagonale troviamo la distribuzione del percentile diviso per cluster

5.2.4 Considerazioni

Dopo le varie considerazioni espresse nei sottocapitoli precedenti, un ultimo test fatto per cercare di associare una guida ad un comportamento normale o sportivo è stato quello del *Kolmogorov-Smirnov test*[3], test statistico in grado di confrontare due insiemi di punti e capire se provengono dalla medesima distribuzione. Il test funziona in maniera molto simile al *Cramer Von Mises test*, se non differire per il calcolo dell'errore.
Data l'ipotesi iniziale H_0: distribuzione empirica = distribuzione fittata, e l'ipotesi alternativa H_a: distribuzione empirica $\neq$ distribuzione fittata, verifico la bontà dell'affermazione H_0 calcolando il p_value che indica la significatività dell'affermazione cioè quanta probabilità c'è di ottenere l'errore ricavato o superiore assumendo H_0 vera.

L'errore si calcola come la distanza tra le funzioni di ripartizione empiriche dei due campioni, viene calcolato come:

$$T = \sup_z |F_{1,n}(x) - F_{2,n}(x)| \tag{5.6}$$

dove $F_{1,n}(x)$ e $F_{2,n}(x)$ sono le distribuzioni empiriche del primo e secondo campione.
Il calcolo dell'errore è rappresentato in Figura 5.8.
L'obiettivo è confrontare un campione estratto dai dati con la guida mo-

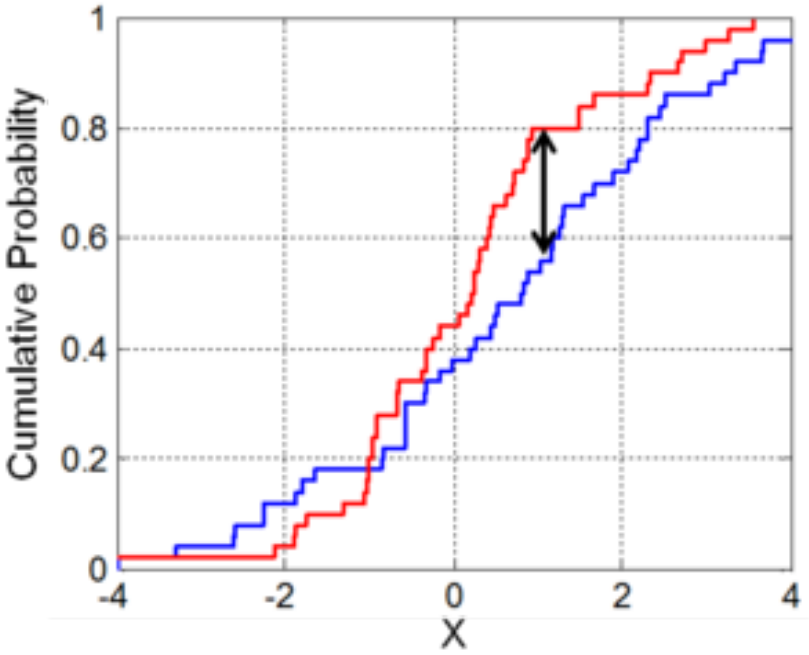


Figura 5.8: calcolo dell'errore secondo il ks-test

derata e quella sportiva campionate in precedenza, cercare di trovare un valore che dica quanto questo nuovo campione si avvicini ad una distribuzione piuttosto che ad un'altra, oppure se è più sportiva, se si trova in mezzo tra sportiva e moderata o se più moderata. Il limite riscontrato con questo metodo è dato dal fatto che il *ks-test* esprime solamente se due insimi di punti provengono dalla stessa distribuzione, perciò non è possibile assegnare dei bounds di appartenenza, solo ricavare se una nuova guida è simile o no da quelle in database, non di quanto si discosta, doverei avere infinite guide

da confrontare ognuna con una label che esprima il tipo di guida.
Grazie a questo problema è stata possibile l'intuizione di cambiare obiettivo, non più definire una guida sportiva o meno, ma un'assegnazione di un punteggio assoluto in base a tutti i parametri espressi nella Sezione 2.3, che poi potrà essere utilizzato come confronto tra veicoli, per capire se un guidatore guida meglio o peggio di un altro.

Capitolo 6

Riconoscimento dossi

Ulteriormente allo studio svolto in precendenza, avendo la possibilità, grazie al Portable real-time data recorder spiegato nel Capitolo 4.4 di acquisire dati con lo stesso campionamento dell'accelerometro, è stato possibile implementare un'analisi approfondita sul riconoscimento dei dossi.
Quest'analisi può essere utilizzata in vari contesti, in particolare i miei scopi sono:

- crash - validation: riconoscere se una rilevazione classificata come incidente è realmente tale. Gli apparecchi sono in grado di rilevare un crash come una sollecitazione degli accelerometri sopra una certa soglia impostabile a priori ed attuare un certo comportameto, però non sempre l'incidente è reale, potrebbe infatti essere legato ad una guida molto sportiva dell'utente, alla scarsa attenzione oppure alla non curanza del veicolo, sottovalutando così la natura della strada percorsa, impattando in modo violento su buche o dossi e perciò facendo avviare una routine di crash.

- stile di guida: utilizzata per attribuire un ulteriore parametro per l'assegnazione del punteggio utilizzato del Capitolo 5.

6.1 Funzionamento crash

In caso in cui il dispositivo rilevi un incidente, in automatico viene generato un report del crash.

In questo report sono salvati tutti i dati che potrebbero essere utili per un'eventuale revisione e contestazione in sede legale, come velocità all'impatto, posizione, accelerazioni/frenate precedenti all'impatto etc...
Un esempio di dati che sono ottenibili dal report sono rappresentati in Figura 6.1.
Ulteriormente a queste informazioni visive, è disponibile anche un database con tutti i valori delle accelerazioni con stesso campionamento dell'accelerometro, dieci secondi prima e dieci secondi dopo l'impatto.
Queste informazioni sono servite a trovare un algoritmo in grado di identificare i rallentatori(dossi) che fosse robusto nel caso di crash (un crash e un dosso possono essere molto simili in certi casi).

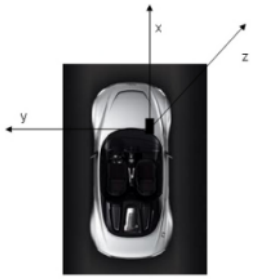
6.2 Analisi e identificazione

Per riuscire a distinguere dei dossi sono state usate varie regole, di seguito spiegherò le più importanti.
Innanzitutto, cominciando l'analisi da un dosso reale, come quello in Figura 6.2 è possibile notare come il segnale sia molto irregolare, perciò, applicando un filtro passabasso, otteniamo Figura 6.3.
Vengono analizzati l'asse x e z perchè nel caso di dosso non ha significato lo spostamento laterale sull'asse y.
Il filtro passabasso è un filtro che permette il passaggio di frequenze solo sotto una certa soglia(chiamata frequenza di taglio), utile per togliere il rumore ai segnali, in modo da poterli così analizzare più facilmente e con maggiore chiarezza. In questo caso è stato utilizzato un filtro passa-basso di secondo ordine con frequenza di taglio $f_{taglio} = 2\ Hz$, valore deciso osservando la Power Spectral Density[5] del segnale di partenza, cioè osservando le componenti in frequenza del segnale, visibile in Figura 6.4. Utilizzando questo filtro con questa frequenza vengono eliminate tutte le variazioni del segnale più veloci di $1/f_{taglio} = 0.5\ secondi$.

Osservando il segnale finale, isolando i due assi x e z e evidenziando i picchi positivi e negativi dopo una certa soglia (soglia che verrà analizzata più avanti), otteniamo Figura 6.5, dove possiamo osservare con più facilità

Dati Sinistro

Data e Ora	07/01/2022 10:06
Latitudine	40.906215
Longitudine	14.300368
Velocità all'impatto	32 km/h
Tipo Strada	Urbana
Luogo	Via Giovanni Boccaccio, 80026 Casoria NA, Italia
Senso di marcia	Non disponibile
Stato del quadro	Acceso
Qualità segnale GPS	Buona
Ultimi secondi prima dell'urto	
Velocità Media	34 km/h
Velocità Massima	41 km/h
Tempo di Arresto dopo l'impatto	3.0 s
Dati accelerazione	
Accelerazione Massima	6.43 g
Asse X	5.98 g
Asse Y	1.02 g
Asse Z	2.13 g



Dettaglio Accelerazione

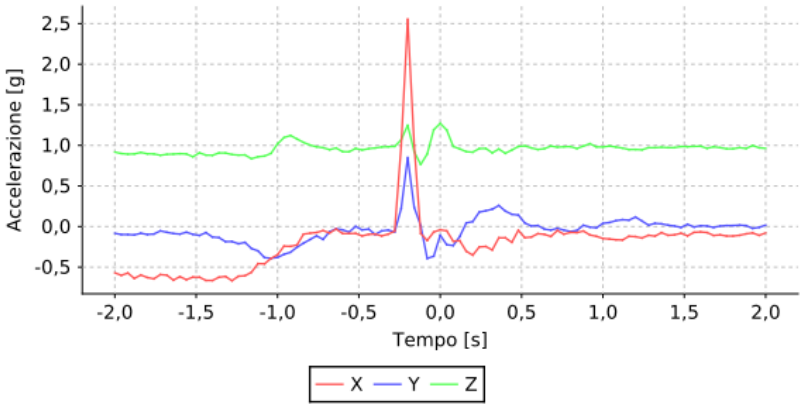


Figura 6.1: Esempi di dati fruibili dal report dopo un crash

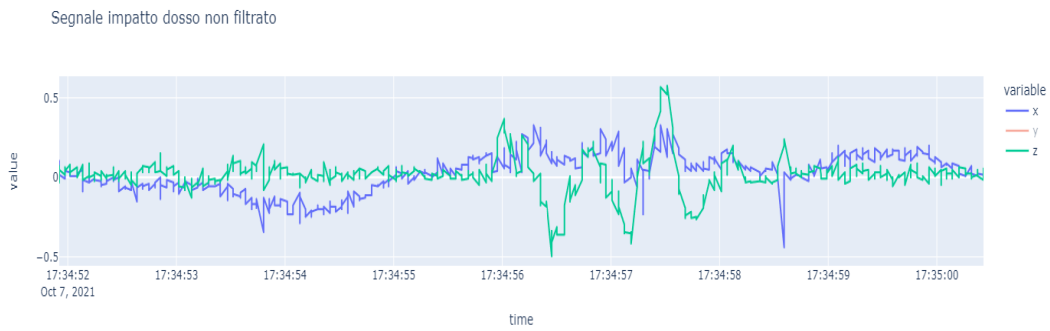


Figura 6.2: Accelerazioni non filtrate che rappresentano l'impatto di un veicolo su un rallentatore

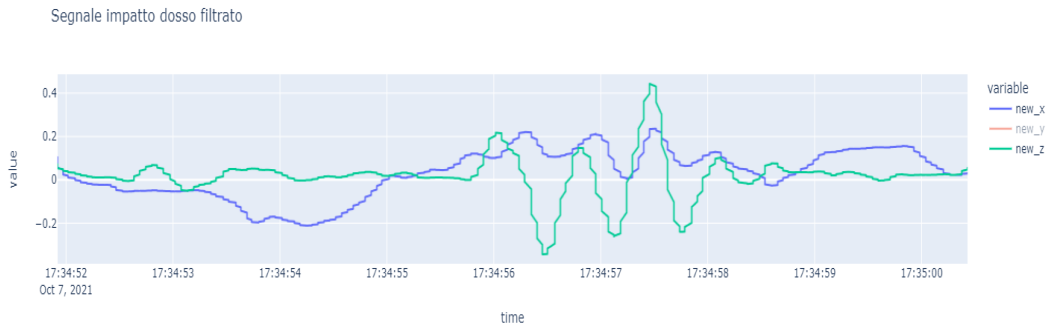


Figura 6.3: Accelerazioni dopo un filtro passabasso che rappresentano l'impatto di un veicolo su un rallentatore

alcune caratterisitche notevoli del segnale:

- sull'asse x, prima del segnale oscillante sull'asse z è presente una "conca", un picco negativo che è dovuto ad un brusco rallenamento del veicolo.
- le oscillazioni sull'asse z sono sempre un numero maggiore o uguale a due.
- i picchi sull'asse x e z sono sfasati tra loro, mai esattamente sullo stesso momento, altrimenti è molto più probabile che sia un incidente.
- il primo picco sulla z dopo la "conca" deve essere positivo, altrimenti è più probabile sia una buca

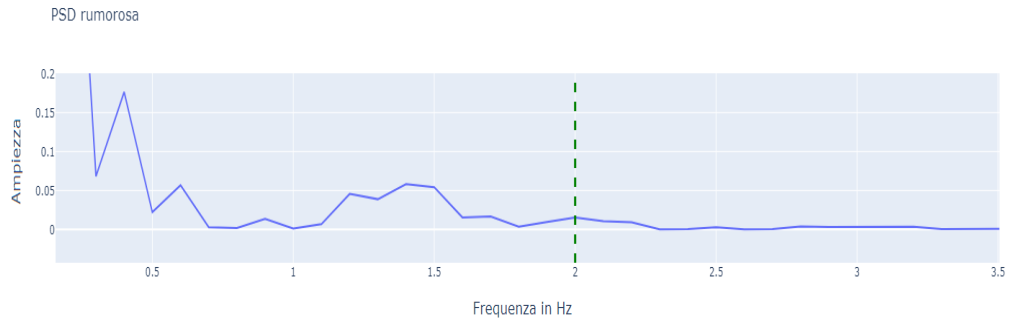


Figura 6.4: Power Spectral Density del segnale in Figura 6.2

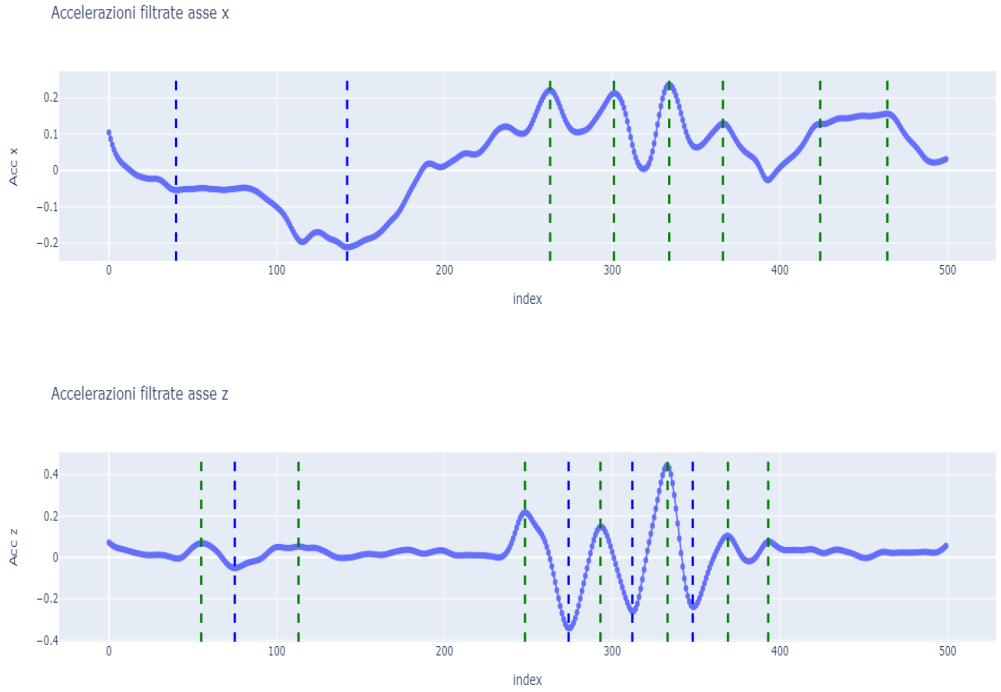


Figura 6.5: Asse x e z isolati

- c'è sempre un picco sull'asse x dopo la "conca"

Queste elencate sono le caratteristiche più significative. Il tutto è stato testato su un dataset di 1500 incidenti verificati (di cui siamo certi sia realmente successo un crash) ed applicandolo a questi segnali l'algoritmo identifica, erroneamente, 43 dossi, quindi su questo dataset l'errore è del 2.86%.

Oltre a questo test sono state effettuate verifiche su dossi reali, 40 dossi, di questi 9 non sono stati riconosciuti, quindi l'errore si attesta intorno al 22%.
Per capire i parametri ottimi da utilizzare, utili a decidere la soglia dei picchi sia positivi che negativi, è stata elaborata la curva ROC, grafico che mette in relazione la sensibilità e la specificità di un test al variare del valore di cut-off (valore soglia).
È stata creata una matrice di confusione che restituisce una rappresentazione dell'accuratezza di classificazione statistica utilizzando 40 casi per descrivere i quattro eventi vero-positivo, vero-negativo, falso-positivo, falso-negativo, Figura 6.6.
Per calcolare l'accuratezza del test è stato utilizzato l'F_1 *score*, calcolato

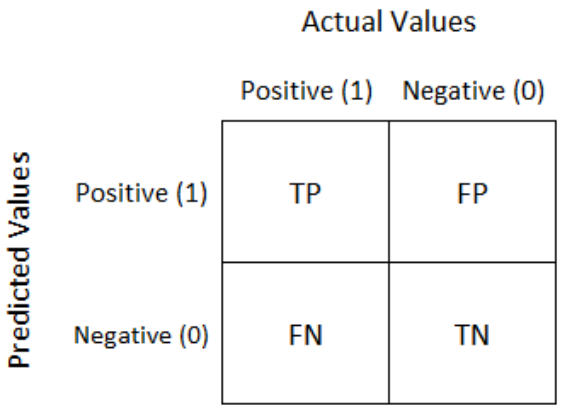


Figura 6.6: Matrice di confusione

tramite la media armonica di precisione e recupero. La formula generale è:

$$F_\beta = (1 + \beta^2) * \frac{p * r}{(\beta^2 * p) + r} \tag{6.1}$$

dove β rappresenta il tipo di score utilizzato(noi usiamo $\beta = 1$), p rappresenta la precisione e r il recupero.
Il vantaggio di utilizzare l'F_1 *score* rispetto alla media convenzionale è che quella armonica attribuisce un peso maggiore ai valori piccoli. Questò fa sì che un classificatore possa ottenere un alto punteggio solo quando precisione e recupero sono entrambi alti.

Sapendo che la precisione può essere scritta come:

$$p = \frac{VeroPositivo}{VeroPositivo + FalsoPositivo} \tag{6.2}$$

e il recupero come:

$$r = \frac{VeroPositivo}{VeroPositivo + FalsoNegativo} \tag{6.3}$$

otteniamo

$$F_1 = \frac{VeroPositivo}{VeroPositivo + \frac{FalsoNegativo + FalsoPositivo}{2}} \tag{6.4}$$

Il risultato si può osservare in Figura 6.7.
Possiamo notare come la soglia di decisione dei picchi migliore sia un valore

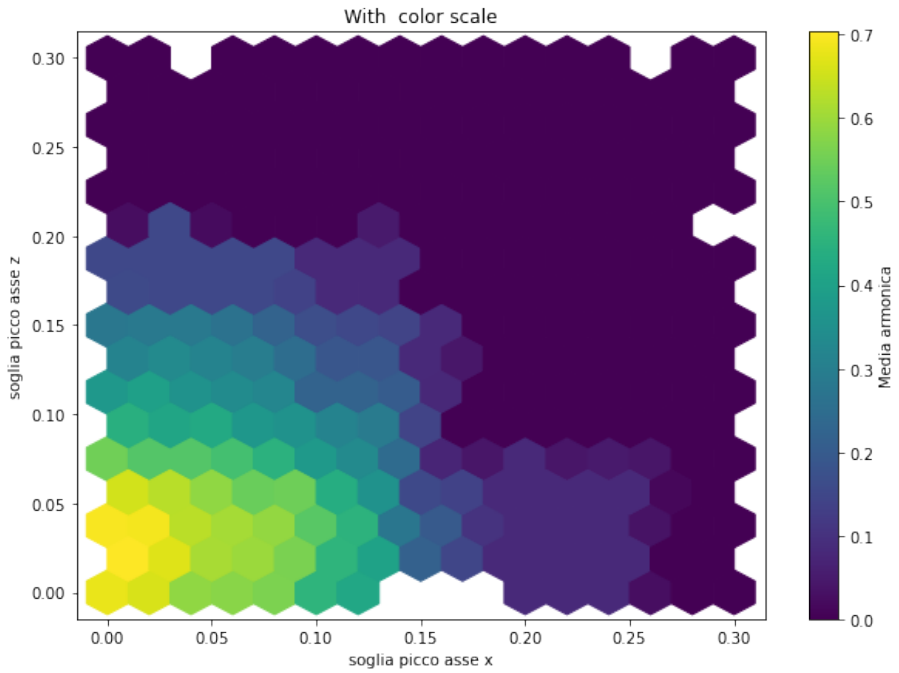


Figura 6.7: Grafico analisi Roc

molto basso, indice che il riconoscimento dei picchi è molto importante, fondamentale per cominciare l'analisi, infatti l'algorimo massimizza il numero di riconoscimenti con questo valore.
Questo algorimo per essere utilizzato come ulteriore punteggio per decidere

la valutazione finale deve essere implementato all'interno della scatola nera, in quanto per fare ques'analisi ha bisogno dei dati grezzi campionati con frequenza elevata, utilizzare solo i percentili non sarebbe sufficiente.

Capitolo 7

Conclusioni

In questo elaborato è stato studiato come ottenere un indicatore della bontà di guida assegnando ad ogni individuo un punteggio.
Partendo dalla descrizione dei dati di un'istogramma costituito dai valori di accelerazione e velocità campionati con una periodicità di 2 *minuti* è stato possibile, utilizzando una regressione non lineare[4], ricostruire il profilo di accelerazione partendo da statistiche descrittive del segnale preso in esame, e così capire il comportamento del soggetto alla guida.
Lo stile di guida della persona è stato valutato tramite l'assegnazione di alcuni punteggi. Per le accelerazioni la valutazione è rappresentata dalla media pesata dell'eccedenza rispetto ad una certa soglia (i pesi sono rappresentati dal tempo in cui si è stati sopra questo limite, e la soglia è fissata a priori. Per la velocità la modalità è simile, solo che la soglia non è fissa ed è rappresentata dal limite di velocità sulla strada percorsa.
Inoltre sono stati implementati alcuni parametri che incidono sull'importanza da dare ad un certo punteggio, ad esempio ad una guida più sportiva nelle ore di punta durante la settimana (con tante auto in circolazione) viene associato un punteggio più alto rispetto alla stessa guida durante le altre ore o rispetto al weekend oppure, di fronte allo stesso superamento medio dei limiti di velocità, il limite inferiore inciderà maggiormente sul punteggio complessivo.
In conclusione l'algoritmo come output genera un dataframe dove ogni riga corrisponde ad un veicolo differente.
Le prime sei colonne rappresentano il punteggio su ogni asse, la settima il

punteggio ottenuto guardando la velocità, nell'ottava colonna c'è la velocità media tenuta sopra i 50 Km/h e sulla nona il tempo per cui si è superato questo limite. La decima, undicesima e dodicesima colonna sono dedicate al tempo trascorso, la durata totale, di cui il periodo trascorso in orario di punta e nel weekend. Infine, come trecicesima colonna troviamo il punteggio ottenuto in tutto il periodo considerato, che tiene conto di ogni colonna di questo dataframe.
In Figura 7.1 troviamo un esempio di dataframe di una serie di veicoli nell'arco di una settimana, in Figura 7.2 la stessa tabella spezzata in due per una maggiore leggibilità.
Con questo punteggio è possibile confrontare tutti i veicoli della stessa flotta e così capire qual'è il migliore ed il peggiore, generando una graduatoria complessiva di tutti i veicoli.
Può essere utile per il gestore di flotte aziendali di un'azienda per decidere dei premi aziendali in corrispondenza ad uno basso punteggio, oppure in campo assicurativo per ridurre o aumentare il costo dell'RCA, infatti una persona a cui è associato un punteggio elevato ha una probabilità maggiore di incidentarsi, in questo modo è più facile associare una persona ad una categoria di rischio.
Alcuni miglioramenti possibili ed implementabili sono:

- Aggiungere all'algoritmo di decisione dello stile di guida il riconoscimento dei dossi spiegato nel Capitolo 6.
  É necessario implementare l'algoritmo analizzato precedentemente all'interno della scatola nera nel suo linguaggio di codice.

- Usare un'algoritmo di clustering per fare un'analisi finale sui punteggi ottenuti, capire se è possibile categorizzare sia come stile di guida bene/male sia secondo insiemi di altro tipo, come città (città diverse, stili diversi), o anche per separare zone di una stessa località, ad esempio riconoscere la città rispetto alla campagna.

- Correlare uno stile di guida sportivo ad un numero maggiore di riparazioni/incidenti riuscendo così a definire un'accurata probabilità di "rischio".

	acc_X	fren_X	acc_Y	fren_Y	acc_Z	fren_Z	score_velocità	eccedenza_media_50km/h	min_out_of_limits_50km/h	total_time [d, h:m:s]	ora di punta	weekend	total_score
0	0.156838	-0.168081	0.204072	-0.171282	0.148595	-0.148681	0.046239	13.209576	27.294875	3:21:00	2:51:00	0:30:00	11.626591
1	0.181538	-0.176532	0.541810	-0.977640	0.144490	-0.981611	0.036105	10.168875	21.081076	6:00:00	5:46:00	0:14:00	22.385755
2	0.149322	-0.175011	0.196800	-0.165907	0.149348	-0.144639	0.048402	12.038167	81.993632	10:41:00	5:23:00	5:18:00	11.710573
3	0.137594	-0.157014	0.169372	-0.175249	0.148456	-0.152373	0.034007	10.210033	73.077062	7:27:00	7:27:00	0:00:00	9.792965
4	0.200787	-0.153676	0.161417	-0.175188	0.149890	-0.149687	0.062122	12.413524	117.035460	15:21:00	11:12:00	4:09:00	13.122842
5	0.152591	-0.172781	0.199397	-0.168926	0.149076	-0.147333	0.053666	12.438586	67.349742	7:04:00	6:10:00	0:54:00	12.303515
6	0.190264	-0.174686	0.391064	-0.880677	0.147135	-1.102745	0.037975	9.408263	49.634898	11:16:00	9:11:00	2:05:00	20.164452
7	0.151920	-0.176645	0.198281	-0.152517	0.147876	-0.144195	0.031570	9.458366	73.412954	9:10:00	4:57:00	4:13:00	9.950646
8	0.139866	-0.157661	0.164227	-0.174103	0.144813	-0.148564	0.020023	7.551702	25.570314	2:04:00	2:04:00	0:00:00	8.360832
9	0.213368	-0.150422	0.163455	-0.175471	0.145402	-0.145986	0.033810	8.900050	66.293028	11:11:00	8:34:00	2:37:00	10.408208
10	0.163954	-0.159710	0.783280	-0.181081	0.147940	-1.105529	0.058311	10.907281	43.302676	5:41:00	5:41:00	0:00:00	18.711376
11	0.185555	-0.692401	0.167813	-0.166581	0.151334	-1.919739	0.049962	9.106258	30.785171	8:10:00	7:10:00	1:00:00	17.119729

Figura 7.1: Dataframe con i punteggi dei vari veicoli

	acc_X	fren_X	acc_Y	fren_Y	acc_Z	fren_Z	score_velocità
0	0.156838	-0.168081	0.204072	-0.171282	0.148595	-0.148681	0.046239
1	0.181538	-0.176532	0.541810	-0.977640	0.144490	-0.981611	0.036105
2	0.149322	-0.175011	0.196800	-0.165907	0.149348	-0.144639	0.048402
3	0.137594	-0.157014	0.169372	-0.175249	0.148456	-0.152373	0.034007
4	0.200787	-0.153676	0.161417	-0.175188	0.149890	-0.149687	0.062122
5	0.152591	-0.172781	0.199397	-0.168926	0.149076	-0.147333	0.053666
6	0.190264	-0.174686	0.391064	-0.880677	0.147135	-1.102745	0.037975
7	0.151920	-0.176645	0.198281	-0.152517	0.147876	-0.144195	0.031570
8	0.139866	-0.157661	0.164227	-0.174103	0.144813	-0.148564	0.020023
9	0.213368	-0.150422	0.163455	-0.175471	0.145402	-0.145986	0.033810
10	0.163954	-0.159710	0.783280	-0.181081	0.147940	-1.105529	0.058311
11	0.185555	-0.692401	0.167813	-0.166581	0.151334	-1.919739	0.049962

min_out_of_limits_50km/h	total_time [d, h:m:s]	ora di punta	weekend	total_score
27.294875	3:21:00	2:51:00	0:30:00	11.626591
21.081076	6:00:00	5:46:00	0:14:00	22.385755
81.993632	10:41:00	5:23:00	5:18:00	11.710573
73.077062	7:27:00	7:27:00	0:00:00	9.792965
117.035460	15:21:00	11:12:00	4:09:00	13.122842
67.349742	7:04:00	6:10:00	0:54:00	12.303515
49.634898	11:16:00	9:11:00	2:05:00	20.164452
73.412954	9:10:00	4:57:00	4:13:00	9.950646
25.570314	2:04:00	2:04:00	0:00:00	8.360832
66.293028	11:11:00	8:34:00	2:37:00	10.408208
43.302676	5:41:00	5:41:00	0:00:00	18.711376
30.785171	8:10:00	7:10:00	1:00:00	17.119729

Figura 7.2: Dataframe con i punteggi dei vari veicoli

Bibliografia

[1] Gareth Halfacree Eben Upton. *Raspberry Pi. La guida completa*. 2013.

[2] J. Kiefer. *K-Sample Analogues of the Kolmogorov-Smirnov and Cramer-V. Mises Tests*. 1959.

[3] H. W. Lilliefors. *On the Kolmogorov-Smirnov Test for Normality with Mean and Variance Unknown*. 1967.

[4] Andreas Ruckstuhl. *Introduction to Nonlinear Regression*. 2010.

[5] S. Tubaro. *Some notes on the power spectral density of random processes*. 2011.

Siti consultati

- 3 Methods for Parallelization in Spark – https://towardsdatascience.com/3-methods-for-parallelization-in-spark/
- Percentile e quartile – https://meetheskilled.com/percentile-e-quartile/
- K-Means clustering – https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html
- K Means Clustering Simplified in Python – https://www.analyticsvidhya.com/blog/2021/04/k-means-clustering-simplified-in-python/
- Generali Ass. – https://www.generali.it/iniziative/iniziative/commerciali/generali-mydrive#cosa
- Cattolica Ass. – https://www.cattolica.it/-/active-auto
- Comunicazione GMS – https://it.wikipedia.org/wiki/Global_System_for_Mobile_Communications
- curve_fit – https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.curve_fit.html
- Trust Region Reflective Algorithm – https://nmayorov.wordpress.com/2015/06/19/trust-region-reflective-algorithm/
- The Two-Sample Cramer-von Mises Criterion – https://projecteuclid.org/journals/annals-of-mathematical-statistics/volume-33/issue-3/On-the-Distribution-of-the-Two-SampleCramer-von-Mises/10.1214/aoms/1177704477.full
- Unsupervised learning for data classification – https://developer.ibm.com/articles/cc-unsupervised-learning-data-classification/
- Computer Raspberry Pi 4 – Guida utente modello B – https://manuals.plus/it/raspberry-pi/raspberry-pi-4-computer-model-b-manual
- Operating system images – https://www.raspberrypi.com/software/operating-systems/

- DS3231 Real Time Clock in Raspberry Pi – https://iot-guider.com/raspberrypi/learn-interfacing-ds3231-real-time-clock-raspberry-pi/
- Analisi esplorativa dei dati – https://www.ibm.com/it-it/cloud/learn/exploratory-data-analysis
- Analisi statistica descrittiva – https://ichi.pro/it/primo-passo-in-eda.analisi-statistica-descrittiva-268953587407381
- ks-test python – https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.kstest.html
- Python pandas – https://pandas.pydata.org/docs/
- ROC -Receiver Operating Characteristics- Curve – https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5
- Understanding Confusion Matrix – https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62
- TomTom mappe – https://www.tomtom.com/products/map-technology/
- Targa Telematics – https://www.targatelematics.com/