Università degli Studi di Padova

Dipartimento di Scienze Statistiche

Corso di laurea triennale in
Statistica per le Tecnologie e le Scienze

Relazione finale

Guida statistica alla Formula Uno

Relatore: **Prof. Mauro Bernardi**

Laureando: **Filippo Scalabrin**
Matricola: **1217665**

Anno Accademico 2021/2022

Indice

1 Creazione dei datasets **8**

1.1 Analisi delle fonti di variabilità di un gran premio 8

1.2 Datasets "Race" e "Qualifying" 9

1.2.1 Descrizione delle variabili del dataset "Race" 9

1.2.2 Il dataset "Race 2" 11

1.2.3 Variabili del dataset "Qualifying" 12

1.3 Analisi della variabilità atmosferica 12

2 Analisi esplorative **15**

2.1 Team, piloti e gran premi dal 2014 al 2019 15

2.2 Statistiche di sintesi per piloti e gran premi 17

2.3 I dati meteorologici 21

2.4 Posizione mediana in classifica dei piloti 24

3 Modelli per gran premi di Formula Uno **25**

3.1 Regressione robusta e tempo sul giro 25

3.2 Albero di classificazione 29

3.2.1 Dataset e metodo di costruzione del CT 31

3.2.2 Risultati dell'albero di classificazione 32

3.2.3 La "*variable selection*" dell'albero di classificazione 33

3.3 Foresta casuale 34

3.3.1 Specificazione 34

3.3.2 Applicazione ai dati 34

3.4 Modelli GAMM 35

3.4.1 Specificazione di un modello GAM 35
3.4.2 Analisi di alcuni gran premi del 2015 mediante GAMM . . . 36
3.4.3 Analisi della strategia di Massa al Gran Premio d'Italia del 2015 38
3.5 Estensione del Modello 1 utlizzando informazioni sui settori 41
3.5.1 Analisi dei tempi sul giro di Monza 2019 42
3.6 Estensione del Modello 2 utilizzando informazioni sui settori e sul meteo 43
3.6.1 Il test del Modello 3 e i punti deboli dell'approccio tramite GAMM 44

4 Conclusioni **48**

Riferimenti bibliografici **50**

Riferimenti sitografici **51**

Appendice: codice R **52**

Ringraziamenti **89**

Introduzione

La Formula Uno è uno sport motoristico nel quale si sfidano vetture monoposto a ruote scoperte. Le scuderie partecipanti competono per la vittoria dei gran premi di un Campionato, che si disputa a cadenza annuale dal 1950, tipicamente nel periodo compreso tra fine marzo e dicembre. I gran premi si svolgono in circuiti sparsi nei cinque continenti del mondo; tra i più celebri, vi sono Silverstone (Gran Bretagna), Spa-Francorchamps (Belgio), l'Autodromo di Monza (Italia), l'Autódromo José Carlos Pace (Brasile), il Circuito di Barcellona (Spagna) e il Circuito di Sakhir (Bahrain). La Figura 1 riporta le forme dei sei circuiti appena citati.

Ciascuna scuderia schiera, per ogni gran premio, due piloti. In palio, vi sono il Titolo Costruttori e il Titolo Piloti, conquistati - rispettivamente - dal team e dal pilota che riescono ad accumulare più punti al termine del Campionato. Di norma, ogni gara assegna punti ai primi 10 classificati; nello specifico, 25 al vincitore, 18 al secondo e 15 al terzo, mentre non sono previsti punti extra per chi conquista la pole position durante la sessione di qualifica [2].

Nel corso del tempo, la Formula Uno - anche grazie alla partnership del 2018 con l'azienda di cloud computing Amazon Web Services (AWS) - ha posto nei dati la pietra miliare delle strategie di gara. Equipaggiata con più di 120 sensori, ogni vettura si configura come una vera e propria miniera di informazioni, capace di trasmettere dalla pista ai box circa un milione di punti dati di telemetria al secondo, per un totale di oltre 800 GB a gran premio [4]. Combinando i dati che le monoposto forniscono in tempo reale con quelli storici e con i "*trackside data*" (provenienti da sensori posizionati lungo il circuito), gli ingegneri e i data scientists delle scuderie possono trarre informazioni utili su aspetti come consumo di carburante, usura degli pneumatici, velocità e angolo di sterzata, condizioni meteorologiche e posizione della vettura sul tracciato. Questo tipo di analisi è vantaggioso non solamente per definire le appropriate tattiche di qualifica o di gara, ma anche per monitorare ed ottimizzare le prestazioni del veicolo e per identificare i punti di forza e debolezza dei veicoli degli avversari.

È chiaro allora che la statistica assume un ruolo di prim'ordine nell'analisi ed interpretazione di queste masse di dati, così come del resto, in quest'ultimo periodo, sta avvenendo nell'ambito di molti altri sport [6] [8]. L'obiettivo della presente Tesi consiste proprio nel fornire una chiave di lettura statistica di alcuni aspetti

legati all'andamento di un gran premio di Formula Uno. I dati a disposizione fanno riferimento ai sei Campionati disputati tra il 2014 e il 2019. In questo periodo, quello della "rivoluzione ibrida" della Formula Uno, una squadra su tutte ha saputo dominare la concorrenza: si tratta della Mercedes di Lewis Hamilton (2014-2019), Nico Rosberg (2014-2016) e Valtteri Bottas (2017-2019), capace di conquistare tutti i Titoli in palio (si veda la Tabella 1). Purtroppo, molti dei dati di telemetria sono custoditi gelosamente dalle scuderie, e si farà riferimento solo a quelli disponibili.

La Tesi è organizzata come segue. Nel Capitolo 1 viene illustrata la procedura di creazione e manipolazione dei datasets di riferimento. Seguono, nel Capitolo 2, alcune analisi esplorative atte a sintetizzare i dati di partenza e a presentare sia i principali protagonisti (polemen e vincitori) dei Mondiali, sia alcune caratteristiche dei circuiti. Il Capitolo 3 è dedicato alla modellizzazione: si è implementato dapprima un metodo di regressione robusta per mostrare la relazione tra tempo sul giro e numero di giri percorsi; sono stati poi messi in pratica dei metodi di machine learning — tra cui, in particolare, quello degli alberi di classificazione — per determinare le variabili in grado di discriminare maggiormente il piazzamento finale di un pilota; infine, si sono elaborati alcuni modelli additivi generalizzati ("*Generalized Additive Models*", GAM) per stimare i tempi sul giro di un gran premio.

Le analisi dei dati svolte per ottenere risultati e grafici sono state compiute mediante software R (versione 4.2.0). Il codice è riportato nell'Appendice.

Tabella 1: Classifica finale di team e piloti per ogni stagione dal 2014 al 2019. Tra parentesi, accanto al nome del pilota, la rispettiva squadra. Legenda delle sigle utilizzate: Mercedes (MER), Red Bull Racing (RBR), Ferrari (FER).

Stagione	Primo posto		Secondo posto		Terzo posto	
	Pilota	Costruttore	Pilota	Costruttore	Pilota	Costruttore
2014	L. Hamilton (MER)	Mercedes	N. Rosberg (MER)	Red Bull	D. Ricciardo (RBR)	Williams
2015	L. Hamilton (MER)	Mercedes	N. Rosberg (MER)	Ferrari	S. Vettel (FER)	Williams
2016	N. Rosberg (MER)	Mercedes	L. Hamilton (MER)	Red Bull	D. Ricciardo (RBR)	Ferrari
2017	L. Hamilton (MER)	Mercedes	S. Vettel (RBR)	Ferrari	V. Bottas (MER)	Red Bull
2018	L. Hamilton (MER)	Mercedes	S. Vettel (RBR)	Ferrari	K. Raikkonen (FER)	Red Bull
2019	L. Hamilton (MER)	Mercedes	V. Bottas (MER)	Ferrari	M. Verstappen (RBR)	Red Bull

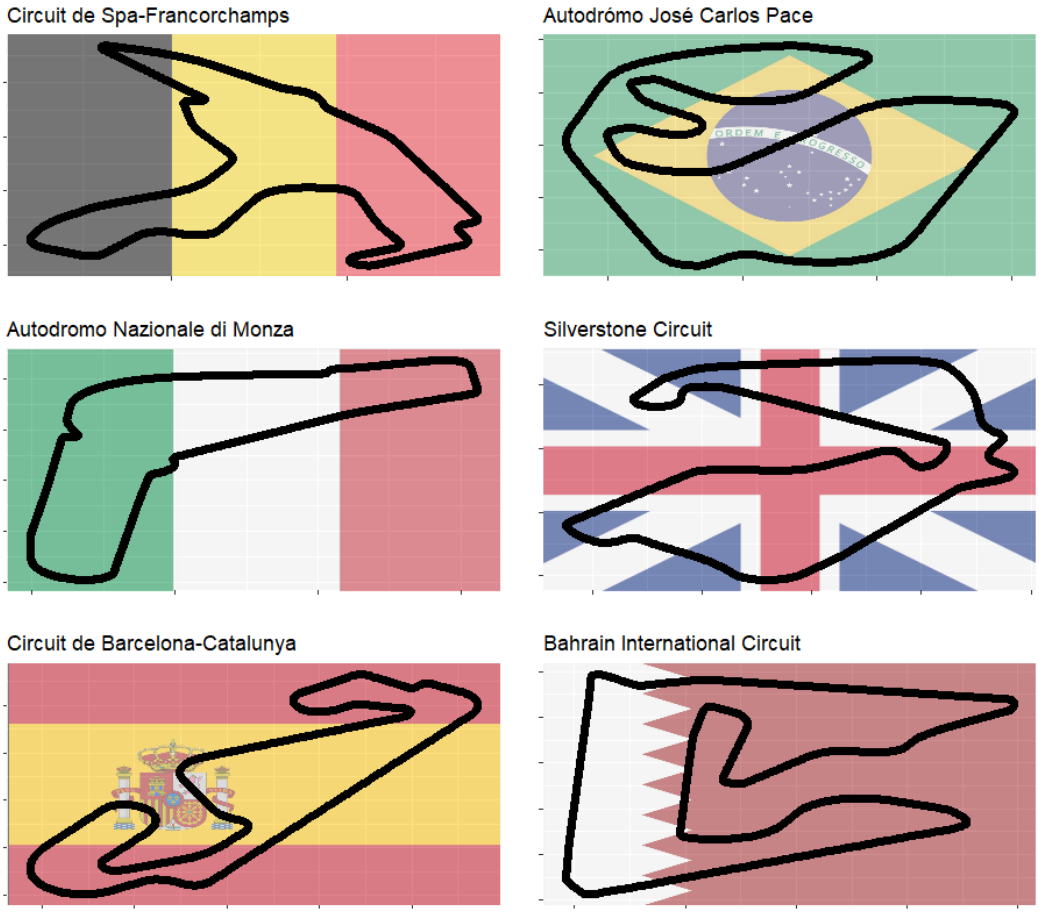

Figura 1: Sei tra i più iconici circuiti del Mondiale di Formula Uno.

1 Creazione dei datasets

In questo capitolo si descrivono la struttura dei datasets a disposizione e le variabili che li compongono. Sono stati considerati sia dati legati ai tempi sul giro e ai risultati dei piloti, sia dati dedicati alle condizioni meteorologiche delle gare. Come verrà illustrato nel paragrafo 1.1, un gran premio di Formula Uno può essere decisamente imprevedibile, per cui ciascuna variabile presente nei datasets ha una propria, fondamentale importanza nello spiegare l'esito di una gara.

1.1 Analisi delle fonti di variabilità di un gran premio

Per poter raggiungere un buon piazzamento in un gran premio occorre, innanzitutto, tentare di conquistare in qualifica la miglior posizione di partenza possibile. Il sistema di qualifica si compone di tre sessioni (dette "*manches*") di durata rispettivamente pari a 20, 15 e 10 minuti; le prime due prevedono l'eliminazione dei cinque piloti che occupano le ultime posizioni nella classifica dei tempi, mentre nell'ultima manche i piloti si giocano la pole position. Una buona posizione di partenza ottenuta in qualifica, tuttavia, non è automaticamente garanzia di una buona posizione di arrivo.

Sicuramente, un aspetto chiave nell'evoluzione di una gara di cui tener conto è quello dalla gestione delle gomme. Nel corso dei Campionati in esame, l'azienda Pirelli, fornitrice unica di pneumatici, ha messo a disposizione dei piloti varie mescole di gomma: dalla A1, la più dura, alla A7, la più soffice, oltre che le gomme Intermedie per tracciato umido o moderatamente bagnato e le gomme da bagnato estremo ("*Full Wet*"). Mescole di pneumatici più dure permettono al pilota di compiere più giri e di fare meno soste ai box (evitando quindi di perdere secondi preziosi), ma allo stesso tempo non permettono tempi sul giro tanto veloci quanto quelli ottenuti con gomme più morbide. L'usura delle gomme dipende, oltre che dalla mescola, dallo stile di guida del pilota, dall'assetto aerodinamico del veicolo e dalla distanza dalla vettura che lo precede in pista. Se un sorpasso non viene compiuto in poco tempo, infatti, l'aria sporca e calda può causare innalzamenti anomali della temperatura d'esercizio della gomma e, conseguentemente, un degrado più marcato.

Non è tutto: un weekend di gara perfetto, contrassegnato da ottima posizione in qualifica, trade-off ottimale tra velocità e usura delle gomme, e assenza di guasti

meccanici, può essere rovinato da una Safety Car dispiegata dalla direzione gara a causa di un incidente, o da un improvviso acquazzone che si abbatte sul circuito. L'ingresso in pista della Safety Car o il segnale di Virtual Safety Car comportano un rallentamento generale della velocità delle vetture, e fanno sì che i piloti possano effettuare delle soste ai box perdendo meno tempo [5]. Ciò va evidentemente a discapito dei piloti che si sono fermati appena prima del periodo di Safety Car (o di Virtual Safety Car), i quali sono costretti a difendersi, alla ripartenza, da avversari con gomme più fresche rispetto alle loro. Si sottolinea che la Virtual Safety Car in Formula Uno entra in funzione in caso di incidente non particolarmente grave, ed impone un limite di velocità senza bisogno dell'ingresso della vettura di sicurezza vera e propria.

Per quanto riguarda le condizioni atmosferiche, sebbene gli ingegneri abbiano modo di monitorare costantemente gli aggiornamenti meteo, è sempre compito del pilota decidere, in base alla sua sensibilità, se fermarsi per cambiare i suoi pneumatici da asciutto ("*slick*") in pneumatici da bagnato, e quali gomme da bagnato — intermedio o estremo — montare. Questa decisione, generalmente, è cruciale per gli sviluppi della corsa.

1.2 Datasets "Race" e "Qualifying"

Per tenere conto, nelle analisi, di tutti gli aspetti elencati al paragrafo 2.1, sono stati selezionati, creati e integrati vari datasets. Il primo, a cui d'ora in avanti si farà riferimento con il nome di dataset "*Race*", comprende i dati relativi ai 134006 giri effettuati da tutti i piloti nel corso di tutti i gran premi dal 2014 al 2019, ed è una rielaborazione del database "*F1 Timing*", realizzato dal Prof. Alexander Heilmeier dell'Institute of Automotive Technology, Università di Monaco di Baviera. F1 Timing attinge dai dati provenienti da un web service sperimentale che consente l'accesso a dati storici del mondo del motorsport, Ergast; sono state aggiunte le variabili race, team, driver e fcy (acronimo di "Full Course Yellow": un regime di bandiere gialle, segno di pericolo, in tutta la pista). Il secondo, che chiameremo dataset "*Qualifying*", è sempre parte del database F1 Timing e contiene i dati sui tempi fatti registrare nelle tre manches di qualifica da parte dei piloti.

1.2.1 Descrizione delle variabili del dataset "Race"

Nella Tabella 2 vengono descritte brevemente tutte le variabili del dataset Race, accompagnate dalla specificazione del loro tipo (carattere, numerica o categoriale).

Vale la pena sottolineare che:

– dal 2014 al 2019 non ci sono stati gran premi corsi in due circuiti diversi nello stesso Stato, dunque le sigle della variabile race sono univoche per ogni anno;

Tabella 2: Descrizione delle variabili presenti nel dataset Race.

#	Nome della variabile	Tipo della variabile	Descrizione della variabile
1	race	carattere	Sigla di tre lettere indicante il circuito dove si è svolto il giro.
2	season	numerica	Anno del Campionato a cui fa riferimento un certo giro.
3	driver	carattere	Sigla di tre lettere indicante il cognome del pilota autore del giro.
4	team	carattere	Sigla di tre lettere indicante la scuderia per cui corre il pilota autore del giro.
5	driverid	numerica	Identificatore numerico di un pilota.
6	lapno	numerica	Numero del giro della gara.
7	raceid	numerica	Identificatore numerico di una gara.
8	fcy	categoriale	Regime di gara (SC, VSC o TC).
9	position	numerica	Posizione del pilota nel corso del giro.
10	laptime	numerica	Tempo sul giro in secondi.
11	racetime	numerica	Tempo sul giro cumulato in secondi.
12	gaptol	numerica	Distacco di un pilota dal leader della corsa, in secondi.
13	interval	numerica	Distacco di un pilota dal pilota precedente in secondi.
14	compound	categoriale	Mescola di pneumatici usata dal pilota in un giro.
15	tireage	numerica	Età della gomma in giri.
16	pitthislap	categoriale	Assume valore 1 se il pilota effettua la sosta nel giro.
17	pitstopduration	numerica	Durata in secondi del pit-stop.

– il regime di Full Course Yellow (variabile fcy) indica la presenza in pista di Safety Car (SC) o Virtual Safety Car (VSC); in caso contrario, il regime con bandiere verdi è quello di pista pulita ("*Track Clear*", TC);

– la variabile compound, relativa alla mescola degli pneumatici, assume un totale di nove modalità: sette di queste vanno su una scala da A1, che indica le gomme più dure in assoluto, ad A7 (le gomme più soffici), mentre le altre due (I e W) denotano rispettivamente mescole intermedie e da bagnato estremo;

– alcuni piloti, alla partenza del gran premio, hanno un valore della variabile tireage maggiore di 0; questo è dovuto alla norma (abolita nel 2022) che prevedeva che i primi 10 classificati usassero per la partenza le gomme adoperate per il giro veloce della seconda sessione di qualifiche.

1.2.2 Il dataset "Race 2"

Si è preso poi in considerazione un dataset sempre riferito alle gare, ma per certi aspetti diverso dal dataset Race. Tramite un'API chiamata "*FastF1*" è stato fatto scraping dei dati provenienti dalla telemetria ufficiale F1 per tutti i gran premi del 2019. I dati sono stati opportunamente trasformati (il codice utilizzato è riportato in Appendice) mediante la conversione dei tempi in secondi, l'eliminazione di alcune variabili non necessarie allo scopo del presente progetto e la costruzione di altre variabili. Le principali differenze tra il dataset Race e quest'altro dataset, a cui d'ora in poi, per semplicità, si farà riferimento come dataset "*Race 2*", sono qui di seguito sintetizzate.

– ogni circuito di Formula Uno è suddiviso in 3 settori. A differenza del dataset Race, nel dataset Race 2 è presente, oltre al tempo sul giro (in secondi) per ciascun pilota, anche il tempo fatto da lui registrare per ogni settore del circuito;

– potendo sfruttare l'informazione derivante dai tempi per ogni settore, sono state create tre variabili dicotomiche denominate FastSector1, FastSector2 e FastSector3, con valore 1 assunto se il pilota fa registrare il suo "*personal best*" in un settore (ovvero, il suo miglior crono nel settore fino a quel momento della corsa), e valore 0 altrimenti;

– nel dataset Race 2 non è presente una variabile equivalente a gaptol e la variabile relativa alla mescola di pneumatici non assume modalità su una scala da A1 a A7, bensì i soli valori "hard", "medium" e "soft";

– la variabile FreshTyre del dataset Race 2, categoriale, indica se le gomme che il pilota usa in un certo giro erano nuove o usate nel momento in cui sono

Tabella 3: Descrizione delle variabili presenti nel dataset Qualifying.

#	Nome della variabile	Tipo della variabile	Descrizione della variabile
1	raceid	numerica	Identificatore di un gran premio.
2	position	numerica	Posizione in griglia del pilota.
3	driverid	numerica	Identificatore numerico di un pilota.

state messe dai meccanici. Si osserva come anche dopo una sosta lo stato della gomma possa essere usato, in quanto in mancanza di pneumatici freschi può essere montato su una vettura un treno di gomme già usato durante le prove libere o le sessioni di qualifica.

1.2.3 Variabili del dataset "Qualifying"

Nella Tabella 3 vengono illustrate le variabili d'interesse del dataset Qualifying. Le variabili raceid e driverid di questo dataset si riferiscono, rispettivamente, alle stesse gare e agli stessi piloti del dataset Race.

1.3 Analisi della variabilità atmosferica

È indispensabile, in Formula Uno, monitorare costantemente il tempo atmosferico durante i weekend di gara, e in particolare durante i gran premi, al fine di modulare la strategia più corretta. Per tener conto di questo importante fattore sono stati costruiti due datasets relativi alle condizioni meteorologiche dei gran premi. Il primo considera i Campionati dal 2014 al 2017 ed è stato creato a partire dai dati scaricati dal sito Visual Crossing - Weather Data Service (consultabile all'indirizzo `www.visualcrossing.com`). A partire da un'ora prima del momento d'inizio della corsa, sono stati estratti i dati meteo rilevati da stazioni meteorologiche posizionate nei pressi del circuito. Una singola rilevazione fa riferimento ad un lasso di tempo pari a 5 minuti.

Abbiamo verificato che i dati inglobassero tutto il periodo dell'evento. Ad esempio, per un GP iniziato alle 14 (ora locale) e durato meno di due ore, è stata considerata la fascia oraria 13–16; per un GP iniziato alle 14 (ora locale) e durato più di due ore, è stata considerata la fascia oraria 13–17. In seguito, abbiamo associato ad ogni giro di ogni gran premio le condizioni della fascia di tempo corrispondente. Ad esempio, ad un giro iniziato alle ore 14:52 sono state associate le condizioni metorologiche rilevate dalle stazioni alle 14:50, e per un giro iniziato alle 14:56 quelle delle 14:55.

Tabella 4: Descrizione delle variabili d'interesse presenti nel dataset Weather 2014-2017, creato a partire dai dati di Visual Crossing.

#	Nome della variabile	Tipo della variabile	Descrizione della variabile
1	name	carattere	Nome del circuito/località.
2	dateTime	data	Data e ora delle rilevazioni.
3	Temperature	numerica	Media tra temperatura minima e massima.
4	windChill	numerica	Temperatura percepita a causa del vento.
5	precipitation	numerica	Quantità in mm di pioggia caduta nell'arco dei 5 minuti considerati.
6	windSpeed	numerica	Velocità del vento in km/h.
7	relativeHumidity	numerica	Umidità relativa percentuale.
8	conditions	carattere	Sintesi delle condizioni meteo.

La scelta di una fascia oraria di 5 minuti, e non più ristretta, è stata fatta perché la qualità delle rilevazioni meteorologiche effettuate dagli strumenti delle centraline risultava migliore — in termini di sensibilità — per dati aggregati. Sfortunatamente, per certe stazioni addirittura l'aggiornamento meteo è avvenuto solo di ora in ora, quindi per i gran premi corrispondenti si sono ottenute condizioni meteo identiche per molte tornate. Per determinare l'orario di inizio di ogni gran premio si è fatto riferimento al sito ufficiale Formula Uno, e per apportare piccole modifiche o correzioni ai dati (sempre dipendenti, appunto, dalla sensibilità degli strumenti) sono state consultate la cronache live del sito FormulaPassion (`https://www.formulapassion.it/`).

Per i Campionati 2018 e 2019 si sono invece reperiti i dati derivanti dalla telemetria ufficiale F1, già relativi ad ogni giro percorso. L'estrazione di questi dati è stata realizzata tramite l'API FastF1 con l'ausilio di codice Phyton. Anche in questo caso sono state fatte manualmente alcune correzioni laddove necessario (ad esempio, per cambiare i valori della variabile categoriale Rainfall — si veda il codice nell'Appendice per maggiori dettagli). Nelle Tabelle 4 e 5 sono descritte le variabili dei due datasets "Weather". Si precisa che la descrizione di alcune variabili — quelle con molti valori mancanti, o non di interesse per le analisi poiché relative, ad esempio, alla neve — è stata omessa dalla Tabella 4.

Tabella 5: Descrizione delle variabili presenti nel dataset Weather 2018-2019, costruito grazie all'API FastF1.

#	Nome della variabile	Tipo della variabile	Descrizione della variabile
1	LapNumber	numerica	Numero del giro.
2	AirTemp	numerica	Temperatura dell'aria in gradi Celsius.
3	Humidity	numerica	Umidità relativa percentuale.
4	Pressure	numerica	Pressione atmosferica in mb (hPa).
5	Rainfall	categoriale	Assume valore 1 se durante il giro piove.
6	TrackTemp	numerica	Temperatura dell'asfalto in gradi Celsius.
7	WindSpeed	numerica	Velocità del vento in m/s.

2 Analisi esplorative

Nel corso di questo capitolo si vedranno alcune analisi esplorative dei datasets descritti al capitolo precedente. Al fine di garantire una migliore comprensione di queste analisi, viene proposta prima di tutto una sintesi delle squadre e dei piloti che hanno partecipato ai Campionati del periodo di riferimento (2014-2019). Successivamente, si propongono grafici a barre, di dispersione e boxplot per illustrare alcune tra le informazioni più interessanti che è possibile derivare dai datasets, e si mostrano i risultati della stima di un modello di regressione logistica costruito specificatamente per il Gran Premio di Monte-Carlo.

2.1 Team, piloti e gran premi dal 2014 al 2019

In Tabella 6 sono riportati team, periodo di attività del team e piloti dei Campionati dal 2014 al 2019. Si sottolinea che nel corso di uno stesso Campionato alcuni piloti hanno gareggiato per scuderie differenti: ad esempio, Carlos Sainz ha cominciato la stagione 2017 con la Toro Rosso ed è poi passato alla Renault; Pierre Gasly ha corso le prime 12 gare del 2019 in Red Bull per poi accasarsi alla Toro Rosso.

Tabella 6: Piloti e team di Formula 1 dal 2014 al 2019.

Team	Periodo di attività	Piloti
Mercedes AMG	2014-2019	Hamilton, Rosberg (2014-2016), Bottas (2017-2019)
Scuderia Ferrari	2014-2019	Alonso (2014), Raikkonen (2014-2018), Vettel (2015-2019), Leclerc (2019)
Red Bull Racing	2014-2019	Vettel (2014), Ricciardo (2014-2018), Verstappen (2016-2019), Kvyat (2015-2016), Albon (2019), Gasly (2019)
McLaren F1 Team	2014-2019	Magnussen (2014-2015), Alonso (2015-2018), Button (2014-2017), Vandoorne (2016-2018), Sainz (2019), Norris (2019)
Scuderia Toro Rosso	2014-2019	Vergne (2014), Kvyat (2014, 2016-2017, 2019), Sainz (2015-2017), Verstappen (2015-2016), Hartley (2017-2018), Gasly (2017-2019), Albon (2019)
Force India (nel 2019 Racing Point)	2014-2019	Perez (2014-2019), Hulkenberg (2014-2016), Ocon (2017-2018), Stroll (2019)
Williams	2014-2019	Massa (2014-2017), Bottas (2014-2016), Di Resta (2017), Stroll (2017-2018), Sirotkin (2018), Kubica (2019), Russell (2019)
Sauber (dal 2018 Alfa-Romeo Sauber)	2014-2019	Sutil (2014), Gutierrez (2014), Nasr (2015-2016), Ericsson (2015-2018), Wehrlein (2017), Leclerc (2018), Raikkonen (2019), Giovinazzi (2017, 2019)
Renault (nel 2014 e 2015 Lotus)	2014-2019	Grosjean (2014-2015), Maldonado (2014-2015), Magnussen (2016), Palmer (2016-2017), Hulkenberg (2017-2019), Sainz (2017-2018), Ricciardo (2019)
Haas	2016-2019	Grosjean (2016-2019), Gutierrez (2016), Magnussen (2017-2019)
Caterham	2014	Ericsson, Lotterer, Stevens
Marussia-Manor	2014-2016	Chilton (2014), Bianchi (2014), Stevens (2015), Rossi (2015), Mehri (2015), Wehrlein (2016), Haryanto (2016), Ocon (2016)

I gran premi di Formula Uno si compongono di un numero di giri legato alla lunghezza del circuito: si passa dai 44 di Spa-Francorchamps ai 78 di Monte-Carlo, e hanno una durata massima di 2 ore effettive (le sospensioni causate dalle bandiere rosse non vengono conteggiate). In ogni caso l'evento, nel complesso, non può durare più di 4 ore.

Nel periodo tra il 2014 e il 2019 si sono svolti 121 gran premi. Nel corso di 65 gare (53.71% del totale) è stata dispiegata dalla direzione gara almeno una volta la Safety Car; dal 2015 al 2019 è stato invece indetto almeno un regime di Virtual Safety Car in 28 gran premi (27.45% del totale).

2.2 Statistiche di sintesi per piloti e gran premi

Intuitivamente, e come già specificato nell'introduzione, partire da una buona posizione in griglia è di grande aiuto per un buon esito della gara. Sulla carta, questo è vero specialmente in circuiti stretti e tortuosi dove il sorpasso è difficile. Ad esempio, molto spesso si dice che nel Gran Premio di Monte-Carlo fare il giro più veloce in qualifica aumenta considerevolmente le probabilità di vittoria finale. L'analisi congiunta del dataset Race e del dataset Qualifying fornisce un quadro della situazione leggermente diverso: si scopre che in sole 3 occasioni su 6 i polemen (Nico Rosberg nel 2014, Daniel Ricciardo nel 2018 e Lewis Hamilton nel 2019) hanno vinto a Monte-Carlo. La proporzione di successi avvenuta partendo dalla pole position, trascurando il dato relativo al GP di Francia (disputatosi solo nel 2018 e nel 2019) è particolarmente elevata per i circuiti del Canada, del Brasile e di Abu Dhabi. Al contrario, com'è possibile osservare in Figura 2, l'Australia è una pista dove il poleman ha vinto in una sola occasione su sei (Hamilton, nel 2015). È opportuno sottolineare che comunque, come si vedrà anche nel Capitolo 3, a Monte-Carlo partire tra i primi porta spesso ad arrivare tra i primi. A tal proposito, possiamo costruire un modello di regressione logistica assumendo come variabile risposta la dicotomica:

$$Y = \begin{cases} 1, & \text{se il pilota ha terminato il GP sul podio} \\ 0, & \text{altrimenti,} \end{cases}$$

e come unica covariata la posizione di partenza, per capire quanto questa incida sul risultato del gran premio che si corre nel Principato. Il modello ha la seguente, semplice struttura:

$$g(\pi_i) = \beta_0 + \beta_1 x_i$$

con $g(\cdot)$ funzione di legame logit, π_i probabilità che Y = 1 per l'unità statistica i, x_i posizione di partenza del pilota e β_0, β_1 coefficienti di regressione [7].

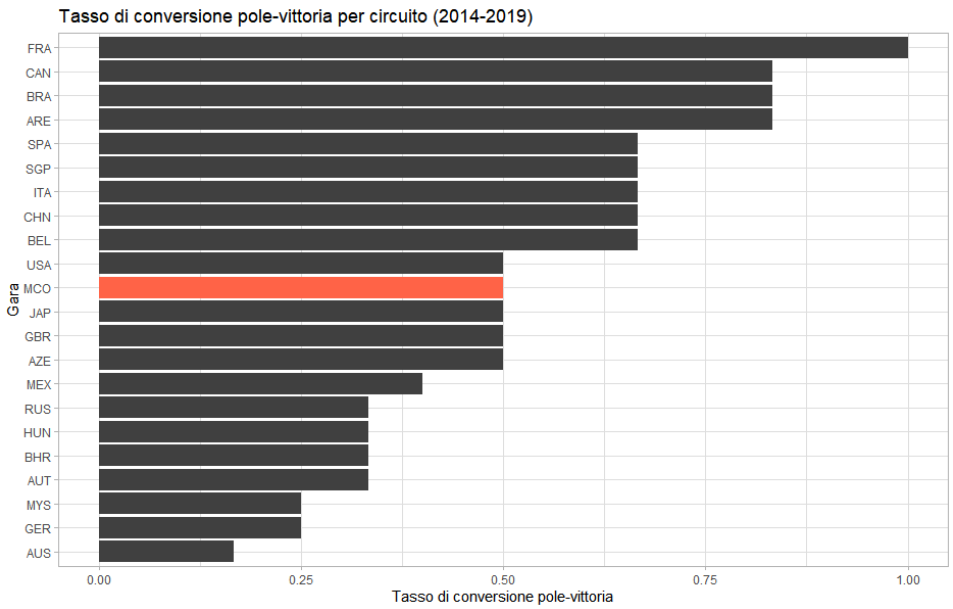

Figura 2: Proporzione di vittorie partendo dalla pole position dal 2014 al 2019, per circuito. In arancione il dato relativo a Monte-Carlo.

Dalla stima dei parametri del modello, effettuata con tutti i dati a disposizione per questa particolare gara, emerge che il valore del coefficiente β_1 è pari a -0.93. Ciò significa che la quota di "successo", ovvero il rapporto tra la probabilità di un arrivo a podio e la probabilità di un arrivo in una posizione uguale o peggiore della quarta, viene moltiplicata per un fattore $exp(-0.93) \approx 0.39$ all'aumentare unitario della posizione di partenza. Nella Tabella 7 sono riportate le stime puntuali delle probabilità che $Y = 1$ per le prime 10 posizioni di partenza sulla griglia, dalle quali si capisce come partire dalla sesta posizione in poi offra ben poche speranze di arrivare sul podio (la probabilità che ciò accada è inferiore al 10%).

Una proporzione analoga a quella rappresentata in Figura 2 (ossia quella del numero di vittorie ottenute da un pilota che partiva dalla pole position, sul totale dei GP disputati nel periodo di riferimento) può essere calcolata riferendosi, anziché ai singoli circuiti, ai piloti. Un alto rapporto tra il numero di vittorie arrivate partendo dalla pole position e il numero di prime piazze conquistate indica, tendenzialmente, che il pilota è in grado di sopportare al meglio la pressione di condurre la gara in testa senza commettere errori, e che la sua vettura offre — oltre ad ottime prestazioni sul giro secco — anche un buon passo gara. Per determinare allora quali piloti hanno saputo concretizzare maggiormente le loro pole position in vittorie si possono manipolare opportunamente i datasets Race e Qualifying. In Figura 3 viene rappresentato mediante un diagramma a barre il suddetto rapporto. Il colore

Tabella 7: Probabilità di arrivo a podio nel gran premio di Monte-Carlo per le prime 10 posizioni di partenza.

Posizione di partenza	Probabilità
1	0.91
2	0.79
3	0.60
4	0.37
5	0.19
6	0.084
7	0.035
8	0.014
9	0.0056
10	0.0022

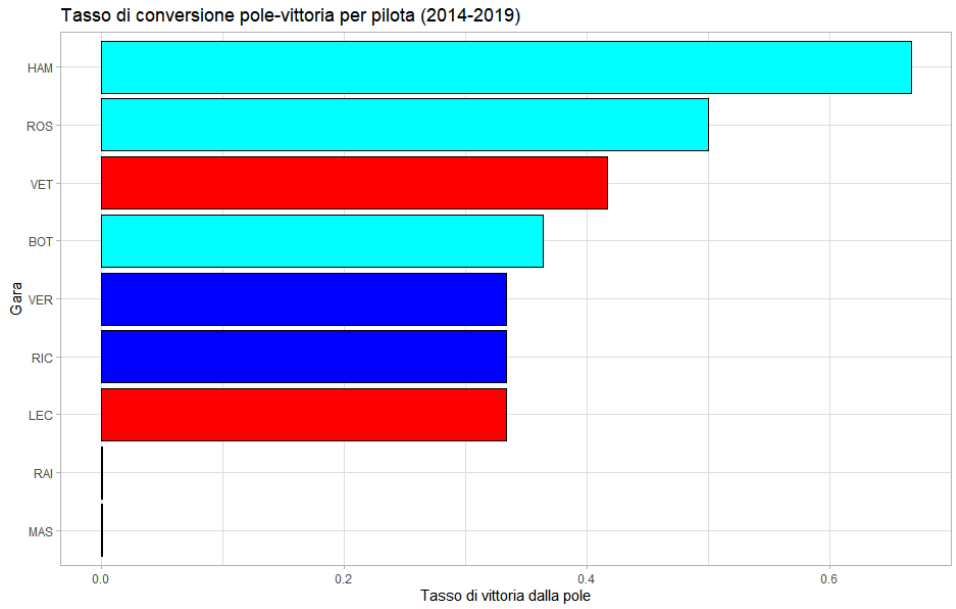

Figura 3: Proporzione di vittorie partendo dalla pole position dal 2014-2019, per pilota.

delle barre richiama quello del team con cui i piloti hanno corso per la maggior parte del periodo di riferimento (sarà così anche per i grafici nelle Figure 4 e 8). Il pilota britannico della Mercedes Lewis Hamilton si pone in testa con un rapporto tra vittorie dalla pole e vittorie elevatissimo (pari al 66%). Seguono Nico Rosberg, Sebastian Vettel, Valtteri Bottas, Max Verstappen, Daniel Ricciardo e Charles Le-

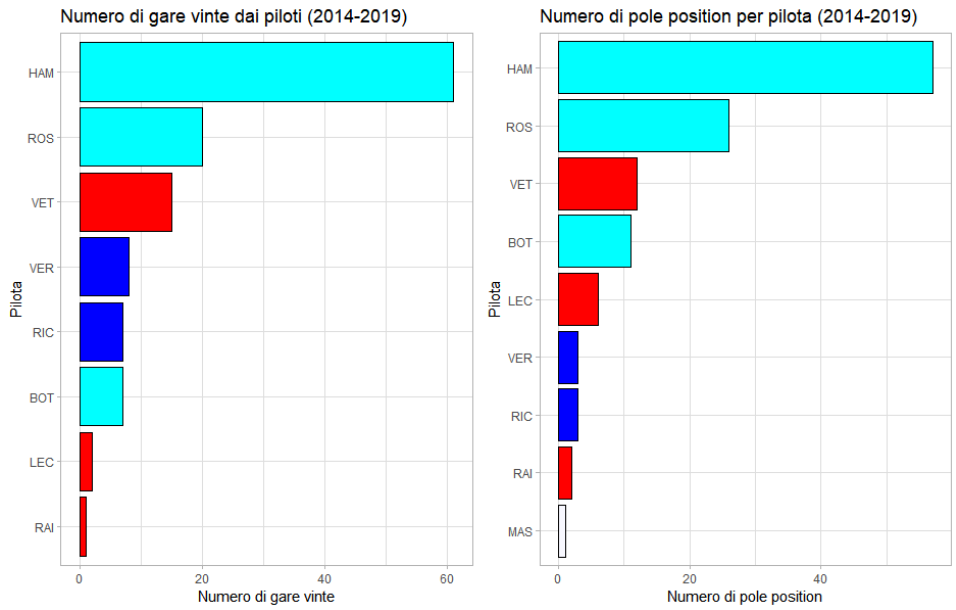

Figura 4: Conteggio di gare e pole position conquistate dai piloti, 2014-2019.

clerc. Felipe Massa, su Williams, e Kimi Raikkonen, su Ferrari, si sono qualificati in testa rispettivamente nei gran premi d'Austria del 2014 (Massa), Monte-Carlo 2017 e Italia 2018 (Raikkonen); tuttavia, in queste occasioni non sono stati in grado di vincere e il loro tasso di vittoria partendo dalla prima casella risulta perciò pari a 0.

Nel conteggio totale di gare vinte nel periodo 2014-2019 (rappresentato in Figura 4), Lewis Hamilton — detentore dei Campionati 2014, 2015, 2017, 2018 e 2019 (si veda la Tabella 1) — non ha rivali: sono ben 62 i trionfi del pilota di punta Mercedes. Risulta come in 121 gran premi vi siano stati solamente 8 differenti vincitori: oltre a Hamilton figurano Rosberg, vincitore del Campionato 2016 (Mercedes), Vettel (Ferrari), Verstappen (Red Bull Racing), Ricciardo (Red Bull Racing), Bottas (Mercedes), Leclerc (Ferrari) e Raikkonen (Ferrari). Inoltre, i piloti citati appartengono a tre squadre soltanto, sulle dodici che hanno partecipato ai Campionati.

Anche per quanto riguarda il numero di pole position conquistate, sempre visibile in Figura 4, Hamilton svetta sugli avversari con 57 primi posti in griglia. Seguono Rosberg e Vettel. Massa è stato l'unico pilota di un team diverso da Mercedes, Ferrari e Red Bull ad aver conquistato una pole position. Notevole il dato relativo a Leclerc, in grado di aggiudicarsi ben 7 prime posizioni tutte nel 2019, al suo primo anno in Ferrari (nel 2018, all'esordio, correva con la meno competitiva Alfa Romeo Sauber).

Tabella 8: Gare disputate in condizioni di bagnato.

Campionato	Gara
2014	Ungheria, Giappone
2015	Gran Bretagna, Stati Uniti
2016	Monte-Carlo, Gran Bretagna, Brasile
2017	Cina, Singapore
2018	Germania
2019	Germania

2.3 I dati meteorologici

L'analisi delle condizioni meteorologiche durante i weekend di gara è fondamentale in Formula Uno. In particolare, la pioggia è il fattore che più di ogni altro può condizionare una gara. Concentrando l'attenzione esclusivamente sui gran premi, nel periodo considerato solamente 11 gare su 121 (9.1%), riportate in Tabella 8, si sono disputate in condizioni di pista bagnata. Si effettua ora un'analisi dei dati inerenti alle condizioni meteorologiche dei gran premi dal 2018 al 2019, potendo contare sulle informazioni della telemetria ufficiale F1. Si considera inizialmente il Campionato 2018, i cui dati sono contenuti nel dataset Weather 2018-2019. È possibile realizzare un grafico di dispersione tra la temperatura dell'aria e la temperatura dell'asfalto, rilevate giro dopo giro per tutti i gran premi in questione. Il grafico, visibile in Figura 5, mostra la presenza di un trend lineare appena abbozzato: al crescere della temperatura dell'aria cresce anche quella dell'asfalto, ma non in maniera netta. Il coefficiente di correlazione di Spearman è positivo e risulta pari a 0.36. Sono ben evidenti, in generale, i "clusters" relativi alle singole gare. Vogliamo a questo punto stabilire quali gare si corrono con elevate temperature dell'asfalto, prendendo in esame i dati più recenti a disposizione, ovvero quelli relativi alla stagione 2019. Il grafico a barre riportato in Figura 6 testimonia come la temperatura media dell'asfalto sia stata particolarmente alta durante le gare svolte in Francia, Canada e Austria. In prima battuta ciò sembra sorprendente, ma in realtà bisogna considerare che i gran premi sopra citati si svolgono nei mesi di giugno e luglio in pieno giorno, a differenza di gran premi quali quelli di Abu Dhabi, Bahrain o Singapore dove la temperatura dell'aria è tendenzialmente più calda, ma si corre al tramonto o addirittura di notte. In Germania (unica gara del Campionato in questione durante la quale ha piovuto), Gran Bretagna e Cina si sono registrate invece le temperature medie dell'asfalto più basse. È noto che la temperatura dell'asfalto è uno dei fattori in grado di condizionare l'usura degli pneumatici. Quando il battistrada genera eccessivo calore nel rotolamento, la gomma può incorrere nel cosiddetto "*bliste-*

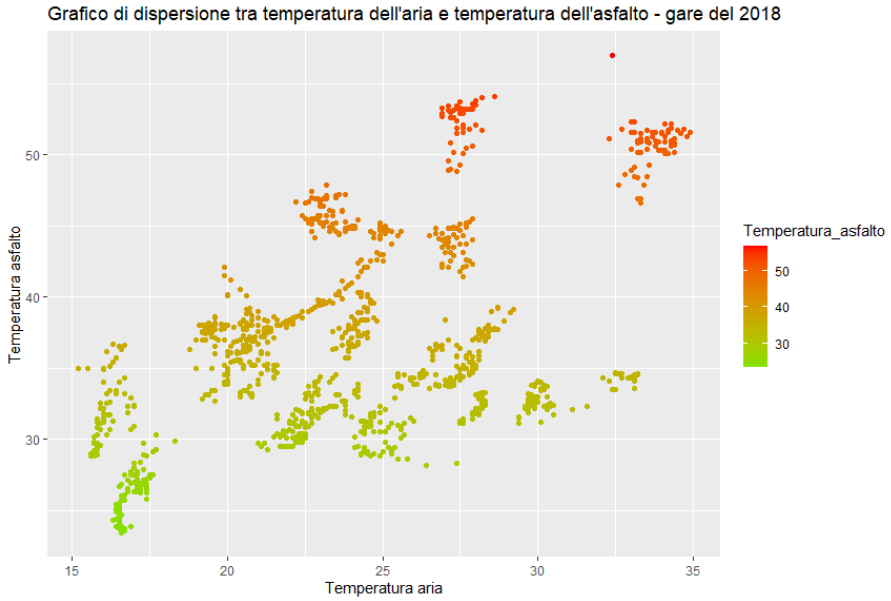

Figura 5: Scatterplot tra temperatura dell'aria e dell'asfalto, Campionato 2018.

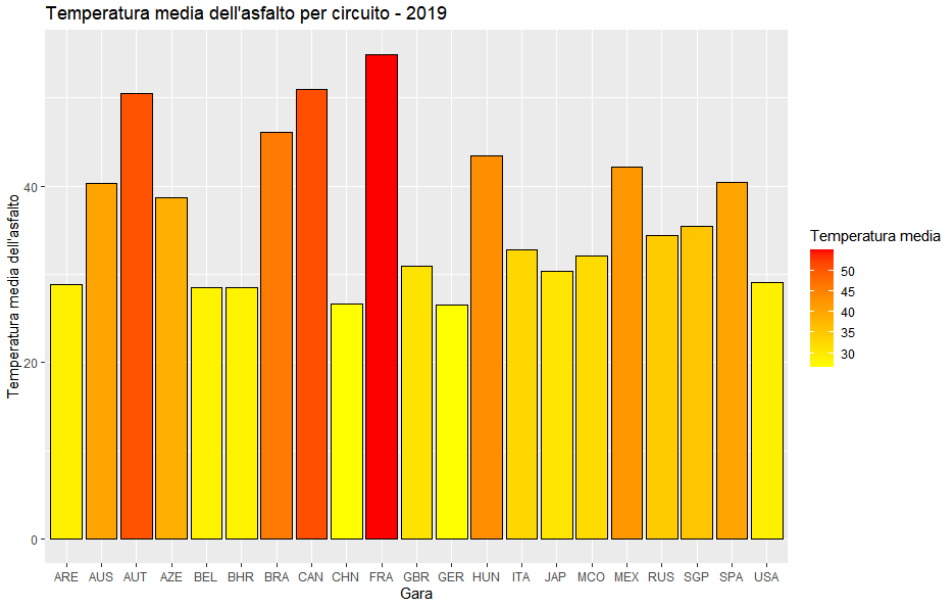

Figura 6: Temperatura media dell'asfalto per cicruito nei gran premi del 2019.

ring", ovvero nella formazione di vesciche tra la carcassa e la superficie a contatto con il suolo, che con il passare dei giri tendono a esplodere generando dei piccoli

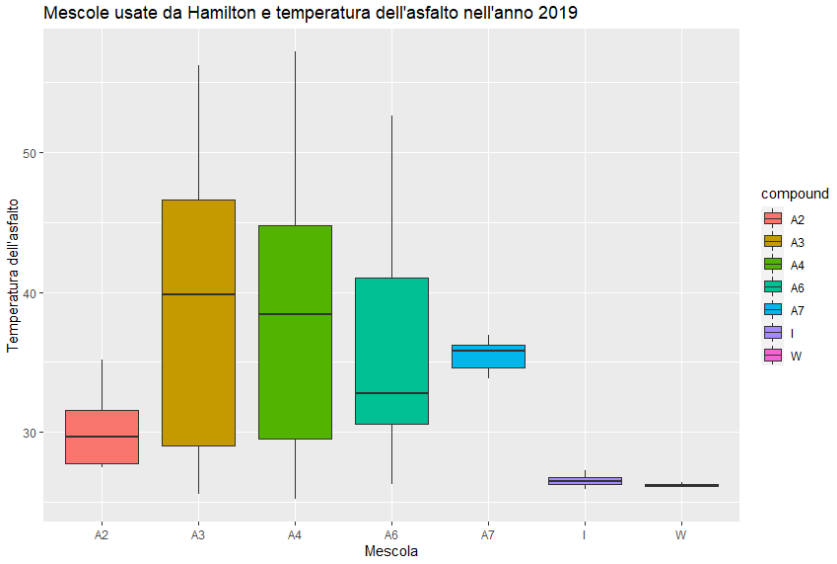

Figura 7: Boxplots della temperatura dell'asfalto in funzione del compound di pneumatici utilizzato da Hamilton nel 2019.

crateri. È lecito aspettarsi che a temperature dell'asfalto più alte corrispondano mescole di pneumatici più duri, in grado di sopportare maggiormente il fenomeno del blistering.

Per verificare questa affermazione a partire dai dati del dataset Race si prendono in esame, per tutti i giri del 2019, le gomme scelte da Lewis Hamilton (unico pilota in grado di portare a termine tutte le gare di questa stagione). I boxplots in Figura 7 non vanno del tutto a supporto dell'affermazione precedente, secondo la quale in condizioni di asfalto caldo sono da utilizzare principalmente mescole dure di pneumatici. Se è vero che la mescola più morbida è stata usata esclusivamente con temperature inferiori ai 40 gradi, si nota come quella più dura tra le mescole a disposizione nel 2019 (la A2) sia stata adoperata esclusivamente in condizioni di asfalto freddo (e non caldo, come si era supposto inizialmente), mentre è apprezzabile la duttilità delle mescole A3, A4 e A6 nell'adattarsi ad un range molto ampio di temperatura del tracciato. È evidente allora che la scelta degli pneumatici è dettata da innumerevoli altri fattori, legati alla strategia, alle eventuali Safety Car, allo stile di guida del pilota e all'assetto ("*setup*") della vettura. In questo senso, l'ottimo bilanciamento della Mercedes W10 del 2019 ha saputo consentire un basso degrado degli pneumatici in praticamente ogni condizione di pista.

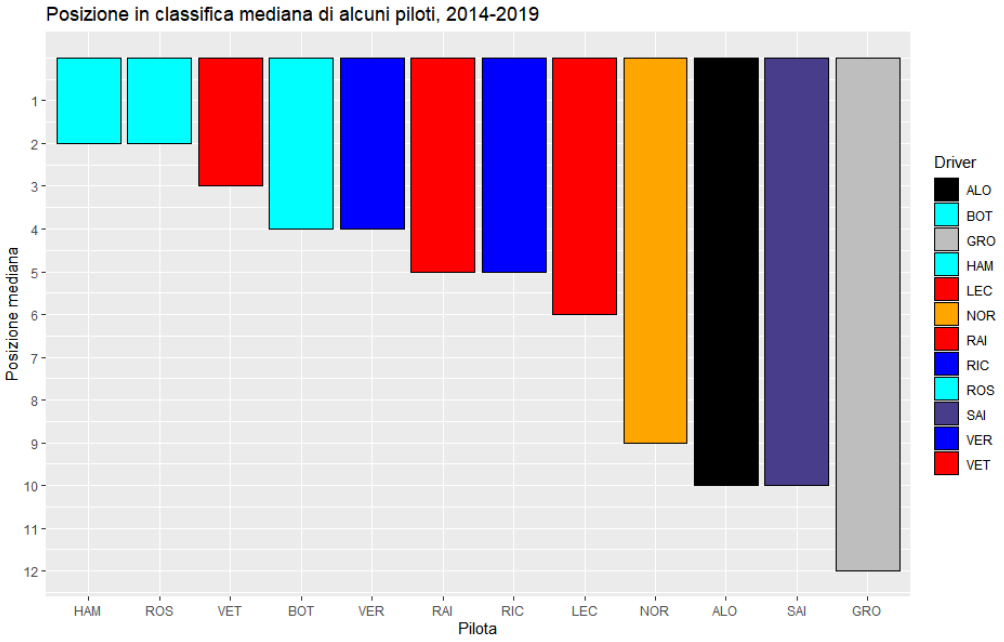

Figura 8: Posizione mediana occupata dai piloti durante i giri dei gran premi dal 2014 al 2019.

2.4 Posizione mediana in classifica dei piloti

L'analisi del dataset Race permette, infine, di determinare qual è stata la posizione mediana occupata dai piloti durante i loro giri. Idealmente, si immagini di visualizzare uno in fila all'altro i numeri corrispondenti alle posizioni occupate da un pilota nel corso delle gare, giro dopo giro, ordinati in maniera crescente, e di isolare il numero centrale. La Figura 8 riporta le posizioni mediane di 12 piloti (da sinistra a destra: Hamilton, Rosberg, Vettel, Bottas, Verstappen, Raikkonen, Ricciardo, Leclerc, Lando Norris, Fernando Alonso, Sainz e Romain Grosjean) ed evidenzia ancora una volta il netto predominio della Mercedes nei confronti delle scuderie avversarie. Per almeno il 50% dei giri che hanno corso, Hamilton e Rosberg o conducevano la gara, o erano in seconda posizione. Al contrario, piloti come Alonso e Sainz hanno trascorso almeno il 50% delle loro gare in una posizione peggiore della nona.

3 Modelli per gran premi di Formula Uno

Ci si appresta ad analizzare ora la dinamica dei gran premi di Formula Uno da un punto di vista prettamente modellistico, presentando metodi via via più complessi: regressione robusta, alberi di classificazione e modelli additivi generalizzati a effetti misti (*Generalized Additive Mixed Models*, GAMM).

3.1 Regressione robusta e tempo sul giro

Il tempo sul giro dei piloti in un gran premio segue un trend discendente: è piuttosto elevato nei giri appena dopo la partenza e più basso nei giri conclusivi. Ciò avviene principalmente in virtù di due fattori. La pista diventa via via più "gommata" con il passare dei giri: a causa del degrado degli pneumatici, i trucioli di gomma persi dalle monoposto tendono ad uniformare le piccole asperità dell'asfalto, generando così una pista più aderente. Inoltre, all'inizio della corsa il serbatoio delle vetture è pieno di benzina; durante la gara, visto che a partire dal Campionato 2010 non si possono effettuare rifornimenti ma solo cambi gomme, la benzina viene consumata e la vettura si alleggerisce. La conseguenza è appunto un abbassamento graduale del tempo sul giro. Si può allora modellare in prima istanza il tempo sul giro tramite un modello di regressione lineare semplice, dove la risposta è spiegata in funzione del numero del giro (covariata).

Per descrivere il fenomeno abbiamo considerato il Gran Premio del Bahrain del 2015 e i tempi sul giro del pilota Fernando Alonso (McLaren-Honda), rappresentati in Figura 9. Poiché in Formula Uno, a differenza di quanto accade in altri sport motoristici come la NASCAR, la partenza avviene a vetture ferme, è stato eliminato il tempo relativo al primo giro, che risulterebbe altrimenti troppo alto in confronto agli altri. Si può constatare che durante il gran premio Alonso ha realizzato tre "*stints*", ossia tre parti di gara separate da un pit stop, con fermate ai giri 15 e 37. Il tempo sul giro è più basso nel terzo stint rispetto al secondo, e più basso nel secondo rispetto al primo. Sono evidenti gli outliers corrispondenti ai giri in cui è stata effettuata la sosta. È da notare come il pit stop non influenzi soltanto il tempo del giro in cui avviene, ma anche di quello precedente, poiché il pilota comincia ad entrare in pit-lane (la corsia dei box) e a rallentare prima di tagliare il traguardo.

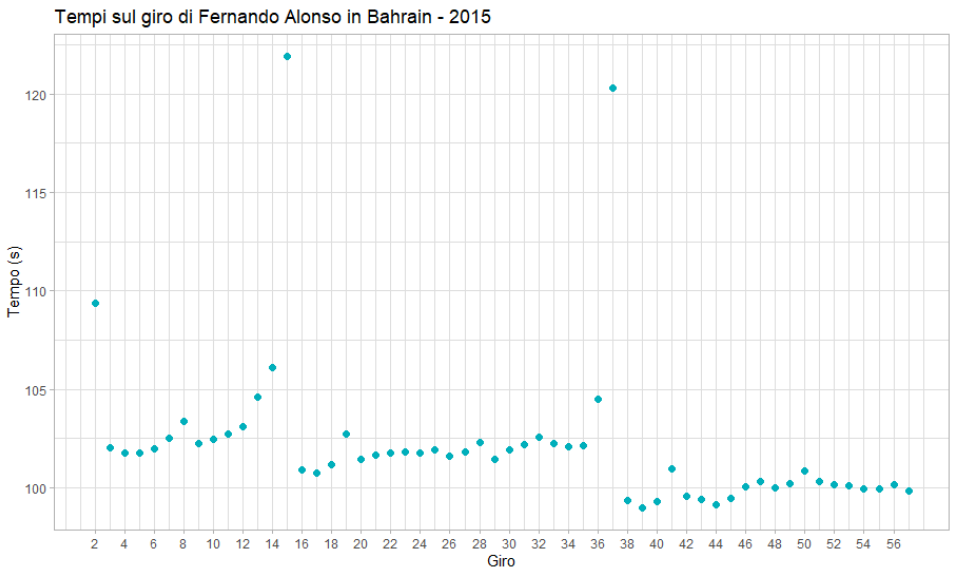

Figura 9: Tempi fatti registrare da Alonso (McLaren-Honda) durante il GP di Bahrain 2015.

Abbiamo anche analizzato i tempi sul giro in un gran premio con rifornimenti, prendendo come riferimento quelli siglati da Michael Schumacher nello stesso circuito, nel 2004.

Nel grafico che li riporta, in Figura 10, sono stati eliminati i tempi del primo giro (influenzato dalla procedura di partenza) e dell'ultimo giro (quando Schumacher ha rallentato di proposito, per farsi avvicinare dal compagno di squadra Rubens Barrichello). In generale, i tempi del tedesco sono nettamente più veloci rispetto a quelli di Alonso — ma questo deriva dal fatto che le vetture del 2004 erano più veloci rispetto a quelle del 2015. In questo contesto è d'interesse constatare come manchi, a differenza del grafico di Figura 9, il miglioramento dovuto al calo del livello della benzina. Addirittura, il tempo di Schumacher — piazzatosi primo in quell'occasione — tende ad aumentare leggermente nel corso della gara, probabilmente in virtù dei numerosi doppiaggi compiuti dalla sua Ferrari. Dato che ci si attende che alcuni valori dei tempi sul giro di Alonso e Schumacher si discostino notevolmente dalla retta di regressione — precisamente, quelli corrispondenti ai giri in cui i piloti hanno fatto una sosta ai box — la scelta migliore per interpolare il pattern dei dati è quella di implementare una tecnica di regressione robusta.

Nello specifico, il metodo "*Least Trimmed Squares*" (LTS) si basa sulla minimizzazione di una porzione $q < n$ dei quadrati dei residui ordinati in ordine crescente. Una tecnica di tale genere sopporta un certo numero di osservazioni molto distante dalla media senza fornire stime distorte dei parametri; detto in altre parole, il

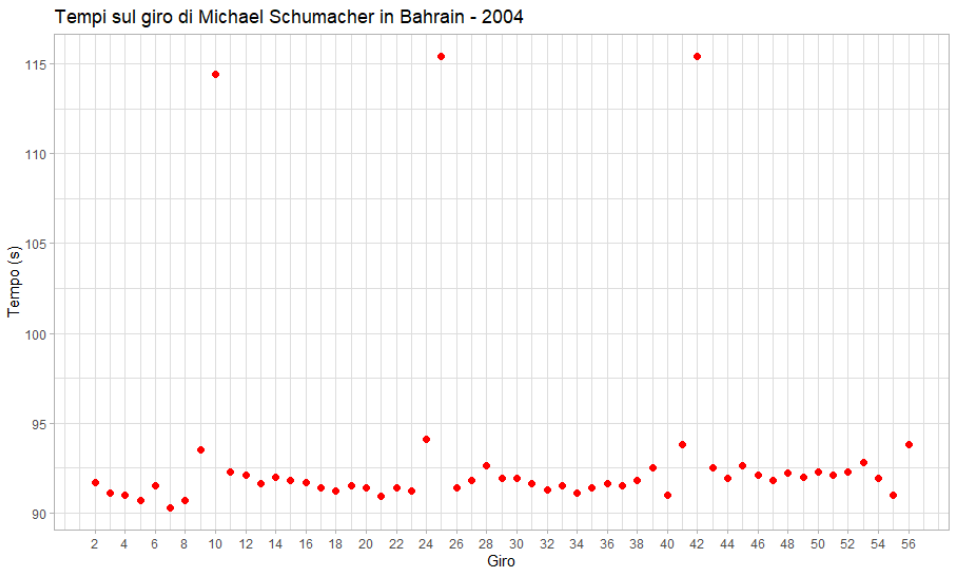

Figura 10: Tempi fatti registrare da Schumacher (Ferrari) durante il GP di Bahrain 2004.

metodo è robusto rispetto a osservazioni anomale come quelle dei tempi sul giro di un gran premio. La scelta ottimale di q va effettuata ricercando un trade-off tra "efficienza" e distorsione.

Con $q = n$, ossia utilizzando il metodo dei minimi quadrati ordinari, la retta interpolante dei tempi sul giro dei due piloti basata su un modello di regressione lineare lascia alquanto a desiderare: essa infatti risulta troppo "elevata" poiché "attratta" dagli outliers (si veda Figura 11).

Anche l'interpolazione del pattern dei dati utilizzando il metodo non-parametrico della regressione locale (LOESS, acronimo di LOcally Estimated Scatterplot Smoothing) è palesemente inadatta, come si desume dall'analisi presentata in Figura 11, in quanto la curva liscia — una polinomiale a tratti, dove i polinomi si congiungono con soluzione di continuità — tende a seguire l'andamento degli outliers. Un risultato molto migliore rispetto ai due appena visti, invece, si ottiene implementando la regressione robusta, com'è chiaro dall'analisi di Figura 12.

Nella Tabella 9 sono riportati i valori puntuali di intercetta e coefficiente angolare delle rette di regressione robusta ottenute per Alonso e altri quattro piloti che hanno partecipato al Gran Premio del Bahrain 2015, accompagnati dai rispettivi intervalli di confidenza al 95%. È possibile notare come, indipendentemente dalla posizione d'arrivo (e quindi, indipendentemente dalla competitività della vettura) le rette di tutti i piloti in questione abbiano un coefficiente angolare negativo, a conferma del fatto che, in ogni caso, durante i gran premi tutti i piloti tendono a diventare più

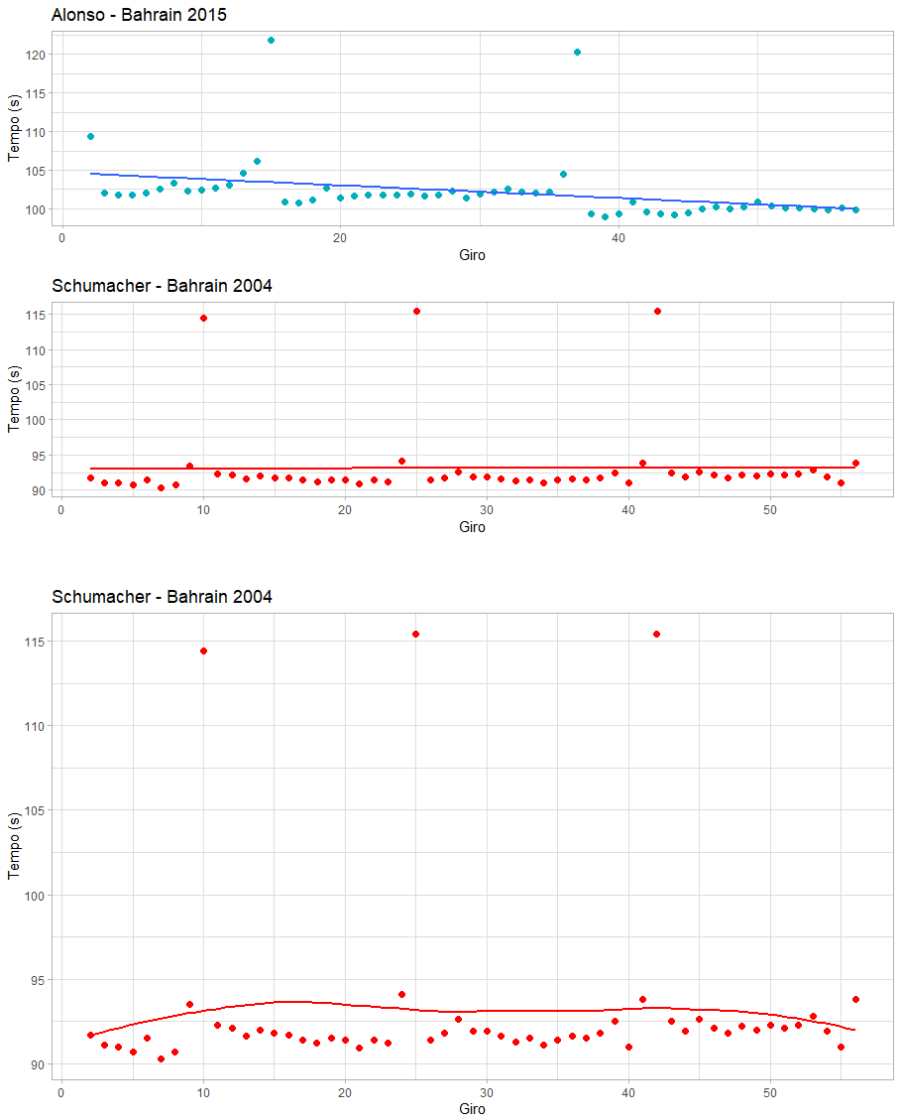

Figura 11: In alto: interpolazione dei tempi di Alonso e Schumacher mediante retta di regressione lineare. In basso: interpolazione dei tempi di Schumacher mediante regressione locale.

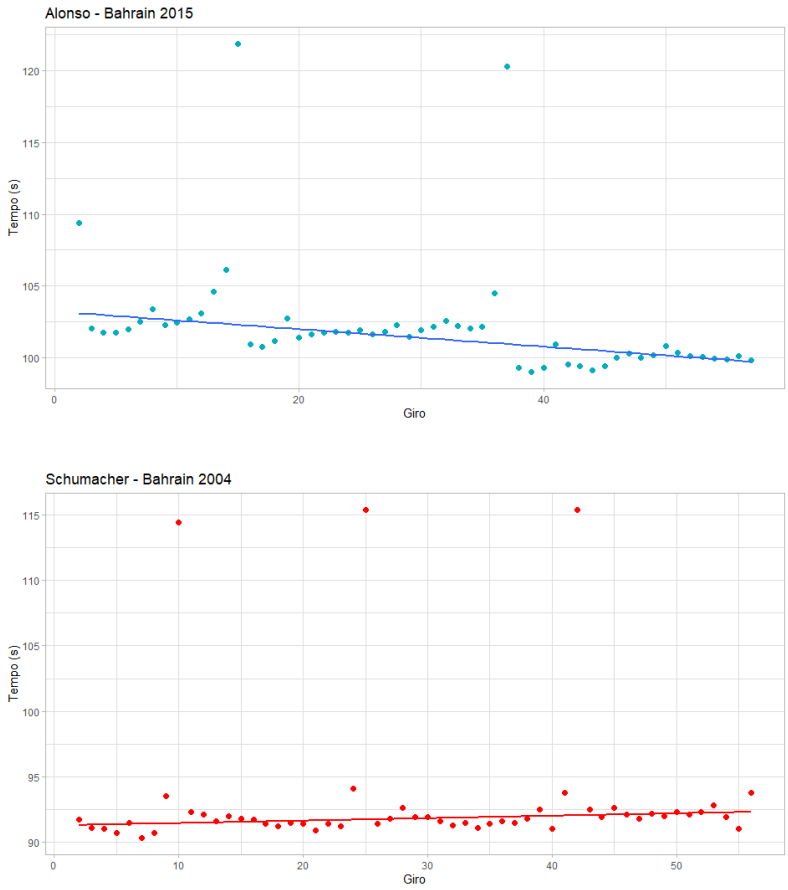

Figura 12: Regressione robusta sui tempi di Alonso (in alto) e Schumacher (in basso).

veloci grazie al consumo della benzina. Nel caso di Alonso, l'interpretazione dei dati di Tabella 9 è la seguente: il tempo "base" del pilota con serbatoio pieno è di 103.21 secondi; al giro 20 (ad esempio), senza tener conto di altri fattori come usura o mescola delle gomme, ci si attende un tempo sul giro di circa un secondo e due decimi più basso. In generale, per ogni giro percorso c'è un decremento del tempo di cinque centesimi di secondo rispetto al passaggio precedente.

3.2 Albero di classificazione

Si desidera, a questo punto, trovare un modo per prevedere i risultati di un gran premio, cioè trarre dai dati a disposizione un'indicazione sulla posizione d'arrivo di un pilota. Un possibile approccio è quello di utilizzare una tecnica non parametrica

Tabella 9: Regressione robusta per i tempi sul giro di alcuni piloti al Gran Premio del Bahrain 2015. Stime puntuali dei coefficienti e intervalli di confidenza al 95%.

		Intercetta		Coefficiente angolare	
Pilota	Posizione d'arrivo	Stima	Intervallo	Stima	Intervallo
Kimi Raikkonen	2	101.44	[100.78, 102.10]	-0.067	[-0.086 -0.047]
Nico Rosberg	3	100.60	[100.07, 101.11]	-0.037	[-0.052, -0.022]
Daniel Ricciardo	6	101.52	[101.11, 101.93]	-0.04	[-0.052, -0.028]
Fernando Alonso	11	103.21	[102.62, 103.80]	-0.06	[-0.079, -0.044]
Pastor Maldonado	15	102.32	[101.74, 102.89]	-0.07	[-0.09, -0.05]

denominata "albero di classificazione" (*Classification Tree*, CT). L'idea alla base di un albero di classificazione è quella di approssimare una funzione $p(x) = P(Y = 1|X = x)$ utilizzandone una a gradini, con i "lati" dei gradini paralleli agli assi coordinati [1]. L'approssimazione di $p(x)$ avviene dunque mediante

$$\hat{p}(x) = \sum_{j=1}^{J} P_j \cdot \mathbb{1}(x \in R_j),$$

dove $P_j \in (0, 1)$ e rappresenta la probabilità che $Y = 1$ nella regione R_j, e $\mathbb{1}(x \in R_j)$ è la funzione indicatrice della regione, che assume valore 1 se e solo se $x \in R_j$. La stima dei valori di P_j coincide con la frequenza relativa dei valori $y_i = 1$ che si trovano in R_j.

Le regioni R_j vanno determinate in base ad un qualche criterio obiettivo. Generalmente, quando si ha a disposizione un cospicuo numero di osservazioni, si procede alla crescita dell'albero su un insieme di dati di addestramento e si minimizza la devianza calcolata su un "*testing set*" comprendente la porzione di osservazioni a disposizione non utilizzate in fase di stima. In questo modo si controlla la complessità dell'albero, che altrimenti crescerebbe indefinitamente facendo corrispondere ad ogni valore osservato di x una regione. Nel caso dei CT, la devianza è, al netto di una costante, una media del grado di "impurità" delle foglie pesata con la numerosità delle foglie stesse, valutata con indici quali quello di Gini o di Shannon. Una foglia è impura quando i suoi elementi non sono omogenei rispetto alla variabile risposta.

Una volta costruita $\hat{p}(x)$, si perviene ad un grafico con struttura gerarchica, composto da un nodo radice dal quale si dipartono due "rami" collegati ad altri nodi. I nodi terminali sono chiamati foglie. Ogni nodo rappresenta una variabile in corrispondenza della quale è stato effettuato uno "*split*", cioè una suddivisione di una regione in "sotto-regioni". L'algoritmo di crescita di un albero di classificazione fa sì che il punto di suddivisione sia ricercato in corrispondenza della variabile che porta

ad una riduzione maggiore della devianza. Di conseguenza, le variabili utilizzate per gli splits possono anche essere considerate le più importanti nel determinare la variabile risposta. Oltre che per prevedere i risultati di un gran premio, allora, utilizzeremo l'albero per capire quali variabili sono importanti nel determinare quegli stessi risultati.

3.2.1 Dataset e metodo di costruzione del CT

In questa sezione applicheremo la tecnica dell'albero di classificazione alla previsione dell'arrivo di un pilota in "*top five*" (cioè, tra i primi cinque), fingendo di non conoscere i risultati del 2019, e utilizzando i dati dal 2014 al 2018 per allenare il modello. In questo caso, si assume come variabile risposta

$$Y = \begin{cases} 1, & \text{se il pilota ha terminato il GP in top five} \\ 0, & \text{altrimenti.} \end{cases}$$

Il vantaggio dell'utilizzo degli alberi di classificazione consiste nella possibilità di gestire contemporaneamente variabili esplicative sia di natura qualitativa, sia di natura quantitativa. Grazie ai CT, dunque, si possono utilizzare per la previsione molte delle informazioni a disposizione. Il dataset costruito, che chiamiamo dataset "*Tree*", attinge dal dataset Race (scuderia del pilota, età della gomma al termine della gara), dal dataset Qualifying (posizione di partenza) e dai dataset delle condizioni meteo (umidità relativa, temperatura dell'aria), e inoltre contiene le variabili dicotomiche descritte in Tabella 10. Per i piloti che non hanno preso parte alle Qualifiche, o che sono stati squalificati, è stata assunta come posizione di partenza la ventesima. Infine, sono stati convertiti in km/h i valori della velocità del vento dei dataset relativi ai Campionati 2018 e 2019 (che erano espressi in m/s). In pratica nel dataset Tree, accanto alle informazioni sul singolo pilota, si trovano variabili relative ai singoli gran premi, che risultano uguali per tutti coloro che hanno partecipato alla competizione.

Ai fini della classificazione si sarebbe potuto scegliere come variabile risposta anche FoP ("Finishes on Podium"). Tuttavia, sebbene il dataset Tree abbia una numerosità relativamente elevata, il numero di dati disponibili per la previsione di ogni singolo gran premio è limitato: l'insieme di addestramento si compone di un massimo di cinque gare soltanto per evento. Per evitare la creazione di un albero troppo sbilanciato, quindi, abbiamo considerato Top5. Di fatto, alcune prove empiriche hanno confermato la difficoltà nella previsione di un albero con variabile riposta FoP. I parametri di controllo dell'albero sono stati stabiliti manualmente in maniera tale da evitare l'overfitting (una classificazione troppo accurata e fedele ai

Tabella 10: Descrizione di alcune delle variabili presenti nel dataset Tree.

#	Nome della variabile	Significato del valore 1
1	rain	Durante il GP ha piovuto.
2	ScVsc	Durante il GP vi sono state fasi di Safety Car (SC) o di Virtual Safety Car (VSC)
3	isPoleman	Il pilota è partito dalla pole position.
4	FoP	"Finishes on Podium": Il pilota è finito sul podio
5	Top5	Il pilota è finito tra i primi 5.

dati, inadatta alla previsione di nuovi dati) e garantire comunque una sufficiente precisione nei risultati.

3.2.2 Risultati dell'albero di classificazione

I risultati dell'albero di classificazione sono riportati in Tabella 11, nella quale vengono affiancate la top five reale e la top five prevista (la sigla del pilota è verde in caso di previsione corretta, rossa in caso contrario). Tenendo conto dell'imprevedibilità che caratterizza ogni gran premio, questi risultati possono essere considerati soddisfacenti: si verifica che per ben 8 gran premi su 20 la previsione dei primi cinque classificati è totalmente corretta (senza tener conto dell'ordine di arrivo). Si tratta dei Gran Premi di Bahrain, Cina, Spagna, Monte-Carlo, Gran Bretagna, Singapore, Messico ed Emirati Arabi Uniti. Non abbiamo fatto una previsione per la gara svolta in Francia, poiché è stata introdotta nel calendario solamente l'anno prima (2018) e l'insieme di addestramento avrebbe potuto contare solamente su un evento.

Le maggiori difficoltà nella classificazione si riscontrano per i circuiti di Australia, Brasile e Germania. Per quanto riguarda il primo, ciò appare coerente con quanto scoperto nell'analisi esplorativa, dove si nota che in una sola occasione su sei il poleman ha vinto la corsa, e ciò è indice di una certa variabilità nei risultati (Figura 2). Inoltre, questo gran premio è il primo della stagione e spesso, a inizio Campionato, il vero potenziale delle squadre fatica ad emergere. Il Gran Premio di Germania del 2019 è stato invece condizionato dalla pioggia e da svariati incidenti. In Brasile sono successi alcuni eventi imprevedibili: a sei giri dalla fine i due piloti della Ferrari, Sebastian Vettel e Charles Leclerc, si sono scontrati e in seguito ritirati mentre occupavano il quarto ed il quinto posto; anche Valtteri Bottas si è ritirato, a venti giri dal termine, a causa di una perdita di pressione dell'olio; infine Lewis Hamilton, arrivato terzo sul traguardo, è stato penalizzato per un contatto avvenuto con Alexander Albon, e si è classificato settimo. Almeno tre piloti tra Hamilton, Bottas, Vettel e Leclerc avrebbero probabilmente terminato la gara tra i primi cin-

Tabella 11: Risultati dell'albero di classificazione sui GP del Campionato 2019. La dicitura "-A" che segue la sigla del GP indica la colonna con i risultati dell'albero.

AUS-A	AUS	BHR-A	BHR	CHN-A	CHN	AZE-A	AZE
RAI	BOT	HAM	HAM	HAM	HAM	BOT	BOT
NOR	HAM	BOT	BOT	BOT	BOT	HAM	HAM
BOT	VER	LEC	LEC	VET	VET	VET	VET
HAM	VET	VER	VER	VER	VER	RIC	VER
VER	LEC	VET	VET	LEC	LEC	GAS	LEC

SPA-A	SPA	MCO-A	MCO	CAN-A	CAN	AUT-A	AUT
HAM	HAM	HAM	HAM	VET	HAM	VER	VER
BOT	BOT	VER	VET	HAM	VET	LEC	LEC
VER	VER	VET	BOT	LEC	LEC	BOT	BOT
VET	VET	BOT	VER	GAS	BOT	MAG	VET
LEC	LEC	GAS	GAS	BOT	VER	HAM	HAM

GBR-A	GBR	GER-A	GER	HUN-A	HUN	BEL-A	BEL
HAM	HAM	VER	VER	VET	HAM	HAM	LEC
BOT	BOT	BOT	VET	GAS	VER	BOT	HAM
GAS	LEC	GAS	KVY	BOT	VET	VER	BOT
VER	GAS	HAM	STR	LEC	LEC	LEC	VET
LEC	VER	VET	SAI	VER	SAI	VET	ALB

ITA-A	ITA	SGP-A	SGP	RUS-A	RUS	JAP-A	JAP
LEC	LEC	VET	VET	HAM	HAM	BOT	BOT
BOT	BOT	LEC	LEC	LEC	BOT	VET	VET
HAM	HAM	VER	VER	VET	LEC	HAM	HAM
PER	RIC	HAM	HAM	VER	VER	VER	ALB
SAI	HUL	BOT	BOT	RUS	ALB	LEC	SAI

MEX-A	MEX	USA-A	USA	BRA-A	BRA	ARE-A	ARE
HAM	HAM	BOT	BOT	VER	VER	HAM	HAM
VET	VET	HAM	HAM	HAM	GAS	VER	VER
BOT	BOT	VER	VER	BOT	SAI	LEC	LEC
LEC	LEC	VET	LEC	VET	RAI	BOT	BOT
ALB	ALB	LEC	ALB	LEC	GIO	VET	VET

que se non fossero incappati in questi problemi, e la procedura di classificazione sarebbe risultata ancora più precisa.

3.2.3 La "*variable selection*" dell'albero di classificazione

Le variabili più importanti nel discriminare la risposta, in generale, sono risultate essere la posizione di partenza e la scuderia del pilota. Con i parametri di complessità scelti, per le gare svoltesi a Monte-Carlo e in Messico l'unico split dell'albero avviene sulla base della posizione di partenza. Nelle analisi esplorative si era visto che a Monte-Carlo in sole 3 occasioni su 6 il poleman ha vinto la gara (si veda la Figura 2), ma nonostante questo, data la tortuosità del tracciato, tendenzialmente

non si sono mai verificate grosse modifiche alla classifica iniziale vista la difficoltà del sorpasso. Abbiamo notato come, tuttavia, per certi gran premi altre variabili oltre a team e posizione di partenza si siano rivelate utili nel processo di classificazione.

Per il Gran Premio di Abu Dhabi, la temperatura dell'aria risulta essere una variabile utile a determinare il piazzamento in top five. In particolare, il CT assegna alta probabilità ad un arrivo tra i primi cinque per un pilota che parte in una posizione non peggiore della sesta, e con temperatura dell'aria inferiore ai 26 gradi. Per quelli di Stati Uniti, Ungheria e Russia è rilevante la conoscenza dell'età della gomma al termine del gran premio. Nello specifico, la probabilità di arrivo in top five è più alta per piloti che finiscono la gara con gomme aventi un'elevato valore della variabile tireage. La conclusione è coerente con le dinamiche che si sono viste nella realtà: l'asfalto poco abrasivo dei circuiti di Stati Uniti e Russia ha spesso premiato i piloti in grado, con una guida pulita, di prolungare per molti giri l'ultimo stint; analogamente, la strategia ad una sosta in Ungheria ha spesso portato i suoi frutti vista la difficoltà di sopravanzare un avversario tra le varianti ("*chicanes*") dell'Hungaroring.

3.3 Foresta casuale

3.3.1 Specificazione

La random forest ("foresta casuale") è un algoritmo di machine learning di tipo supervisionato che consiste nel costruire sui dati una moltitudine di alberi di classificazione o di regressione (da qui il termine "foresta"). Per ognuno degli alberi viene selezionato, ad ogni nodo, un piccolo gruppo di covariate, le quali sono esaminate per trovare il migliore punto di suddivisione in accordo ad un criterio di splitting — a differenza di quanto accade per i CT, nei quali vengono esaminate tutte le variabili ad ogni nodo. Gli alberi vengono fatti crescere fino alla massima grandezza impostata e non sono potati (ovvero, non sono adattati in modo che avvenga una minimizzazione della devianza penalizzata per la dimensione dell'albero), ma la combinazione di diversi alberi evita l'overfitting. I parametri di controllo ("*tuning*") di una random forest sono il numero di alberi e il numero q di variabili da selezionare ad ogni nodo [1].

3.3.2 Applicazione ai dati

Abbiamo implementato una random forest sui gran premi per i quali l'albero di classificazione non aveva fornito una classificazione perfetta dei piloti arrivati tra i primi cinque. Abbiamo osservato dei significativi miglioramenti per il Gran Premio d'Australia, dove le previsioni di Raikkonen e Norris vengono, correttamente,

Tabella 12: Risultati dell'albero di classificazione e della random forest su due GP del Campionato 2019. Le diciture "-A" e "-RF" che seguono la sigla del GP indicano, rispettivamente, la colonna con i risultati dell'albero e della random forest.

AUS-RF	AUS-A	AUS	ITA-RF	ITA-A	ITA
BOT	RAI	BOT	LEC	LEC	LEC
HAM	NOR	HAM	BOT	BOT	BOT
VER	BOT	VER	VET	HAM	HAM
VET	HAM	VET	HAM	PER	RIC
LEC	VER	LEC	RIC	SAI	HUL

sostituite da quelle delle due Ferrari di Vettel e Leclerc, e per il Gran Premio d'Italia. Qui l'algoritmo prevede ancora erroneamente Vettel in top five, ma comunque assegna una probabilità maggiore di 0.5 anche a Nico Hulkenberg (su Renault) di terminare la gara in una posizione non peggiore della quinta (si veda la Tabella 12). Per gli altri gran premi, purtroppo, non si riescono ad eliminare gli errori di classificazione, soprattutto per quelli "imprevedibili" come Germania e Brasile.

3.4 Modelli GAMM

La ricchezza dei datasets a disposizione risiede nell'avere i tempi sul giro di ogni pilota per ogni giro di tutti i gran premi tra il 2014 al 2019. Dopo aver implementato alberi di classificazione e random forest, ricavando dai datasets delle informazioni di sintesi, vogliamo quindi trovare una maniera per modellare i tempi sul giro singolarmente.

3.4.1 Specificazione di un modello GAM

I modelli additivi generalizzati (Generalized Additive Models, GAM) sono una classe di modelli di regressione semiparametrici nei quali la relazione tra la variabile dipendente e le covariate può seguire andamenti non lineari. La loro struttura è del tipo

$$g(\mu) = \beta_0 + \beta_1 x_1 + \ldots + \beta_j x_j + s_1(x_{j+1}) + s_2(x_{j+2}, x_{j+3}) + \ldots$$

dove β_j è il coefficiente di regressione associato alla covariata $x_j, j = 1, \ldots, J$. Le $s_j(\cdot)$ sono funzioni splines delle covariate, definite come $\sum_{j=1}^{p} b_j x_j$, dove i termini b_j sono detti "funzioni base". La funzione di legame ("*link function*") $g(\cdot)$ lega alle esplicative x_j la media della variabile risposta Y, la quale segue una distribuzione appartenente ad una famiglia esponenziale. I modelli additivi generalizzati a effetti misti (Generalized Additive Mixed Models, GAMM) estendono i GAM permettendo l'introduzione di un effetto casuale [9].

Tabella 13: Descrizione delle covariate presenti nel Modello 1.

#	Variabile	Descrizione
1	lapno	Frazione di giri coperti durante una gara.
2	interval	Gap dalla vettura che precede, in secondi.
3	compound	Mescola di pneumatici del pilota.
4	team	Sigla di tre lettere indicante la squadra del pilota.
5	driver	Sigla di tre lettere indicante il cognome del pilota.
6	tireage	Età (in giri) delle gomme del pilota
7	pitThisLap	Il pilota rientra ai box nel corso del giro.
8	FollowingPit	Il pilota esce dalla pit-lane nel corso del giro
9	race	Sigla di tre lettere indicante il circuito.

3.4.2 Analisi di alcuni gran premi del 2015 mediante GAMM

Come anticipato, il GAMM consente di modellare i tempi sul giro di ogni singolo pilota per ogni gran premio, tenendo conto della variabilità intrinseca al pilota stesso grazie alla presenza del termine casuale. Seguendo la traccia di lavoro definita da Casella e Vidoni nell'articolo "Formula 1 lap time modeling using generalized additive models" [3], si considerano i seguenti gran premi della stagione 2015: Abu Dhabi, Australia, Italia, Malesia e Russia. Il dataset Race è stato appositamente modificato per la costruzione del modello come segue: in prima istanza, si sono eliminati tutti i giri effettuati sotto regime di Safety Car o di Virtual Safety Car. Sono stati rimossi anche tutti i giri numero 1 dei piloti, alti e chiaramente condizionati dalla procedura di partenza (come già spiegato nel Paragrafo 3.1). Inoltre, abbiamo tolto tutti i tempi più alti di una soglia fissata a 130 secondi, imputabili a errori del pilota come testacoda, problemi meccanici o difficoltà occorse ai box. Le variabili di cui abbiamo tenuto conto sono quelle di Tabella 13.

Si evidenzia che la variabile lapno è misurata come frazione di giri coperti durante la corsa in modo da tener conto delle diverse lunghezze di gara; inoltre, per quanto riguarda la variabile interval, per distacchi superiori a 3.7 secondi dalla vettura che precede in pista si assume un distacco pari proprio a 3.7 secondi, poiché si verifica empiricamente che l'effetto di questa variabile — in termini di turbolenze — svanisce oltre la suddetta soglia.

Tabella 14: Stime di alcuni coefficienti del Modello 1.

Parametro	Stima puntuale	Parametro	Stima puntuale
team = MER	−0.0047	race = ITA	−0.19
team = RBR	0.011	race = MYS	−0.0075
team = MAR	0.05	race = AUS	−0.14
team = WIL	0.0053	race = RUS	−0.045
team = STR	0.014	FollowingPit	0.17
team = SAU	0.015	pitThisLap	0.22

Il modello finale, che indicheremo con "Modello 1", assume la seguente struttura:

$$g(\mu) = \beta_k + \bar{\beta}_1 \cdot \mathsf{team} + \bar{\beta}_2 \cdot (\mathsf{pitThisLap} \times \mathsf{race}) + \bar{\beta}_3 \cdot (\mathsf{FollowingPit} \times \mathsf{race}) + s_1(\mathsf{lapno}) + s_2(\mathsf{tireage}, \mathsf{compound}) + s_1(\mathsf{interval}), \quad (1)$$

dove β_k indica l'effetto casuale del k-esimo pilota, avente una distribuzione Gaussiana di media nulla e varianza σ^2. Le interazioni tra le variabili race e FollowingPit e tra race e pitThisLap sono dovute al fatto che ogni circuito ha una lunghezza diversa della pit-lane, quindi le soste sono più o meno lunghe a seconda del tracciato. Nella Tabella 14 vengono riportati i risultati più importanti ottenuti dopo la stima. Tutti i parametri e le splines (queste ultime del tipo "thin plate") sono ampiamente significativi. In particolare, per la variabile team è stata considerata come baseline la Ferrari; l'unico coefficiente per team con segno negativo è quello relativo alla Mercedes, che quindi risulta essere l'unica squadra più veloce della Ferrari. Il modello assegna invece un elevato coefficiente alle squadre più lente come la Marussia. Il circuito baseline è quello di Abu Dhabi, che tra i cinque considerati è quello che i piloti impiegano più tempo a percorrere, visto il segno negativo dei coefficienti della variabile race. Da evidenziare anche l'utilità delle variabili FollowingPit e pitThisLap, entrambe con coefficiente positivo, a testimoniare l'aumento del tempo sul giro nelle tornate dove il pilota transita in pit-lane.

L'effetto delle covariate lapno e tireage sulla risposta non è lineare, nel senso che non è possibile fare affermazioni del tipo "all'aumentare unitario del valore di una di queste due covariate, la risposta aumenta o diminuisce di un certo valore", come era stato fatto con la regressione robusta. In questo caso, risulta chiaro dall'analisi della Figura 13 che all'aumentare della distanza percorsa il logaritmo del tempo sul giro (e, di conseguenza, il tempo sul giro) diminuisce, salvo riprendere a salire leggermente verso fine gara. Negli ultimi giri, infatti, solitamente i piloti non in battaglia per la posizione effettuano "*fuel saving*" (cioè risparmiano carburante) o

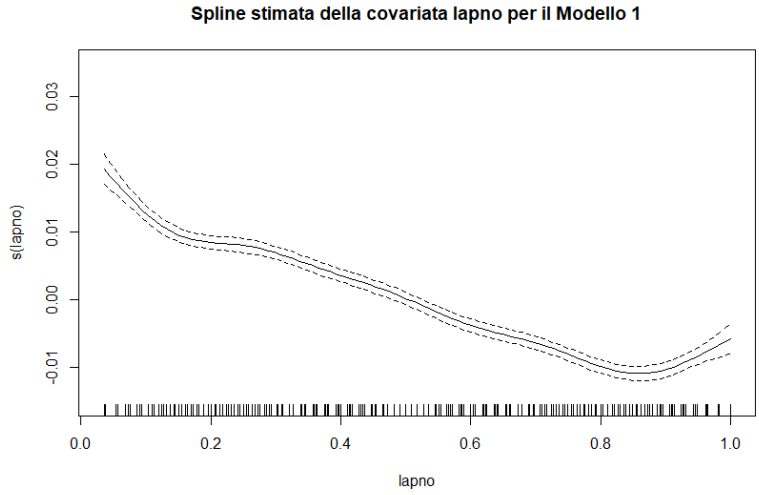

Figura 13: Spline del Modello 1 per la covariata lapno. Le linee tratteggiate rappresentano l'intervallo di confidenza della curva liscia.

incontrano molti doppiati, e quelli in battaglia perdono tempo con gli ultimi sorpassi e controsorpassi.

Dal punto di vista delle gomme (Figura 14), invece, si constata come le mescole più dure garantiscano un aumento del tempo sul giro lieve e costante, al contrario delle soffici dove l'usura avviene più rapidamente. In particolare, per la mescola A3 è da rilevare un drastico crollo di prestazioni intorno al giro 30.

3.4.3 Analisi della strategia di Massa al Gran Premio d'Italia del 2015

Analizziamo in dettaglio la strategia di un pilota, Felipe Massa, durante un gran premio, quello svoltosi a Monza nel 2015. Il brasiliano della Williams ha effettuato una sosta al giro 19, passando dagli pneumatici A3 (Soft) agli A2 (Medium). Proviamo a stabilire se questa è stata la strategia ottimale, ovvero quella che gli ha consentito di terminare la gara nel minor tempo possibile, utilizzando il Modello 1. Ne stimiamo i parametri eliminando dall'insieme di addestramento i giri di Massa a Monza, onde evitare l'overfitting. Il tempo compiuto da Massa per ultimare il Gran Premio, tolto il primo giro, risulta pari a 4635.837 secondi. La previsione effettuata con il Modello 1 è di 4641.36 secondi, più "pessimista" di 5.523 secondi. La sequenza di operazioni eseguite per realizzare la simulazione di una gara con pit-stop al generico giro x è descritta nell'Algoritmo 1.

Questa procedura ha il difetto di non portare alla modifca della variabile interval, perché purtroppo, con i dati in possesso, non è possibile arrivare a capire quale

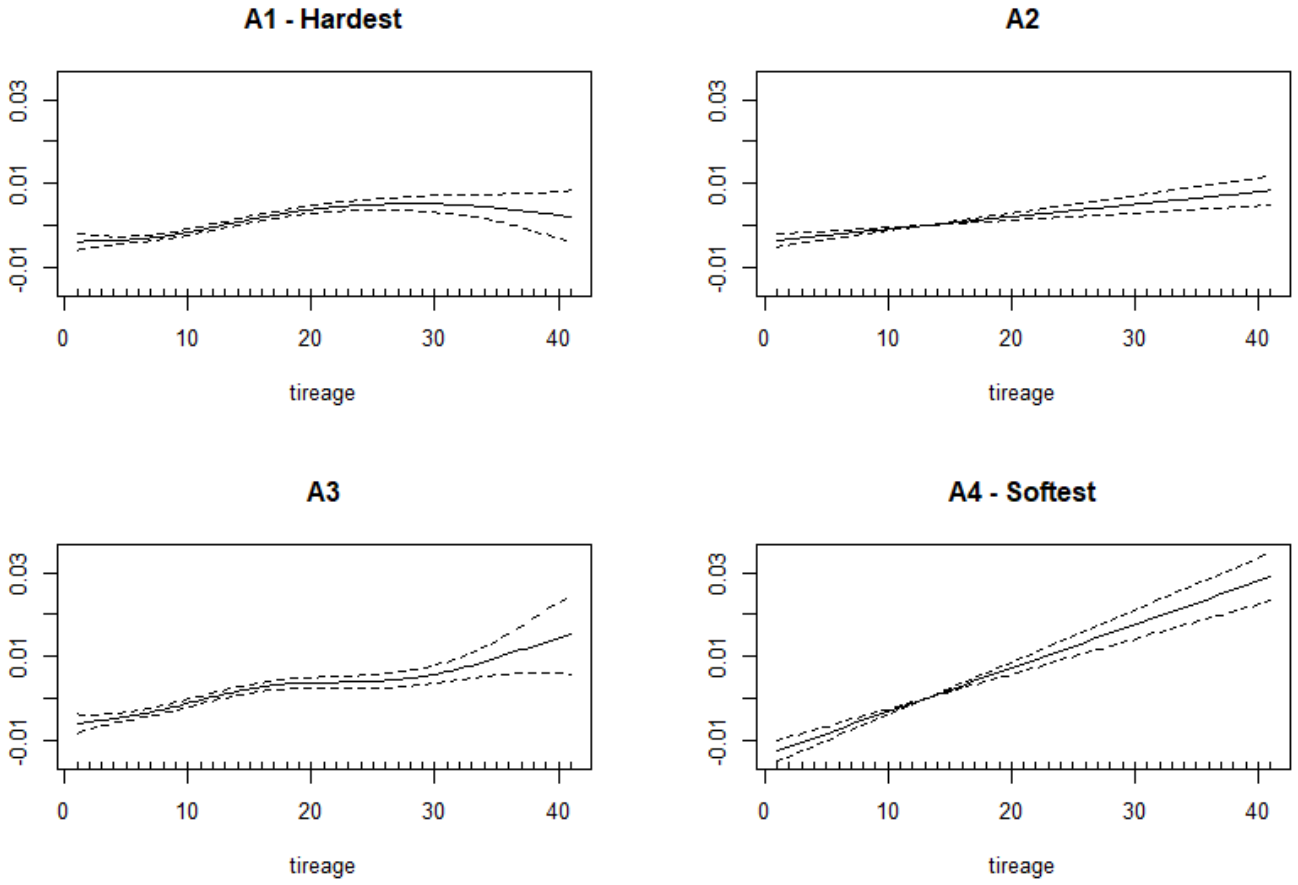

Figura 14: Splines del Modello 1 per la covariata tireage in base alla mescola di pneumatici (dalla A1, la più dura, alla A4, la più morbida).

Algoritmo 1 Simulazione di gara di Massa a Monza - Campionato 2015

1: a partire dal testing set (contenente le variabili del dataset Race utilizzate in fase di stima per Massa, Italia 2015) si crea un nuovo dataset ponendo uguali a FALSE tutte le occorrenze della variabile pitThisLap e a 0 tutte le occorrenze della variabile FollowingPit;
2: per il giro x del testing set si imposta uguale a TRUE la variabile PitThisLap e a 1 la variabile FollowingPit del giro $x + 1$;
3: si modifica il valore della variabile compound imponendo mescola "A3" per i giri dal primo al numero x, e mescola "A2" per tutti gli altri;
4: i valori della variabile tireage dal primo giro al giro x vengono mantenuti pari a quelli originali, mentre dal giro $x+1$ la variabile tireage riparte da 1 (si immagina che i meccanici montino gomme nuove);
5: si stima il Modello 1 sul dataset appena creato e si tiene traccia del tempo di gara cumulato ottenuto;
6: si restaura il dataset riportandolo alla forma del punto 1;
7: si itera il procedimento per tutti i giri del Gran Premio di Monza 2015.

sarebbe la distanza di Massa dal pilota che precede in caso di pit stop ad un giro arbitrario. Dalla simulazione di gara emerge che il tempo di gara sarebbe stato minimo nel caso il pilota brasiliano si fosse fermato al giro 23 anziché al giro 19 (Figura 15), tuttavia le differenze nel tempo totale di gara in caso di fermate dal giro 17 al giro 35 sono dell'ordine dei decimi di secondo. Nella realtà, ciò è altamente improbabile: si deduce che questo modello non è molto adatto per elaborare strategie di gara efficaci. Tuttavia, una cosa è effettivamente plausibile: se Massa si fosse fermato qualche giro più tardi avrebbe impiegato meno tempo per coprire la distanza di gara. Uno sguardo al replay del Gran Premio chiarifica le motivazioni dietro alla scelta del box Williams di fermare il pilota al giro 19. Infatti, al giro 18, Massa occupa la terza posizione davanti al compagno di squadra Bottas, che segue a circa un secondo e mezzo, e a Rosberg (Mercedes), separato da Bottas da circa un secondo. Rosberg viene richiamato ai box e, nel giro di rientro, inizia a spingere sfruttando il fatto di avere una gomma fresca e performante. Per rimediare alla strategia di Rosberg, in gergo definita "*undercut*", Massa è costretto ad effettuare la sosta al giro immediatamente successivo, il 19, (e nonostante questo non riesce a rientrare davanti al tedesco). Se Massa fosse rimasto in pista fino al giro 23, con tutta probabilità sarebbe rientrato molto dietro a Rosberg, ma verso la fine del GP avrebbe potuto contare su gomme più fresche e lo avrebbe passato comunque, minimizzando il suo tempo di gara.

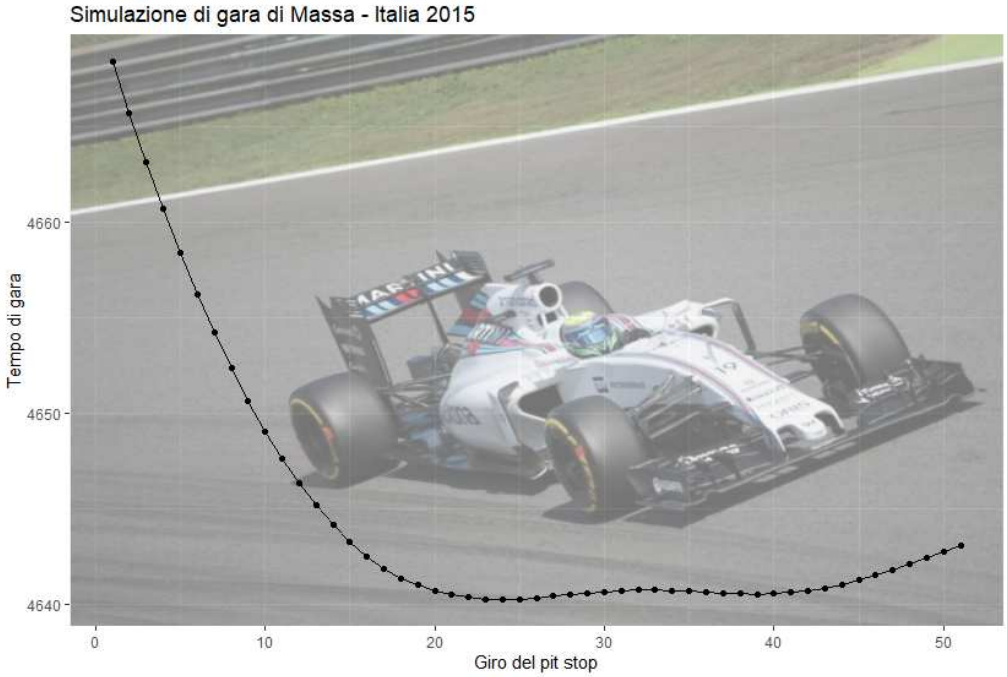

Figura 15: Simulazione di gara di Felipe Massa, Gran Premio d'Italia 2015. In ascissa il giro della sosta "virtuale", in ordinata il tempo "virtuale" di gara.

3.5 Estensione del Modello 1 utlizzando informazioni sui settori

Una possibile estensione del Modello 1 prevede la stima del tempo sul giro tenendo conto dei tempi fatti registrare dal pilota nei singoli settori di cui si compone un giro. Ogni circuito si divide in tre settori, il primo dei quali parte appena dopo la linea del traguardo, che è anche il punto di fine del terzo. Un pilota che entra in pit-lane per effettuare una sosta sigla tempi alti nell'ultimo settore del giro d'ingresso e nel primo settore del giro d'uscita. Se il pilota riesce a scaldare in fretta le gomme, il terzo settore del giro d'uscita, e in certi casi anche il secondo, sono tipicamente molto veloci in virtù dell'aderenza ("*grip*") extra garantita dalla mescola appena montata. Successivamente, i tempi di solito si stabilizzano. Questa informazione, preziosa per comprendere al meglio il valore della strategia di undercut descritta al paragrafo 3.4.3, si perde quasi completamente se si usa per la stima del modello il dataset Race, poiché il tempo del giro dopo il pit stop risulta viziato dal primo settore alto e non rispecchia l'effettiva "*over-performance*" del pilota.

Risulta allora opportuno stimare i parametri di un nuovo modello sulla base del dataset Race 2. In questo caso, la struttura di quello che individueremo con

Tabella 15: Stime di alcuni coefficienti del Modello 2.

Parametro	**Stima puntuale**	**Parametro**	**Stima puntuale**
Team = MER	-0.0036	Circuit = ITA	-0.20
Team = RBR	0.0038	Circuit = CHN	-0.046
Team = REN	0.014	Circuit = AUS	-0.15
Team = WIL	0.033	Circuit = RUS	-0.036
Team = STR	0.014	ExitingPit	0.20
Team = ALF	0.019	EnteringPit	0.21

"Modello 2" diventa la seguente:

$$\begin{gathered} g(\mu) = \beta_k + \bar{\beta}_1 \cdot \mathsf{Team} + s(\mathsf{LapNumber}) + \bar{\beta}_2 \cdot (\mathsf{ExitingPit} \times \mathsf{Circuit}) + \\ \bar{\beta}_3 \cdot (\mathsf{EnteringPit} \times \mathsf{Circuit}) + s(\mathsf{Tyrelife}, \mathsf{Compound} \times \mathsf{FreshTyre}) + \\ \bar{\beta}_4 \cdot (\mathsf{FastSector1} \times \mathsf{ExitingPit}) + \bar{\beta}_5 \cdot (\mathsf{FastSector2} \times \mathsf{ExitingPit}) + \\ \bar{\beta}_6 \cdot (\mathsf{FastSector3} \times \mathsf{ExitingPit}), \end{gathered} \tag{2}$$

dove, come nel Modello 1, β_k indica l'effetto casuale del $k-$esimo pilota, con distribuzione Gaussiana di media nulla e varianza σ^2.

3.5.1 Analisi dei tempi sul giro di Monza 2019

Come prima cosa si ripete il lavoro fatto per il Modello 1 utilizzando solamente cinque dei gran premi del dataset Race 2: Australia, Italia, Cina (in sostituzione del GP di Malesia), Russia e Emirati Arabi Uniti.

Dalle stime dei parametri del Modello 2 si nota subito che i coefficienti relativi all'interazione tra l'uscita dai box e i settori veloci sono altamente significativi, così come tutti gli altri parametri del modello. Nella Tabella 15 sono riportate le stime di alcuni tra i parametri relativi alle variabili Team e Circuit (analogamente a quanto visto in Tabella 14) e di quelli relativi al passaggio del pilota ai box. La squadra di riferimento per la variabile Team è, come per il Modello 1, Ferrari; il tempo sul giro si abbassa se il team del pilota è Mercedes, mentre si alza molto, ad esempio, se il pilota guida una Williams (molto peggiorata rispetto al 2014 e al 2015, quand'era stata in grado di piazzarsi al terzo posto nella classifica finale dei Costruttori, come si può vedere in Tabella 1). Dopo il pit stop, come ampiamente preannunciato, il tempo sul giro si alza nel primo settore e si abbassa negli altri settori. Per testare le performance del Modello 2, come testing set si considerano stavolta i giri di Charles Leclerc a Monza, unico pilota che in quell'occasione scelse di percorrere un primo

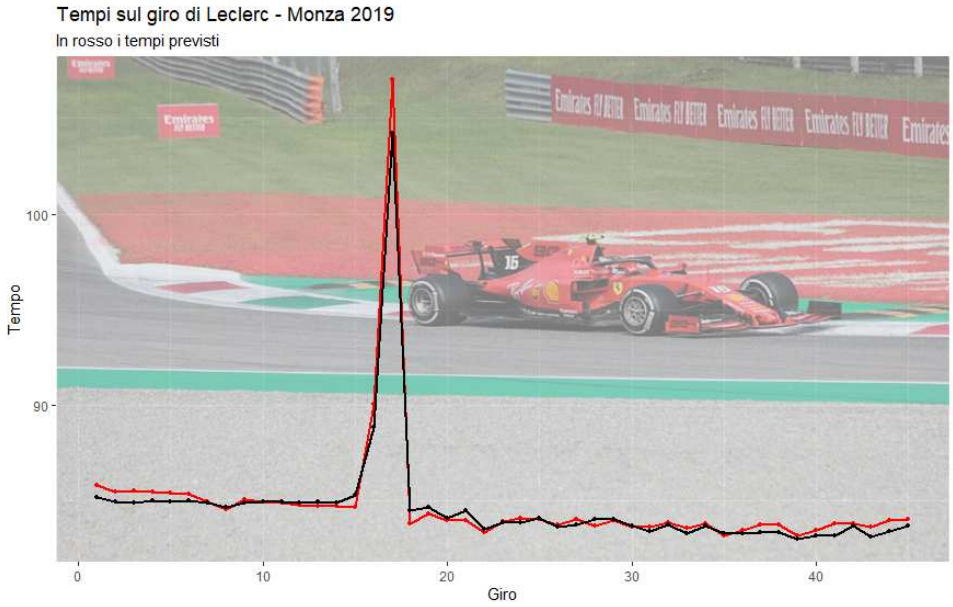

Figura 16: Tempi sul giro di Leclerc a Monza nel 2019. In nero i tempi osservati e in rosso quelli previsti dal Modello 2.

stint con pneumatici di mescola Soft e un secondo stint con pneumatici di mescola Hard, nonché vincitore davanti al pubblico di casa. In generale, la previsione dei tempi sul giro appare piuttosto aderente alla realtà, come si evince dalla Figura 16; questo nonostante il Modello 2 sia stato costruito su un dataset non comprendente l'informazione sulla distanza del pilota dal pilota che precede. La variabile relativa ai settori è dunque molto importante nell'aumentare la precisione delle stime.

3.6 Estensione del Modello 2 utilizzando informazioni sui settori e sul meteo

A questo punto si prende in considerazione un modello con informazioni sui settori e sulle condizioni meteorologiche della pista. Anziché considerare un "campione" di cinque gran premi, si considerano direttamente tutti i gran premi del 2019 (ossia tutto il dataset Race 2) nel tentativo di ottenere un adattamento ancora migliore. Analogamente a quanto fatto in precedenza, si eliminano dal dataset i giri con bandiere gialle o rosse, Safety Car o Virtual Safety Car. Il Modello 3 ha la seguente struttura:

$$\begin{aligned} g(\mu) = \beta_k + \bar{\beta}_1 \cdot \mathsf{Team} + s(\mathsf{LapNumber}) + \ldots + \beta_7 \cdot \mathsf{Humidity} + \\ \beta_8 \cdot \mathsf{AirTemp} + \beta_9 \cdot \mathsf{TrackTemp} + \beta_{10} \cdot \mathsf{WindSpeed} + \beta_{11} \cdot \mathsf{Rainfall}, \end{aligned} \tag{3}$$

dove i puntini di sospensione indicano le altre variabili e splines presenti nel Modello 2. L'adattamento del Modello 3 sorprendentemente rivela che, tra le variabili meteorologiche, solo AirTemp è significativa. È stata adattata anche una variante del Modello 3 che possiede, in sostituzione di AirTemp e TrackTemp, una variabile denominata HeatIndex, definita come la media aritmetica di queste due temperature, ma HeatIndex non è risultata significativa. È allora la temperatura dell'aria ad avere maggior influenza nei tempi sul giro, più che quella dell'asfalto, come forse sarebbe risultato più intuitivo pensare. Il coefficiente relativo alla variabile AirTemp risulta positivo, anche se molto basso in valore assoluto, e pari a 0.00023, a segnalare un incremento del tempo sul giro all'aumentare della temperatura. Evidentemente, l'aria più calda costringe i meccanici a scegliere configurazioni del motore meno aggressive per evitare problemi di surriscaldamento, e ciò si traduce in una riduzione della velocità massima. Per quanto riguarda la relazione tra tempo sul giro e tipologia di mescola utilizzata, descritta in Figura 17, si apprezza come la mescola Hard, sia nuova sia usata, subisca un degrado praticamente uniforme. Un treno usato di pneumatici di mescola Medium garantisce tempi piuttosto veloci per i primi 11/12 giri, poi inizia lentamente ad usurarsi. Le mescole Soft usate hanno un crollo verticale dopo un numero di giri maggiore a 40. Solo in un'occasione, infatti, un pilota (Albon, al GP di Monte-Carlo) ha effettuato uno stint con Soft usate per più di quaranta giri.

3.6.1 Il test del Modello 3 e i punti deboli dell'approccio tramite GAMM

Analogamente a quanto fatto per il Modello 1 e per il Modello 2, si è messo alla prova il Modello 3 ponendo nel testing set i dati del Gran Premio di Monza. I risultati ottenuti sono molto simili a quelli del Modello 2. Si considera adesso come pilota di riferimento Valtteri Bottas (Mercedes) per evidenziare alcuni punti deboli dell'approccio utilizzato. I tempi sul giro di Bottas dalla sosta in poi sono previsti dal Modello 3 in maniera abbastanza soddisfacente. Appare evidente, invece, la discrepanza nei giri precedenti al pit stop, dove i tempi ipotizzati dal Modello 3 sono molto inferiori a quelli effettivamente realizzati da Bottas (Figura 18). Nel corso del primo stint, il pilota stava tentando di conservare il più a lungo possibile le sue gomme, al fine di compiere una strategia diversificata rispetto a quella del compagno di squadra Hamilton. Forse il pilota è stato troppo cauto, oppure, al fine di favorire la lotta al Mondiale di Hamilton, ha voluto evitare di impensierirlo troppo avvicinandosi. In ogni caso, è evidente che l'esito e le strategie di un gran premio dipendono da fattori che esulano dalle performance effettivamente raggiungibili dalla macchina.

A conferma di quest'ultima considerazione, l'esempio finale che si propone prende spunto da quanto accaduto durante il Gran Premio di Abu Dhabi nel 2016,

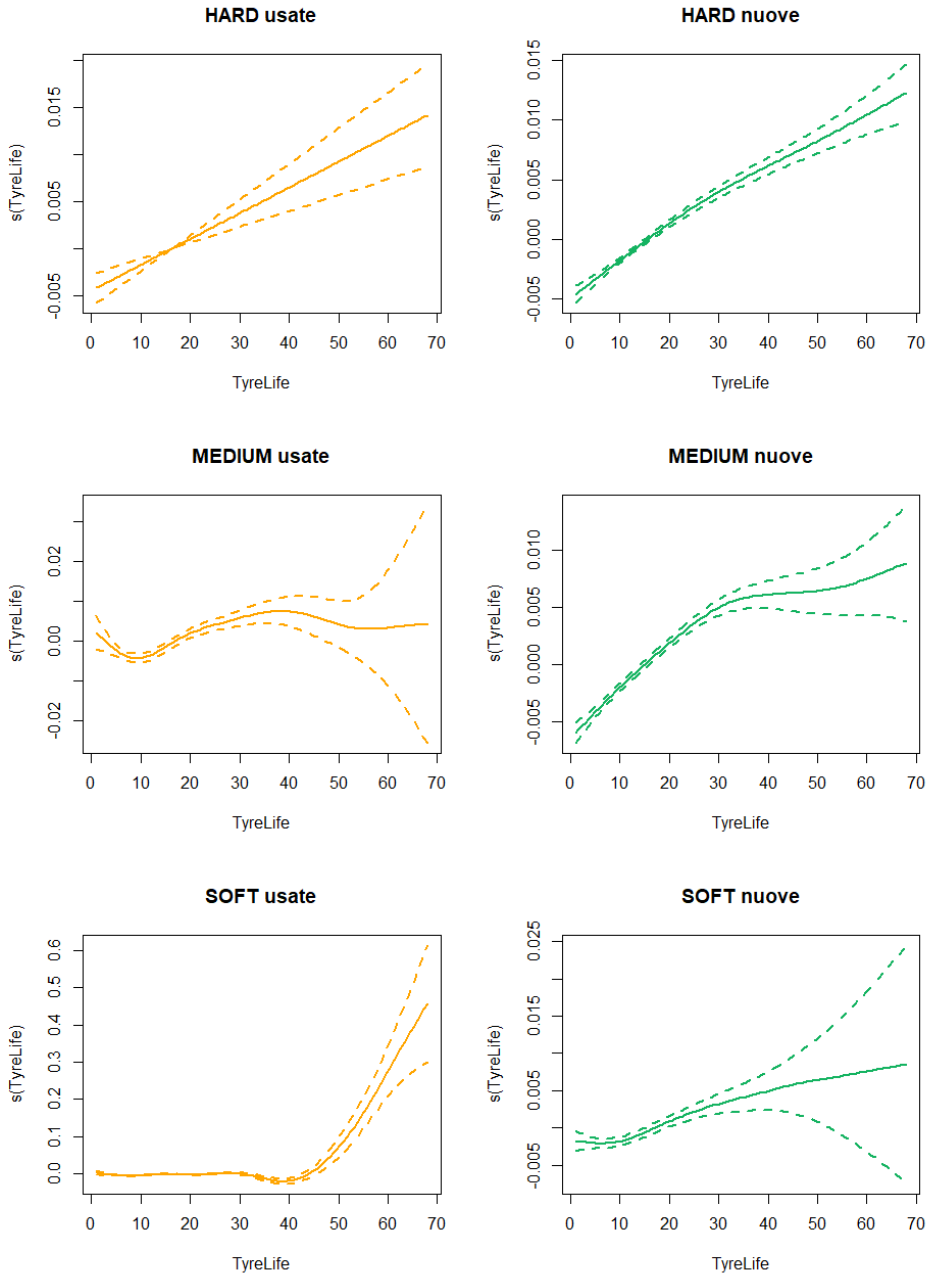

Figura 17: Splines del Modello 3 per la covariata TyreLife, in base alla mescola della gomma (in alto: gomme Hard; in mezzo: gomme Medium; in basso; gomme Soft).

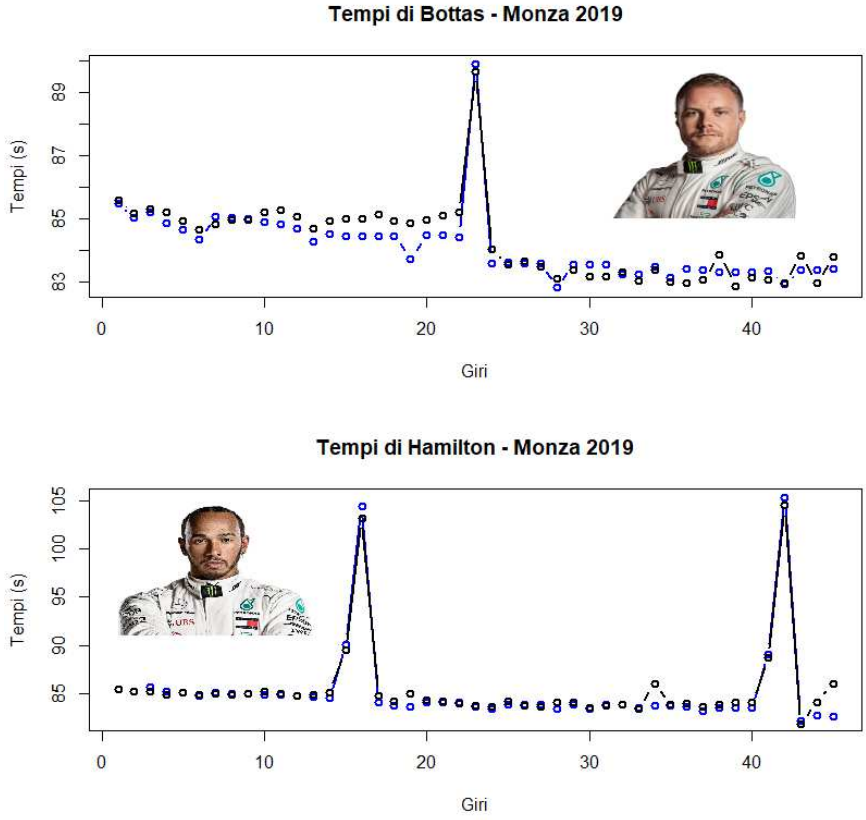

Figura 18: Tempi sul giro di Bottas e Hamilton a Monza nel 2019. In blu quelli previsti dal Modello 3.

gara conclusiva del Campionato, decisiva per il Mondiale Piloti. Hamilton era in testa seguito da Rosberg, conscio del fatto che al compagno di squadra sarebbe bastato arrivare secondo per diventare Campione. Il britannico rallentava allora appositamente per compattare il gruppo dietro di sé, senza però farsi sorpassare dal compagno di squadra, sperando invece che qualche altro pilota sopravanzasse Rosberg (cosa alla fine non accaduta). In Figura 19 si può vedere come l'andatura di Hamilton dalla seconda sosta in poi sia stata in media un secondo più lenta di quanto previsto dal Modello 2.

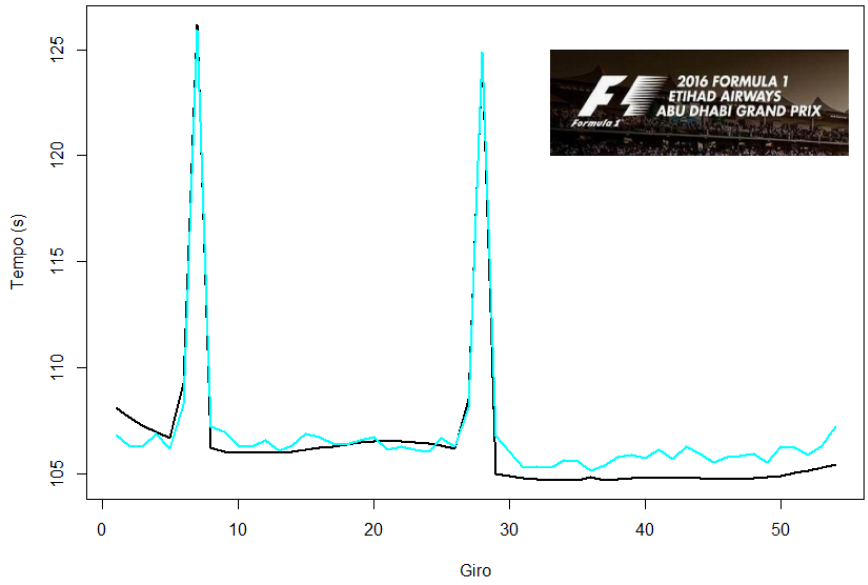

Figura 19: Tempi sul giro di Hamilton ad Abu Dhabi nel 2016. In nero quelli previsti dal Modello 2.

4 Conclusioni

Nel corso di questa Tesi abbiamo visto alcuni tentativi di modellizzazione di alcuni aspetti dei gran premi di Formula Uno, prendendo come riferimento i Campionati dal 2014 al 2019. Sono stati inizialmente descritti i dataset di riferimento, ottenuti tramite varie fonti e metodi (repositories online e scraping). Le elaborazioni effettuate per l'arricchimento ed adattamento delle masse di dati a disposizione sono state particolarmente laboriose, perché molte informazioni sono riservate e non possono essere condivise per finalità di ricerca dalle scuderie (nemmeno, come si è provato a fare, su esplicita richiesta). Inizialmente, sono state effettuate alcune semplici analisi esplorative per avere un quadro più chiaro della situazione competitiva del periodo, dalle quali si è dedotta la netta supremazia in termini di pole position e vittorie di tre scuderie: Ferrari, Red Bull Racing e soprattuto Mercedes.

In seguito, è stato mostrato tramite tecniche di regressione robusta come il tempo sul giro di ciascun pilota — indipendentemente dalle performance della sua vettura e dalla posizione di partenza — subisca un trend discendente nel corso di una gara. Un confronto con i tempi sul giro nel corso dei Campionati nei quali era ancora concesso il rifornimento in pista ha rivelato che la motivazione sottostante a questo fenomeno risiede sostanzialmente nell'alleggerimento progressivo della vettura dopo l'unico, grande rifornimento prima dello spegnimento dei semafori consentito nel corso dei Mondiali considerati.

L'attenzione si è in seguito focalizzata sull'uso di tecniche non parametriche per prevedere le prime cinque posizioni di arrivo dei gran premi del 2019, utilizzando i dati dei Campionati precedenti in possesso per allenare un albero di classificazione. Sostanzialmente, dai risultati - peraltro molto precisi, grazie al fatto che non vi è stato un particolare mutamento nei valori in gioco nel 2019 rispetto agli anni precedenti - è emerso che le due variabili più importanti per prevedere la top five sono il team del pilota e la posizione di partenza. In certi casi l'albero indicava come unica variabile di discriminazione la posizione di partenza (ad esempio, a Monte-Carlo); in altri, sono subentrate variabili come l'età della gomma alla fine del GP. In ogni caso, difficoltà previsive sono state constatate per gran premi con condizioni meteorologiche mutabili o ricchi di eventi imprevedibili come incidenti o guasti. In questo senso, per migliorare il modello potrebbero essere utili informazioni circa la propensione del singolo pilota all'errore nel singolo circuito, o circa l'usura delle

componenti della power unit utilizzate dai piloti, così come potrebbe essere utile un'indicazione sullo "stato di forma" del pilota. Ad esempio: "vincitore dell'ultimo gran premio", "in striscia positiva", "fresco di rinnovo con la squadra".

Infine, sono stati realizzati tre modelli additivi generalizzati a effetti misti per stimare il tempo sul giro dei piloti nel corso delle gare. Le variabili inserite nel predittore, quali, ad esempio, compound della gomma, età della gomma, frazione di giri percorsi sul totale, sono risultate tutte altamente significative. Nessuno dei modelli è però in grado di consentire affidabili simulazioni di gara, ottenibili facendo variare il giro della/e sosta/e. Si è notato comunque un miglioramento notevole nelle capacità predittive comprendendo anche le informazioni sui tempi settore per settore e la loro interazione con la fermata ai box, la quale consente ai piloti di effettuare giri molto veloci subito dopo sfruttando l'aderenza extra fornita dalle gomme fresche. Un'ulteriore, possibile specificazione del modello GAMM potrebbe comprendere la variabile dummy relativa all'ingresso della Safety Car. Tuttavia, in questo caso bisognerebbe avere a disposizione anche il momento esatto del giro (sotto forma di tempo) in cui effettivamente viene instaurato il regime di bandiere gialle e Safety Car, perché se ciò capitasse alla fine della tornata il tempo complessivo non ne risentirebbe molto, a differenza di quanto succederebbe nel caso la Safety Car fosse dispiegata quando il pilota si trova poco dopo la linea del traguardo (situazione in cui egli sarebbe costretto a procedere lentamente per tutto il giro).

In conclusione, si è visto come i dati a disposizione consentano di modellizzare in modo apprezzabile gran premi disputati in "condizioni ideali", nei quali non piove e non succedono incidenti tali da richiedere l'ingresso della vettura di sicurezza (reale o virtuale). Per realizzare simulazioni di gara più complesse occorrerebbero più dati e, forse, una conoscenza ancor più approfondita delle dinamiche di gara. Ciononostante, sarebbe comunque impossibile costruire il modello "perfetto": come tener conto, d'altronde, di piloti che guidano al limite della vettura e oltre, o di altri che disobbediscono alle strategie fidandosi della loro sensibilità; di situazioni in cui i compagni di squadra si aiutano reciprocamente rallentando deliberatamente gli avversari, e di molte altre circostanze di questo tipo? Impossibile; ma in fondo, proprio qui risiede il bello di questo sport.

Riferimenti bibliografici

[1] Azzalini, A. and Scarpa, B. (2012). *Data analysis and data mining: An introduction*. OUP USA.

[2] Bekker, J. and Lotz, W. (2009). Planning formula one race strategies using discrete-event simulation. *Journal of the Operational Research Society*, 60(7):952–961.

[3] Casella, C. and Vidoni, P. (2017). Formula 1 lap time modeling using generalized additive models. In *Proceedings of MathSport International 2017 Conference*, page 87.

[4] Debicki, T. (2008). Challenges for logistics in the pinnacle of motorsports-formula 1. *Archives of Transport System Telematics*, 1:3–7.

[5] Heilmeier, A., Thomaser, A., Graf, M., and Betz, J. (2020). Virtual strategy engineer: Using artificial neural networks for making race strategy decisions in circuit motorsport. *Applied Sciences*, 10(21):7805.

[6] Kwartler, T. (2022). *Sports Analytics in Practice with R*. Wiley.

[7] Salvan, A., Sartori, N., and Pace, L. (2020). *Modelli Lineari Generalizzati*. UNITEXT. Springer Milan.

[8] Severini, T. (2020). *Analytic Methods in Sports: Using Mathematics and Statistics to Understand Data from Baseball, Football, Basketball, and Other Sports*. CRC Press.

[9] Wood, S. N. (2017). *Generalized additive models*. Texts in Statistical Science Series. CRC Press, Boca Raton, FL. An introduction with R, Second edition.

Riferimenti sitografici

Accelerating the fan experience. How FORMULA 1 is driving the future of racing using machine learning and AWS.
URL: https://pages.awscloud.com/rs/112-TZM-766/images/AWS_Formula_1_eBook_Accelerating_the_Fan_Experience_Final.pdf

Ergast. *Ergast Developer API.*
URL: http://ergast.com/mrd/

F1 Official Live Timing.
URL: https://www.formula1.com/en/f1-live.html

F1 Timing Database.
URL: https://github.com/TUMFTM/f1-timing-database

Fast F1 API.
URL: https://theoehrly.github.io/Fast-F1/

Visual Crossing Weather Data. *Visual Crossing Corporation. (2022). Visual Crossing Weather (2014-2017). [data service].*
URL: https://www.visualcrossing.com/legacy/weather/weather-data-services

Appendice: codice R

In questa Appendice viene riportato il codice R che è stato utilizzato per svolgere le analisi descritte nei capitoli precedenti.

```
library(tidyverse)
library(png)
library(grid)
library(ggpubr)

# Si disegnano, sfruttando la library ggplot2(), le forme di
# alcuni tracciati "storici" come Silverstone, Hockenheim,
# Monza, Spa-Francorchamps.

# Spa-Francorchamps.
coor.spa <- read.csv("Tracks_coordinates/spa.csv",
                     sep = ",")

img.spa <- readPNG("bel.PNG")
img.spa2 <- matrix(rgb(img.spa[,,1],
                       img.spa[,,2],
                       img.spa[,,3],
                       img.spa[,,4] * 0.5),
                   nrow = dim(img.spa)[1])

plotspa <- ggplot() +
  annotation_custom(rasterGrob(img.spa2,
                               width = unit(1, "npc"),
                               height = unit(1, "npc")),
                               -Inf, Inf, -Inf, Inf) +
  geom_point(data = coor.spa, aes(x = x_m, y = y_m),
             color = "black", size = 2) +
  labs(x = "", y = "") +
  theme(legend.position = "none",
        axis.text.x = element_blank(),
        axis.text.y = element_blank()) +
  ggtitle("Circuit de Spa-Francorchamps")
```

```
# Silverstone.
coor.sil <- read.csv("Tracks_coordinates/silverstone.csv",
                     sep = ",")

img.gbr <- readPNG("gbr.PNG")
img.gbr2 <- matrix(rgb(img.gbr[,,1],
                       img.gbr[,,2],
                       img.gbr[,,3],
                       img.gbr[,,4] * 0.5),
                   nrow = dim(img.gbr)[1])

plot.sil <- ggplot() +
  annotation_custom(rasterGrob(img.gbr2,
                               width = unit(1, "npc"),
                               height = unit(1, "npc")),
                    -Inf, Inf, -Inf, Inf) +
  geom_point(data = coor.sil, aes(x = x_m, y = y_m),
             color = "black", size = 2) +
  labs(x = "", y = "") +
  theme(legend.position = "none",
        axis.text.x = element_blank(),
        axis.text.y = element_blank()) +
  ggtitle("Silverstone circuit")


# Monza.
coor.ita <- read.csv("Tracks_coordinates/monza.csv",
                     sep = ",")

img.ita <- readPNG("ita.PNG")
img.ita2 <- matrix(rgb(img.ita[,,1],
                       img.ita[,,2],
                       img.ita[,,3],
                       img.ita[,,4] * 0.5),
                   nrow = dim(img.ita)[1])

plot.ita <- ggplot() +
  annotation_custom(rasterGrob(img.ita2,
                               width = unit(1, "npc"),
                               height = unit(1, "npc")),
                    -Inf, Inf, -Inf, Inf) +
  geom_point(data = coor.ita, aes(x = x_m, y = y_m),
             color = "black", size = 2) +
  labs(x = "", y = "") +
  theme(legend.position = "none",
        axis.text.x = element_blank(),
        axis.text.y = element_blank()) +
  ggtitle("Autodromo Nazionale di Monza")
```

```
# Interlagos.
coor.bra <- read.csv("Tracks_coordinates/interlagos.csv",
                     sep = ",")

img.bra <- readPNG("bra.PNG")
img.bra2 <- matrix(rgb(img.bra[,,1],
                       img.bra[,,2],
                       img.bra[,,3],
                       img.bra[,,4] * 0.4),
                   nrow = dim(img.bra)[1])

plot.bra <- ggplot() +
  annotation_custom(rasterGrob(img.bra2,
                               width = unit(1, "npc"),
                               height = unit(1, "npc")),
                    -Inf, Inf, -Inf, Inf) +
  geom_point(data = coor.bra, aes(x = x_m, y = y_m),
             color = "black", size = 2) +
  labs(x = "", y = "") +
  theme(legend.position = "none",
        axis.text.x = element_blank(),
        axis.text.y = element_blank()) +
  ggtitle("Autodromo Jose Carlos Pace")

figure <- ggarrange(plotspa, plot.bra, plot.ita,
                    plot.sil)
figure
```

Codice 1: Creazione delle immagini in Figura 1.

Per l'associazione ad ogni giro dei Gran Premi dal 2014 al 2017 delle condizioni meteorologiche corrispondenti, rilevate per fasce orarie di 5 minuti ciascuna, si riporta a titolo di esempio la conversione dei dati del Gran Premio di Melbourne 2017. Un analogo procedimento è stato fatto per tutti gli altri gran premi.

```
library(lubridate)
library(tidyverse)

# Melbourne 2017
mel <- read.csv("aus2017.csv", header = T, sep = ",")

t <- as.POSIXlt(mel$Date.time, tz = "UTC",
                format = "%m/%d/%Y %H:%M:%S")

t0 <- t[13]
t0 # l'ora d'inizio del Gran Premio

# tesiDSLT
gare <- read.csv(file.choose(), header = T, sep = ",")
```

```r
# racetime
rt <- gare %>% filter(season == "2017",
                      race == "AUS",
                      driver == "VET",
                      lapno != "0") %>% dplyr::select(racetime)

# Convertiamo il racetime in un tempo, a partire dalle 17
# (t0).

tig <- t0 + as.numeric(rt$racetime) + 120

# Dobbiamo associare ad ognuno di questi giri una
# condizione meteo, "pescandola" dal file con i dati meteo,
# in base alla fascia oraria in cui cade il giro.

mel2 <- mel %>% mutate(Date.time = mdy_hms(Date.time))
mel2 <- mel2[13:nrow(mel2), ]

# Trova il timestamp maggiormente vicino nel tempo.
findRow <- function(dt, df){
  max(which(df$Date.time <= dt))
}

# Righe:
rows <- sapply(tig, findRow, df = mel2)

# Dataset finale:
res <- cbind(1:length(tig), tig, mel2[rows, ])
names(res)[c(1, 2)] <- c("lapno", "racetime")

write.csv(res,
          "path/waus2017.csv",
          row.names = FALSE)
```

Codice 2: Conversione dei dati meteo 2014-2017.

Nel Codice 3 sono riportati i comandi per realizzare i grafici di tutte le figure presentate in fase di analisi esplorativa. È stato fatto largo uso del pacchetto ggplot2; ogni grafico è accompagnato dal relativo dataframe di riferimento.

```r
library(tidyverse)
source("source.lib")

# Caricamento datasets gare e qualifiche.
dir <- "..."
r <- read_delim(paste(dir, "/tesidsLT.csv", sep = ""),
                delim = ",",
                col_types = "cccccdcidddcild")
q <- read_delim(paste(dir, "/qualifyings.csv", sep = ""),
```

```
                 delim = ",",
                 col_types = "cicdddd")


# Creazione dataframe driver-driver_id.
drdrid <- tibble(r %>% select(driver_id, driver) %>% unique)

# Creazione dataframe race-race_id.
rrid <- tibble(r %>% select(race_id, race, season) %>%
               unique)

# Creazione dataframe con numero di giri dei GP.
nlaps <- rep(NA, nrow(rrid))
for(i in 1:nrow(rrid)){
  g <- r %>% filter(race_id == i) %>%
       summarise(max(lapno)) %>% as.numeric()
  nlaps[i] <- g
}
nl <- tibble(nlaps, rrid$race_id, rrid$season)
names(nl) <- c("nlaps", "race_id", "season")

# Numero di GP corsi in un certo circuito.
circuits <- tibble(sigle = r %>% select(race) %>% unique())
circuits <- circuits %>% mutate(ngps = rrid %>%
                                       select(race) %>%
                                       table())

# Molto spesso si dice che a Montecarlo il pilota in grado
# di fare il giro piu' veloce in qualifica ha la vittoria
# in tasca. Quante volte e' successo questo nelle stagioni
# dal 2014 al 2019?

# ID vincitori GP di Monaco.
mc <- winner(r, "MCO")

# Vincitori GP Monaco.
find_dr(mc)

# ID polemen GP di Monaco.
mcq <- polemen(q, find_rid("MCO"))

# Polemen GP Monaco.
find_dr(mcq)

# Ecco a confronto polemen e vincitori dei sei Gran Premi
# di Monaco considerati.

find_dr(mc); find_dr(mcq)
```

```r

# Si scopre che in sole 3 occasioni su 6 i polemen (Nico
# Rosberg, Daniel Ricciardo e Lewis Hamilton)
# hanno vinto a Monte-Carlo.

# Dataset con vincitori e polemen di ogni GP.
winpol <- tibble(race = NA, winner = NA, poleman = NA)
sigle <- as.character(rrid$race) %>% unique()
for(i in 1:length(sigle)){
  gp <- winner(r, sigle[i])
  qu <- polemen(q, find_rid(sigle[i]))
  winpol <- winpol %>% add_row(race = sigle[i],
                               winner = find_dr(gp),
                               poleman = find_dr(qu))
}
winpol <- winpol %>% na.omit()

# GP vinti dal poleman.
swp <- winpol %>% filter(winner == poleman) %>%
       group_by(race)

ngps <- rrid %>% select(race) %>% table()

# "Tasso di vittoria" 2014-2019 per circuito.
tasso <- table(swp$race)/ngps

# thl: "to highlight"
dft <- tibble(race = names(tasso),
              tasso = as.numeric(tasso))
dft <- dft %>% mutate(thl = ifelse(race == "MCO", 1, 0))

# Definiamo come tasso di conversione pole-vittoria la
# proporzione di successi avvenuta partendo dalla pole
# position. Come si pone Montecarlo rispetto a tutte le
# altre piste?
ggplot(data = dft, aes(x = reorder(race, tasso),
                       y = tasso,
                       fill = as.factor(thl))) +
  geom_bar(stat = 'identity') + coord_flip() +
  theme_light() + ylab("Tasso di conversione pole-vittoria") +
  xlab("Gara") +
  ggtitle("Tasso di conversione pole-vittoria per circuito
    (2014-2019)") +
  scale_fill_manual(values = c("gray25",
                               "tomato"), guide = "none")
# In Canada, in Brasile e negli Emirati Arabi uniti, per
# 5 Gran Premi su 6 il pilota partito dalla prima casella
# ha vinto la gara.
```

```r

# E' d'interesse scoprire quali sono i piloti ad aver
# concretizzato maggiormente le pole position in vittorie.

# Conteggio di GP vinti e non vinti in caso di pole position.
tvp <- winpol %>% group_by(poleman) %>%
       count((winner == poleman))
names(tvp) <- c("driver", "vfp", "nraces")

# Per Raikkonen e Massa si mette 0.001, per una migliore
# resa grafica.
tvpfd <- c(4/(4+7), 38/(38+19), 2/6, 0.001, 0.001, 1/3,
           13/26, 1/3, 5/12)
dftvp <- data.frame(cbind(c("BOT", "HAM", "LEC",
                            "MAS", "RAI", "RIC", "ROS",
                            "VER", "VET"),
                          tvpfd))
names(dftvp) <- c("driver", "wfp_rate")
dftvp$wfp_rate <- as.numeric(dftvp$wfp_rate)

# Il grafico seguente mostra come Lewis Hamilton
# sia il pilota con il rapporto
# pole su vittorie dalla pole piu' elevato. Seguono
# Rosberg e Sebastian Vettel. Felipe Massa e Kimi
# Raikkonen, invece, non sono mai stati in grado di
# vincere quando partivano dalla prima piazza.
ggplot(data = dftvp, aes(x = reorder(driver, wfp_rate),
                         y = wfp_rate,
                         fill = driver)) +
  geom_bar(stat = 'identity',
           colour = "black") + coord_flip() +
  theme_light() + ylab("Tasso di vittoria dalla pole") + xlab("Gara
    ") +
  ggtitle("Tasso di conversione pole-vittoria per pilota
    (2014-2019)") +
  scale_fill_manual(values = c("cyan", "cyan", "red",
                               "ghostwhite", "red",
                               "blue", "cyan", "blue",
                               "red"), guide = "none")

# Frequenza assoluta di gare vinte dai piloti.
nrw <- winpol %>% select(winner) %>% table()
dfnrw <- tibble(driver = names(nrw),
                nrw = as.numeric(nrw))

# Nel conteggio di gare vinte Lewis Hamilton non
# ha rivali. Interessante notare come in 121 Gran Premi
# vi siano stati solamente 8 differenti vincitori.
```

```
ggplot(data = dfnrw, aes(x = reorder(driver, nrw),
                         y = nrw,
                         fill = driver)) +
  geom_bar(stat = 'identity',
           colour = "black") + coord_flip() +
  theme_light() + ylab("Numero di gare vinte") +
  xlab("Pilota") +
  ggtitle("Numero di gare vinte dai piloti (2014-2019)") +
  scale_fill_manual(values = c("cyan", "cyan", "red",
                               "red", "blue", "cyan",
                               "blue", "red"),
                    guide = "none")

# Frequenze assolute di pole position.
npp <- winpol %>% select(poleman) %>% table()
dfnpp <- tibble(driver = names(npp),
                npp = as.numeric(npp))

# Anche per quanto riguarda il numero di pole position
# conquistate, il britannico Hamilton svetta sugli
# avversari.
ggplot(data = dfnpp, aes(x = reorder(driver, npp),
                         y = npp, fill = driver)) +
  geom_bar(stat = 'identity',
           colour = "black") + coord_flip() +
  theme_light() + ylab("Numero di pole position") +
  xlab("Pilota") +
  ggtitle("Numero di pole position per pilota (2014-2019)") +
  scale_fill_manual(values = c("cyan", "cyan", "red",
                               "ghostwhite", "red", "blue",
                               "cyan", "blue", "red"),
                    guide = "none")

# Caricamento dati meteo 2018-2019.
w2018 <- read_delim(paste(dir, "/all2018.csv", sep = ""),
                    delim = ",",
                    col_types = "ddddlddc")

w2019 <- read_delim(paste(dir, "/all2019.csv", sep = ""),
                    delim = ";",
                    col_types = "ddddlddc")

# Si vogliono ora esaminare alcune delle variabili relative
# alle condizioni atmosferiche dei Gran Premi di Formula 1.
# Si considera, ad esempio, il campionato 2018.

# Correlazione tra temperatura dell'aria e dell'asfalto:
cor(w2018$AirTemp, w2018$TrackTemp,
```

```
    method = "spearman") # 0.355

# Il seguente grafico di dispersione mostra la presenza
# di un lieve trend lineare, in quanto al crescere della
# temperatura dell'aria cresce anche quella dell'asfalto.
ggplot(data = w2018, aes(x = AirTemp, y = TrackTemp)) +
  geom_point(aes(colour = TrackTemp)) +
  scale_colour_gradient2(low = "blue", mid = "green" ,
                         high = "red", midpoint = 16) +
  ggtitle("Grafico di dispersione tra AirTemp e TrackTemp") +
  xlab(label = "Temperatura aria") +
  ylab(label = "Temperatura asfalto")

# Temperatura media dell'asfalto:
t2018 <- w2018 %>% group_by(Race) %>%
         summarise(mean(TrackTemp))
t2019 <- w2019 %>% group_by(Race) %>%
         summarise(mean(TrackTemp))
names(t2019) <- c("Race", "Mean_Track_Temperature")

# Si vuole determinare quali gare si corrono con alte
# temperature dell'asfalto. Si prendono in esame i dati
# relativi alla stagione 2019.
ggplot(t2019, aes(x = Race, y = Mean_Track_Temperature,
                  fill = Mean_Track_Temperature)) +
       geom_bar(stat = "identity", colour = "black") +
  ggtitle("Temperatura media dell'asfalto per circuito - 2019") +
  labs(fill = "Temperatura media") +
  scale_fill_gradient(high = "red", low = "yellow") +
  xlab(label = "Gara") + ylab(label = "Temperatura media dell'
   asfalto")

# Come la temperatura dell'asfalto influenza la scelta
# delle gomme? Si considerano quelle scelte da
# Lewis Hamilton per tutti i giri della stagione 2019.
ct <- tibble(compound = r[(r$season == "2019" &
                           r$lapno != 0 &
                           r$driver == "HAM"), ]$compound,
             race = r[(r$season == "2019" &
                       r$lapno != 0 &
                       r$driver == "HAM"), ]$race)
w2019 <- w2019 %>% arrange(Race)
ct <- ct %>% arrange(race)
ct <- ct %>% mutate(trackt = w2019$TrackTemp)

# Nel 2019 si usavano solo i compound A2 (il piu'
# duro), A3, A4, A6 e A7.
ggplot(ct, aes(x = compound, y = trackt,
```

```
                fill = compound)) +
  geom_boxplot() +
  ggtitle("Compound e temperatura asfalto - 2019") +
  xlab(label = "Compound") + ylab(label = "Temperatura dell'asfalto
   ")

# Si nota come la mescola piu' dura (A2) sia stata usata
# esclusivamente in condizioni di asfalto freddo,
# mentre e' apprezzabile la duttilita' delle mescole
# A3, A4 e A6 nell'adattarsi ad un range molto ampio
# di temperatura del tracciato. In ogni caso, la
# scelta degli pneumatici e' dettata da innumerevoli altri
# fattori, legati alla strategia, alle eventuali safety
# car, ai fenomeni di blistering e graining (usura).

# Infine, ci si chiede quale sia la posizione mediana
# occupata dai piloti nel totale dei loro giri percorsi
# durante i Gran Premi a cui hanno partecipato.
median_pos <- r %>% group_by(driver) %>%
              summarise(median(position))
names(median_pos) <- c("Driver", "Posizione_mediana")
median_pos <- median_pos %>% filter(Driver == "HAM" |
                                    Driver == "VET" |
                                    Driver == "RAI" |
                                    Driver == "RIC" |
                                    Driver == "VER" |
                                    Driver == "ROS" |
                                    Driver == "BOT" |
                                    Driver == "LEC" |
                                    Driver == "SAI" |
                                    Driver == "NOR" |
                                    Driver == "GRO" |
                                    Driver == "ALO")

ggplot(median_pos,
       aes(x =  reorder(Driver, Posizione_mediana),
                        y = Posizione_mediana,
           fill = Driver)) +
  geom_bar(stat = "identity") +
  scale_y_reverse(breaks = 1:13) +
  ggtitle("Posizione in classifica mediana di alcuni piloti,
   2014-2019") +
  xlab(label = "Pilota") + ylab(label = "Posizione mediana")
```

Codice 3: Analisi esplorative.

Per la modellizzazione del tempo sul giro mediante regressione robusta (Capitolo 3) sono stati considerati i tempi di Alonso e Michael Schumacher fatti registrare nel

circuito del Bahrain. Il listato seguente comprende anche i comandi per realizzare i grafici delle Figure 9, 10, 11 e 12.

```
# REGRESSIONE ROBUSTA

library(tidyverse)
library(robustbase)
library(ggpubr)
source("source.lib")
load(".RData")

# Selezioniamo i tempi sul giro di un pilota a caso,
# Fernando Alonso, durante il GP del Bahrain 2015.

aloLaps <- r %>% filter(driver == "ALO",
                        season == "2015",
                        race == "BHR") %>% pull(laptime)

aloLaps <- aloLaps[-c(1)]

dfaloLaps <- tibble(tempi = aloLaps,
                    nlap = seq(2, length(aloLaps)+1, by = 1))

ggplot(dfaloLaps, aes(x = nlap, y = tempi)) +
  geom_point(color = "#00AFBB", size = 2) +
  theme_light() + xlab("Giro") + ylab("Tempo (s)") +
  ggtitle("Tempi sul giro di Fernando Alonso in Bahrain - 2015") +
  scale_x_continuous(breaks = seq(min(dfaloLaps$nlap),
                                  max(dfaloLaps$nlap),
                                  by = 2))

ggplot(dfaloLaps, aes(x = nlap, y = tempi)) +
  geom_point(color = "#00AFBB", size = 2) +
  geom_smooth(method = "loess", se = F) +
  theme_light() + xlab("Giro") + ylab("Tempo (s)") +
  ggtitle("Tempi sul giro di Fernando Alonso in Bahrain - 2015")

fitA <- rlm(aloLaps ~ dfaloLaps$nlap)
summary(fitA)
confint.default(fitA)

raiLaps <- r %>% filter(driver == "RAI",
                        season == "2015",
                        race == "BHR") %>% pull(laptime)

raiLaps <- raiLaps[-c(1)]

fitR <- rlm(raiLaps ~ c(dfaloLaps$nlap, 58))
summary(fitR)
```

```r
confint.default(fitR)

ricLaps <- r %>% filter(driver == "RIC",
                        season == "2015",
                        race == "BHR") %>% pull(laptime)

ricLaps <- ricLaps[-c(1)]

fitRic <- rlm(ricLaps ~ c(dfaloLaps$nlap, 58))
summary(fitRic)
confint.default(fitRic)

malLaps <- r %>% filter(driver == "MAL",
                        season == "2015",
                        race == "BHR") %>% pull(laptime)

malLaps <- malLaps[-c(1)]

fitM <- rlm(malLaps ~ dfaloLaps$nlap)
summary(fitM)
confint.default(fitM)

rosLaps <- r %>% filter(driver == "ROS",
                        season == "2015",
                        race == "BHR") %>% pull(laptime)

rosLaps <- rosLaps[-c(1)]

fitRos <- rlm(rosLaps ~ c(dfaloLaps$nlap, 58))
fitRos # 100.5865, -0.0369
summary(fitRos)
confint.default(fitRos)

# Tempi di Schumacher in Bahrain nel 2004.
# Primo giro: 1.34.9 (eliminato)
# Ultimo giro: 1.39.9 (eliminato)
schumiLaps <- read.table("schumiBHR2004.txt", header = F)
names(schumiLaps) <- "tempo"
schumiLaps2 <- c(91.7, 91.1, 91, 90.7, 91.5, 90.3,
                 90.7, 93.5, 114.4, 92.3, 92.1, 91.6, 92,
                 91.8, 91.7, 91.4, 91.2, 91.5, 91.4, 90.9,
                 91.4, 91.2, 94.1, 115.4, 91.4, 91.8,
                 92.6, 91.9, 91.9, 91.6, 91.3, 91.5, 91.1,
                 91.4, 91.6, 91.5, 91.8, 92.5, 91, 93.8,
                 115.4, 92.5, 91.9, 92.6, 92.1, 91.8, 92.2,
                 92, 92.3, 92.1, 92.3, 92.8, 91.9, 91, 93.8)

dfSch <- tibble(tempi = schumiLaps2,
```

```
                nlap = 2:(length(schumiLaps2)+1))

ggplot(dfSch, aes(x = nlap, y = tempi)) +
  geom_point(color = "red", size = 2) +
  theme_light() + xlab("Giro") + ylab("Tempo (s)") +
  ggtitle("Tempi sul giro di Michael Schumacher in Bahrain - 2004") +
  scale_x_continuous(breaks = seq(min(dfSch$nlap),
                                  max(dfSch$nlap),
                                  by = 2))

# Interpolazione con modello lineare.
g1 <- ggplot(dfaloLaps, aes(x = nlap, y = tempi)) +
        geom_point(color = "#00AFBB", size = 2) +
        geom_smooth(method = "lm", se = F) +
        theme_light() + xlab("Giro") + ylab("Tempo (s)") +
        ggtitle("Alonso - Bahrain 2015")

g2 <- ggplot(dfSch, aes(x = nlap, y = tempi)) +
        geom_point(color = "red", size = 2) +
        geom_smooth(method = "lm", se = F, col = "red") +
        theme_light() + xlab("Giro") + ylab("Tempo (s)") +
        ggtitle("Schumacher - Bahrain 2004")

ggarrange(g1, g2, nrow = 2)

ggplot(dfSch, aes(x = nlap, y = tempi)) +
  geom_point(color = "red", size = 2) +
  geom_smooth(method = "loess", se = F, col = "red") +
  theme_light() + xlab("Giro") + ylab("Tempo (s)") +
  ggtitle("Schumacher - Bahrain 2004")


# Regressione robusta.

g3 <- ggplot(dfaloLaps, aes(x = nlap, y = tempi)) +
  geom_point(color = "#00AFBB", size = 2) +
  stat_smooth(method = "rlm", se = F) +
  theme_light() + xlab("Giro") + ylab("Tempo (s)") +
  ggtitle("Alonso - Bahrain 2015")

g4 <- ggplot(dfSch, aes(x = nlap, y = tempi)) +
  geom_point(color = "red", size = 2) +
  stat_smooth(method = "rlm", se = F, col = "red") +
  theme_light() + xlab("Giro") + ylab("Tempo (s)") +
  ggtitle("Schumacher - Bahrain 2004")

ggarrange(g3, g4, nrow = 2)
```

```
142
143 fitS <- ltsreg(schumiLaps2 ~ dfSch$nlap)
144 fitS
```

Codice 4: Regressione robusta per i tempi sul giro.

Di seguito, vi sono i comandi per costruire un dataset adatto alla crescita di un albero di classificazione e di una random forest. Viene riportato il procedimento per ottenere le previsioni per un singolo gran premio; modificando il nome del tracciato che compare nelle righe 312, 314, 333, 335 si possono trovare le previsioni per altre gare.

```
1  # Riduzione del dataset.
2  library(tidyverse)
3  source("source.lib")
4  load(".RData")
5
6  # Trova i vincitori di tutti i Gran Premi.
7  findWinner <- function(){
8    df <- r %>% group_by(season, race) %>%
9          filter(lapno == max(lapno), position == 1) %>%
10         select(race, season, driver, team,
11                racetime, tireage, lapno, position)
12 }
13
14 # Trova i secondi classificati di ogni gara.
15 findPodium2 <- function(){
16   df <- r %>% group_by(season, race) %>%
17         filter(lapno == max(lapno),
18                position == 2) %>%
19         select(race, season, driver, team,
20                racetime, tireage, lapno, position)
21 }
22
23 # Trova i terzi classificati di ogni gara.
24 findPodium3 <- function(){
25   df <- r %>% group_by(season, race) %>%
26         filter(lapno == max(lapno),
27                position == 3) %>%
28         select(race, season, driver, team,
29                racetime, tireage, lapno, position)
30 }
31
32 # Gli altri classificati:
33 findStandings <- function(){
34   df <- r %>% group_by(season, race, driver) %>%
35         filter(lapno == max(lapno)) %>%
36         select(race, season, driver, team,
37                racetime, tireage, lapno, position)
```

```
}

# Tibble con i 121 vincitori per ogni Gran Premio.
winners <- findWinner()

# Tibble con i secondi classificati.
podium2 <- findPodium2()

# Tibble con i terzi classificati.
podium3 <- findPodium3()

# Tibble con i piloti fuori dal podio.
stand <- findStandings()


# Caricamento dei dati meteo 2014, 2015, 2016, 2017.

meteo14 <- read_delim("all2014.csv", delim = ";")
meteo15 <- read_delim("all2015.csv", delim = ";")
meteo16 <- read_delim("all2016.csv", delim = ";")
meteo17 <- read_delim("all2017.csv", delim = ";")

dfRid <- rbind(winners, podium2, podium3, stand)
dfRid <- cbind(dfRid,
               airt    = rep(NA, nrow(dfRid)),
               rain    = rep(NA, nrow(dfRid)),
               relhum  = rep(NA, nrow(dfRid)),
               windsp  = rep(NA, nrow(dfRid)))

# Attenzione: questo ciclo for e i cicli for seguenti
# richiedono molto tempo computazionale per essere eseguiti.
ROT <- NULL
for(i in 1:2842){
  if(dfRid$season[i] == "2014"){
    meteo14g <- meteo14 %>%
      filter(Race == as.character(dfRid[i, 1]),
             lapno == ifelse(as.numeric(dfRid[i, 7]) == 0,
                             1, as.numeric(dfRid[i, 7])))
    dfRid$airt[i] <- meteo14g %>% pull(Temperature)
    ROT[i]        <- meteo14g %>% pull(Precipitation)
    dfRid$rain[i] <- ifelse(ROT[i] != 0 &
                            !is.na(ROT[i]), 1, 0)
    dfRid$relhum[i] <- meteo14g %>% pull(Relative.Humidity)
    dfRid$windsp[i] <- meteo14g %>% pull(Wind.Speed)
  }
}

ROT <- NULL
```

```r
for(i in 1:2842){
  if(dfRid$season[i] == "2015"){
    meteo15g <- meteo15 %>%
      filter(Race == as.character(dfRid[i, 1]),
             lapno == ifelse(as.numeric(dfRid[i, 7]) == 0,
                             1, as.numeric(dfRid[i, 7])))
    dfRid$airt[i] <- meteo15g %>% pull(Temperature)
    ROT[i]        <- meteo15g %>% pull(Precipitation)
    dfRid$rain[i] <- ifelse(ROT[i] != 0 &
                            !is.na(ROT[i]), 1, 0)
    dfRid$relhum[i] <- meteo15g %>% pull(Relative.Humidity)
    dfRid$windsp[i] <- meteo15g %>% pull(Wind.Speed)
  }
}

ROT <- NULL
for(i in 1:2842){
  if(dfRid$season[i] == "2016"){
    meteo16g <- meteo16 %>%
      filter(Race == as.character(dfRid[i, 1]),
             lapno == ifelse(as.numeric(dfRid[i, 7]) == 0,
                             1, as.numeric(dfRid[i, 7])))
    dfRid$airt[i] <- meteo16g %>% pull(Temperature)
    ROT[i]        <- meteo16g %>% pull(Precipitation)
    dfRid$rain[i] <- ifelse(ROT[i] != 0 &
                            !is.na(ROT[i]), 1, 0)
    dfRid$relhum[i] <- meteo16g %>% pull(Relative.Humidity)
    dfRid$windsp[i] <- meteo16g %>% pull(Wind.Speed)
  }
}

ROT <- NULL
for(i in 1:2842){
  if(dfRid$season[i] == "2017"){
    meteo17g <- meteo17 %>%
      filter(Race == as.character(dfRid[i, 1]),
             lapno == ifelse(as.numeric(dfRid[i, 7]) == 0,
                             1, as.numeric(dfRid[i, 7])))
    dfRid$airt[i] <- meteo17g %>% pull(Temperature)
    ROT[i]        <- meteo17g %>% pull(Precipitation)
    dfRid$rain[i] <- ifelse(ROT[i] != 0 &
                            !is.na(ROT[i]), 1, 0)
    dfRid$relhum[i] <- meteo17g %>% pull(Relative.Humidity)
    dfRid$windsp[i] <- meteo17g %>% pull(Wind.Speed)
  }
}

meteo18 <- read_delim("all2018.csv", delim = ",")
```

```
meteo19 <- read_delim("all2019.csv", delim = ",")

for(i in 1:2842){
  if(dfRid$season[i] == "2018"){
    meteo18g <- meteo18 %>%
      filter(Race == as.character(dfRid[i, 1]),
             LapNumber == ifelse(as.numeric(dfRid[i, 7]) == 0,
                                 1, as.numeric(dfRid[i, 7])))
    dfRid$airt[i] <- meteo18g %>% pull(AirTemp)
    dfRid$rain[i] <- ifelse(sum(meteo18g %>%
                                pull(Rainfall)) > 0, 1, 0)
    dfRid$relhum[i] <- meteo18g %>% pull(Humidity)
    dfRid$windsp[i] <- meteo18g %>% pull(WindSpeed)
  }
}

for(i in 1:2842){
  if(dfRid$season[i] == "2019"){
    meteo19g <- meteo19 %>%
      filter(Race == as.character(dfRid[i, 1]),
             LapNumber == ifelse(as.numeric(dfRid[i, 7]) == 0,
                                 1, as.numeric(dfRid[i, 7])))
    dfRid$airt[i] <- meteo19g %>% pull(AirTemp)
    dfRid$rain[i] <- ifelse(sum(meteo19g %>%
                                pull(Rainfall)) > 0, 1, 0)
    dfRid$relhum[i] <- meteo19g %>% pull(Humidity)
    dfRid$windsp[i] <- meteo19g %>% pull(WindSpeed)
  }
}

# Correzione manuale: gare bagnate.
# Ungheria 2014 - Giappone 2014
# Gran Bretagna 2015 - USA 2015
# Monaco 2016 - Gran Bretagna 2016 - Brasile 2016
# Cina 2017 e Singapore 2017
# Germania 2018
# Germania 2019

# Ciclo for lunghissimo.
for(i in 1:2842){
  if(dfRid[i, 1] == "HUN" & dfRid[i, 2] == "2014" |
     dfRid[i, 1] == "JAP" & dfRid[i, 2] == "2014" |
     dfRid[i, 1] == "GBR" & dfRid[i, 2] == "2015" |
     dfRid[i, 1] == "USA" & dfRid[i, 2] == "2015" |
     dfRid[i, 1] == "MCO" & dfRid[i, 2] == "2016" |
     dfRid[i, 1] == "GBR" & dfRid[i, 2] == "2016" |
     dfRid[i, 1] == "BRA" & dfRid[i, 2] == "2016" |
     dfRid[i, 1] == "CHN" & dfRid[i, 2] == "2017" |
```

```
     dfRid[i, 1] == "SGP" & dfRid[i, 2] == "2017" |
     dfRid[i, 1] == "GER" & dfRid[i, 2] == "2018" |
     dfRid[i, 1] == "GER" & dfRid[i, 2] == "2019" ){
    dfRid$rain[i] <- 1
  }
}

# Conversione in km orari della velocita' mediana del vento
# nel 2018 e nel 2019.
for(i in 1:2842){
  if(dfRid[i, 2] == "2018" | dfRid[i, 2] == "2019"){
    dfRid$windsp[i] <- 3.6 * dfRid$windsp[i]
    dfRid$windsp[i] <- 3.6 * dfRid$windsp[i]
  }
}

# FoP: Finishes on Podium
dfRid <- cbind(dfRid, FoP = rep(0, nrow(dfRid)))

# Top5: piloti finiti tra i primi 5
dfRid <- cbind(dfRid, Top5 = rep(0, nrow(dfRid)))

for(i in 1:2842){
  if(dfRid[i, 8] == 1 | dfRid[i, 8] == 2 |
     dfRid[i, 8] == 3 | dfRid[i, 8] == 4 |
     dfRid[i, 8] == 5){
    dfRid$Top5[i] <- 1
  }
  if(dfRid[i, 8] == 1 | dfRid[i, 8] == 2 |
     dfRid[i, 8] == 3 ){
    dfRid$FoP[i] <- 1
  }
}

View(dfRid)

# Il vettore assume valore TRUE se in un Gran Premio sono
# intervenute Safety Car o Virtual Safety Car e FALSE
# altrimenti.
isScVsc <- NULL
for(i in 1:2842){
  fcy <- r %>% filter(race   == as.character(dfRid[i, 1]),
                      season == as.character(dfRid[i, 2])) %>% pull
   (fcy)
  isScVsc[i] <- any(fcy == "SC" | fcy == "VSC")
  print(i)
}

```

```
dfRid <- cbind(dfRid, ScVsc = isScVsc)

dfRid <- dfRid %>% distinct()
# Per evitare che vengano inseriti due volte i piloti
# finiti sul podio.

# Bisogna aggiungere una variabile che indica se il
# pilota partiva dalla pole position o no.

pole <- rep(0, nrow(dfRid))

for(i in 1:nrow(dfRid)){
  if(find_dr(polemen(q, find_rid(dfRid$race[i],
                          y = dfRid$season[i]),
             y = dfRid$season[i])) == dfRid$driver[i]){
    pole[i] <- 1
  }
  print(i)
}

dfRid <- cbind(dfRid, isPoleman = pole)

# Troviamo la posizione in griglia di ogni pilota.
qualipos <- rep(0, nrow(dfRid))
for(i in 1:nrow(dfRid)){
  qp <- q %>% filter(driver_id ==
                       as.character(find_drid(dfRid$driver[i])),
                     race_id ==
                       as.character(find_rid(dfRid$race[i],
                                             dfRid$season[i]))) %>% pull(
   position)
  qualipos[i] <- qp
  print(i)
}
qualipos[704] <- 19 # Ricciardo squalificato
for(i in 705:707){
  qp <- q %>% filter(driver_id ==
                       as.character(find_drid(dfRid$driver[i])),
                     race_id ==
                       as.character(find_rid(dfRid$race[i],
                                             dfRid$season[i]))) %>%
    pull(position)
  qualipos[i] <- qp
  print(i)
}
qualipos[708] <- 20 # Vettel squalificato
for(i in 709:808){
  qp <- q %>% filter(driver_id ==
```

```
                        as.character(find_drid(dfRid$driver[i])),
                      race_id ==
                        as.character(find_rid(dfRid$race[i],
                                              dfRid$season[i]))) %>%
    pull(position)
  qualipos[i] <- qp
  print(i)
}
qualipos[809] <- 20 # Button fuori dal tempo limite

# Per risparmiare tempo computazionale assegnamo a
# piloti squalificati o non qualificati entro il
# tempo limite la posizione in griglia 20.
for(i in 810:nrow(dfRid)){
  qp <- q %>% filter(driver_id ==
                        as.character(find_drid(dfRid$driver[i])),
                      race_id ==
                        as.character(find_rid(dfRid$race[i],
                                              dfRid$season[i]))) %>%
    pull(position)
  ifelse(!is.na(qp), qualipos[i] <- qp, qualipos[i] <- 20)
  print(i)
}

dfRid <- cbind(dfRid, startPos = qualipos)

# Posizioni zero vanno ricodificate come 20.
for(i in 1:nrow(dfRid)){
  if(dfRid$startPos[i] == 0)
    dfRid$startPos[i] = 20
}

write.csv(dfRid, "datasetAlbero.csv",
          row.names = F) # dataset albero

library(rpart)
library(rpart.plot)

# Si riporta l'esempio del GP Australia 2019. Procedimento analogo
   per tutti gli altri Gran Premi.
train3 <- dfRid %>% filter(race == "AUS",
                           season != "2019")
test3  <- dfRid %>% filter(race == "AUS",
                           season == "2019")
row.names(test3) = test3$driver

t3 <- rpart(Top5 ~ team + tireage + airt + rain +
                   relhum + windsp + ScVsc + startPos +
```

```
                    isPoleman,
            data = train3,
            control = rpart.control(minsplit = 10,
                                    cp = .05,
                                    xval = 10))
rpart.plot(t3)

p3 <- predict(t3, newdata = test3)
sort(p3, decreasing = T)

# RANDOM FOREST - ESEMPIO GP AUSTRIA
library(randomForest)

traingp <- dfRid %>% filter(race == "AUT",
                            season != "2019")
testgp  <- dfRid %>% filter(race == "AUT",
                            season == "2019")

rf_fit <- randomForest(as.factor(Top5) ~ team + tireage +
                                         airt + rain +
                                         relhum + windsp +
                                         ScVsc + startPos +
                                         driver,
                       data = traingp,
                       mtry = 6, ntrees = 500)
rf_fit

plot(rf_fit)

pred.rf <- predict(rf_fit, newdata = testgp)
pred.rf
```

Codice 5: Costruzione dell'albero di classificazione e della Random Forest.

Nel Capitolo 3 sono presentati tre diversi modelli della classe GAMM, a cui si fa riferimento con i nomi di Modello 1, Modello 2 e Modello 3. Vengono illustrate la creazione del Modello 1, che ricalca quello costruito dai Prof. Vidoni e Casella nell'articolo Formula 1 lap time modeling using generalized additive models [3], e della simulazione di gara di Felipe Massa nel Gran Premio d'Italia del 2015 (per ulteriori dettagli si rimanda al paragrafo 3.4.3).

```
# Caricamento dei dati.
library(tidyverse)
# library(devtools)
# install_github("m-clark/gammit")
library(mgcv)
library(gammit)
source("source.lib")
load(".RData")
```

```
9
10 # Dataset con i tempi sul giro.
11 View(r)
12
13 # <<We take into account only lap times under a given
14 # threshold, since high values are usually related to
15 # unpredictable events, such as driver's error,
16 # car breakdown or car crash, and they can be interpreted
17 # as outliers which could generate a distortion in the
18 # model fitting procedure. Moreover, we do not consider
19 # as well the time of the first lap of the race since
20 # it can be viewed as a further outlier, strongly
21 # related to the qualifying position, and not
22 # particularly relevant for the specification of a good
23 # pit stop strategy.>>
24 r %>% select(laptime) %>% summary()
25
26 data <- r %>% filter(laptime < 130,
27                      !is.na(laptime),
28                      lapno != "0", lapno != "1")
29
30 # <<The covariate Distance is considered since the traffic
31 # could be a problem for drivers. We have empirically
32 # noticed that this effect is detectable only for distances
33 # in seconds less than a suitable value and then we decide
34 # to use this value also for greater distances, assuming
35 # that this is substantially equivalent to a clear track
36 # state.>>
37
38 # Distance e' il nostro "interval".
39
40 data <- data %>% mutate(interval = ifelse(interval >= 3.7,
41                                           3.7, interval))
42 data %>% select(interval) %>% summary()
43
44 # <<We aim at specifying a model for describing a race
45 # without safety car deployment and, for this reason,
46 # in the model fitting procedure, we omit all data related
47 # to laps completed under safety car regime.>>
48
49 data <- data %>% filter(fcy == "TC")
50
51 # Manca FollowingPit.
52 dataFP <- data %>% mutate(FollowingPit = 0)
53
54 for(i in 1:nrow(dataFP)){
55   if(dataFP$pit_this_lap[i] == TRUE){
56     pil <- as.character(dataFP$driver[i])
```

```
    gar <- as.character(dataFP$race[i])
    sta <- as.character(dataFP$season[i])
    gir <- dataFP$lapno[i]
    dataFP[dataFP$driver == pil &
             dataFP$season == sta &
             dataFP$race == gar &
             dataFP$lapno == gir+1, 18] = 1
  }
}

# <<The explanatory variable Lap is measured by evaluating
# the fraction of laps already covered by a particular
# driver in a specific grand prix.>>

data2 <- dataFP %>% group_by(race, season) %>%
  mutate(lapno = lapno/max(lapno))

data2 <- data2 %>% mutate(compound = factor(compound))

# Covariate del modello:
# lapno, team, driver, compound, tireage, pit_this_lap,
# interval, race, FollowingPit.

# <<We define a Generalized Additive Mixed Model (GAMM)
# with a random intercept term describing the driver
# effect.>>

# Consideriamo la stagione 2015 e le gare in Australia,
# Malesia, Italia, Russia ed Emirati Arabi Uniti.

dv <- data2 %>% filter(season == "2015",
                       race == "ITA" | race == "MYS" |
                       race == "ARE" | race == "AUS" |
                       race == "RUS")

fit2 <- gamm(log(laptime) ~ as.factor(team) + s(lapno) +
               FollowingPit*as.factor(race) +
               pit_this_lap*as.factor(race) +
               s(interval) +
               s(tireage, by = as.factor(compound)),
             random = list(driver = ~ 1),
             data = dv)
summary(fit2$gam)

plot(fit2$gam, select = 1, col = "blue",
     main = "Spline stimata della covariata lapno per il Modello 1"
   ,
     ylab = "s(lapno)")
```

```
104
105 # Per vedere i grafici con splines di tireage e compound
106 par(mfrow = c(2, 2))
107 plot(fit2$gam, ylab = "", select = 3, main = "A1 - Hardest")
108 plot(fit2$gam, ylab = "", select = 4, main = "A2")
109 plot(fit2$gam, ylab = "", select = 5, main = "A3")
110 plot(fit2$gam, ylab = "", select = 6, main = "A4 - Softest")
111
112
113 # Previsione dei tempi di Felipe Massa in Italia nel 2015.
114
115 massa <- dv %>% filter(race == "ITA",
116                        driver == "MAS")
117
118 dvRid <- dv %>% filter(!(race == "ITA" & driver == "MAS"))
119
120 fit3 <- gamm(log(laptime) ~ as.factor(team) + s(lapno) +
121                FollowingPit*as.factor(race) +
122                pit_this_lap*as.factor(race) +
123                s(interval) +
124                s(tireage, by = as.factor(compound)),
125              random = list(driver = ~ 1),
126              data = dvRid)
127
128 logT <- predict_gamm(fit3$gam, newdata = massa)
129 TMassa <- exp(logT) # Massa si e‘  fermato al giro 19
130
131 sum(TMassa) # 4641.36: tempo previsto di Massa
132
133 r %>% filter(race == "ITA", season == "2015",
134              driver == "MAS", lapno == 53) %>% pull(racetime) -
135 r %>% filter(race == "ITA", season == "2015",
136              driver == "MAS", lapno == 1) %>% pull(racetime)
137 # 4635.837: tempo effettivo di Massa (tolto il primo giro)
138
139
140 # Simulazione di gara di Massa
141 massaSim <- massa %>% mutate(pit_this_lap = FALSE,
142                              FollowingPit = 0)
143 tempiMAS <- NULL
144
145 originalTireAge <- massaSim %>% pull(tireage)
146
147 for(i in 1:(nrow(massa)-1)){
148   massaSim[i, 16] = TRUE   # pit_this_lap
149   massaSim[i+1, 18] = 1    # FollowingPit
150   massaSim[1:i, 14] = "A3" # compound prima del pit
151   massaSim[(i+1):nrow(massa), 14] = "A2" # compound dopo pit
```

```
  for(j in 1:i){
    massaSim[j, 15] = originalTireAge[1] + j - 1
  } # fix tireage
  for(j in (i+1):nrow(massa)){
    massaSim[j, 15] = 1 + (j-i-1)
  } # tireage after the pit
  logTSim <- predict_gamm(fit3$gam, newdata = massaSim)
  TMassaSim <- exp(logTSim)
  tempiMAS <- c(tempiMAS, sum(TMassaSim))
  massaSim <- massa %>% mutate(pit_this_lap = FALSE,
                               FollowingPit = 0,
                               tireage = originalTireAge)
}

tempiMAS # se Massa si fosse fermato al giro 23
         # avrebbe finito la gara prima, stando
         # al modello...ma doveva difendersi
         # dall'undercut di Rosberg

TMassa$prediction - massa$laptime
plot(1:52, TMassa$prediction)
points(1:52, massa$laptime, col = "red")

order(tempiMAS)-1 # il -1 deriva dal fatto che i giri
                  # partono dal secondo

dFtempiMAS <- tibble(tempiMAS,
                     giroPit = seq(1, 51, by = 1))

# Grafico.
library(png)
library(grid)
img <- readPNG(file.choose())
img.2 <- matrix(rgb(img[,,1], img[,,2], img[,,3], img[,,4] * 0.5),
                nrow = dim(img)[1])

plot.mas <- ggplot(dFtempiMAS, aes(y = tempiMAS, x = giroPit)) +
  annotation_custom(rasterGrob(img.2,
                               width = unit(1, "npc"),
                               height = unit(1, "npc")),
                    -Inf, Inf, -Inf, Inf) +
  geom_line() + geom_point() +
  ggtitle("Simulazione di gara di Massa - Italia 2015") +
  labs(x = "Giro del pit stop", y = "Tempo di gara")


# Con un procedimento analogo si possono vedere le strategie e le
   simulazioni di altri piloti.
```

```
199
200
201 ############ AMPLIAMENTO DEL DATASET ############
202
203 # Tempi sul giro - stagione 2019 (tolto il tempo di
204 # Leclerc a Monza). Rimuoviamo anche i giri fatti
205 # con gomme da bagnato, altrimenti l'algoritmo non
206 # converge.
207 lt2019 <- data2 %>% filter(season == "2019",
208                            compound != "I", compound != "W")
209 lt2019v2 <- lt2019 %>% mutate(compound = fct_collapse(compound,
210                                          hard = c("A2", "A3"),
211                                          medium = "A4",
212                                          soft = c("A6", "A7")))
213 training <- lt2019v2 %>% filter(!(race == "ITA" &
214                                   driver == "LEC"))
215 leclercv2 <- lt2019v2 %>% filter(race == "ITA",
216                                  driver == "LEC")
217
218 fit4 <- gamm(log(laptime) ~ as.factor(team) + s(lapno) +
219                FollowingPit*as.factor(race) +
220                pit_this_lap*as.factor(race) +
221                s(interval) +
222                s(tireage, by = as.factor(compound)),
223              random = list(driver = ~ 1),
224              data = training)
225 # Attenzione: la stima del modello richiede un lungo tempo.
226
227 par(mfrow = c(3, 1))
228 plot(fit4$gam, scale = 0, ylab = "", select = 3,
229      main = "hard")
230 plot(fit4$gam, scale = 0, ylab = "", select = 4,
231      main = "medium")
232 plot(fit4$gam, scale = 0, ylab = "", select = 5,
233      main = "soft")
234 # Si impone scale = 0 per vedere ogni spline con
235 # ylim diversi (migliore rappresentazione).
236
237 LeclogT <- predict_gamm(fit4$gam, newdata = leclercv2)
238 TLec <- exp(LeclogT)
239 TLec
240
241 cbind(TLec, leclercv2$laptime)
242 TLec - leclercv2$laptime # risultati poco soddisfacenti
243
244 # Proviamo modello tenendo mescole originali di compound.
245
246 training <- lt2019 %>% filter(!(race == "ITA" &
```

```
                                    driver == "LEC"))
leclerc <- lt2019 %>% filter(race == "ITA",
                             driver == "LEC")

fit4 <- gamm(log(laptime) ~ as.factor(team) + s(lapno) +
               FollowingPit*as.factor(race) +
               pit_this_lap*as.factor(race) +
               s(interval) +
               s(tireage, by = as.factor(compound)),
             random = list(driver = ~ 1),
             data = training)

LeclogT <- predict_gamm(fit4$gam, newdata = leclercv2)
TLec <- exp(LeclogT)
TLec

cbind(TLec, leclerc$laptime)
TLec - leclerc$laptime

plot(1:50, TLec$prediction)
points(1:50, leclerc$laptime, col = "red")
# Meglio previsioni per gomma hard rispetto a gomma soft.
# Passo molto buono di Leclerc a inizio gara.
```

Codice 6: Modello GAMM 1.

Nel Codice 7 è riportato il procedimento per realizzare il Modello 2 a partire dai dati del dataset Race 2. Dopo il caricamento dei files .csv si manipolano i dati rimuovendo i NA, convertendo i tempi in secondi e aggiungendo alcune variabili d'interesse per l'analisi. In seguito si stimano i parametri del modello e si realizza il grafico dei tempi di Leclerc a Monza nel 2019 (Figura 16).

```
library(tidyverse)
library(gammit)
library(png)
library(grid)
library(ggpubr)

# Load data
china <- read.csv("C:/formulauno/chnFull.csv", header = T,
                  na.strings = c(""))
australia <- read.csv("C:/formulauno/ausFull.csv", header = T,
                      na.strings = c(""))
russia <- read.csv("C:/formulauno/rusFull.csv", header = T,
                   na.strings = c(""))
abudhabi <- read.csv("C:/formulauno/areFull.csv", header = T,
                     na.strings = c(""))
monza <- read.csv("C:/formulauno/itaFull.csv", header = T,
                  na.strings = c(""))
```

```
18
19 # Clean data
20 clean.data <- function(df){
21   df <- df %>% filter(LapNumber != 1)
22   df <- df %>% select(-c(Sector1SessionTime,
23                          Sector2SessionTime,
24                          Sector3SessionTime,
25                          LapStartDate, LapStartTime,
26                          IsAccurate, Time))
27   df <- df %>%
28     mutate(PitOutTime = replace(PitOutTime,
29                                 which(!is.na(PitOutTime)),
30                                 1))
31   df <- df %>%
32     mutate(PitOutTime = replace(PitOutTime,
33                                 which(is.na(PitOutTime)),
34                                 0))
35   df <- df %>%
36     mutate(PitInTime = replace(PitInTime,
37                                which(!is.na(PitInTime)),
38                                1))
39   df <- df %>%
40     mutate(PitInTime = replace(PitInTime,
41                                which(is.na(PitInTime)),
42                                0))
43   df <- df %>% rename(EnteringPit = PitInTime)
44   df <- df %>% rename(ExitingPit = PitOutTime)
45 }
46
47 china <- clean.data(china)
48 abudhabi <- clean.data(abudhabi)
49 monza <- clean.data(monza)
50 australia <- clean.data(australia)
51 russia <- clean.data(russia)
52
53 # Convert times
54 library(qdapTools)
55 fix.times <- function(df){
56   df$LapTime <- gsub("0 days ", "", df$LapTime)
57   df$Sector1Time <- gsub("0 days ", "", df$Sector1Time)
58   df$Sector2Time <- gsub("0 days ", "", df$Sector2Time)
59   df$Sector3Time <- gsub("0 days ", "", df$Sector3Time)
60   df$LapTime <- hms2sec(df$LapTime)
61   df$Sector1Time <- hms2sec(df$Sector1Time)
62   df$Sector2Time <- hms2sec(df$Sector2Time)
63   df$Sector3Time <- hms2sec(df$Sector3Time)
64   return(df)
65 }
```

```
66
67 china <- fix.times(china)
68 abudhabi <- fix.times(abudhabi)
69 monza <- fix.times(monza)
70 australia <- fix.times(australia)
71 russia <- fix.times(russia)
72
73 # Add circuit
74 china <- china %>% mutate(Circuit = "CHN")
75 monza <- monza %>% mutate(Circuit = "ITA")
76 australia <- australia %>% mutate(Circuit = "AUS")
77 abudhabi <- abudhabi %>% mutate(Circuit = "ARE")
78 russia <- russia %>% mutate(Circuit = "RUS")
79
80 # Add fast sectors
81 add.sectors <- function(df){
82   df <- df %>% add_column(FastSector1 = 0,
83                           .after = "Sector1Time")
84   df <- df %>% add_column(FastSector2 = 0,
85                           .after = "Sector2Time")
86   df <- df %>% add_column(FastSector3 = 0,
87                           .after = "Sector3Time")
88   df
89 }
90
91 china <- add.sectors(china)
92 australia <- add.sectors(australia)
93 monza <- add.sectors(monza)
94 russia <- add.sectors(russia)
95 abudhabi <- add.sectors(abudhabi)
96
97 piloti <- c(unique(australia %>% pull(Driver)), "GRO", "GAS")
98
99 dataset <- tibble()
100
101 find.personal.best <- function(gara){
102   for(i in 1:20){
103     df <- gara %>% filter(Driver == piloti[i])
104     for(j in 1:nrow(df)){
105       if(isTRUE(df$Sector1Time[j] <= min(df$Sector1Time[1:j])))
106         df$FastSector1[j] = 1
107       if(isTRUE(df$Sector2Time[j] <= min(df$Sector2Time[1:j])))
108         df$FastSector2[j] = 1
109       if(isTRUE(df$Sector3Time[j] <= min(df$Sector3Time[1:j])))
110         df$FastSector3[j] = 1
111     }
112     dataset <- dataset %>% bind_rows(df)
113   }
```

```
  dataset
}

chn <- find.personal.best(china)
aus <- find.personal.best(australia)
ita <- find.personal.best(monza)
rus <- find.personal.best(russia)
are <- find.personal.best(abudhabi)

# Join datasets
data <- bind_rows(chn, ita, aus, are, rus)

# Select only track clear laps
data2 <- data %>% filter(TrackStatus == "1")

# Lap-time model
library(mgcv)

fit <- gamm(log(LapTime) ~ as.factor(Team) + s(LapNumber) +
              ExitingPit*as.factor(Circuit) +
              EnteringPit*as.factor(Circuit) +
              s(TyreLife,
                by = interaction(Compound, FreshTyre)),
            random = list(Driver = ~ 1),
            data = data2)
plot(fit$gam, scale = 0)

# Add sector times effects...
fit2 <- gamm(log(LapTime) ~ as.factor(Team) + s(LapNumber) +
               ExitingPit*as.factor(Circuit) +
               EnteringPit*as.factor(Circuit) +
               s(TyreLife,
                 by = interaction(Compound, FreshTyre)) +
               as.factor(FastSector1)*ExitingPit +
               as.factor(FastSector2)*ExitingPit +
               as.factor(FastSector3)*ExitingPit,
             random = list(Driver = ~ 1),
             data = data2)
summary(fit2$gam)

plot(fit2$gam, scale = 0)

# Race pace - Leclerc, Monza

training <- data2 %>% filter(!(Circuit == "ITA" &
                                 Driver == "LEC"))
test     <- data2 %>% filter(Circuit == "ITA" & Driver == "LEC")

```

```
fit3 <- gamm(log(LapTime) ~ as.factor(Team) + s(LapNumber) +
               ExitingPit*as.factor(Circuit) +
               EnteringPit*as.factor(Circuit) +
               s(TyreLife,
                 by = interaction(Compound, FreshTyre)) +
               as.factor(FastSector1)*ExitingPit +
               as.factor(FastSector2)*ExitingPit +
               as.factor(FastSector3)*ExitingPit,
             random = list(Driver = ~ 1),
             data = training)

logT <- predict_gamm(fit3$gam, newdata = test)
tExpect <- exp(logT)
tExpect

cbind(tExpect, test$LapTime)

plot(1:45, tExpect$prediction, col = "blue", type = "b")
lines(1:45, test$LapTime, type = "b")

dfLecPred <- data.frame(cbind(pred = tExpect$prediction,
                              lap = test$LapTime))

img <- readPNG("lecMonza2019.png")
img.2 <- matrix(rgb(img[,,1], img[,,2], img[,,3], img[,,4] * 0.5),
                nrow = dim(img)[1])

plot.lec <- ggplot() +
  annotation_custom(rasterGrob(img.2,
                               width = unit(1, "npc"),
                               height = unit(1, "npc")),
                    -Inf, Inf, -Inf, Inf) +
  geom_line(data = dfLecPred, aes(x = seq(1:45), y = pred),
            color = "red", size = 1) +
  geom_point(data = dfLecPred, aes(x = seq(1:45), y = pred),
            color = "red", size = 1) +
  geom_line(data = dfLecPred, aes(x = seq(1:45), y = lap),
           color = "black", size = 1) +
  geom_point(data = dfLecPred, aes(x = seq(1:45), y = lap),
            color = "black", size = 1) +
  labs(x = "Giro", y = "Tempo",
       subtitle = "In rosso i tempi previsti") +
  ggtitle("Tempi sul giro di Leclerc - Monza 2019")
```

Codice 7: Modello GAMM 2.

Infine, il Codice 8 è dedicato al Modello 3. I files che terminano in "joined.csv" sono già comprensivi delle informazioni meteo giro dopo giro, per ogni pilota. La

procedura di "pulizia" del dato per arrivare al dataset utilizzato per la stima è analoga a quella effettuata per il Modello 2.

```
library(tidyverse)
library(qdapTools)
library(mgcv)
library(gammit)

# Impostare la cartella contenente tutti i singoli files .csv.
setwd(choose.dir())

# Load data
temp = list.files(pattern = "*.csv")

# Singoli files
for (i in 1:length(temp)) assign(temp[i],
                                 read.csv(temp[i], header = T,
                                          na.strings = c("")))
abujoined.csv <- abujoined.csv %>% mutate(Circuit = "ARE")
australiajoined.csv <- australiajoined.csv %>% mutate(Circuit = "
    AUS")
austriajoined.csv <- austriajoined.csv %>% mutate(Circuit = "AUT")
azerbaijanjoined.csv <- azerbaijanjoined.csv %>% mutate(Circuit = "
    AZE")
bahrainjoined.csv <- bahrainjoined.csv %>% mutate(Circuit = "BHR")
belgiumjoined.csv <- belgiumjoined.csv %>% mutate(Circuit = "BEL")
braziljoined.csv <- braziljoined.csv %>% mutate(Circuit = "BRA")
canadajoined.csv <- canadajoined.csv %>% mutate(Circuit = "CAN")
chinajoined.csv <- chinajoined.csv %>% mutate(Circuit = "CHN")
francejoined.csv <- francejoined.csv %>% mutate(Circuit = "FRA")
germanyjoined.csv <- germanyjoined.csv %>% mutate(Circuit = "GER")
hungaryjoined.csv <- hungaryjoined.csv %>% mutate(Circuit = "HUN")
itajoined.csv <- itajoined.csv %>% mutate(Circuit = "ITA")
japanjoined.csv <- japanjoined.csv %>% mutate(Circuit = "JAP")
mexicojoined.csv <- mexicojoined.csv %>% mutate(Circuit = "MEX")
monacojoined.csv <- monacojoined.csv %>% mutate(Circuit = "MCO")
russiajoined.csv <- russiajoined.csv %>% mutate(Circuit = "RUS")
silverstonejoined.csv <- silverstonejoined.csv %>% mutate(Circuit =
     "GBR")
singaporejoined.csv <- singaporejoined.csv %>% mutate(Circuit = "
    SGP")
spainjoined.csv <- spainjoined.csv %>% mutate(Circuit = "ESP")
usajoined.csv <- usajoined.csv %>% mutate(Circuit = "USA")

# Dataframe con tutte le gare
races <- tibble()
races <- races %>% bind_rows(abujoined.csv) %>%
  bind_rows(australiajoined.csv) %>% bind_rows(azerbaijanjoined.csv
    ) %>%
```

```
  bind_rows(bahrainjoined.csv) %>% bind_rows(belgiumjoined.csv) %>%
  bind_rows(braziljoined.csv) %>% bind_rows(canadajoined.csv) %>%
  bind_rows(chinajoined.csv) %>% bind_rows(francejoined.csv) %>%
  bind_rows(germanyjoined.csv) %>% bind_rows(hungaryjoined.csv) %>%
  bind_rows(itajoined.csv) %>% bind_rows(japanjoined.csv) %>%
  bind_rows(mexicojoined.csv) %>% bind_rows(monacojoined.csv) %>%
  bind_rows(russiajoined.csv) %>% bind_rows(silverstonejoined.csv)
   %>%
  bind_rows(singaporejoined.csv) %>% bind_rows(spainjoined.csv) %>%
  bind_rows(usajoined.csv)

clean.data <- function(df){
  df <- df %>% filter(LapNumber != 1)
  df <- df %>% select(-c(Sector1SessionTime,
                         Sector2SessionTime,
                         Sector3SessionTime,
                         LapStartDate, LapStartTime,
                         IsAccurate, Time))
  df <- df %>%
    mutate(PitOutTime = replace(PitOutTime,
                                which(!is.na(PitOutTime)),
                                1))
  df <- df %>%
    mutate(PitOutTime = replace(PitOutTime,
                                which(is.na(PitOutTime)),
                                0))
  df <- df %>%
    mutate(PitInTime = replace(PitInTime,
                               which(!is.na(PitInTime)),
                               1))
  df <- df %>%
    mutate(PitInTime = replace(PitInTime,
                               which(is.na(PitInTime)),
                               0))
  df <- df %>% rename(EnteringPit = PitInTime)
  df <- df %>% rename(ExitingPit = PitOutTime)
}

races <- clean.data(races)

# Convert times
fix.times <- function(df){
  df$LapTime <- gsub("0 days ", "", df$LapTime)
  df$Sector1Time <- gsub("0 days ", "", df$Sector1Time)
  df$Sector2Time <- gsub("0 days ", "", df$Sector2Time)
  df$Sector3Time <- gsub("0 days ", "", df$Sector3Time)
  df$LapTime <- hms2sec(df$LapTime)
  df$Sector1Time <- hms2sec(df$Sector1Time)
```

```
  df$Sector2Time <- hms2sec(df$Sector2Time)
  df$Sector3Time <- hms2sec(df$Sector3Time)
  return(df)
}

races <- fix.times(races)

# Add fast sectors
add.sectors <- function(df){
  df <- df %>% add_column(FastSector1 = 0,
                          .after = "Sector1Time")
  df <- df %>% add_column(FastSector2 = 0,
                          .after = "Sector2Time")
  df <- df %>% add_column(FastSector3 = 0,
                          .after = "Sector3Time")
  df
}

races <- add.sectors(races)

piloti <- unique(australiajoined.csv %>% pull(Driver))
gare   <- cbind("AUS", "ARE", "AUT", "AZE", "BHR", "BEL", "BRA",
                "CAN", "CHN", "FRA", "GER", "HUN", "ITA", "JAP",
                "MEX", "MCO", "RUS", "GBR", "SGP", "ESP", "USA")

dataset <- tibble()

find.personal.best <- function(g){
  for(j in 1:21){
    df <- g %>% filter(Circuit == gare[j])
    for(i in 1:20){
      df2 <- df %>% filter(Driver == piloti[i])
      for(j in 1:nrow(df)){
        if(isTRUE(df2$Sector1Time[j] <= min(df2$Sector1Time[1:j])))
          df2$FastSector1[j] = 1
        if(isTRUE(df2$Sector2Time[j] <= min(df2$Sector2Time[1:j])))
          df2$FastSector2[j] = 1
        if(isTRUE(df2$Sector3Time[j] <= min(df2$Sector3Time[1:j])))
          df2$FastSector3[j] = 1
      }
      dataset <- dataset %>% bind_rows(df2)
    }
  }
  dataset
}

prova <- races

```

```r
find.personal.best <- prova %>% group_by(Circuit, Driver) %>%
  summarise(PB1)

races2 <- find.personal.best(races)

# Select only track clear laps
races3 <- races2 %>% filter(TrackStatus == "1")

# Aggiunta covariata lapno (frazione di giri completati)
races4 <- races3 %>% group_by(Circuit) %>%
  mutate(lapno = LapNumber/max(LapNumber))

races5 <- races4 %>% filter(LapTime < 130, !is.na(LapTime))

write.csv(races5, file = "2019sectorsWeather.csv",
          row.names = F)

# Per caricamento successivo:
races5 <- read.csv(file.choose(), header = T)
View(races5)
#################################################################

# Aggiungiamo un "HeatIndex" calcolato come media
# tra temperatura dell'aria e dell'asfalto.
races5 <- races5 %>% mutate(HeatIndex = (TrackTemp + AirTemp)/2)

# GAMM model with weather effects
fit <- gamm(log(LapTime) ~ as.factor(Team) + s(lapno) +
              ExitingPit*as.factor(Circuit) +
              EnteringPit*as.factor(Circuit) +
              s(TyreLife,
                by = interaction(Compound, FreshTyre)) +
              as.factor(FastSector1)*ExitingPit +
              as.factor(FastSector2)*ExitingPit +
              as.factor(FastSector3)*ExitingPit +
              Humidity + HeatIndex + WindSpeed +
              Rainfall,
            random = list(Driver = ~ 1),
            data = races5)
summary(fit$gam)

# WindSpeed e TrackTemp non risultano significativa
# nello spiegare i tempi sul giro.
# Nemmeno RainFall, ma sembra strano non includerla
# nel modello...

cor(races5$TrackTemp, races5$AirTemp, method = "spearman")
cor(races5$HeatIndex, races5$Humidity, method = "spearman")
```

```
185
186 fit <- gamm(log(LapTime) ~ as.factor(Team) + s(lapno) +
187               ExitingPit*as.factor(Circuit) +
188               EnteringPit*as.factor(Circuit) +
189               s(TyreLife,
190                 by = interaction(Compound, FreshTyre)) +
191               as.factor(FastSector1)*ExitingPit +
192               as.factor(FastSector2)*ExitingPit +
193               as.factor(FastSector3)*ExitingPit +
194               AirTemp,
195             random = list(Driver = ~ 1),
196             data = races5)
197 summary(fit$gam)
198 summary(fit$gam)$p.coeff
199
200 # Al crescere della temperatura dell'aria emerge una
201 # crescita del tempo sul giro.
202 library(png)
203 hard <- readPNG(file.choose())
204 medium <- readPNG(file.choose())
205 soft <- readPNG(file.choose())
206
207 par(mfrow = c(1, 2))
208 plot(fit$gam, scale = 0, select = 2, main = "HARD usate",
209      ylab = "s(TyreLife)", col = "orange", lwd = 2)
210 # rasterImage(hard, 3, 0.008, 15, 0.018)
211 plot(fit$gam, scale = 0, select = 5, main = "HARD nuove",
212      ylab = "s(TyreLife)", col = "#15B361", lwd = 2)
213 # rasterImage(hard, 3, 0.006, 15, 0.014)
214 plot(fit$gam, scale = 0, select = 3, main = "MEDIUM usate",
215      ylab = "s(TyreLife)", col = "orange", lwd = 2)
216 # rasterImage(medium, 3, 0.007, 15, 0.030)
217 plot(fit$gam, scale = 0, select = 7, main = "MEDIUM nuove",
218      ylab = "s(TyreLife)", col = "#15B361", lwd = 2)
219 # rasterImage(medium, 3, 0.004, 15, 0.012)
220 plot(fit$gam, scale = 0, select = 4, main = "SOFT usate",
221      ylab = "s(TyreLife)", col = "orange", lwd = 2)
222 # rasterImage(soft, 3, 0.31, 15, 0.55)
223 plot(fit$gam, scale = 0, select = 8, main = "SOFT nuove",
224      ylab = "s(TyreLife)", col = "#15B361", lwd = 2)
225 # rasterImage(soft, 3, 0.01, 15, 0.022)
226
227 # All'aumentare dei giri percorsi diminuisce il tempo sul
228 # giro. Hard usate: usura uniforme. Le medium usate
229 # garantiscono tempi veloci per i primi 11/12 giri,
230 # poi iniziano lentamente a degradarsi. Le soft usate
231 # hanno un crollo verticale dopo un numero di giri
232 # maggiore a 40 (successo infatti solo con Albon
```

```
# a Montecarlo). Le gomme nuove hanno usura uniforme.

races5 %>% filter(Compound == "SOFT" & TyreLife > 40)


# Altri piloti.

training <- races5 %>% filter(!(Circuit == "ITA" &
                                Driver == "RAI"))
test     <- races5 %>% filter(Circuit == "ITA" & Driver == "RAI")

fit3 <- gamm(log(LapTime) ~ as.factor(Team) + s(lapno) +
               ExitingPit*as.factor(Circuit) +
               EnteringPit*as.factor(Circuit) +
               s(TyreLife,
                 by = interaction(Compound, FreshTyre)) +
               as.factor(FastSector1)*ExitingPit +
               as.factor(FastSector2)*ExitingPit +
               as.factor(FastSector3)*ExitingPit +
               AirTemp,
             random = list(Driver = ~ 1),
             data = races5)

logT <- predict_gamm(fit3$gam, newdata = test)
tExpect <- exp(logT)
tExpect

cbind(tExpect, test$LapTime)

plot(1:45, tExpect$prediction, col = "blue", type = "b")
lines(1:45, test$LapTime, type = "b")

# Ottimi risultati per Norris, Leclerc, Hamilton, Vettel,
# Kubica, Raikkonen.
```

Codice 8: Modello GAMM 3.

Ringraziamenti

Desidero ringraziare innanzitutto il Prof. Mauro Bernardi per avermi seguito in questo lavoro con estrema disponibilità.

Grazie ai miei genitori e a mia sorella Ilaria per il supporto, e grazie anche ai miei nonni. Grazie ai miei compagni di corso e amici, in particolare a Federico, Fabio, Vittorio, Arianna, Emma.

Infine un ringraziamento speciale a mio zio, Don Giuseppe, e ai professori che ho incontrato durante il mio percorso scolastico e universitario, tra cui la maestra Annamaria Nucibella, le Prof.sse Antonella Capparotto, Cristina Trivellin, Alessandra Salvan e i Prof. Francesco Mantoan, Marco Rinaldi, Mario Zanellato.