

F1WP – Formula 1 Winner Prediction

Progetto per il corso di Fondamenti di Intelligenza Artificiale A.A.

2021/2022 di Giorgio Angelo Esposito, matricola 0512107389

<https://github.com/giorgio-angelo-esposito/F1WP.git>

1. INTRODUZIONE

La Formula Uno è uno sport automobilistico nato ufficialmente nel 1950 ed è attualmente lo sport automobilistico di più alta categoria per quanto riguarda le vetture monoposto a ruota scoperta da corsa su circuito.

Il termine “*Formula*” si riferisce all’insieme di regole che i partecipanti (team e piloti) devono rispettare.

Nel tempo le vetture si sono evolute molto diventando, nell’ultimo decennio, l’apice della tecnologia nelle corse automobilistiche.

Una gara di Formula Uno non comprende però, solo la gara stessa: un *Gran Premio* occupa un intero *weekend*: si inizia il giovedì con le interviste ai piloti, prove libere il venerdì e il sabato mattina, qualifiche il sabato pomeriggio e gara la domenica. Data la natura estremamente dinamica dello sport, fatta di sorpassi, incidenti e guasti, viene naturale provare a prevedere chi tra i partecipanti alla gara sarà il vincitore, cosa non sempre scontata.

2. DESCRIZIONE DELL’AGENTE E SCELTA DEGLI STRUMENTI

2.1 DESCRIZIONE DELL’AGENTE

L’obiettivo del progetto è quindi quello di realizzare un agente capace di determinare quale pilota vincerà un *Gran Premio*.

Andiamo ora a definire la misura PEAS (**P**erformance, **E**nvironment, **A**ctuators, **S**ensors) dell’agente:

TABELLA DELLA MISURA PEAS

PERFORMANCE	La misura di performance dell'agente è la sua capacità di predire correttamente il vincitore di un Gran Premio di Formula Uno
ENVIRONMENT	L'ambiente in cui opera l'agente è: <ul style="list-style-type: none"> • OSSERVABILE: l'agente ha sempre accesso a tutti i dati che ha a disposizione • DISCRETO: l'agente ha un numero limitato di dati da cui apprendere • AGENTE SINGOLO • STATICO: l'ambiente non cambia mentre l'agente sta apprendendo
ACTUATORS	L'attuatore dell'agente corrisponde alla predizione effettuata
SENSORS	I sensori dell'agente sono l'insieme di dati passati in input al machine learner per apprendere

2.2 STRUMENTI UTILIZZATI

Nell'ambiente del Machine Learning negli ultimi anni è divenuto popolare il linguaggio di programmazione Python. Dotato di una vasta gamma di librerie, si è rivelato essere ottimo per tutti gli appassionati della materia.

Per questo si è scelto di utilizzare tale linguaggio nello sviluppo del progetto.

Come ambiente si è utilizzato Google Colab: Colab permette di eseguire codice Python direttamente nel browser, sotto forma di notebook, fornendo accesso gratuito alle GPU di Google, senza nessuna configurazione necessaria.

Nell'ambito del Machine Learning abbiamo già detto che Python offre un'ampia gamma di librerie utili a processare visualizzare i dati, fare predizioni su di essi, ecc... Le librerie che verranno utilizzate sono le seguenti:

- pandas: libreria utilizzata per l'analisi dei dati e la loro manipolazione
- matplotlib: libreria utilizzata per la creazione di grafici in Python
- seaborn: libreria per la visualizzazione di dati che si basa su matplotlib
- scikit-learn: libreria che permette di utilizzare gli algoritmi di Machine Learning in Python

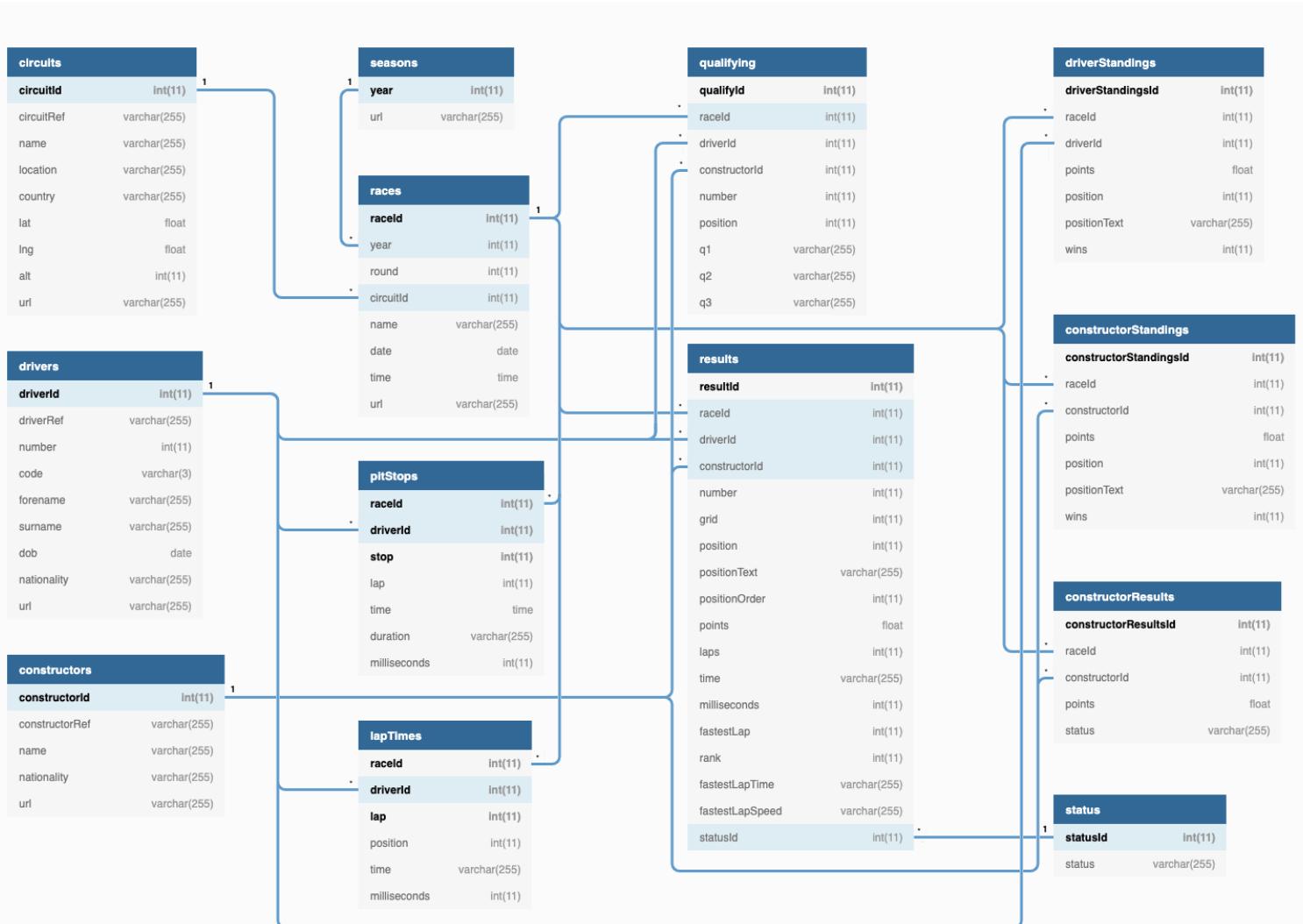
3. RACCOLTA E ANALISI DEI DATI

3.1 SCELTA DEL DATASET

Passiamo ora al dataset che andremo a utilizzare per i nostri scopi. Dopo aver cercato sui vari siti dedicati (Kaggle, Google Dataset Search, ecc...), la scelta è ricaduta su un *web service* chiamato Ergast Developer API (<http://ergast.com/mrd/>). Ergast fornisce dati storici relativi alle corse automobilistiche e, oltre a poter interrogare il database mediante query parametriche, permette di scaricare i dati in formato .csv (comma separated value).

Trattandosi di un database relazionale, avremo diverse tabelle su cui eseguire le tecniche di Data Analysis.

Di seguito, la struttura del database.



Procediamo ora ad analizzare tabella per tabella, esaminando la loro struttura, i dati contenuti e eventuali relazioni tra i dati.

3.2 DESCRIZIONE E ANALISI TABELLE

Andiamo adesso a descrivere e esaminare le tabelle che compongono il database. Ergast mette a disposizione una descrizione delle sue tabelle, e per ognuna di queste andremo a esaminare il contenuto, visualizzando i dati, cercando relazioni tra di essi ed eventuali dati mancanti.

Come strumenti useremo il linguaggio Python e le sue librerie pandas e matplotlib.

TABELLA CIRCUITS

circuits table							
Field	Type	Null	Key	Default	Extra	Description	
circuitId	int (11)	NO	PRI	NULL	auto_increment	Primary key	
circuitRef	varchar(255)	NO				Unique circuit identifier	
name	varchar(255)	NO				Circuit name	
location	varchar(255)	YES		NULL		Location name	
country	varchar(255)	YES		NULL		Country name	
lat	float	YES		NULL		Latitude	
lng	float	YES		NULL		Longitude	
alt	int (11)	YES		NULL		Altitude (metres)	
url	varchar(255)	NO	UNI			Circuit Wikipedia page	

La tabella circuits riporta le informazioni sui circuiti.

	circuitId	...	url
0	1	...	http://en.wikipedia.org/wiki/Melbourne_Grand_P...
1	2	...	http://en.wikipedia.org/wiki/Sepang_Internatio...
2	3	...	http://en.wikipedia.org/wiki/Bahrain_Internati...
3	4	...	http://en.wikipedia.org/wiki/Circuit_de_Barc...
4	5	...	http://en.wikipedia.org/wiki/Istanbul_Park
..
74	75	...	http://en.wikipedia.org/wiki/Algarve_Internati...
75	76	...	http://en.wikipedia.org/wiki/Mugello_Circuit
76	77	...	http://en.wikipedia.org/wiki/Jeddah_Street_Cir...
77	78	...	http://en.wikipedia.org/wiki/Losail_Internatio...
78	79	...	https://en.wikipedia.org/wiki/Miami_Internatio...

[79 rows x 9 columns]

```
circuitId    79
circuitRef   79
name         79
location     79
country      79
lat          79
lng          79
alt          79
url          79
dtype: int64
```

La tabella circuits contiene 79 record, nessuno dei quali contiene valori nulli.

TABELLA CONSTRUCTOR

constructors table						
Field	Type	Null	Key	Default	Extra	Description
constructorId	int(11)	NO	PRI	NULL	auto_increment	Primary key
constructorRef	varchar(255)	NO				Unique constructor identifier
name	varchar(255)	NO	UNI			Constructor name
nationality	varchar(255)	YES		NULL		Constructor nationality
url	varchar(255)	NO				Constructor Wikipedia page

La tabella constructor riporta le informazioni sui costruttori, ovvero i team.

	constructorId	...	url
0	1	...	http://en.wikipedia.org/wiki/McLaren
1	2	...	http://en.wikipedia.org/wiki/BMW_Sauber
2	3	...	http://en.wikipedia.org/wiki/Williams_Grand_P...
3	4	...	http://en.wikipedia.org/wiki/Renault_in_Formul...
4	5	...	http://en.wikipedia.org/wiki/Scuderia_Toro_Rosso
..
206	209	...	http://en.wikipedia.org/wiki/Manor_Motorsport
207	210	...	http://en.wikipedia.org/wiki/Haas_F1_Team
208	211	...	http://en.wikipedia.org/wiki/Racing_Point_F1_Team
209	213	...	http://en.wikipedia.org/wiki/Scuderia_AlphaTauri
210	214	...	http://en.wikipedia.org/wiki/Alpine_F1_Team

[211 rows x 5 columns]

constructorId	211
constructorRef	211
name	211
nationality	211
url	211
dtype: int64	

La tabella contiene 211 record, di cui nessuno contiene valori nulli. Ma come si può notare dall'immagine, i valori di constructorId sono più alti di quelli dell'indice della riga: ciò è dovuto al fatto che nella tabella mancano ben tre valori di constructorId (43, 165 e 212):

```
array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13,
       14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26,
       27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39,
       40, 41, 42, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53,
       54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66,
       67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79,
       80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92,
       93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105,
      106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118,
      119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131,
      132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144,
      145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157,
      158, 159, 160, 161, 162, 163, 164, 167, 166, 168, 169, 170, 171,
      172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184,
      185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197,
      198, 199, 200, 201, 202, 203, 204, 205, 206, 207, 208, 209, 210,
      211, 213, 214])
```

Si è deciso di non cambiare valori constructorId, dovendo poi andare ad alterare questi valori anche nelle tabelle constructors_standings e constructors_results, cosa non banale visto l'alto numero di record che contengono.

TABELLA CONSTRUCTOR RESULTS

constructor_results table						
Field	Type	Null	Key	Default	Extra	Description
constructorResultsId	int(11)	NO	PRI	NULL	auto_increment	Primary key
raceId	int(11)	NO		0		Foreign key link to races table
constructorId	int(11)	NO		0		Foreign key link to constructors table
points	float	YES		NULL		Constructor points for race
status	varchar(255)	YES		NULL		"D" for disqualified (or null)

La tabella constructor_results riporta le informazioni sui risultati ottenuti dai costruttori nelle gare.

	constructorResultsId	raceId	constructorId	points	status
0		1	18	1	14.0
1		2	18	2	8.0
2		3	18	3	9.0
3		4	18	4	5.0
4		5	18	5	2.0
...
11945	16445	1073	214	6.0	\N
11946	16446	1073	117	0.0	\N
11947	16447	1073	210	0.0	\N
11948	16448	1073	3	0.0	\N
11949	16449	1073	51	0.0	\N
[11950 rows x 5 columns]					
	constructorResultsId	11950			
	raceId	11950			
	constructorId	11950			
	points	11950			
	status	11950			
	dtype:	int64			

All'interno della tabella sono presenti 11950 record, nessuno dei quali contiene valori nulli. Ma si può subito osservare che la colonna status sembra contenere tutti valori pari a \N, ma così non è infatti da un'analisi più approfondita si può notare che alcuni record contengono anche valori diversi:

	constructorResultsId	raceId	constructorId	points	status
185		186	36	1	14.0
195		196	37	1	18.0
207		208	38	1	12.0
218		219	39	1	14.0
228		229	40	1	18.0
239		240	41	1	12.0
250		251	42	1	18.0
262		263	43	1	8.0
273		274	44	1	14.0
283		284	45	1	10.0
294		295	46	1	15.0
306		307	47	1	10.0
316		317	48	1	18.0
328		329	49	1	11.0
338		339	50	1	10.0
350		351	51	1	8.0
361		362	52	1	8.0

TABELLA CONSTRUCTOR STANDINGS

constructor_standings table						
Field	Type	Null	Key	Default	Extra	Description
constructorStandingsId	int(11)	NO	PRI	NULL	auto_increment	Primary key
raceId	int(11)	NO		0		Foreign key link to races table
constructorId	int(11)	NO		0		Foreign key link to constructors table
points	float	NO		0		Constructor points for season
position	int(11)	YES		NULL		Constructor standings position (integer)
positionText	varchar(255)	YES		NULL		Constructor standings position (string)
wins	int(11)	NO		0		Season win count

La tabella `constructor_standings` riporta le informazioni sulla classifica costruttori.

	constructorStandingsId	raceId	...	positionText	wins
0		1	18	...	1 1
1		2	18	...	3 0
2		3	18	...	2 0
3		4	18	...	4 0
4		5	18	...	5 0
...	
12711		27938	1074	...	- 0
12712		27939	1074	...	- 0
12713		27940	1074	...	- 0
12714		27941	1074	...	- 0
12715		27942	1074	...	- 0
[12716 rows x 7 columns]					
	constructorStandingsId	12716			
	raceId	12716			
	constructorId	12716			
	points	12716			
	position	12716			
	positionText	12716			
	wins	12716			
	dtype:	int64			

La tabella contiene 12716 record e nessuno di essi contiene valori nulli.
Bisogna però fare una precisazione:
nell'immagine, gli ultimi cinque valori di `positionText` sono contrassegnati come "-": questo perché i record relativi al `raceId` 1074

sono inerenti alla prima gara del Campionato del 2022, che comincerà a Marzo.

TABELLA DRIVERS

drivers table							
Field	Type	Null	Key	Default	Extra	Description	
driverId	int(11)	NO	PRI	NULL	auto_increment	Primary key	
driverRef	varchar(255)	NO				Unique driver identifier	
number	int(11)	YES		NULL		Permanent driver number	
code	varchar(3)	YES		NULL		Driver code e.g. "ALO"	
forename	varchar(255)	NO				Driver forename	
surname	varchar(255)	NO				Driver surname	
dob	date	YES		NULL		Driver date of birth	
nationality	varchar(255)	YES		NULL		Driver nationality	
url	varchar(255)	NO	UNI			Driver Wikipedia page	

La tabella drivers riporta le informazioni sui piloti.

driverId	...	url
0	1	http://en.wikipedia.org/wiki/Lewis_Hamilton
1	2	http://en.wikipedia.org/wiki/Nick_Heidfeld
2	3	http://en.wikipedia.org/wiki/Nico_Rosberg
3	4	http://en.wikipedia.org/wiki/Fernando_Alonso
4	5	http://en.wikipedia.org/wiki/Heikki_Kovalainen
..
849	851	http://en.wikipedia.org/wiki/Jack_Aitken
850	852	http://en.wikipedia.org/wiki/Yuki_Tsunoda
851	853	http://en.wikipedia.org/wiki/Nikita_Mazepin
852	854	http://en.wikipedia.org/wiki/Mick_Schumacher
853	855	https://en.wikipedia.org/wiki/Guanyu_Zhou
[854 rows x 9 columns]		
driverId	854	
driverRef	854	
number	854	
code	854	
forename	854	
surname	854	
dob	854	
nationality	854	
url	854	
dtype: int64		

La tabella contiene 854 record di cui nessuno contiene valori nulli.

TABELLA DRIVER STANDINGS

driver_standings table							
Field	Type	Null	Key	Default	Extra	Description	
driverStandingsId	int(11)	NO	PRI	NULL	auto_increment	Primary key	
raceId	int(11)	NO		0		Foreign key link to races table	
driverId	int(11)	NO		0		Foreign key link to drivers table	
points	float	NO		0		Driver points for season	
position	int(11)	YES		NULL		Driver standings position (integer)	
positionText	varchar(255)	YES		NULL		Driver standings position (string)	
wins	int(11)	NO		0		Season win count	

La tabella `driver_standings` riporta le informazioni sulla classifica ottenuta dai piloti in una gara.

	driverStandingsId	raceId	driverId	points	position	positionText	wins	
0		1	18	1	10.0	1	1	1
1		2	18	2	8.0	2	2	0
2		3	18	3	6.0	3	3	0
3		4	18	4	5.0	4	4	0
4		5	18	5	4.0	5	5	0
...
33389		70776	1074	840	0.0	16	-	0
33390		70777	1074	852	0.0	17	-	0
33391		70778	1074	830	0.0	18	-	0
33392		70779	1074	20	0.0	19	-	0
33393		70780	1074	855	0.0	20	-	0

[33394 rows x 7 columns]

La tabella contiene 33394 record, nessuno contenente valori nulli.

Allo stesso modo di `constructor_standings` i valori relativi a `raceId` 1074 sono relativi alla prima gara del Campionato del 2022 che comincerà a Marzo.

TABELLA LAP TIMES

lap_times table							
Field	Type	Null	Key	Default	Extra	Description	
raceId	int(11)	NO	PRI	NULL		Foreign key link to races table	
driverId	int(11)	NO	PRI	NULL		Foreign key link to drivers table	
lap	int(11)	NO	PRI	NULL		Lap number	
position	int(11)	YES		NULL		Driver race position	
time	varchar(255)	YES		NULL		Lap time e.g. "1:43.762"	
milliseconds	int(11)	YES		NULL		Lap time in milliseconds	

La tabella `lap_times` riporta le informazioni sui tempi ottenuti dai piloti durante le gare.

	raceId	driverId	lap	position	time	milliseconds
0	841	20	1	1	1:38.109	98109
1	841	20	2	1	1:33.006	93006
2	841	20	3	1	1:32.713	92713
3	841	20	4	1	1:32.803	92803
4	841	20	5	1	1:32.342	92342
...
514587	1073	847	22	15	1:30.821	90821
514588	1073	847	23	15	1:30.647	90647
514589	1073	847	24	14	1:31.577	91577
514590	1073	847	25	16	1:32.794	92794
514591	1073	847	26	18	2:46.262	166262

[514592 rows x 6 columns]

raceId	514592
driverId	514592
lap	514592
position	514592
time	514592
milliseconds	514592
dtype:	int64

All'interno della tabella sono presenti 514592 record, di cui nessuno è nullo. Inoltre, per i piloti che non hanno concluso la gara i tempi sono riportati fino al giro in cui si sono ritirati.

TABELLA PIT STOPS

pit_stops table						
Field	Type	Null	Key	Default	Extra	Description
raceId	int(11)	NO	PRI	NULL		Foreign key link to races table
driverId	int(11)	NO	PRI	NULL		Foreign key link to drivers table
stop	int(11)	NO	PRI	NULL		Stop number
lap	int(11)	NO		NULL		Lap number
time	time	NO		NULL		Time of stop e.g. "13:52:25"
duration	varchar(255)	YES		NULL		Duration of stop e.g. "21.783"
milliseconds	int(11)	YES		NULL		Duration of stop in milliseconds

La tabella pit_stops riporta le informazioni sui pit stops effettuati dai piloti durante le gare.

	raceId	driverId	stop	lap	time	duration	milliseconds
0	841	153	1	1	17:05:23	26.898	26898
1	841	30	1	1	17:05:52	25.021	25021
2	841	17	1	11	17:20:48	23.426	23426
3	841	4	1	12	17:22:34	23.251	23251
4	841	13	1	13	17:24:10	23.842	23842
...
8823	1073	840	2	52	18:22:55	22.661	22661
8824	1073	815	3	53	18:23:09	21.385	21385
8825	1073	854	2	52	18:23:42	22.070	22070
8826	1073	852	2	53	18:24:01	21.909	21909
8827	1073	842	2	54	18:25:56	21.920	21920

[8828 rows x 7 columns]

raceId	8828
driverId	8828
stop	8828
lap	8828
time	8828
duration	8828
milliseconds	8828
dtype:	int64

All'interno della tabella sono presenti 8828 e nessuno ha valore nullo.

TABELLA QUALIFYING

qualifying table						
Field	Type	Null	Key	Default	Extra	Description
qualifyId	int(11)	NO	PRI	NULL	auto_increment	Primary key
raceId	int(11)	NO		0		Foreign key link to races table
driverId	int(11)	NO		0		Foreign key link to drivers table
constructorId	int(11)	NO		0		Foreign key link to constructors table
number	int(11)	NO		0		Driver number
position	int(11)	YES		NULL		Qualifying position
q1	varchar(255)	YES		NULL		Q1 lap time e.g. "1:21.374"
q2	varchar(255)	YES		NULL		Q2 lap time
q3	varchar(255)	YES		NULL		Q3 lap time

La tabella qualifying riporta le informazioni sulle qualifiche dei Gran Premi.

	qualifyId	raceId	driverId	...	q1	q2	q3
0	1	18	1	...	1:26.572	1:25.187	1:26.714
1	2	18	9	...	1:26.103	1:25.315	1:26.869
2	3	18	5	...	1:25.664	1:25.452	1:27.079
3	4	18	13	...	1:25.994	1:25.691	1:27.178
4	5	18	2	...	1:25.960	1:25.518	1:27.236
...
9130	9171	1073	849	...	1:24.338	\N	\N
9131	9172	1073	847	...	1:24.423	\N	\N
9132	9173	1073	8	...	1:24.779	\N	\N
9133	9174	1073	854	...	1:24.906	\N	\N
9134	9175	1073	853	...	1:25.685	\N	\N
[9135 rows x 9 columns]							
qualifyId	9135						
raceId	9135						
driverId	9135						
constructorId	9135						
number	9135						
position	9135						
q1	9127						
q2	9001						
q3	8880						
dtype: int64							

Nella tabella qualifying sono presenti 9135 record, ma risulta subito evidente che nella tabella mancano dei dati, maggiormente nelle colonne q1 e q2: questo è motivato dal fatto che le metodologie di qualifiche negli anni sono cambiate e di conseguenza i tempi non coincidono con la struttura data alla tabella. Inoltre, è presente il valore speciale "\N".

	qualifyId	raceId	driverId	constructorId	number	position	q1	q2	q3
3880	3882	114	23		3	4	1 1:15.259	NaN	NaN
3881	3883	114	8		1	6	2 1:15.295	NaN	NaN
3882	3884	114	31		3	3	3 1:15.415	NaN	NaN
3883	3885	114	15		4	7	4 1:15.500	NaN	NaN
3884	3886	114	30		6	1	5 1:15.644	NaN	NaN
...
8669	8710	1046	825		210	20	16 0:54.705	NaN	NaN
8670	8711	1046	849		3	6	17 0:54.796	NaN	NaN
8671	8712	1046	851		3	89	18 0:54.892	NaN	NaN
8672	8713	1046	8		51	7	19 0:54.963	NaN	NaN
8673	8714	1046	850		210	51	20 0:55.426	NaN	NaN

255 rows × 9 columns

Sono presenti 255 record che contengono valori NaN

TABELLA RACES

races table						
Field	Type	Null	Key	Default	Extra	Description
raceId	int(11)	NO	PRI	NULL	auto_increment	Primary key
year	int(11)	NO		0		Foreign key link to seasons table
round	int(11)	NO		0		Round number
circuitId	int(11)	NO		0		Foreign key link to circuits table
name	varchar(255)	NO				Race name
date	date	NO		0000-00-00		Race date e.g. "1950-05-13"
time	time	YES		NULL		Race start time e.g."13:00:00"
url	varchar(255)	YES	UNI	NULL		Race Wikipedia page

La tabella races riporta le informazioni sui Gran Premi.

	raceId	year	...	time	url
0	1	2009	...	06:00:00	http://en.wikipedia.org/wiki/2009_Australian_Grand_Prix
1	2	2009	...	09:00:00	http://en.wikipedia.org/wiki/2009_Malaysian_Grand_Prix
2	3	2009	...	07:00:00	http://en.wikipedia.org/wiki/2009_Chinese_Grand_Prix
3	4	2009	...	12:00:00	http://en.wikipedia.org/wiki/2009_Bahrain_Grand_Prix
4	5	2009	...	12:00:00	http://en.wikipedia.org/wiki/2009_Spanish_Grand_Prix
...
1075	1092	2022	...	05:10:00	https://en.wikipedia.org/wiki/2022_Japanese_Grand_Prix
1076	1093	2022	...	19:00:00	https://en.wikipedia.org/wiki/2022_United_States_Grand_Prix
1077	1094	2022	...	19:00:00	https://en.wikipedia.org/wiki/2022_Mexican_Grand_Prix
1078	1095	2022	...	17:00:00	https://en.wikipedia.org/wiki/2022_S%F3t%C3%A3o_Portuguese_Grand_Prix
1079	1096	2022	...	13:00:00	https://en.wikipedia.org/wiki/2022_Abu_Dhabi_Grand_Prix

[1080 rows × 8 columns]

raceId	1080
year	1080
round	1080
circuitId	1080
name	1080
date	1080
time	1080
url	1080
dtype:	int64

Sono presenti 1080 record e nessuno di questi contiene valori nulli. Da un'analisi più approfondita non risultano esserci dati con valori speciali.

TABELLA RESULTS

results table						
Field	Type	Null	Key	Default	Extra	Description
resultId	int(11)	NO	PRI	NULL	auto_increment	Primary key
raceId	int(11)	NO		0		Foreign key link to races table
driverId	int(11)	NO		0		Foreign key link to drivers table
constructorId	int(11)	NO		0		Foreign key link to constructors table
number	int(11)	YES		NULL		Driver number
grid	int(11)	NO		0		Starting grid position
position	int(11)	YES		NULL		Official classification, if applicable
positionText	varchar(255)	NO				Driver position string e.g. "1" or "R"
positionOrder	int(11)	NO		0		Driver position for ordering purposes
points	float	NO		0		Driver points for race
laps	int(11)	NO		0		Number of completed laps
time	varchar(255)	YES		NULL		Finishing time or gap
milliseconds	int(11)	YES		NULL		Finishing time in milliseconds
fastestLap	int(11)	YES		NULL		Lap number of fastest lap
rank	int(11)	YES		0		Fastest lap rank, compared to other drivers
fastestLapTime	varchar(255)	YES		NULL		Fastest lap time e.g. "1:27.453"
fastestLapSpeed	varchar(255)	YES		NULL		Fastest lap speed (km/h) e.g. "213.874"
statusId	int(11)	NO		0		Foreign key link to status table

La tabella results riporta le informazioni sui risultati dei Gran Premi.

	resultId	raceId	driverId	...	fastestLapTime	fastestLapSpeed	statusId
0		1	18	1	...	1:27.452	218.300
1		2	18	2	...	1:27.739	217.586
2		3	18	3	...	1:28.090	216.719
3		4	18	4	...	1:28.603	215.464
4		5	18	5	...	1:27.418	218.385
...
25394	25400	1073	815	...	1:26.419	219.993	5
25395	25401	1073	849	...	1:29.293	212.912	3
25396	25402	1073	841	...	1:29.442	212.557	6
25397	25403	1073	847	...	1:30.647	209.732	6
25398	25404	1073	8	...	1:29.698	211.951	23
[25399 rows x 18 columns]							
	resultId	25399					
	raceId	25399					
	driverId	25399					
	constructorId	25399					
	number	25399					
	grid	25399					
	position	25399					
	positionText	25399					
	positionOrder	25399					
	points	25399					
	laps	25399					
	time	25399					
	milliseconds	25399					
	fastestLap	25399					
	rank	25399					
	fastestLapTime	25399					
	fastestLapSpeed	25399					
	statusId	25399					
	dtype: int64						

Come possiamo vedere sono presenti 25399 record nella tabella e nessuno di questi contiene dati mancanti.

	resultId	raceId	driverId	constructorId	number	grid	position	positionText	positionOrder	points	laps	time	milliseconds
0	1	18	1		1	22	1	1	1	10.0	58	1:34.50.616	5690616
1	2	18	2		2	3	5	2	2	8.0	58	+5.478	5696094
2	3	18	3		3	7	7	3	3	6.0	58	+8.163	5698779
3	4	18	4		4	5	11	4	4	5.0	58	+17.181	5707797
4	5	18	5		1	23	3	5	5	4.0	58	+18.014	5708630
5	6	18	6		3	8	13	6	6	3.0	57	\N	\N
6	7	18	7		5	14	17	7	7	2.0	55	\N	\N
7	8	18	8		6	1	15	8	8	1.0	53	\N	\N
8	9	18	9		2	4	2	\N	R	0.0	47	\N	\N
9	10	18	10		7	12	18	\N	R	0.0	43	\N	\N
10	11	18	11		8	18	19	\N	R	0.0	32	\N	\N
11	12	18	12		4	6	20	\N	R	0.0	30	\N	\N
12	13	18	13		6	2	4	\N	R	0.0	29	\N	\N
13	14	18	14		9	9	8	\N	R	0.0	25	\N	\N

Da un'analisi più approfondita notiamo, però, che all'interno della tabella è presente all'interno di più colonne il valore speciale "\N".

TABELLA SEASON

seasons table							
	Field	Type	Null	Key	Default	Extra	Description
	year	int(11)	NO	PRI	0		Primary key e.g. 1950
	url	varchar(255)	NO	UNI			Season Wikipedia page

La tabella season riporta le informazioni sulla singola stagione.

	year	url
0	2009	https://en.wikipedia.org/wiki/2009_Formula One...
1	2008	https://en.wikipedia.org/wiki/2008_Formula One...
2	2007	https://en.wikipedia.org/wiki/2007_Formula One...
3	2006	https://en.wikipedia.org/wiki/2006_Formula One...
4	2005	https://en.wikipedia.org/wiki/2005_Formula One...
..
68	2018	https://en.wikipedia.org/wiki/2018_Formula One...
69	2019	https://en.wikipedia.org/wiki/2019_Formula One...
70	2020	https://en.wikipedia.org/wiki/2020_Formula One...
71	2021	https://en.wikipedia.org/wiki/2021_Formula One...
72	2022	https://en.wikipedia.org/wiki/2022_Formula One...

[73 rows x 2 columns]

year	73
url	73
dtype:	int64

All'interno della tabella sono presenti 73 record di cui nessuno contiene valori nulli.

TABELLA STATUS

status table						
Field	Type	Null	Key	Default	Extra	Description
statusId	int(11)	NO	PRI	NULL	auto_increment	Primary key
status	varchar(255)	NO				Finishing status e.g. "Retired"

La tabella `status` riporta le informazioni riguardanti lo stato dei piloti a fine gara.

```
      statusId      status
0            1    Finished
1            2 Disqualified
2            3     Accident
3            4   Collision
4            5     Engine
..
..          ...
132         135 Brake duct
133         136       Seat
134         137     Damage
135         138     Debris
136         139    Illness

[137 rows x 2 columns]
statusId    137
status      137
dtype: int64
```

La tabella contiene 137 record di cui nessuno contiene valori nulli.

3.3 FEATURE SELECTION

Passiamo ora alla fase di Feature Selection, ovvero alla scelta delle *feature* (caratteristiche) che potranno essere più utili ai nostri scopi. Chiediamoci quindi quali sono le informazioni in nostro possesso prima dell'inizio di una gara.

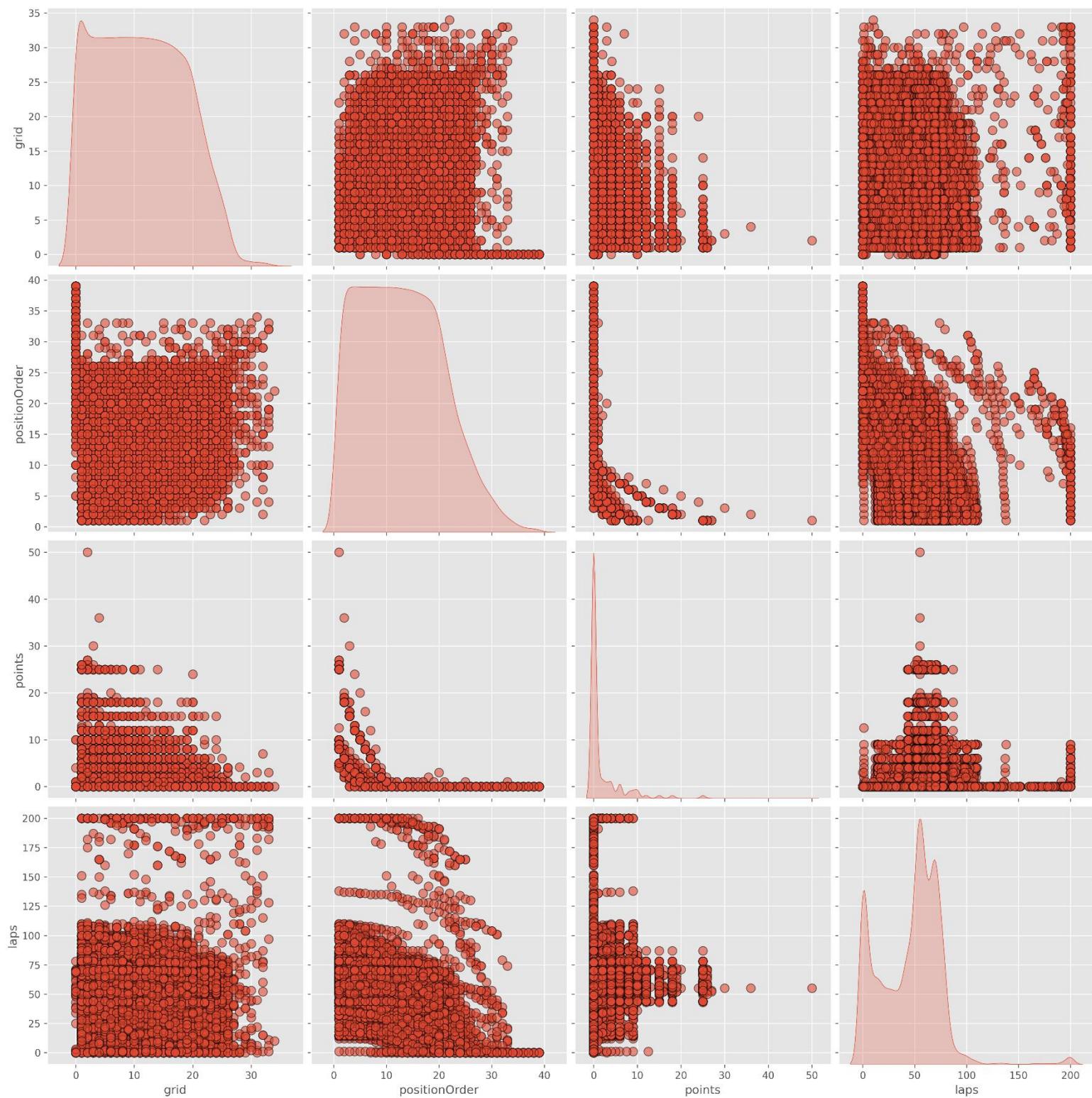
Iniziamo dicendo che le tabelle `seasons`, `status`, `lap_times`, `pit_stops` sono state scartate perché potremmo incappare nei cosiddetti *leaky predictor*: queste informazioni sono infatti disponibili **durante** o **dopo** il Gran Premio, non prima. Quindi averle all'interno delle nostre *feature* ci porterà al problema del *Data Leakage*: il modello utilizzerà quei dati per allenarsi, ma quando arriverà una nuova istanza da predire non ne sarà capace. Quindi, è stato deciso di non prendere in considerazione le tabelle.

Un po' di considerazione in più va fatta per la tabella `qualifying`: oltre ai vari campi contenenti gli `Id`, si potrebbe pensare di prendere in considerazione i campi `q1,q2` e `q3`. Ma abbiamo già visto che sono proprio questi campi a contenere i valori nulli, cosa più che normale date le varie metodologie con cui si effettuate le qualifiche.

Poiché alla fine anche avere i tempi con cui i piloti ottengono la posizione sulla griglia di partenza non fornisce particolari conoscenze che ci possono essere utili all'atto della predizione e poiché i restanti campi sono presenti anche in altre tabelle, decidiamo di non utilizzare la tabella `qualifying`.

La tabella che sembra esserci più utile è la tabella `results`, che contiene le informazioni relative alle gare. Come osservato nella sezione precedente, questa tabella contiene tre campi `position`,`positionText` e `positionOrder` che rappresentano la stessa informazione, ovvero la posizione d'arrivo del pilota. Per l'utilizzo che vogliamo fare dei dati la scelta ricade su `positionOrder`, poiché le altre due contengono sia valori speciali “\N” che testo, il che può creare problemi durante la fase di addestramento.

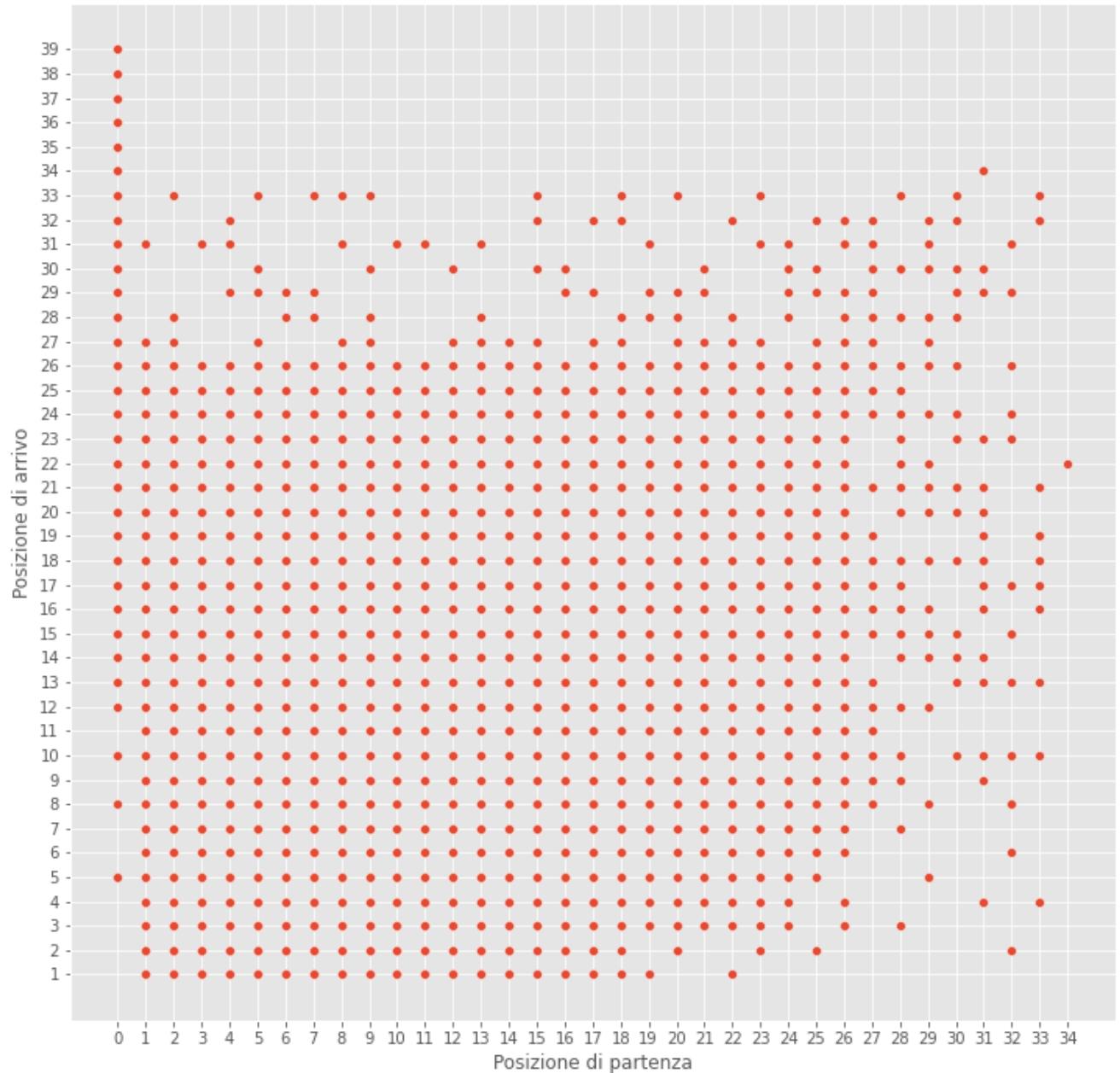
Andiamo quindi ora a scoprire come i dati si relazionano tra di loro:



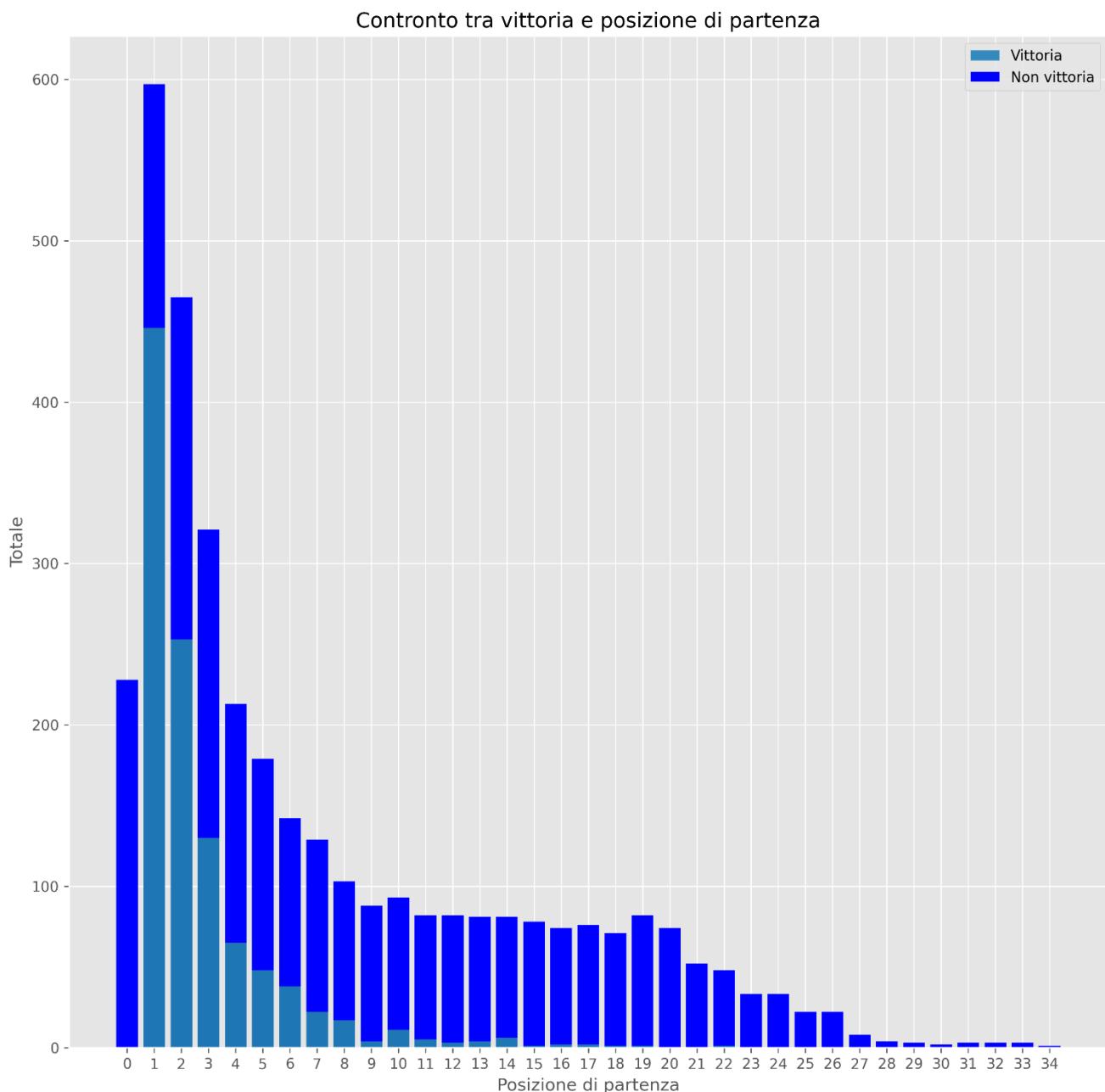
In figura è rappresentato un pairplot, che permette di visualizzare tutti i grafici tra due variabili in un dataset.

Una relazione che si è andati a esaminare è quella tra grid e positionOrder.

Correlazione tra posizione di partenza e di arrivo



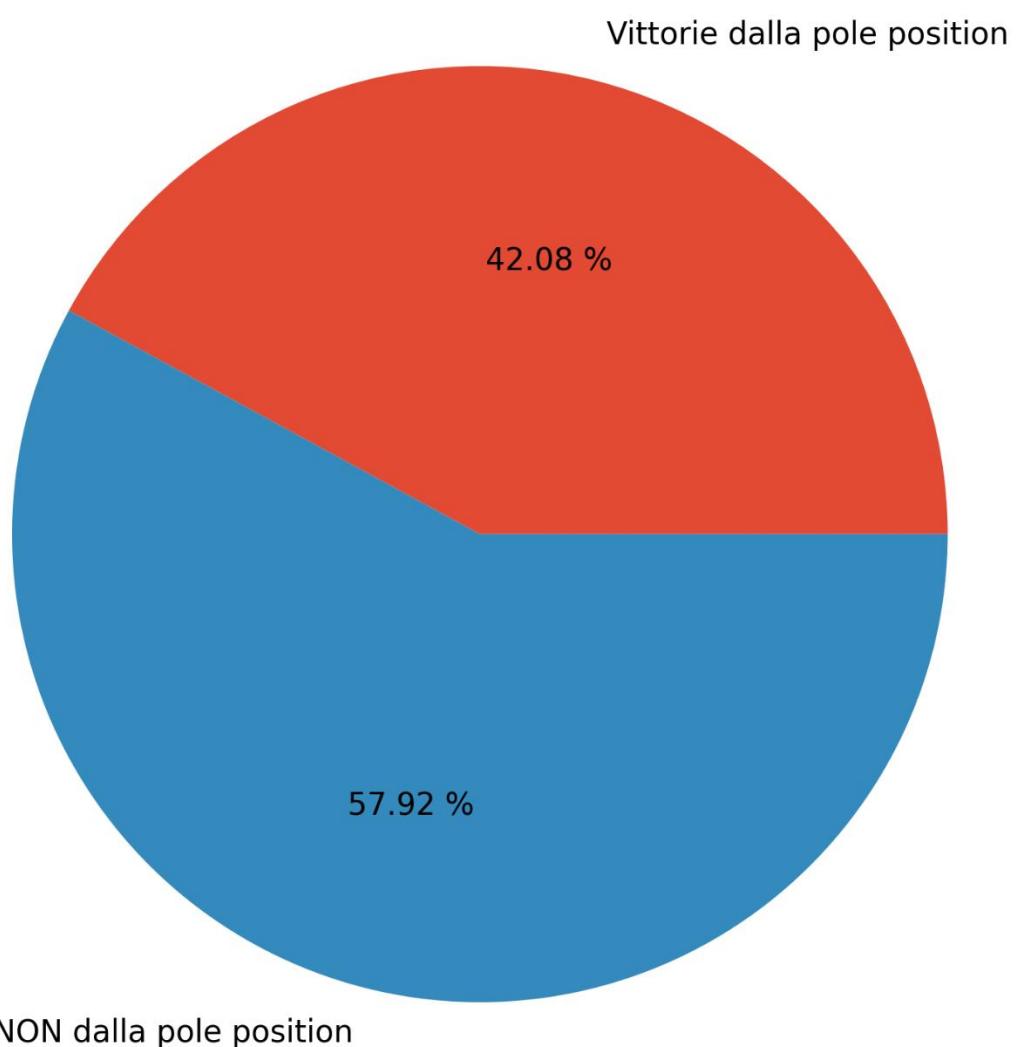
Sorge spontanea una domanda: quante volte partire da una determinata posizione conduce alla vittoria?



Nel grafico sono rappresentati il numero di volte che partire da una determinata posizione ha portato alla vittoria contro il numero di volte che, invece, non si ha vinto. Come si può evincere dalla figura, avere una posizione di partenza abbastanza “alta” porta a un maggior successo rispetto a una “bassa”.

In particolare, la Pole Position sembra essere la posizione che porta a un maggior numero di successi. Esaminiamo quindi il rapporto che può esistere tra vincere partendo dalla Pole e vincere non partendo dalla Pole.

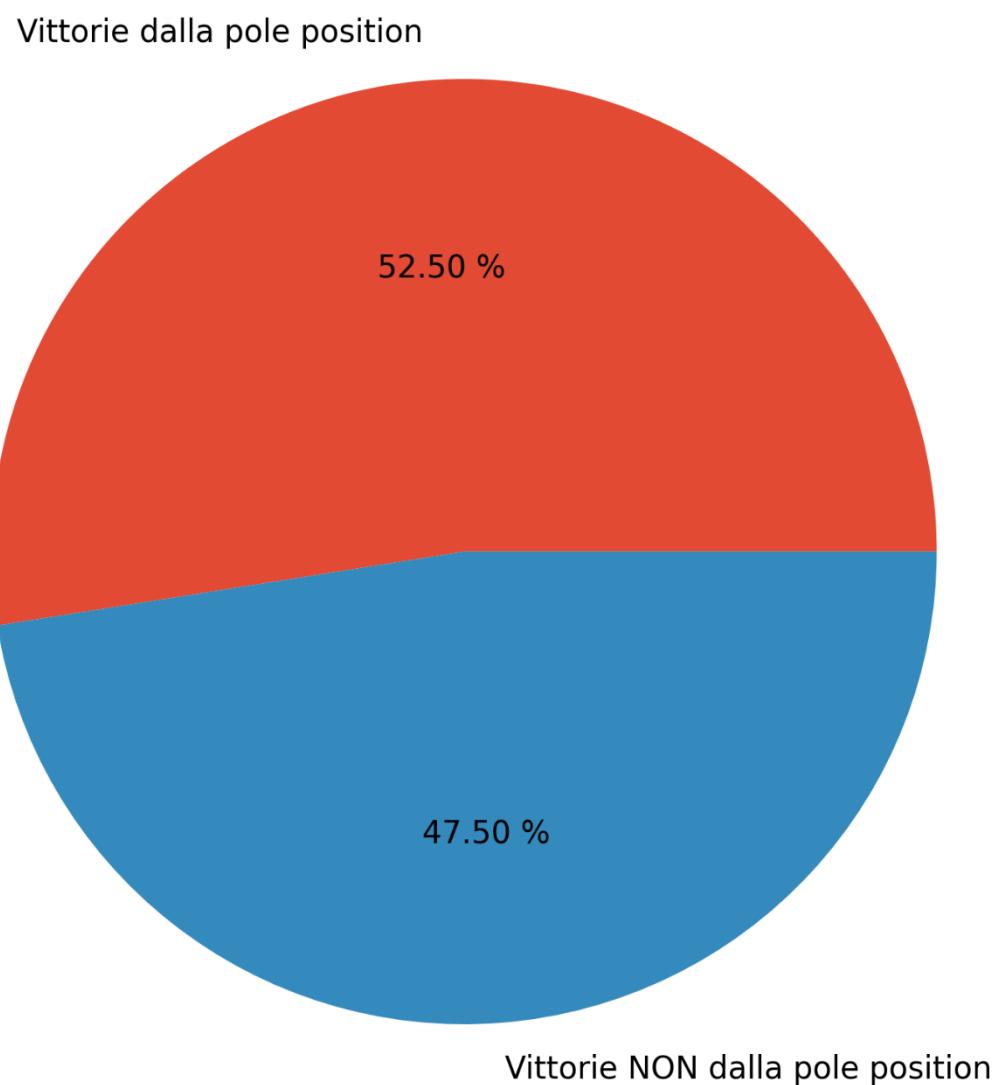
Confronto tra la percentuale di vittorie dalla pole e non dalla pole



Il grafico riporta in confronto in percentuale tra vittorie dalla Pole e vittorie NON dalla Pole.

Sembra quindi che nella maggior parte dei casi partire dalla Pole non assicura una vittoria. Però questa analisi è stata effettuata su TUTTI i risultati dei Gran Premi: non tiene conto delle evoluzioni tecnologiche che si sono avute nel corso del tempo. Concentriamoci quindi sugli anni 2014-2021, ovvero negli anni in cui è stato introdotto l'utilizzo del motore V6 e il blocco dello sviluppo:

Confronto tra la percentuale di vittorie dalla pole e non dalla pole 2014-2021

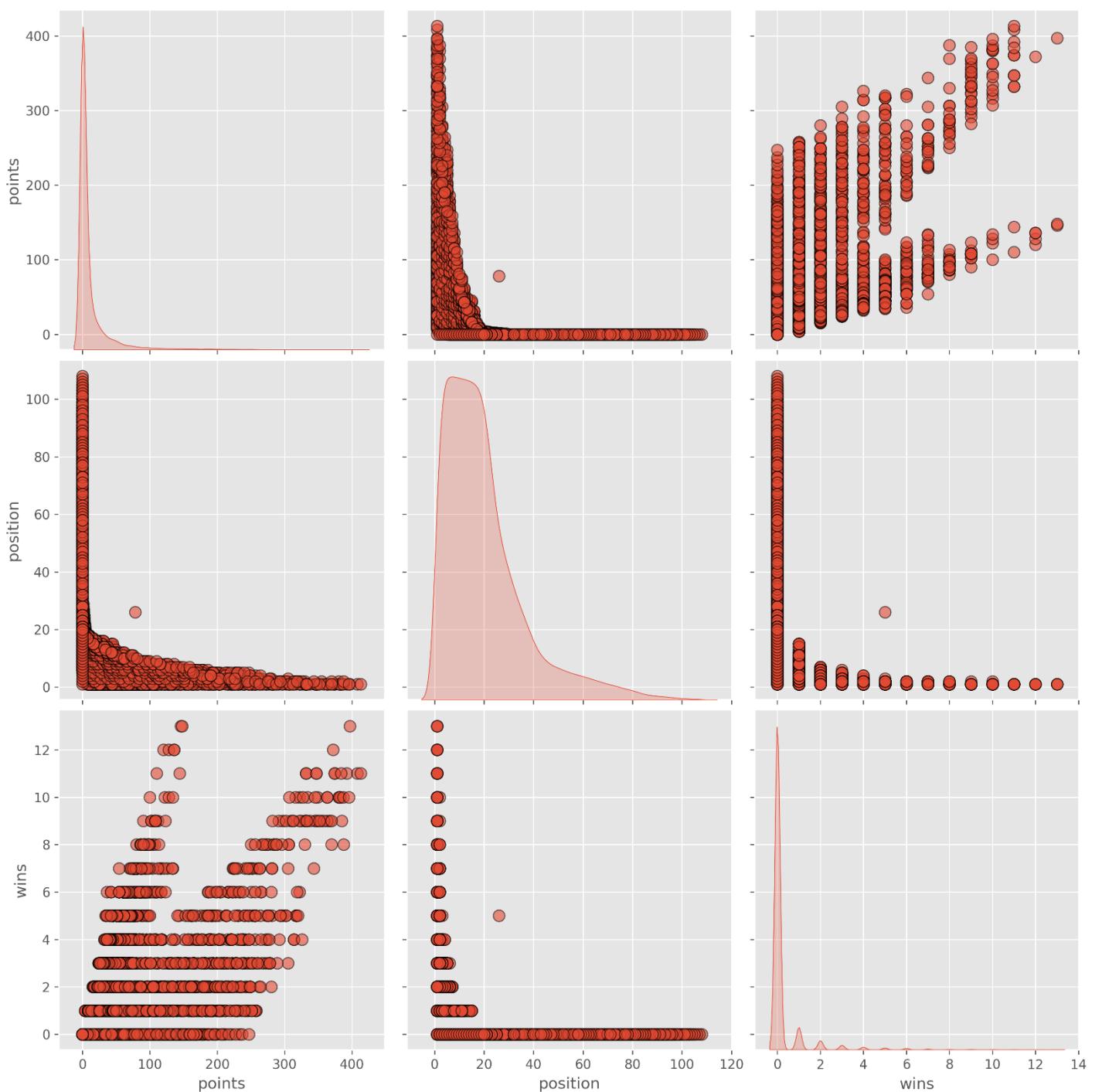


Come possiamo osservare negli ultimi sette anni la tendenza sembra essersi invertita e nella maggior parte dei casi il pilota che parte dalla Pole ottiene la vittoria.

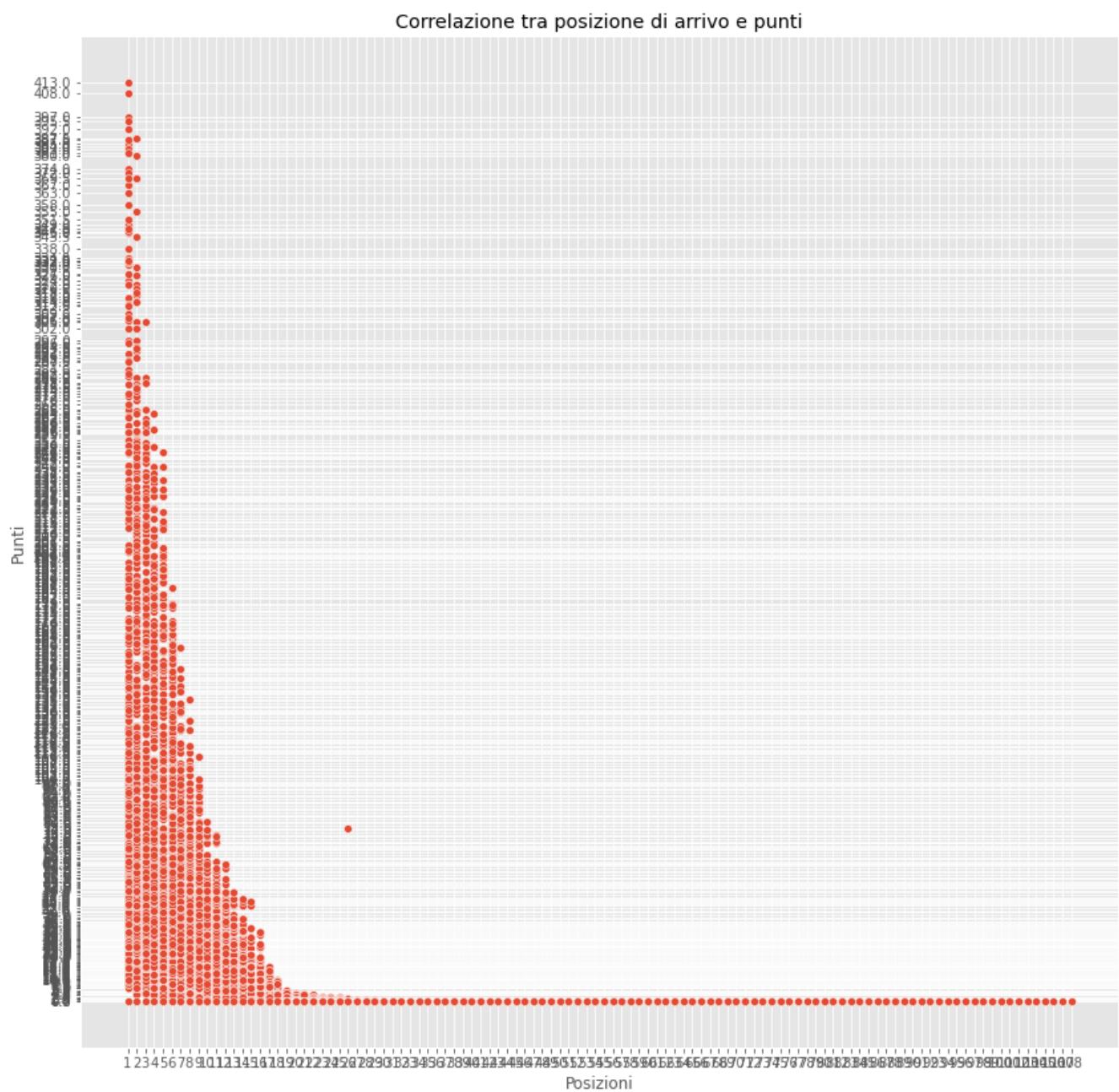
Quindi abbiamo identificato le prime due feature che ci potranno essere utili: `grid` e `positionOrder`.

Altra relazione che ci può tornare utile è quella tra posizione d'arrivo e punti ottenuti. Esploriamo questa relazione però non nella tabella `results`, ma nella tabella `driver_standings`, che riporta le classifiche dei piloti.

Infatti, anche qui appare evidente una relazione:

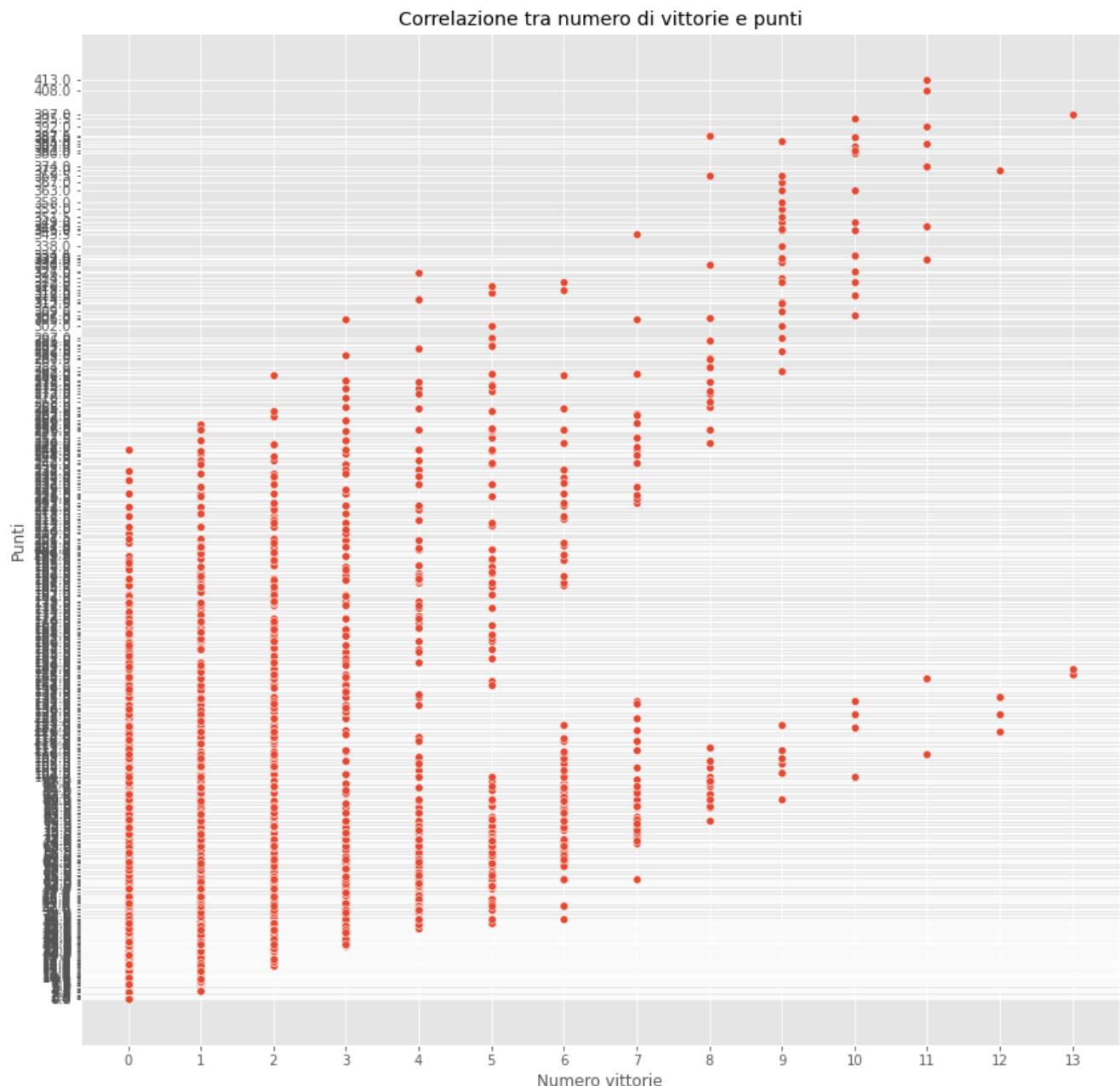


Più in dettaglio:

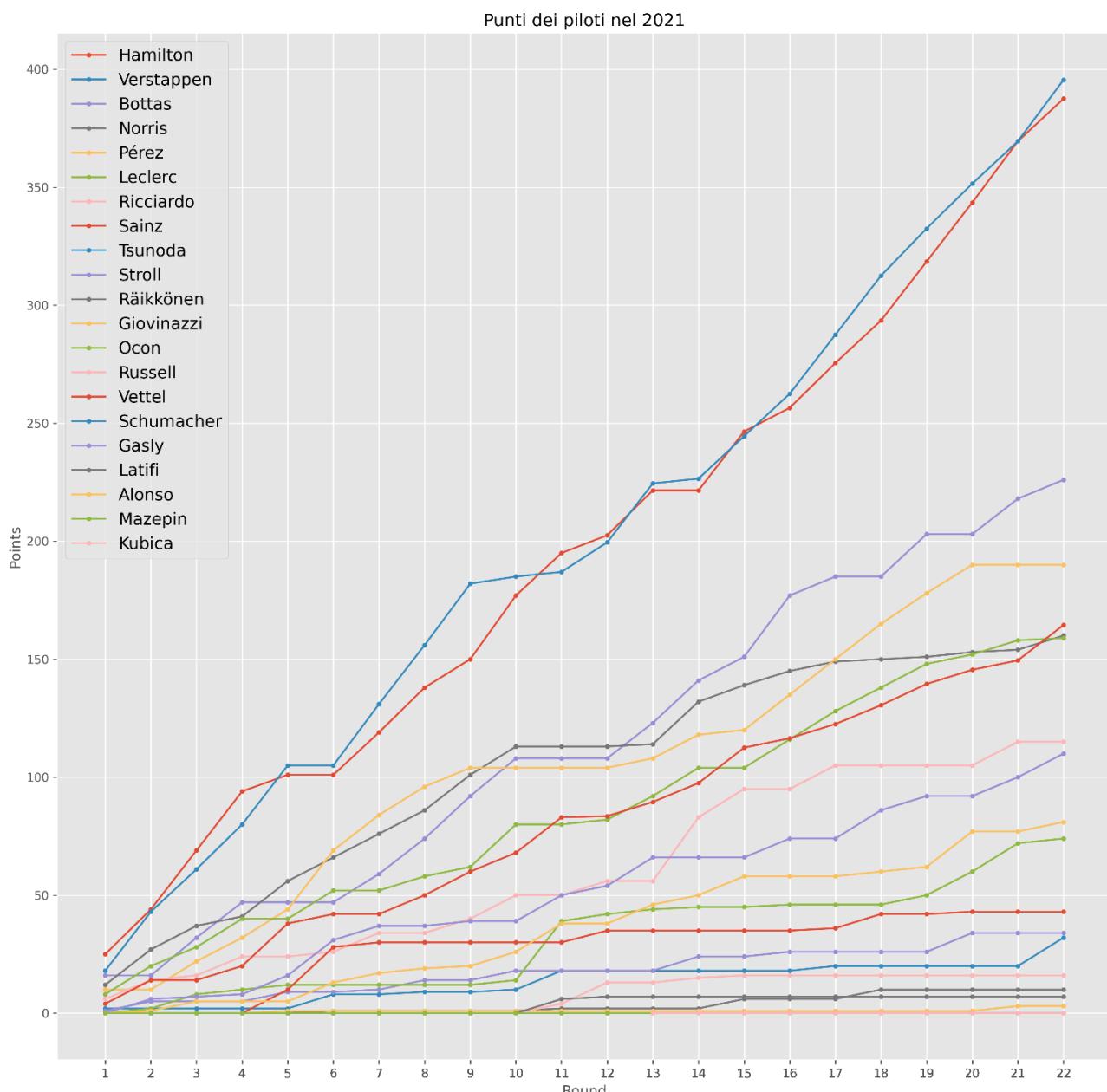


Come si può evincere dalla figura, un pilota che ottiene un buon piazzamento ha più punti. Quindi possiamo assumere che un pilota con un alto numero di punti abbia “più

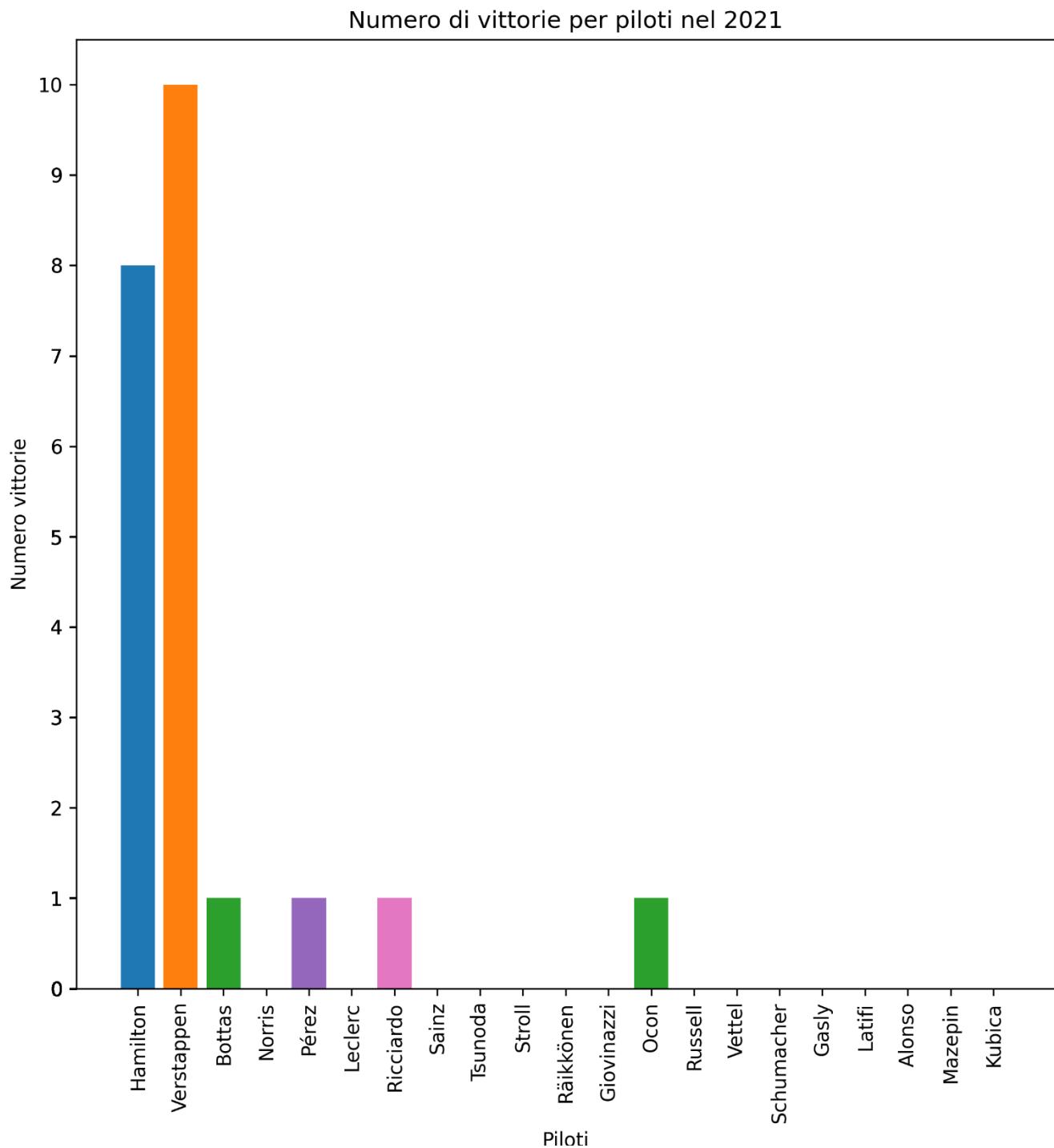
probabilità" di vincere rispetto a uno che ha pochi punti. Quindi la caratteristica `points` di `driver_standings` può tornarci più che utile per la nostra predizione. Sulla stessa linea è la relazione tra numero di vittorie e punti:



Anche in questo caso, come prima, ci sembra più promettente che a vincere sia un pilota che ha un maggior numero di vittorie rispetto a uno che ne ha poche. Come esempio si consideri il Campionato 2021:

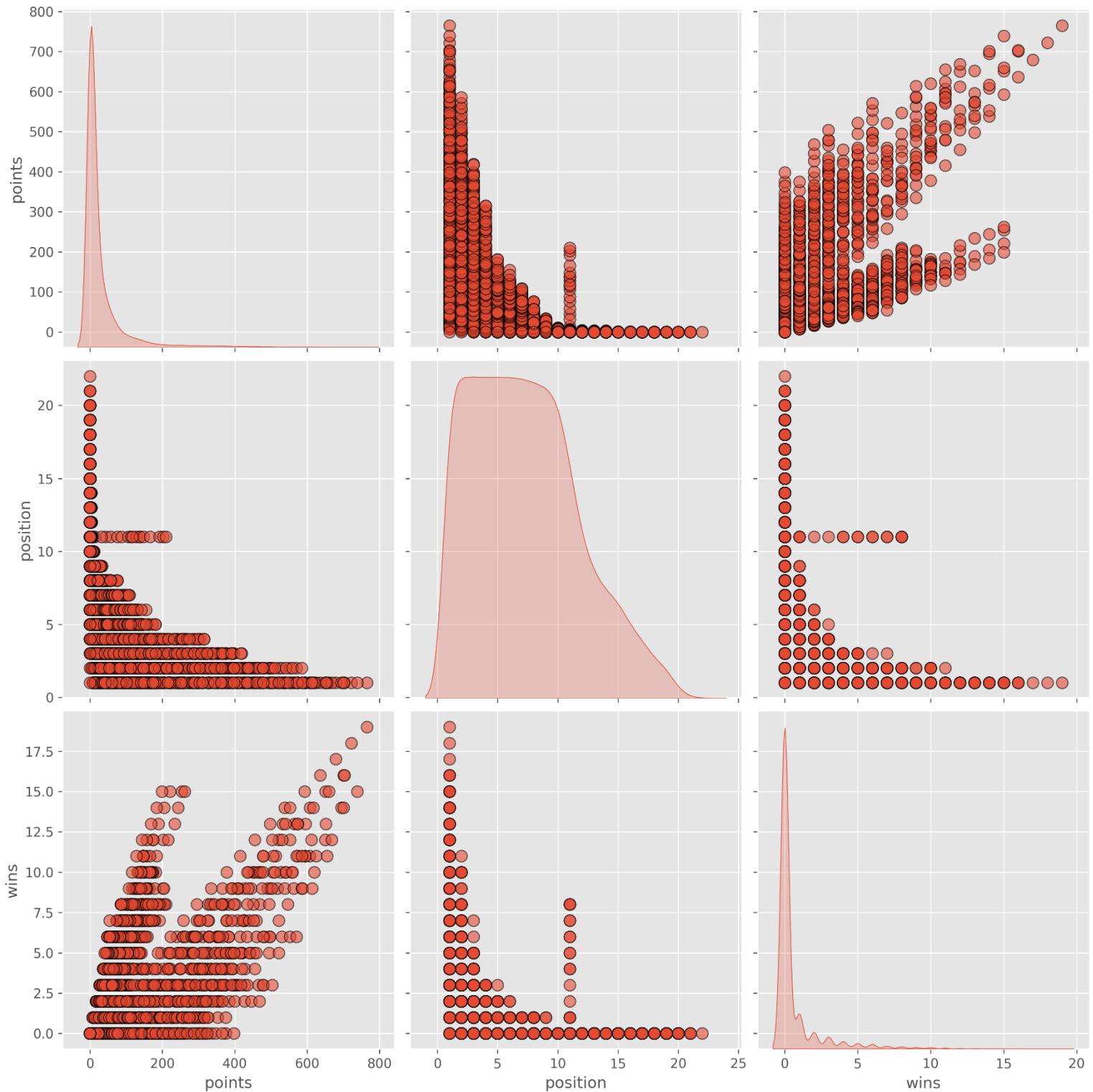


Come si può evincere dalla figura, ci sono stati due piloti, Verstappen e Hamilton che sono nettamente distaccati dal resto; infatti, sono coloro che hanno ottenuto il maggior numero di vittorie (10 e 8 rispettivamente):



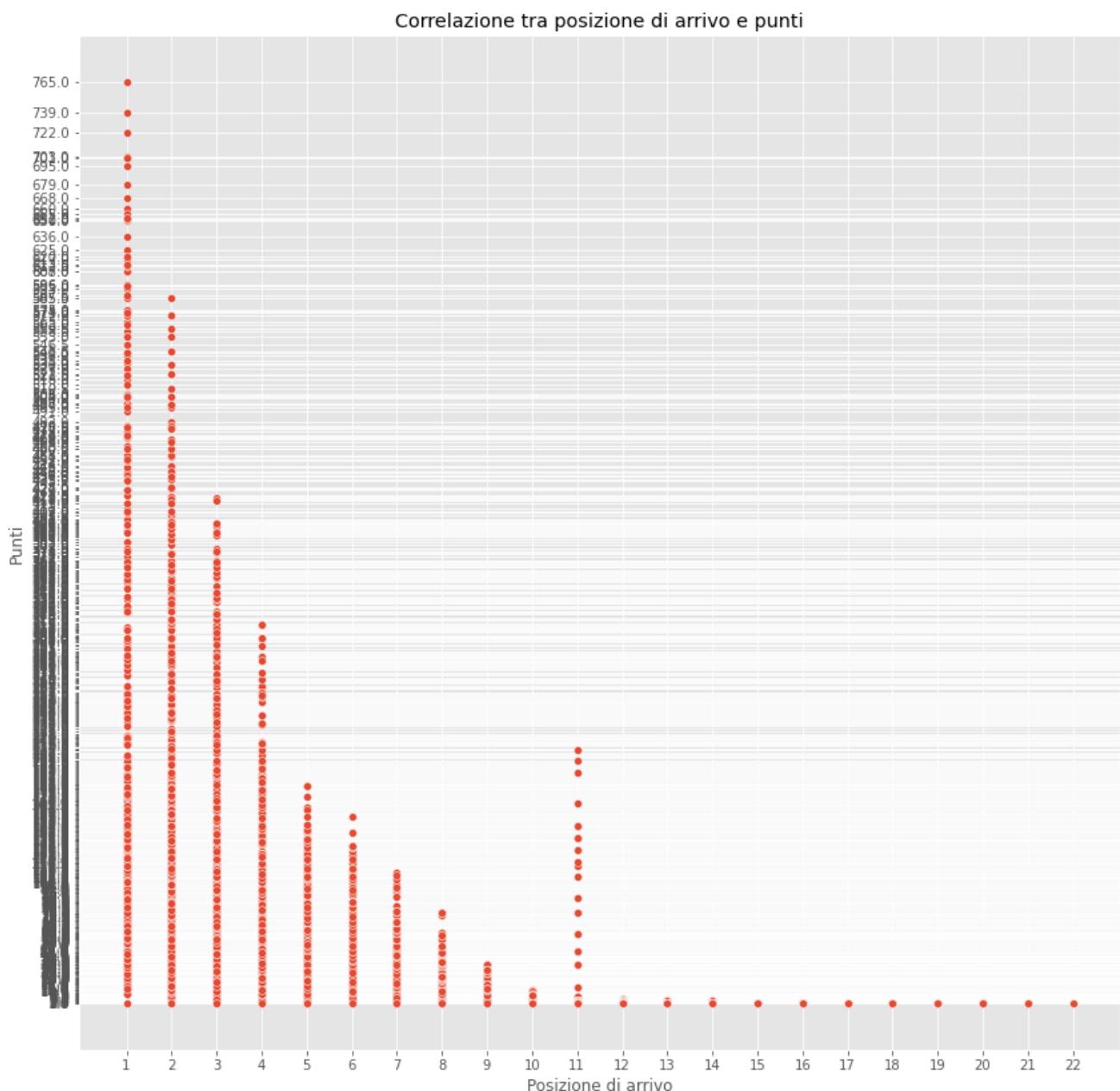
Abbiamo quindi identificato altri due *feature* che ci potranno essere utili nella nostra predizione: `points` e `wins`.

Un discorso analogo può essere fatto sulla tabella `constructor_standings`:



In figura è riportato lo scatterplot relativo alla tabella. Come avvenuto per la tabella `driver_standings` la prima relazione che andiamo a esaminare è quella tra posizione d'arrivo e punti.

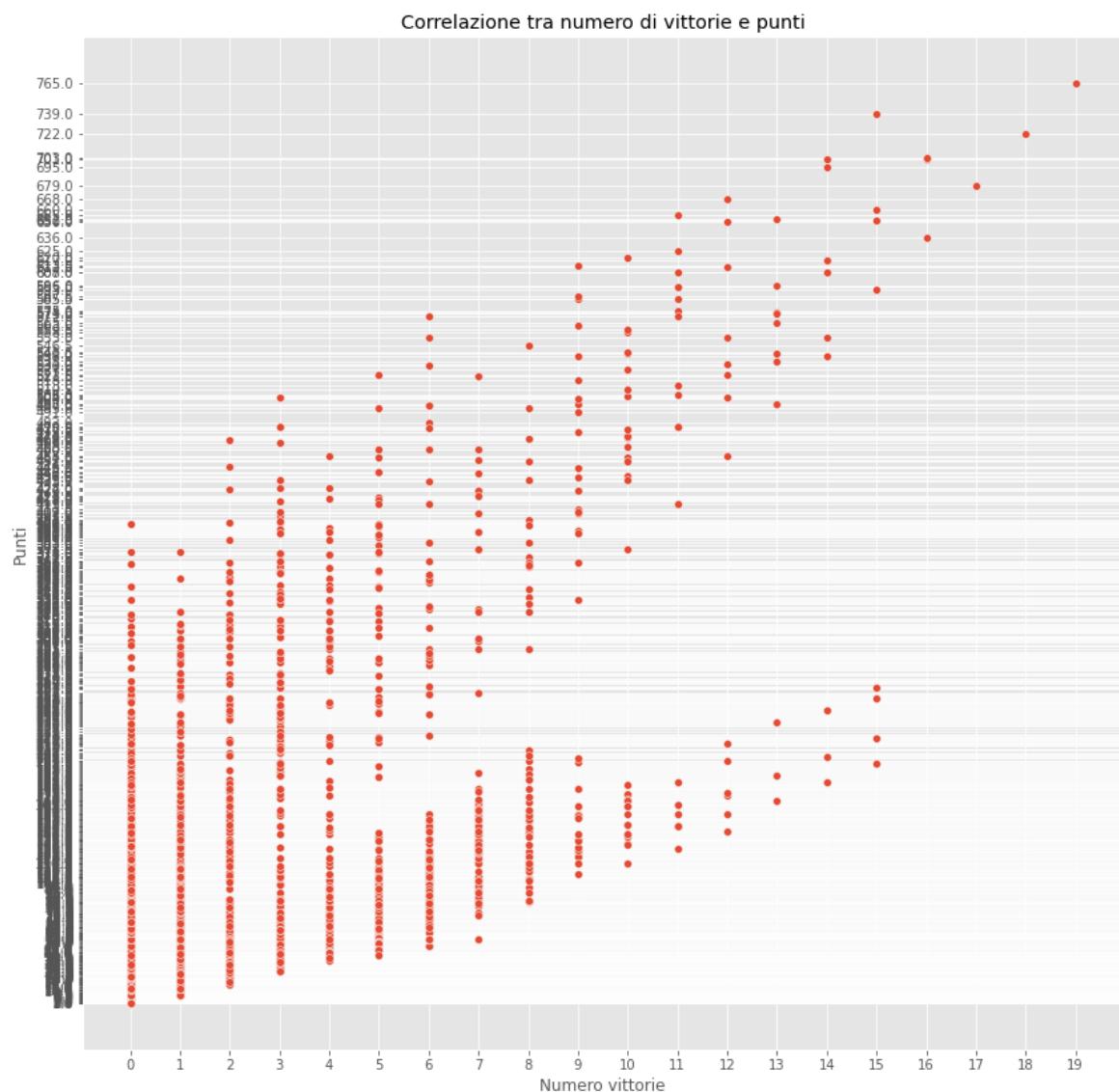
Più in dettaglio:



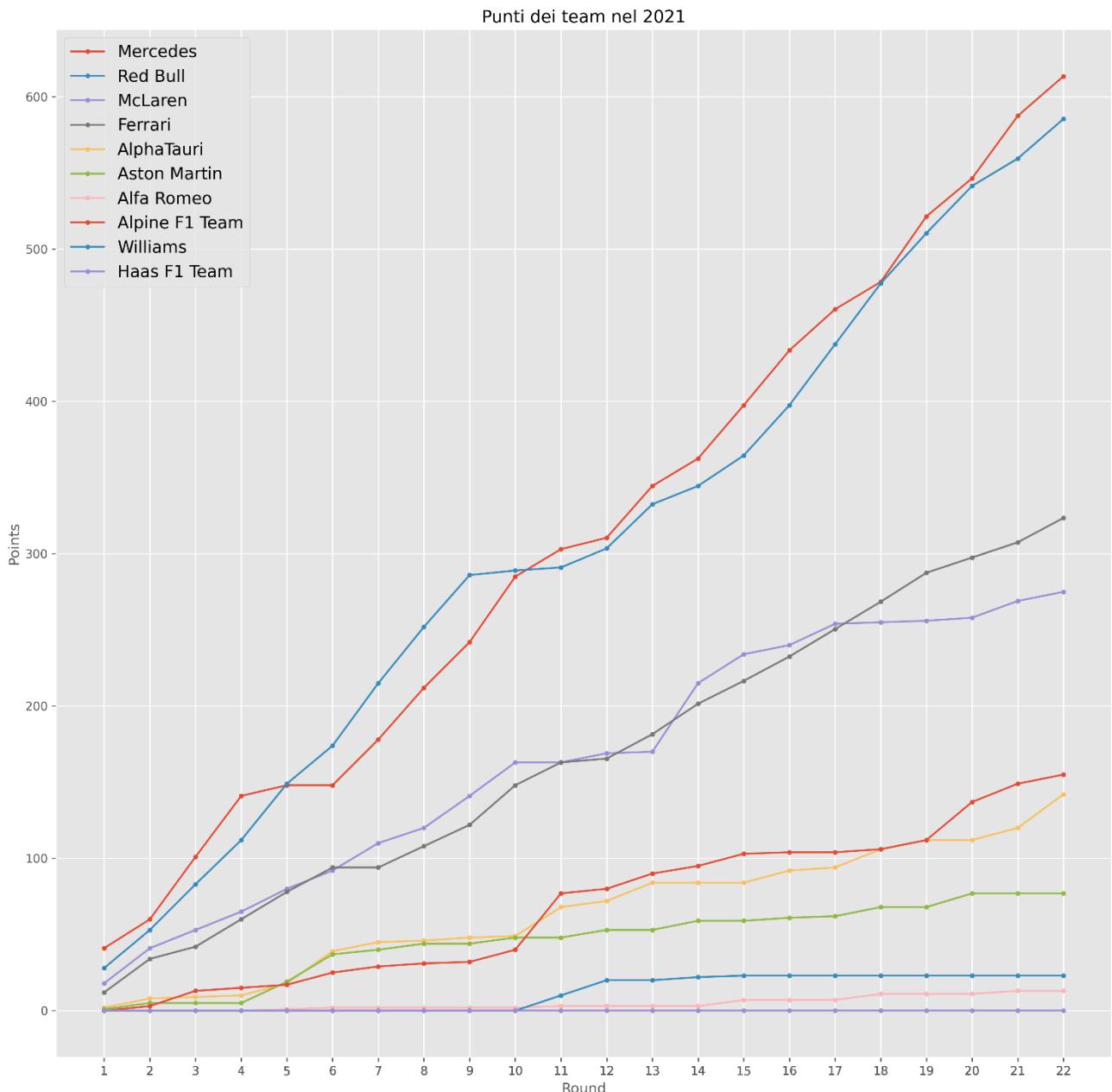
Rispetto al caso precedente va però fatta una precisazione: i punti assegnati ai Team sono la somma dei punti assegnati ai piloti che corrono per quel Team. Ad esempio: se i piloti Leclerc e Sainz, che corrono per la Ferrari, ottengono rispettivamente la prima e la seconda posizione, quindi 25 e 18 punti, la Ferrari guadagna 43 punti.

Quindi è più probabile che per un Team che ha molti punti corra un pilota che abbia ottenuto buoni piazzamenti.

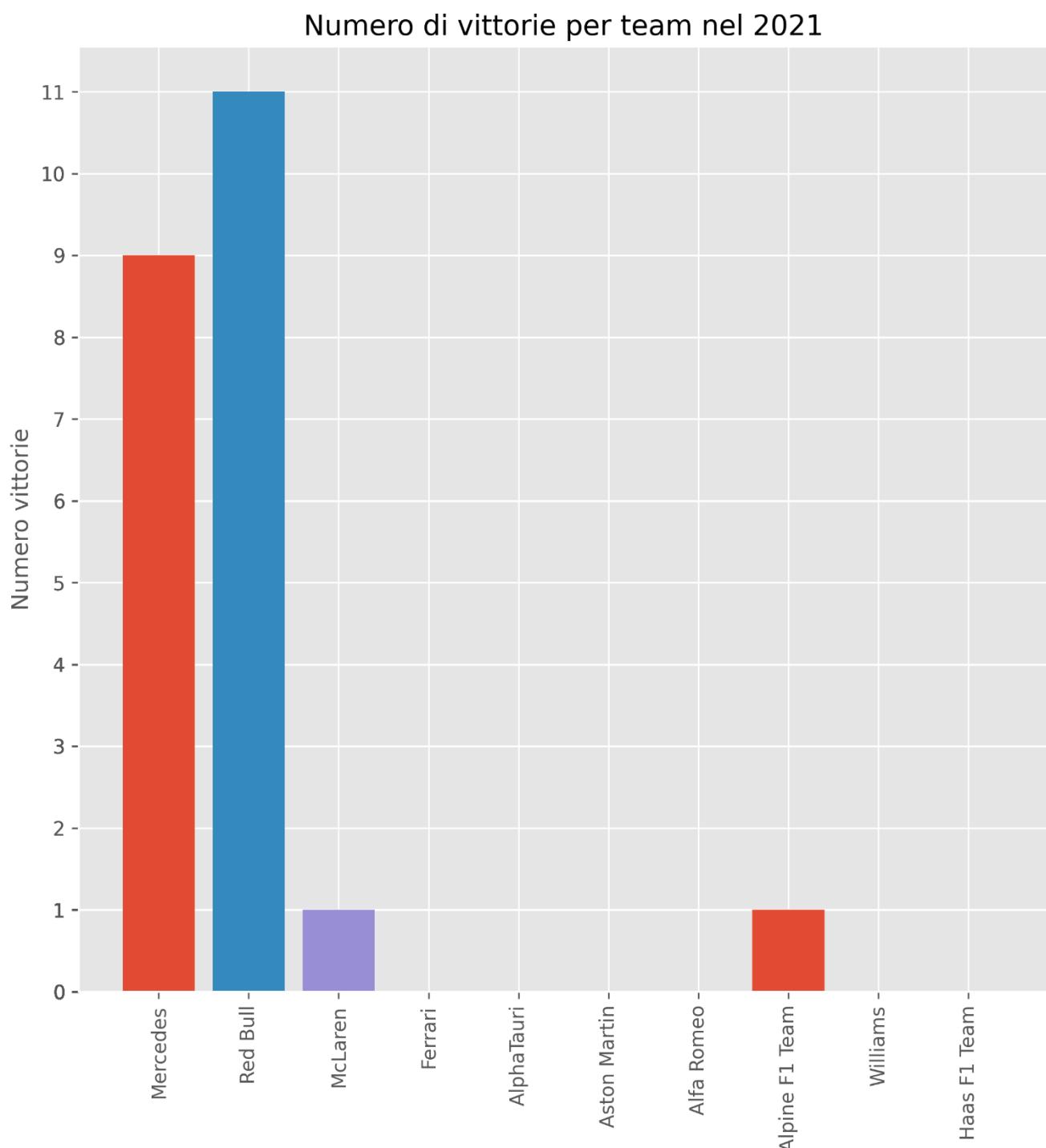
Può essere quindi utile visualizzare la relazione che esiste anche in questa tabella tra punti e vittorie.



Come possiamo osservare dal grafico, un alto numero di vittorie porta a un alto numero di punti. Riprendiamo il Campionato 2021 come esempio:



Si nota subito che i due Team che hanno maggiori punti sono i corrispettivi Team di Hamilton e Verstappen, Mercedes e Red Bull, che sono anche i due piloti con il maggior numero di vittorie nel 2021. Infatti:



Abbiamo quindi identificato altre due *feature* che ci potranno essere utili per la predizione: *wins* e *points* di *constructor_standings*.

Passiamo ora a un altro aspetto importante che abbiamo a disposizione all'atto della predizione, ovvero il circuito su cui si svolge la gara.

Andiamo quindi ad analizzare la tabella *circuits*: presa singolarmente questa tabella non fornisce niente di utile ai nostri scopi, ma se unita con *results* possiamo ottenere varie importanti informazioni: abbiamo già discusso del perché alcune tabelle siano state scartate, perché legate a eventi che non permettevano di aiutare il modello a effettuare la predizione prima della gara, come pit stop, tempi sul giro e incidenti. Quest'ultimo in particolare richiede però particolare attenzione: un incidente può cambiare in modi inaspettati l'esito della gara.

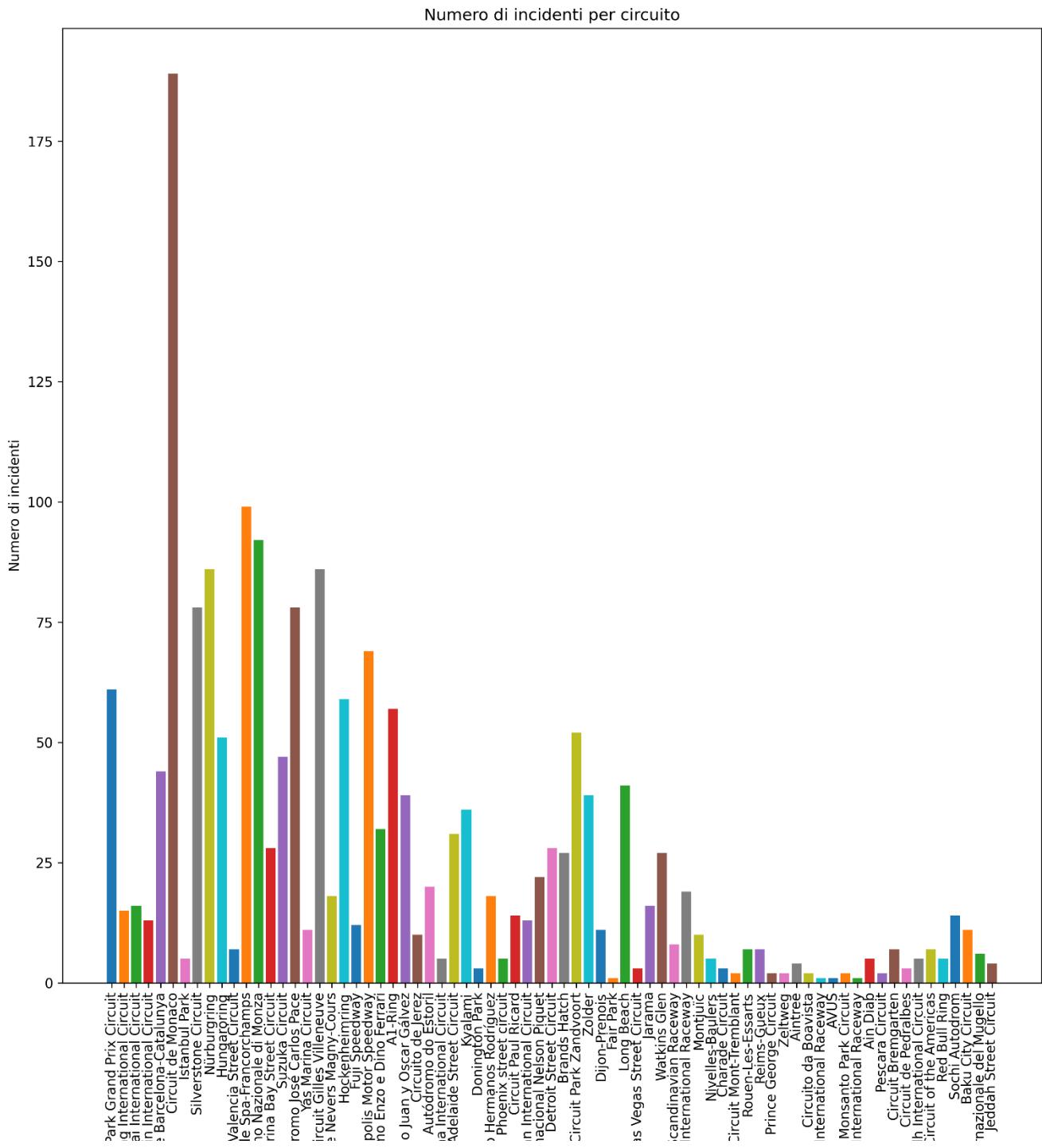
Prendiamo, ancora una volta, come esempio la stagione appena conclusasi: durante il Gran Premio d'Italia Verstappen, al venticinquesimo giro, nel tentativo di superare Hamilton, che era in testa, effettua una manovra azzardata, che causa una collisione tra i due, costringendo entrambi al ritiro.



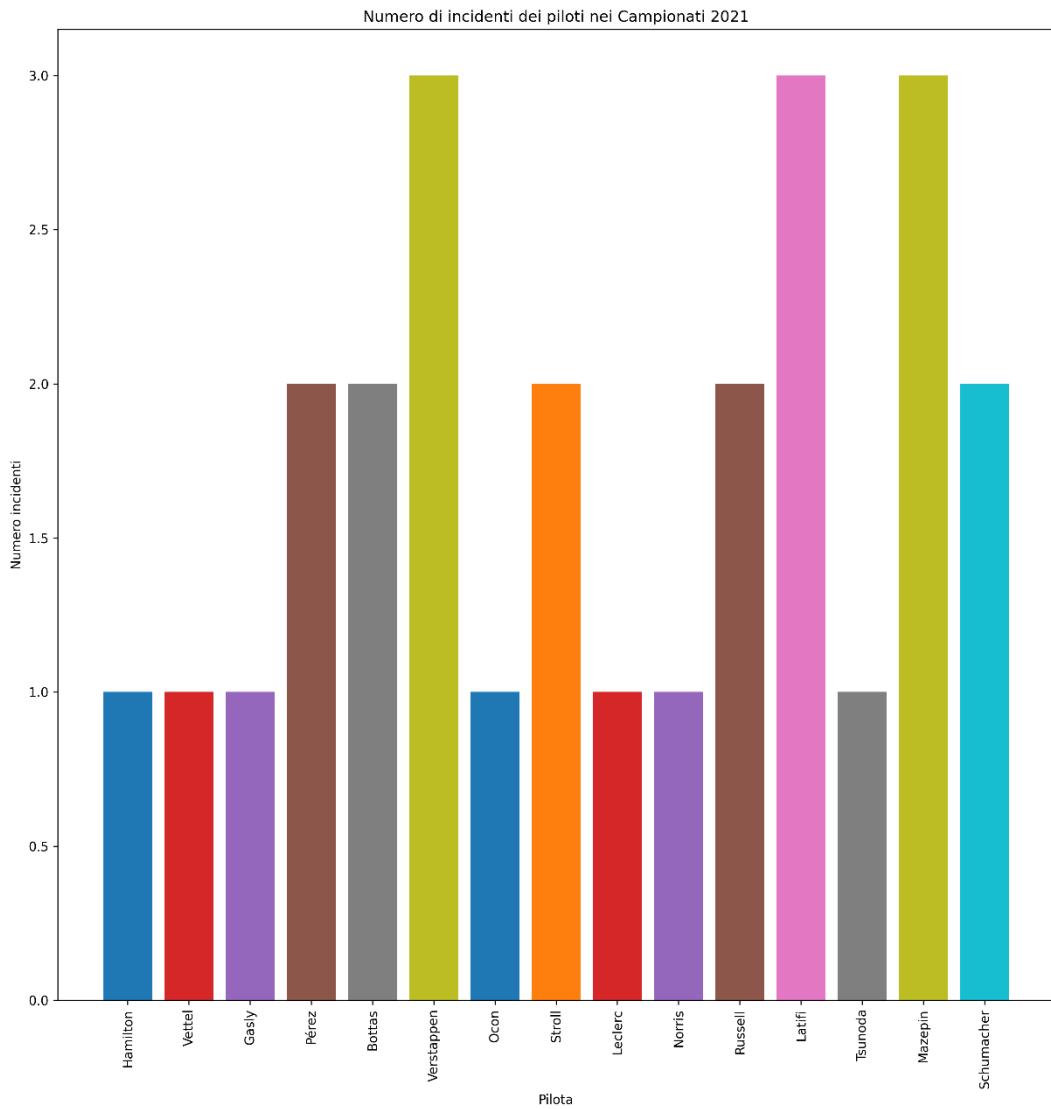
Mentre l'episodio accaduto ha a che fare con l'errore umano, dobbiamo tenere conto che non tutti i circuiti sono uguali: alcuni sono molto più difficili da percorrere. Si pensi ai cosiddetti *circuiti cittadini*: il layout di questi circuiti sono effettivamente ottenuti da strade cittadine. Esempio lampante è il Gran Premio di Monaco, caratterizzato da percorsi stretti che rendono difficile il sorpasso e che richiedono la massima concentrazione da parte del pilota.

Sulla base di queste osservazioni, sorge spontanea una domanda: quali sono quei circuiti che hanno una maggiore probabilità di causare un incidente?

Il seguente grafico ci dà la risposta.



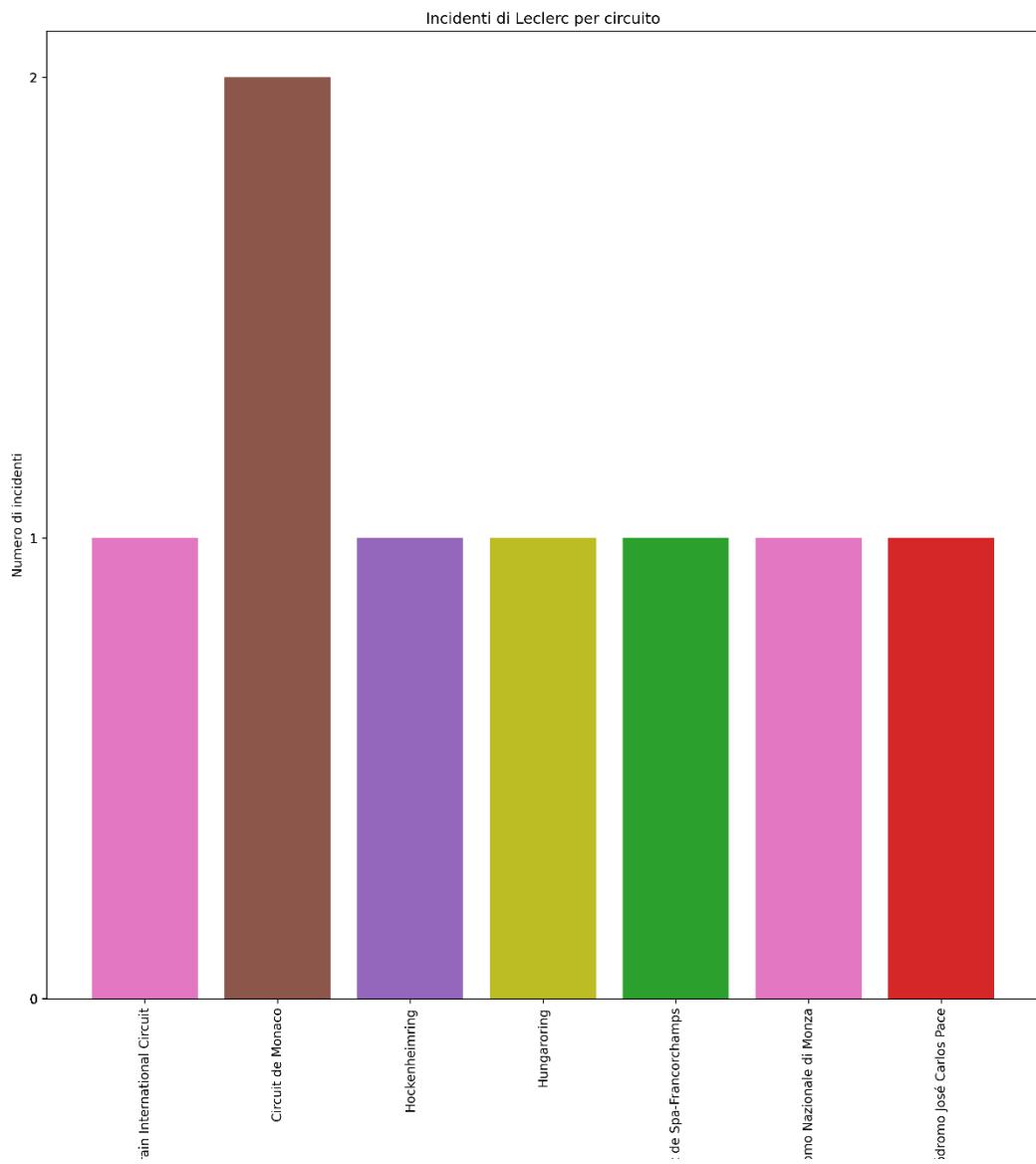
A noi però interessano i piloti, non il circuito in sé. Quindi dobbiamo vedere chi è più prone all'errore e ha, di conseguenza una minore probabilità di vittoria. Come esempio si è preso il Campionato 2021:



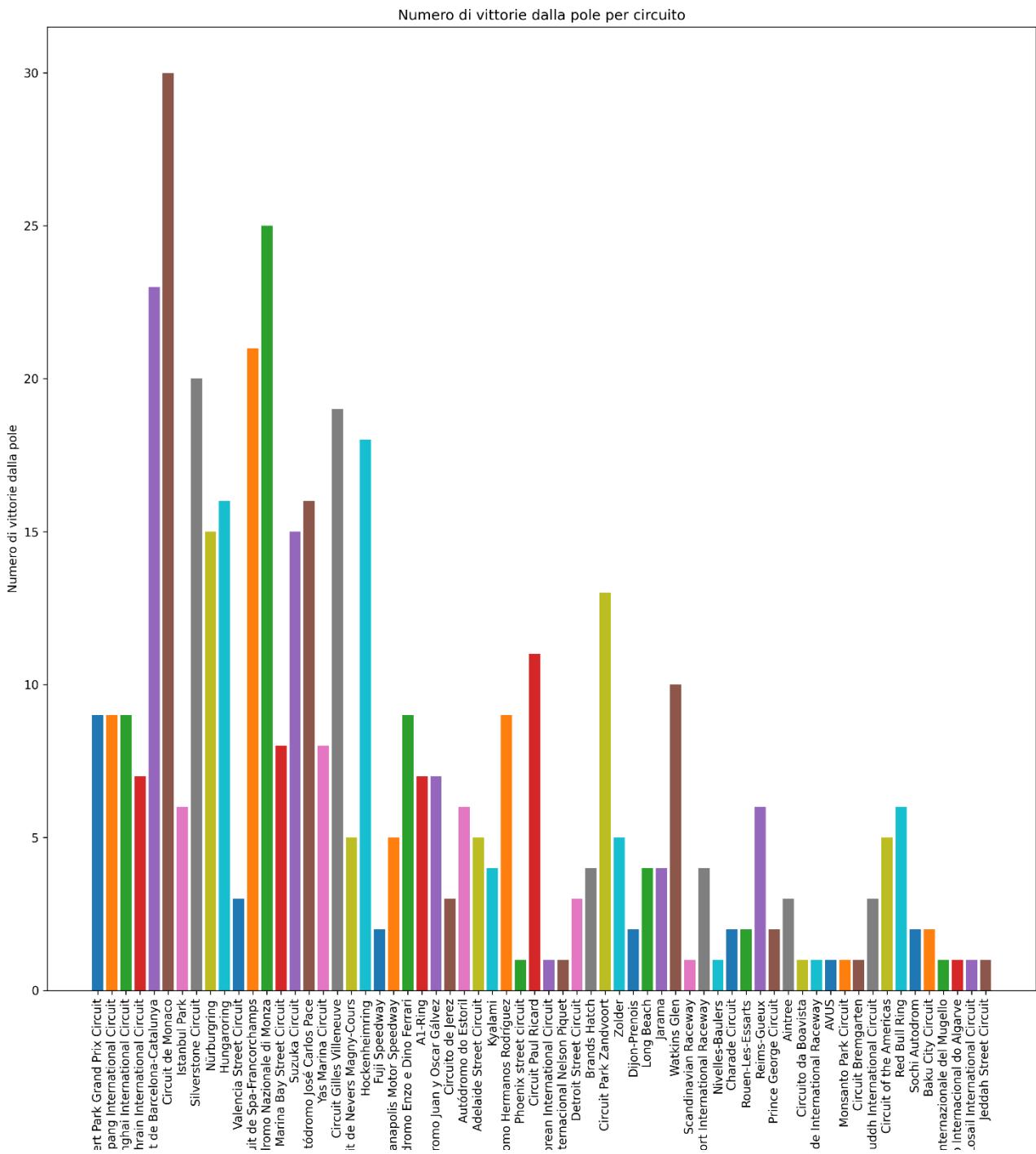
È abbastanza curioso osservare come Verstappen, che è il pilota con più vittorie nel 2021 e anche tra quelli con più incidenti.

Può quindi sembrare utile tener conto di due cose: il numero di incidenti avuti dai piloti in tutta la loro carriera e il numero di incidenti che un pilota ha avuto su un determinato circuito.

Si pensi al pilota Leclerc: come riportato nel grafico della pagina successiva, da quando corre il Formula 1, ha avuto un incidente in Bahrain, Germania, Ungheria, Belgio, Italia e Brasile, mentre 2 a Monaco, per un totale di 8 incidenti.



Abbiamo parlato di circuiti cittadini e della loro enorme difficoltà. In questi circuiti è estremamente difficile effettuare sorpassi e il minimo errore può causare un incidente. Per questo partire il più avanti possibile in questi casi è necessario. Si è quindi andato ad esaminare quante volte partire dalla Pole Position in un determinato circuito ha portato alla vittoria.



Come si può vedere nel grafico l' 1/3 dei circuiti hanno un numero maggiore di 10 vittorie dalla Pole. Come già affermato prima, a noi interessa il pilota e non il circuito in sé; quindi, andremo a mantenere per ogni pilota il numero di volte che ha vinto su quel circuito partendo dalla Pole e non partendo dalla Pole.

Rimangono le tabelle `driver_results`, `constructor_results`, `driver`, `constructors` e `races`. Le prime due non forniscono informazioni in più rispetto a `driver_standings` e `constructor_standings`.

Allo stesso modo `driver` e `constructors`, manteniamo gli id associati sia a piloti che Team, poiché sarebbe inutile dover utilizzare poi una codifica che va a mappare nome del pilota e del team quando ci viene già fornita.

Da parte di `races` invece andiamo a selezionare il campo `year` che ci permetterà di distinguere tra i vari Campionati.

Ricapitoliamo quindi quali sono le *feature* selezionate:

- `grid`: indica la posizione di partenza
- `positionOrder`: indica la posizione d'arrivo
- `points`: indica i punti ottenuti dal pilota nella gara
- `driverId`: id associato al pilota
- `constructorId`: id associato al pilota
- `year`: anno di svolgimento della gara
- `round`: indica l'ordine delle gare nel Campionato
- `circuitId`: id associato al circuito
- `points (driver_standings)`: indica i punti ottenuti dal pilota durante il Campionato. Per evitare confusione con `points` di `constructor_standings` si è proceduto a rinominare il campo come `punti_campionato_pilota`
- `wins (driver_standings)`: indica il numero di vittorie del pilota ottenute durante il Campionato. Per evitare confusione con `wins` di `constructor_standings` si è proceduto a rinominare il campo come `vittorie_pilota`
- `points (constructor_standings)`: indica i punti ottenuti dal Team del pilota durante il Campionato. Per evitare confusione con `points` di `driver_standings` si è proceduto a rinominare il campo come `punti_campionato_team`
- `wins (constructor_standings)`: indica il numero di vittorie del Team del pilota ottenute durante il Campionato. Per evitare confusione con `points` di `driver_standings` si è proceduto a rinominare il campo come `punti_campionato_team`
- `incidenti_pilota`: numero di incidenti del pilota da inizio carriera in Formula Uno
- `incidenti_circuito`: numero di incidenti del pilota sul circuito su cui si svolge la gara

- **vincitori_dalla_pole**: numero di volte che il pilota ha vinto sul circuito partendo dalla pole
- **vincitori_NON_dalla_pole**: numero di volte che il pilota NON ha vinto sul circuito partendo dalla pole

```

        driverId constructorId ... vincitori_dalla_pole vincitori_NON_dalla_pole
0            642           51 ...                   1                     0
1            786           51 ...                   0                     0
2            686           51 ...                   0                     0
3            704           154 ...                  0                     0
4            627           154 ...                  0                     0
...
...          ...
25394         815           9 ...                   0                     0
25395         849           3 ...                   0                     0
25396         841           51 ...                  0                     0
25397         847           3 ...                   0                     0
25398          8           51 ...                  0                     1

[25399 rows x 17 columns]
Index(['driverId', 'constructorId', 'grid', 'positionOrder', 'points',
       'pilota', 'year', 'round', 'circuitId', 'punti_campionato_pilota',
       'vittorie_pilota', 'punti_campionato_team', 'vittorie_team',
       'incidenti_pilota', 'incidenti_circuito', 'vincitori_dalla_pole',
       'vincitori_NON_dalla_pole'],
      dtype='object')

```

driverId	25399
constructorId	25399
grid	25399
positionOrder	25399
points	25399
pilota	25399
year	25399
round	25399
circuitId	25399
punti_campionato_pilota	25399
vittorie_pilota	25399
punti_campionato_team	25399
vittorie_team	25399
incidenti_pilota	25399
incidenti_circuito	25399
vincitori_dalla_pole	25399
vincitori_NON_dalla_pole	25399
dtype:	int64

Salviamo il tutto in un file .csv chiamato F1WP, presente all'interno della tabella Dataset del repository.

Alcune osservazioni: il dataset è estremamente sbilanciato: la posizione a cui siamo interessati è ovviamente la prima posizione, che è in numero minore rispetto alle altre posizioni. Per cercare di mitigare questo problema si è tentato di andare a predire non la singola posizione di vittoria, ma tutte le posizioni e poi si è presa quella relativa alla prima.

4. ALGORITMI UTILIZZATI

Prima di andare a esaminare quali algoritmi sono stati scelti, comprendiamo il tipo di problema che stiamo trattando: poiché i nostri dati sono etichettati, possiamo subito affermare che si tratta di un problema di apprendimento supervisionato. I problemi di apprendimento supervisionato si distinguono in due tipologie sulla base dell'output prodotto dal modello:

- se l'output del modello è continuo allora si tratta di un problema di **regressione**
- se l'output del modello è discreto allora stiamo parlando di un problema di **classificazione**

Per il nostro problema si è scelto di andare a usare sia algoritmi di regressione che di classificazione: come già spiegato si utilizzerà la libreria sklearn di Python, che mette a disposizione una vasta gamma di algoritmi: per la regressione utilizzeremo la regressione lineare e gli alberi decisionali (DecisionTreeRegressor) , mentre per la classificazione utilizzeremo gli alberi decisionali (DecisionTreeClassifier) e Naive Bayes.

Per quanto riguarda la divisione del dataset in training – set e test – set, non possiamo dividere il dataset in maniera randomica poiché, essendo presenti dati che sono temporali (ad esempio, il Campionato 2021 non può essere preso come allenamento e il risultato di quest'ultimo utilizzato sul Campionato 2020), si è deciso di andare a utilizzare tutti i Campionati fino al 2019 come training – set, il 2020 come test – set e useremo il Campionato 2021 per vedere come il nostro modello si comporta.

4.1 REGRESSIONE LINEARE E ALBERI DECISIONALI

Nel caso della regressione andiamo a prevedere il valore della posizione di arrivo (`positionOrder`): l'algoritmo andrà a produrre un valore numerico associato a ogni istanza del dataset di test e il valore più piccolo corrisponderà a quello del pilota vincitore.

4.2 ALBERI DECISIONALI E NAIVE BAYES

Nel caso degli alberi decisionali e Naive Bayes invece come variabile dipendente si è scelto il nome del pilota (pilota) da predire per ogni istanza: il nome associato all'istanza con posizione di arrivo 1 sarà quello che ci interessa.

5 VALUTAZIONI SUL TEST SET

Per valutare gli algoritmi di regressione sono state usate la MSE (Mean Squared Error) , MAE (Mean Absolute Error) e RMSE (Root Mean Squared Error).

Per valutare gli algoritmi classificazione si è andati a ottenere il punteggio di accuratezza e la matrice di confusione.

5.1 REGRESSIONE LINEARE

```
Mean squared error: 24.90
Mean absolute error: 3.74
Root Mean Squared Error: 4.99
```

5.2 ALBERI DECISIONALI (REGRESSIONE)

```
Mean squared error: 10.49
Mean absolute error: 1.84
Root Mean Squared Error: 3.24
```

5.3 ALBERI DECISIONALI (CLASSIFICAZIONE)

```
0.888235294117647
[[ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0 ]
 [ 0 17  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 ]
 [ 0  0 17  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 ]
 [ 0  0  0  0 17  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 ]
 [ 0  0  0  0 17  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 ]
 [ 0  0  0  0  0 15  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 ]
 [ 0  0  0  0  0  0 16  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 ]
 [ 0  0  0  0  0  0  0  3  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 ]
 [ 0  0  0  0  0  0  0  0  0 17  0  0  0  0  0  0  0  0  0  0  0  0  0  0 ]
 [ 0  0  0  0  0  0  0  0  0  0  0 17  0  0  0  0  0  0  0  0  0  0  0  0 ]
 [ 0  0  0  0  0  0  0  0  0  0 17  0  0  0  0  0  0  0  0  0  0  0  0  0 ]
 [ 0  0  0  0  0  0  0  0  0  0  0  0 17  0  0  0  0  0  0  0  0  0  0  0 ]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0 17  0  0  0  0  0  0  0  0  0  0 ]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0 17  0  0  0  0  0  0  0  0  0 ]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 17  0  0  0  0  0  0  0  0 ]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 17  0  0  0  0  0  0  0 ]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 17  0  0  0  0  0  0 ]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 17  0  0  0  0  0 ]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 17  0  0  0  0 ]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 17  0  0  0 ]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 17  0  0 ]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 17  0 ]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 17 ]]
```

5.4 NAIVE BAYES

```
0.9352941176470588
[[ 0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 ]
 [ 0 17  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 ]
 [ 0  0 17  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 ]
 [ 0  0  0 17  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 ]
 [ 0  0  0  0 17  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 ]
 [ 0  0  0  0  0 15  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 ]
 [ 0  0  0  0  0  0 16  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 ]
 [ 0  0  0  0  0  0  0  3  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 ]
 [ 0  0  0  0  0  0  0  0 17  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 ]
 [ 0  0  0  0  0  0  0  0  0 17  0  0  0  0  0  0  0  0  0  0  0  0  0  0 ]
 [ 0  0  0  0  0  0  0  0  0  0 17  0  0  0  0  0  0  0  0  0  0  0  0  0 ]
 [ 0  0  0  0  0  0  0  0  0  0  0 17  0  0  0  0  0  0  0  0  0  0  0  0 ]
 [ 0  0  0  0  0  0  0  0  0  0  0  0 17  0  0  0  0  0  0  0  0  0  0  0 ]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0 17  0  0  0  0  0  0  0  0  0  0 ]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0 17  0  0  0  0  0  0  0  0  0 ]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 17  0  0  0  0  0  0  0  0 ]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 17  0  0  0  0  0  0  0 ]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 17  0  0  0  0  0  0 ]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 17  0  0  0  0  0 ]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 17  0  0  0  0 ]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 17  0  0  0 ]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 17  0  0 ]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 17  0 ]]
```

Come possiamo osservare gli alberi decisionali, sia per la regressione che classificazione, sono quelli che sembrano dare i migliori risultati. Segue poi la regressione lineare e per ultimo il Naive Bayes.

6 VALUTAZIONE SU DATI SCONOSCIUTI

Gran Premio	Vincitore	Regressione Lineare	Alberi Decisionali (Regressione)	Alberi Decisionali (Classificazione)	Naive Bayes		
Gran Premio del Bahrein	Hamilton	Hamilton	Hamilton	Hamilton	Hamilton		
Gran Premio dell' Emilia Romagna	Verstappen	Verstappen	Verstappen	Verstappen	Verstappen		Predizione Corretta
Gran Premio del Portogallo	Hamilton	Hamilton	Hamilton	Hamilton	Hamilton		Predizione Errata
Gran Premio di Spagna	Hamilton	Hamilton	Hamilton	Hamilton	Hamilton		
Gran Premio di Monaco	Verstappen	Verstappen	Verstappen	Verstappen	Verstappen		
Gran Premio d'Azerbaigian	Perez	Perez	Perez	Perez	Ricciardo		
Gran Premio di Francia	Verstappen	Verstappen	Verstappen	Verstappen	Verstappen		
Gran Premio di Stiria	Verstappen	Verstappen	Verstappen	Verstappen	Verstappen		
Gran Premio d'Austria	Verstappen	Verstappen	Verstappen	Verstappen	Verstappen		
Gran Premio di Gran Bretagna	Hamilton	Hamilton	Hamilton	Hamilton	Hamilton		
Gran Premio d'Ungheria	Ocon	Ocon	Ocon	Ocon	Ocon		
Gran Premio del Belgio	Verstappen	Verstappen	/	Verstappen	Verstappen		
Gran Premio d'Olanda	Verstappen	Verstappen	Verstappen	Verstappen	Verstappen		
Gran Premio d'Italia	Ricciardo	Ricciardo	Ricciardo	Ricciardo	Ricciardo		
Gran Premio di Russia	Hamilton	Hamilton	Hamilton	Hamilton	Hamilton		
Gran Premio di Turchia	Bottas	Bottas	Bottas	Bottas	Bottas		
Gran Premio degli Stati Uniti	Verstappen	Verstappen	Verstappen	Verstappen	Verstappen		
Gran Premio di Città del Messico	Verstappen	Verstappen	Verstappen	Verstappen	Verstappen		
Gran Premio di San Paolo	Hamilton	Hamilton	Hamilton	Hamilton	Hamilton		
Gran Premio del Qatar	Hamilton	Hamilton	Hamilton	Hamilton	Hamilton		
Gran Premio d'Arabia Saudita	Hamilton	Hamilton	Hamilton	Hamilton	Hamilton		
Gran Premio di Abu Dhabi	Verstappen	???	???	???	???		

Verrà utilizzato il Gran Premio di Abu Dhabi come test per la demo durante la presentazione del progetto.

7 SVILUPPI FUTURI

In questa sezione andiamo a esaminare eventuali miglioramenti che possono essere fatti al progetto e che non sono stati utilizzati per mancanza di conoscenze e di tempo.

- Utilizzo di più modelli: sono stati presi in esame solo quattro modelli, che sono quelli visti a lezione. Per poter effettuare una analisi più approfondita sia per quanto riguarda la regressione che la classificazione, utilizzare ulteriori modelli può sicuramente tornarci utile.
- Dati sulle condizioni metereologiche: avere a disposizione dei dati sulle condizioni metereologiche in cui si sono svolte le gare può sicuramente essere utile ai fini della predizione: tipicamente le gare svoltesi in condizioni di pioggia sono quelle che hanno portato a risultati inaspettati e quindi possono influenzare molto la gara.
-

8 GLOSSARIO

- **Formula:** si riferisce all'insieme di regole che i partecipanti (team e piloti) devono rispettare.
- **Prove Libere:** le prove libere sono tre sessioni durante le quali i piloti possono prendere confidenza con la pista e gli ingegneri aggiustare e adattare meglio l'assetto della vettura. Attualmente ogni sessione dura 60 minuti, contro i 90 e i 45 adottati nelle precedenti stagioni.
- **Qualifiche:** le qualifiche servono a stabilire l'ordine di partenza della gara della domenica, la griglia di partenza. Attualmente la qualifica si compone di tre sessioni, denominate Q1, Q2 e Q3 della durata di 18, 15 e 12 minuti rispettivamente, con un sistema *knock-out*: le 5 vetture più lente al termine della Q1 sono eliminate, allo stesso modo le ultime 5, tra le vetture rimaste, al termine delle Q2 sono eliminate e le restanti combattono per la *Pole Position* nella Q3.
Il formato delle qualifiche è quello che ha subito più variazioni nel corso della storia della Formula Uno.
- **Pole Position:** è il termine che indica la prima posizione nella griglia di partenza.

- **Gran Premio:** erroneamente confuso con la gara della domenica, il Gran Premio indica tutti gli eventi che si svolgono dal giovedì alla domenica.
- **Costruttore:** un Costruttore può essere visto come il team stesso. Il termine deriva dal fatto che, inizialmente, un team costruiva sia telaio che motore per le monoposto di Formula Uno. Col tempo però, a causa delle grandi quantità di denaro da dover spendere per poter sviluppare una vettura, molti team hanno deciso di utilizzare un motore costruito da altri team, costruendo però da se il telaio, così da poter prendere parte al Mondiale Costruttori.
- **Mondiale Piloti:** il Mondiale Piloti è il titolo che si contendono ogni anno i piloti di Formula Uno. Alla fine di ogni gara, i primi dieci classificati ottengono un punteggio proporzionato alla posizione ($1^{\circ} = 25$ punti, $2^{\circ} = 18$ punti, ..., $10^{\circ} = 1$ punto, secondo l'assegnazione attuale). Il pilota che a fine Campionato avrà totalizzato più punti si aggiudicherà il titolo.
- **Mondiale Costruttori:** il Mondiale Costruttori è l'analogo del Mondiale Piloti per i team di Formula Uno. I punteggi sono calcolati come la somma dei punti ottenuti dai piloti di un team al termine di una gara. Il team che avrà totalizzato più punti a fine Campionato avrà totalizzato più punti si aggiudicherà il titolo.