



DATA-DRIVEN COMPARISON OF FORMULA 1 DRIVERS: A MACHINE LEARNING APPROACH

Prepared by: **SİNEM FAİDE ALTUN**
Student No: **090200360**
Submission Date: **June 14, 2024**
Course: **MAT 4902E**
Supervisor: **PROF. DR. ATABEY KAYGUN**

Contents

1	Introduction	3
2	Methodology	4
2.1	Fundamentals of Data in Formula 1 Racing	4
2.2	Data Preprocessing and Quality Assurance Techniques	5
2.2.1	Evaluation of Timestamps	6
2.2.2	Importance of Standardization	6
2.2.3	Handling Imbalanced Data	7
2.3	Modeling Approaches	9
2.3.1	Choice of Classification Algorithms	9
2.3.2	Evaluation Metrics	10
2.3.3	Dimensionality Reduction	12
2.4	Advanced Analytical Techniques for Comparison	14
2.4.1	Cosine Similarity	14
2.4.2	Hierarchical Clustering	15
3	Experiments	19
3.1	Collection and Organization of Data	19
3.1.1	Team Radio Audio Data	19
3.1.2	Telemetry Data	20
3.1.3	Merge of Team Radio and Telemetry Data	21
3.2	Track Visualization	22
3.3	Detecting Cause of Team Radio: Classification	23
3.4	Comparing Drivers with Hierarchical Clustering	24
4	Analysis	25

4.1	Team Radio Classification	25
4.1.1	Logistic Regression	25
4.1.2	Extreme Gradient Boosting	25
4.2	Comparison of Formula 1 Drivers	26
4.2.1	Within-Season Comparison	26
4.2.2	Across-Season Comparison	30
5	Conclusion	32
A	Appendix	35
A.1	Detailed Track Visualizations	35

Introduction

Formula 1 is an extreme racing sport where 20 drivers from 10 teams compete every weekend during a season, each time at a different track and country. From the pit stop crew to the mechanical engineers, strategists, team principals, and the driver, it is a combination of exceptional teamwork. Formula 1 cars contain over 300 sensors, and all team units analyze this data to constantly optimize and upgrade their strategies.

In the fast-paced and dynamic environment of Formula 1, data and its analysis are crucial. We were curious to gain access to this data and understand what makes a team win. In Formula 1, it is not just about who is the fastest but who best utilizes their abilities, whether through car mechanics or driver skill. We aim to pinpoint the optimal approach to compare Formula 1 drivers, considering that some drive the same car but differ in rankings. Can we identify similarities between point earners and non-scorers? Is the car really holding some drivers back? What reasons can we identify when races end prematurely due to DNFs?

To answer these questions, we will conduct a data-driven comparison of Formula 1 drivers, hoping to uncover new insights into this physically and mentally challenging sport.

All the code and detailed outputs for this project can be reached through the GitHub repository **Data-Driven Comparison of Formula 1 Drivers: A Machine Learning Approach**.



Figure 1.1: 2024 Formula 1 Drivers and Teams

Methodology

In this chapter, we aim to review the methodologies used in this project, understand the mathematical logic and algorithms behind them, and gain insights into their contributions.

First, we will understand the type of data we are handling. Next, we will process and prepare this data for the following phase: creating classification models to identify patterns between Formula 1 pilots making team calls and their car or track conditions. Finally, we will compare the pilots.

2.1 Fundamentals of Data in Formula 1 Racing

Telemetry data of a Formula 1 car, unlocks immense knowledge to its condition, at almost every millisecond it moves, it refers to the system built on the constant wireless data transmission from the race car to the team's engineers live[1]. Aston Martin Aramco Formula 1 Team, explains the pre-telemetry stages of Formula 1, as a constant receive-give feedback cycle based on latest laps of the driver. The communication channels between the pit and the driver were simple dashboards, instead of the smooth radio communication we have now. Data, of course, has always been crucial, even if around 600 sensors have not existed yet, there would be engineers on the track visually inspecting the engine and the situation of the car. These early stages have now progressed to race engineers constantly tracking live telemetry data, analysing strategies, and maintaining a continuous transmission with the driver.



Figure 2.1: Scuderia Ferrari Formula 1 Team Post-Race Dashboard (AWS): Drivers Charles Leclerc and Carlos Sainz

It is crucial for us to really grasp what telemetry data entails, as it is the primary data we have

in this project. Key areas of telemetry data would be brake and tyre temperatures, insights on how the pilot is driving the car with throttle and brake usages.

Term	Description
Date	The date when the data was recorded.
SessionTime	The time elapsed since the start of the session.
DriverAhead	The identifier of the driver ahead on the track.
DistanceToDriverAhead	The distance to the driver ahead, measured in meters.
Time	The exact time at which the data point was recorded.
RPM	The revolutions per minute of the car's engine.
Speed	The speed of the car in kilometers per hour.
nGear	The gear number in which the car is currently running.
Throttle	The percentage of throttle applied.
Brake	The percentage of brake applied.
DRS	The status of the Drag Reduction System (open or closed).
Source	The source of the data (e.g., telemetry, sensor).
Distance	The total distance traveled by the car.
RelativeDistance	The distance relative to another reference point or car.
Status	The current status of the car (e.g., running, pit stop).
X	The X-coordinate of the car's position on the track.
Y	The Y-coordinate of the car's position on the track.
Z	The Z-coordinate of the car's position on the track.

Table 2.1: Descriptions of Telemetry Data Terms

Another big aspect of our project is the team radio. In Formula 1, radio communication is crucial, connecting all members of the team—from mechanics and pit crew to drivers and race engineers. This communication is essential for developing and updating strategies, but it also involves a high level of encryption, with teams maintaining a complex matrix of permissions to control who can hear what. The radio data we have obtained consists of short clips (under 10 seconds) that are often noisy and hard to interpret. Although Formula 1 teams sometimes assign trainees to transcribe these audios, converting them using speech recognition models remains a challenge. Therefore, for now, we will focus solely on the existence of a team radio, rather than the specific content of the communications.

2.2 Data Preprocessing and Quality Assurance Techniques

As Formula 1 cars compose immense amount of raw data, it is our duty to organize and preprocess them for the next steps of our project.

2.2.1 Evaluation of Timestamps

At first glance, interpreting UTC timings can be challenging. Therefore, when merging team radio data (in UTC) with telemetry data (in YYYY-MM-DD HH:DD:ss.SSS format), we converted both to the latter time format. However, machine learning algorithms cannot process timestamp data types, necessitating conversion. Therefore, we revert to UTC to obtain an integer type.

To convert a timestamp to UTC, we first need to represent the timestamp in a format that allows us to perform the conversion. Then, we can use appropriate tools or methods to perform the conversion.

For example, consider the timestamp 2023-03-05 15:03:38.501. To convert this timestamp to UTC, we can use the Unix timestamp format, which represents the number of seconds since the Unix epoch (January 1, 1970, 00:00:00 UTC).

The Unix timestamp corresponding to the timestamp 2023-03-05 15:03:38.501 is 1678028618. This means that 1678028618 seconds have elapsed since the Unix epoch.

2.2.2 Importance of Standardization

When training classification models, one should be aware of the distribution and statistical structure of their data. Standardization allows us to transform data to have a mean of zero and a standard deviation of one, converting all features to a common format. When inspecting the models we intend to train, it's notable that while tree based algorithms such as XGBoost do not necessarily demand standardization and are unaffected by monotonic transformations of the data, not assuming any particular distribution, Logistic Regression assumes predictors to be approximately normally distributed and operates effectively with standardized data.

Two of the standardization techniques we utilized are Z-Score Standardization and Normalization.

Z-score Standardization

The z-score (standard score) returns transformed values that represent the z-scores of the original data, where they have a mean of zero and a standard deviation of one. In a vast amount of and crowded data, Z-scores provided us a way of comparing individual data points to a standard normal distribution, measuring how many standard deviations that particular data point is away from the mean of the dataset. This technique is often preferred because programmer can compute the probability of a score occurring within our normal distribution.[2]

The formula for Z-score standardization is given by:

$$Z = \frac{x - \mu}{\sigma}$$

Where:

- Z is the standardized value (the z-score).
- x is the original value.
- μ is the mean of the dataset.
- σ is the standard deviation of the dataset.

Normalization

While not as immune to outliers as z-score standardization, Normalization (Min-Max Scaling) is another option to standardize our data, this time to a fixed range from zero to one. In Min-Max Scaling, the minimum value of the feature is transformed to 0, the maximum to 1, and all other values in between are proportionally distributed within this range.

The formula for min-max scaling is given by:

$$X_{\text{scaled}} = \frac{X - X_{\min}}{X_{\max} - X_{\min}}(max - min) + min$$

Where:

- X_{scaled} is the scaled value.
- X is the original value.
- X_{\min} is the minimum value of the feature in the dataset.
- X_{\max} is the maximum value of the feature in the dataset.
- min is the minimum value of the desired range after scaling.
- max is the maximum value of the desired range after scaling.

2.2.3 Handling Imbalanced Data

In real-world problems, rare events, unusual patterns, and abnormal behavior do occur. The infrequent nature of these events creates highly imbalanced datasets.[3]. This situation applies to our dataset as well. In a fast-paced racing environment such as Formula 1, it is not unusual for drivers to avoid contacting the pit. Team radio calls usually occur when there are issues

with the car, when the driver requests race updates, engineers call for a pit stop, or strategy discussions take place. However, there is controversy surrounding radio calls, with issues related to broadcasting them live or the driver sometimes asking for silence. Considering all these factors and the limited availability of data, it is understandable that we do not have much positive team radio data.

Nonetheless, as imbalanced data problem is very common in machine learning applications, there are numerous ways to handle them. One of being, stratified sampling. Stratifying is done when splitting the data to its training and testing portions. It helps us maintain the same class distribution as the original dataset. Therefore, the minority and majority classes are both represented and potential bias issues are prevented.

We will visualize the process of stratified train-test splitting using a bar chart as an example. These values are not accurate to our project.

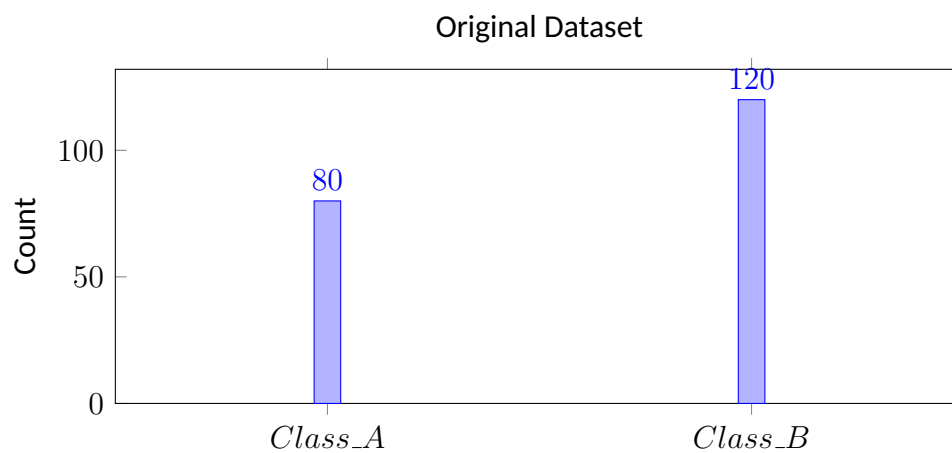


Figure 2.2: Class Distribution in Original Dataset

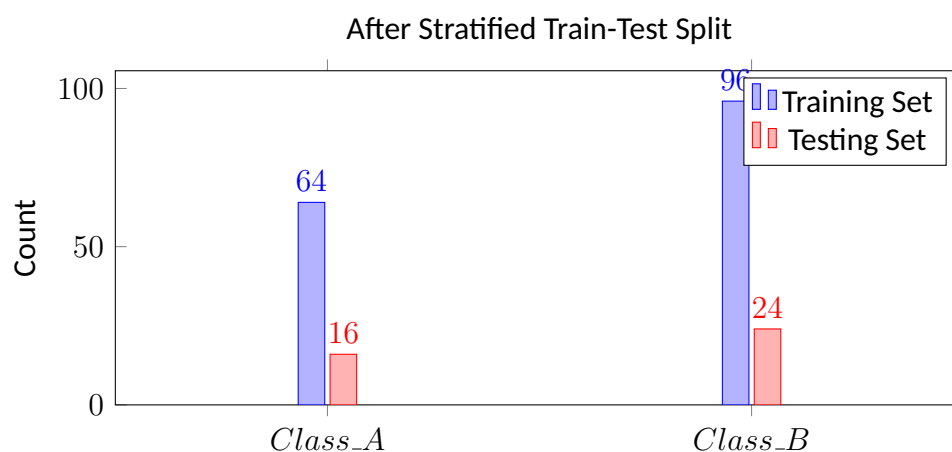


Figure 2.3: Class Distribution in Training and Testing Sets

2.3 Modeling Approaches

2.3.1 Choice of Classification Algorithms

Logistic Regression

Logistic regression is a very classic and reliable algorithm with binary classification problems, because of its simplicity and relatively fast computation time. In our project, this algorithm is used to determine if there is a distinct formula linking a cause and an effect for the pilot to make a team call. If so, what contributes more to that cause, track conditions (X-Y-Z position data) or car conditions (telemetry data).

The logistic regression formula for binary classification is given by the sigmoid function:

$$P(Y = 1) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_n X_n)}} \quad (2.1)$$

where:

- Y is the binary outcome (e.g., 1 for the positive class, 0 for the negative class),
- X_1, X_2, \dots, X_n are the features,
- $\beta_0, \beta_1, \dots, \beta_n$ are the coefficients,
- e is the base of the natural logarithm.

This formula transforms the linear combination of features into probabilities. The logistic regression model predicts $Y = 1$ if the probability is greater than a threshold (typically 0.5) and $Y = 0$ otherwise.

Extreme Gradient Boosting (XGBoost)

Extreme gradient boosting's (XGBoost) fundamental lies within the Gradient Boosting framework by (Friedman, 2001)[4], which is a combination of predictions from multiple weak decision models, usually trees. XGBoosting's speed and performance by parallel processing along with it's built in regularization techniques to prevent overfitting makes it a preferred prediction algorithm. It deals well with large scales of data, which is why we implemented it in our binary classification problem as well.

XGBoost algorithm operates as the following:

Step 1: Initialize Model

- Initialize an empty tree ensemble model: $F_0(x) = 0$.

Step 2: Iterate Over Trees

1. For each iteration $t = 1, 2, \dots, T$:

(a) Compute the gradient and Hessian of the loss function:

$$g_i = \frac{\partial L(y_i, F(x_i))}{\partial F(x_i)} \quad \text{and} \quad h_i = \frac{\partial^2 L(y_i, F(x_i))}{\partial F(x_i)^2}.$$

(b) For each leaf node j in tree $t - 1$, compute:

- Gain:

$$\text{gain} = \frac{1}{2} \left[\frac{\sum_{i \in I_j} g_i}{\sum_{i \in I_j} h_i + \lambda} \right]^2$$

- Similarity:

$$\text{similarity} = \frac{(\sum_{i \in I_j} g_i)^2}{\sum_{i \in I_j} h_i + \lambda}$$

(c) Select the best split for each leaf node based on maximum gain or similarity.

(d) Update the leaf values:

$$\gamma_j = \frac{\sum_{i \in I_j} g_i}{\sum_{i \in I_j} h_i + \lambda}$$

(e) Prune the tree if necessary using regularization parameters.

Step 3: Ensemble Model

- Final model prediction: $F(x) = \sum_{t=1}^T f_t(x)$.

2.3.2 Evaluation Metrics

It is crucial to establish the criteria by which your model's performance will be assessed, determining its efficiency and the accuracy of its predictions. For classification problems, various evaluation metrics are available and we will review them in this section.

Accuracy

The accuracy metric represents the ratio of correctly predicted instances to the total number of instances. It is a proficient and reliable metric for balanced datasets. In our case, as our dataset is very much imbalanced, spite of the techniques we utilized to overcome this issue, accuracy may not be a good metric option.

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}} \times 100 \quad (2.2)$$

where:

Number of Correct Predictions : The count of correctly classified instances

Total Number of Predictions : The total number of instances

Figure 2.4: Accuracy Calculation Formula

Precision

Precision is the ratio of true positives to the sum of true positives and false positives. It is a smooth indicator of the model's efficiency if our priority is minimizing false positives. In our case false positives are equivalent to falsely expecting team radio calls, thus precision is a matter we should be aware of.

$$Precision = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \quad (2.3)$$

where:

True Positives : The count of correctly predicted positive instances

False Positives : The count of incorrectly predicted positive instances

Figure 2.5: Precision Calculation Formula

Recall

Recall is the ratio of true positives to the sum of true positives and false negatives. It is important when capturing as many positive instances as possible is a priority. In our context, positive instances refer to team radio calls occurring. Detecting as many team calls as possible could be a beneficial approach, ensuring constant communication with the pit about potential issues or strategy updates.

$$Recall = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \quad (2.4)$$

where:

True Positives : The count of correctly predicted positive instances

False Negatives : The count of incorrectly predicted negative instances

Figure 2.6: Recall Calculation Formula

F1 Score

F1 score proposes a harmonic mean of precision and recall, providing a balance perspective on false positives and false negatives. It is an adequate evaluation metric in our case as there is an imbalance between positive and negative classes within our dataset.

$$F1\ Score = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (2.5)$$

Figure 2.7: F1 Score Calculation Formula

ROC-AUC (Receiver Operating Characteristic - Area Under the Curve)

This metric represents the area under the ROC curve, which plots the true positive rate against the false positive rate. It is used to evaluate the model's ability to distinguish between classes.

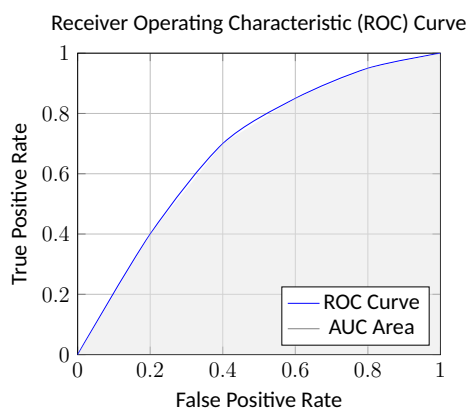


Figure 2.8: Receiver Operating Characteristic (ROC) Curve with Area Under the Curve (AUC) highlighted.

2.3.3 Dimensionality Reduction

Principal Component Analysis (PCA)

Principal component analysis (PCA) is a mathematical technique that is designed to reduce the dimensionality of the data while keeping most of the variation in the data set.[5]

This reduction is achieved by identifying directions along which the variation in the data is maximal, these directions are referred as principal components. Consequently, samples can

be represented by a considerably smaller number of variables instead of values for thousands of variables.[6]. Deciding on an optimal sample number, is the tricky part. We decided that applying dimensionality reduction to a dataset, reducing it to n components while maintaining an explained variance of around 90%, yields the most optimal solutions. Essentially, we aim for a reduction process where the resulting model still explains a large portion of the data.

We measured explained variance values for different number of components, on both the 2022 and 2023 season. Below are the results.

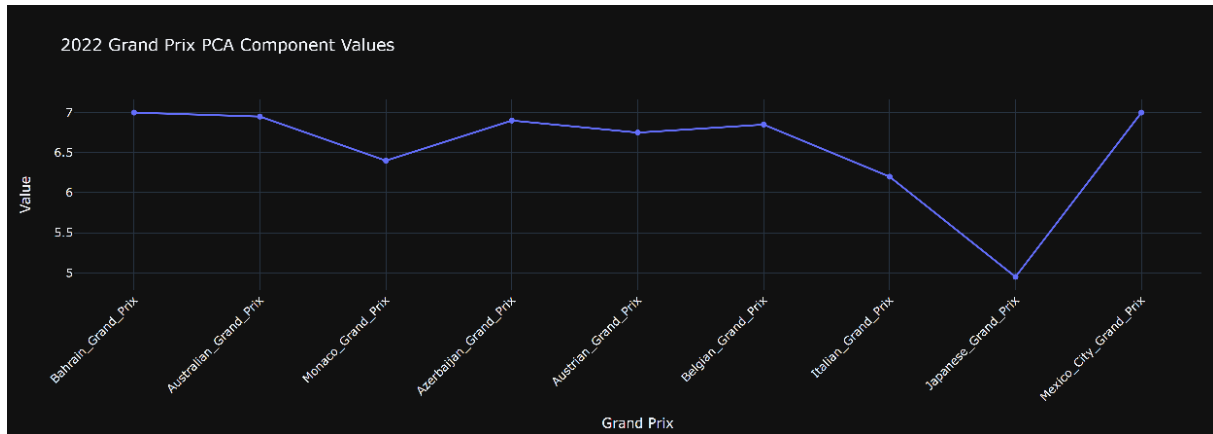


Figure 2.9: Optimal PCA Components by Grand Prix, 2022 Season

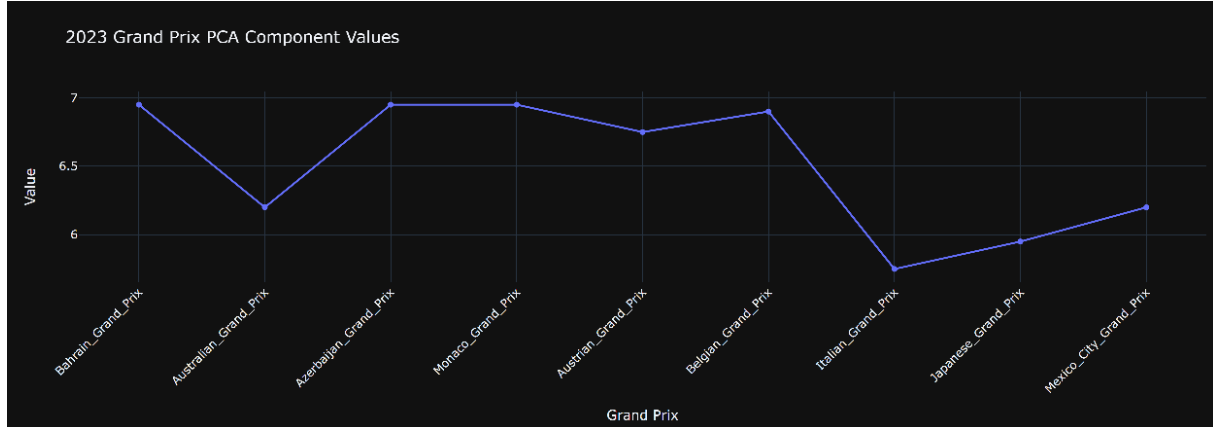


Figure 2.10: Optimal PCA Components by Grand Prix, 2023 Season

The principal components PC_1, PC_2, \dots, PC_p are formed by arranging the eigenvectors in decreasing order of their corresponding eigenvalues.

$$PC_i = v_i \quad (2.6)$$

The transformed data Y is obtained by projecting the original data onto the principal components:

$$Y = XV \quad (2.7)$$

where:

X : Original data matrix

V : Matrix of principal components

Figure 2.11: Principal Component Analysis (PCA) Process

2.4 Advanced Analytical Techniques for Comparison

2.4.1 Cosine Similarity

In the realm of Mathematics, cosine similarity measures the similarity between two vectors, related to the cosine of the angle between them.

The formula for cosine similarity is given by:

$$\text{Cosine Similarity}(A, B) = \frac{A \cdot B}{\|A\| \|B\|}$$

Where:

- A and B are the vectors to be compared.
- $A \cdot B$ denotes the dot product of vectors A and B .
- $\|A\|$ and $\|B\|$ represent the Euclidean norms of vectors A and B , respectively.

While it is a much preferred method in Natural Language Processing problems in detecting the similarities between tokens or texts, for our problem of comparing 20 Formula 1 drivers, it will be beneficial as well. For every season and Grand Prix, we evaluate the similarity between pairs of pilots using telemetry and date data, which we have dimensionally reduced. For each pilot comparison, we compute their average similarity and append it to a comprehensive comparison matrix that includes all 20 pilots.

What we end up with is a 20x20 cosine similarity matrix, where each value represents the similarity between the pilot in its respective row and column. Values closer to 1 indicate a higher similarity, while those closer to 0 suggest no resemblance.

Table 2.2: Illustration of a 5x5 Cosine Similarity Matrix

	x_1	x_2	x_3	x_4	x_5
x_1	1				
x_2		1			
x_3			1		
x_4				1	
x_5					1

2.4.2 Hierarchical Clustering

Hierarchical clustering, is algorithm is a widely used tool in data science for grouping similar data points. This grouping is done through distance calculations whether that is the standard Euclidean distance or Manhattan distance. Hierarchical clustering takes on an iterative approach and keeps merging or splitting clusters based on the distance between data points until all data points belong to a single cluster.

Euclidean Distance

The Euclidean distance between two points $\mathbf{p} = (p_1, p_2, \dots, p_n)$ and $\mathbf{q} = (q_1, q_2, \dots, q_n)$ in an n -dimensional space is given by the formula:

$$\text{Euclidean Distance} = \sqrt{\sum_{i=1}^n (q_i - p_i)^2} \quad (2.8)$$

Manhattan Distance

The Manhattan distance (also known as the taxicab distance) between two points $\mathbf{p} = (p_1, p_2, \dots, p_n)$ and $\mathbf{q} = (q_1, q_2, \dots, q_n)$ in an n -dimensional space is given by the formula:

$$\text{Manhattan Distance} = \sum_{i=1}^n |q_i - p_i| \quad (2.9)$$

Types of Linkages in Hierarchical Clustering

While distance metrics such as Euclidean distance or Manhattan distance, refer to the distance between individual data points, the term **linkage** determines how the distance between clusters is calculated when merging them into larger clusters. There are several common linkage methods: Single, average, complete, centroid and ward's linkage. However in this project, we only utilized the first three.

Single Linkage

In single linkage hierarchical clustering, the distance between two clusters A and B is defined

as the minimum distance between any single data point in cluster A and any single data point in cluster B :

$$\text{Single Linkage Distance}(A, B) = \min_{x \in A, y \in B} \text{distance}(x, y) \quad (2.10)$$

Average Linkage

In average linkage hierarchical clustering, the distance between two clusters A and B is defined as the average distance between all pairs of data points where one point belongs to cluster A and the other belongs to cluster B :

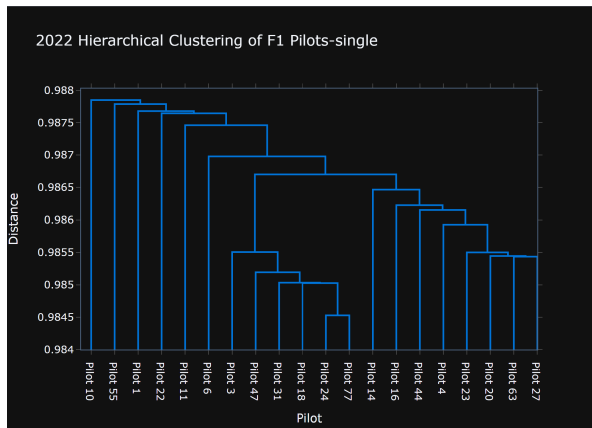
$$\text{Average Linkage Distance}(A, B) = \frac{1}{|A| \cdot |B|} \sum_{x \in A} \sum_{y \in B} \text{distance}(x, y) \quad (2.11)$$

Complete Linkage

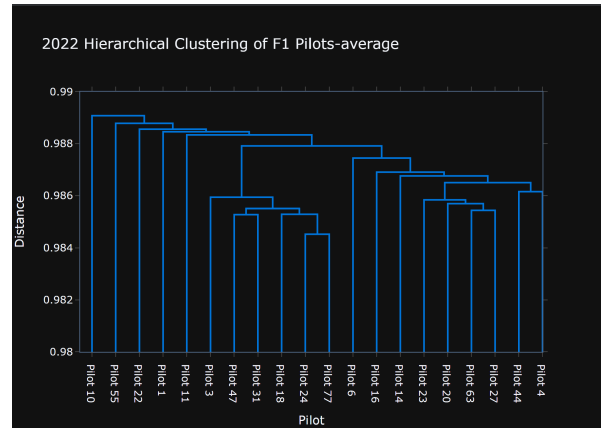
In complete linkage hierarchical clustering, the distance between two clusters A and B is defined as the maximum distance between any single data point in cluster A and any single data point in cluster B :

$$\text{Complete Linkage Distance}(A, B) = \max_{x \in A, y \in B} \text{distance}(x, y) \quad (2.12)$$

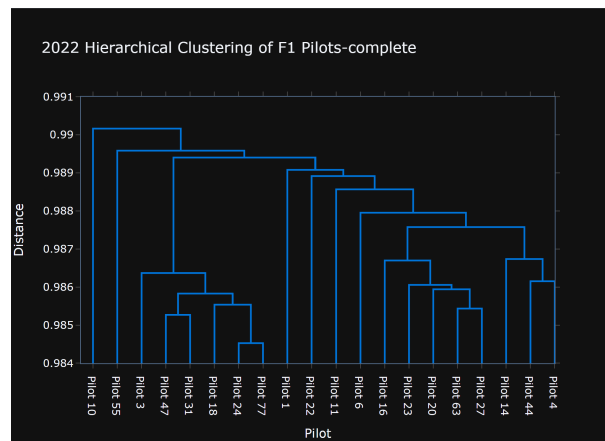
Building clusters with different types of linkages yielded not very different results when comparing pilots.



(a) Single Linkage



(b) Average Linkage



(c) Complete Linkage

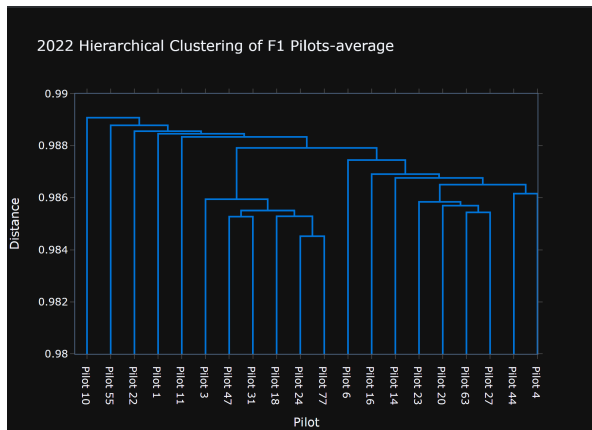
Figure 2.12: Pilot Comparison Dendrograms using different linkage methods, 2022 Bahrain Grand Prix

Dendrograms

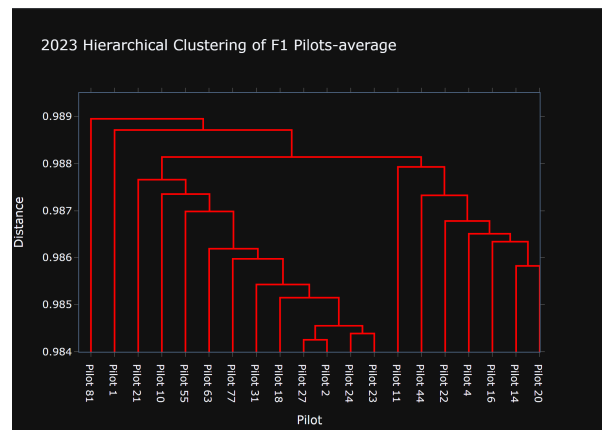
Dendrograms are hierarchical tree-like visualizations used in cluster analysis. They allow us to visually perceive the arrangement of clusters produced by hierarchical clustering algorithms. Dendrograms are formed by grouping the most similar elements and then progressively merging clusters with the highest similarity. This iterative grouping continues until similarities lessen and no further grouping can occur.

In our case, since we converted our sample points in clustering to distances, longer lines in the dendrogram represent greater distances between elements, indicating less or no similarity.

Below are the pilot similarity dendrograms for the Bahrain Grand Prix, illustrating visuals for both the 2022 and 2023 Formula 1 seasons.



(a) Formula 1, 2022 Season



(b) Formula 1, 2023 Season

Figure 2.13: Pilot Comparison Dendrograms using average method, Bahrain Grand Prix

Tanglegrams

Tanglegrams are often used in biology, aiming to compare the structure of two distinct phylogenetic trees, where the connection links illustrate the same specie's structure across different trees. A tanglegram consists of two trees sharing the same leaf set, where corresponding leaves in both trees are connected by an edge.[7]

Since dendrograms are types of tree structures which we used in hierarchical clustering visualizations, tanglegrams could also be used on dendrograms and see how they align or differ in their hierarchical clustering patterns. Instead of inspecting the evolutionary steps for a specie, we will be examining the evolution of Formula 1 pilots across different seasons. We can assess how the clustering of pilots has changed over time.

Experiments

3.1 Collection and Organization of Data

3.1.1 Team Radio Audio Data

Formula 1 offers live timing data to users through general or premium subscriptions. This data includes crucial race details such as pit stop timings, tire degradation, onboard camera feeds and more. In our project, our focus lies on team radio conversations, for which the JSON files suffice. However, before delving into the audio content, it's important to grasp the structure of the JSON stream containing team radio data for each Grand Prix in 2023. To initiate this process, we began by thoroughly analyzing the racing schedule for 2023.

```
1 race_schedule_2023 = fastf1.get_event_schedule(2023)
```

RoundNumber	Country	Location	OfficialEventName	EventDate	EventName
0	Bahrain	Sakhir	FORMULA 1 ARAMCO PRE-SEASON TESTING 2023	2023-02-25	Pre-Season Testing
1	Bahrain	Sakhir	FORMULA 1 GULF AIR SAUDI ARABIAN GRAND PRIX 2023	2023-03-05	Bahrain Grand Prix
2	Saudi Arabia	Jeddah	FORMULA 1 STC SAUDI ARABIAN GRAND PRIX 2023	2023-03-19	Saudi Arabian Grand Prix
3	Australia	Melbourne	FORMULA 1 ROLEX AUSTRALIAN GRAND PRIX 2023	2023-04-02	Australian Grand Prix
4	Azerbaijan	Baku	FORMULA 1 AZERBAIJAN GRAND PRIX 2023	2023-04-30	Azerbaijan Grand Prix
5	United States	Miami	FORMULA 1 CRYPTO.COM MIAMI GRAND PRIX 2023	2023-05-07	Miami Grand Prix
6	Monaco	Monaco	FORMULA 1 GRAND PRIX DE MONACO 2023	2023-05-28	Monaco Grand Prix
7	Spain	Barcelona	FORMULA 1 AWS GRAN PREMIO DE ESPAÑA 2023	2023-06-04	Spanish Grand Prix
8	Canada	Montréal	FORMULA 1 PIRELLI GRAND PRIX DU CANADA 2023	2023-06-18	Canadian Grand Prix
9	Austria	Spielberg	FORMULA 1 ROLEX GROSSER PREIS VON ÖSTERREICH 2023	2023-07-02	Austrian Grand Prix
10	Great Britain	Silverstone	FORMULA 1 ARAMCO BRITISH GRAND PRIX 2023	2023-07-09	British Grand Prix
11	Hungary	Budapest	FORMULA 1 QATAR AIRWAYS HUNGARIAN GRAND PRIX 2023	2023-07-23	Hungarian Grand Prix
12	Belgium	Spa-Francorchamps	FORMULA 1 MSC CRUISES BELGIAN GRAND PRIX 2023	2023-07-30	Belgian Grand Prix
13	Netherlands	Zandvoort	FORMULA 1 HEINEKEN DUTCH GRAND PRIX 2023	2023-08-27	Dutch Grand Prix
14	Italy	Monza	FORMULA 1 PIRELLI GRAN PREMIO D'ITALIA 2023	2023-09-03	Italian Grand Prix
15	Singapore	Marina Bay	FORMULA 1 SINGAPORE AIRLINES SINGAPORE GRAND PRIX 2023	2023-09-17	Singapore Grand Prix
16	Japan	Suzuka	FORMULA 1 LENovo JAPANESE GRAND PRIX 2023	2023-09-24	Japanese Grand Prix
17	Qatar	Lusail	FORMULA 1 QATAR AIRWAYS QATAR GRAND PRIX 2023	2023-10-08	Qatar Grand Prix
18	United States	Austin	FORMULA 1 LENovo UNITED STATES GRAND PRIX 2023	2023-10-22	United States Grand Prix
19	Mexico	Mexico City	FORMULA 1 GRAN PREMIO DE LA CIUDAD DE MEXICO 2023	2023-10-29	Mexico City Grand Prix
20	Brazil	São Paulo	FORMULA 1 ROLEX GRANDE PRÊMIO DE SÃO PAULO 2023	2023-11-05	São Paulo Grand Prix
21	United States	Las Vegas	FORMULA 1 HEINEKEN SILVER LAS VEGAS GRAND PRIX 2023	2023-11-18	Las Vegas Grand Prix
22	Abu Dhabi	Yas Island	FORMULA 1 ETIHAD AIRWAYS ABU DHABI GRAND PRIX 2023	2023-11-26	Abu Dhabi Grand Prix

Figure 3.1: Creating an EventSchedule object for 2023 and the resulting output.

Upon initial inspection we notice minor details that require attention to prevent potential issues.

- Firstly, we have decided to exclude pre-season testing from our project to streamline our focus.
- Additionally, the 2023 Emilia Romagna Grand Prix, listed on the Formula 1 race calendar, was canceled due to flooding, and thus, it is not included in the data table.

- As our project progresses, we plan to compile a comprehensive dictionary containing all relevant data. It is important to note that the keys of dictionaries cannot be countries, as certain countries may host multiple races (e.g., the United States), potentially leading to overwrite issues and the loss of valuable data. Therefore, we intend to declare variables for event names, specifically the names of the Grand Prix.

Finally, this results in a total of 22 races' worth of team radio data for our project.

Moving forward, we proceed to generate the necessary links for accessing team radio data.

```
1 team_radio_links[event_name] = "https://livetiming.formula1.com/static/"
2 + date.strftime('%Y') + "/" + date.strftime('%Y-%m-%d') + "_"
3 + event_name + "/" + date.strftime('%Y-%m-%d') + "_Race/TeamRadio.json"
```

Listing 3.2: General format of the JSON Stream Link for accesing team radio data.

The JSON Streams for each Grand Prix looks like the following format:

```
1 {
2     "Utc": "2023-03-05T14:20:25.563Z",
3     "RacingNumber": "20",
4     "Path": "TeamRadio/KEVMAG01_20_20230305_141949.mp3"
5 }
```

We create a dictionary which has Grand Prix names as keys, containing JSON stream data. However, we convert UTC date types to timestamp format, to match with telemetry data's date effortlessly.

Team radio audios are short, shorter than 30 seconds, noisy segments, where most of the time it is hard to comprehend the conversation. There could be many reasons to trigger the driver to communicate with the race engineers: race strategies, penalty discussions, team hierarchy, problems with the car, track or even simple entertainment. Later on we will conduct tests to try and spot a comprehensive relation but for now, it is enough to understand the structure of team radio audio.

3.1.2 Telemetry Data

Data collection and analysis is one of the most important aspects of Formula 1, even leading to discussions whether it is the pilot, the mechanic or the race engineers who analyse the data, make the team win. Telemetry data is, without a doubt, at the top of the data hierarchy in Formula 1 racing. With around 600 sensors equipped with, Formula 1 race cars produce 35 megabytes of raw telemetry per 2-minute lap [8]. Speeding 220mph on average, telemetry data is the best direct source of catching up to and observing the car.

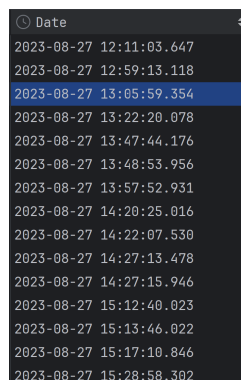
Fast-F1 Python package, contributes enormous amount of telemetry data for each race of 2023. We organize this data matching telemetry with pilots, and then grand prix.

Variable	Type	Description
Speed	float	Car speed [km/h]
RPM	int	Car RPM
nGear	int	Car gear number
Throttle	float	0-100 Throttle pedal pressure [%]
Brake	bool	Brakes are applied or not
DRS	int	DRS indicator
X	float	X position [1/10 m]
Y	float	Y position [1/10 m]
Z	float	Z position [1/10 m]
Status	string	Flag - OffTrack/OnTrack
Time	timedelta	Time (o is start of the data slice)
SessionTime	timedelta	Time elapsed since the start of the session
Date	datetime	The full date + time at which this sample was created
Source	str	Flag indicating how this sample was created

Table 3.1: Description of Car Data and Position Data Variables in Telemetry Data

3.1.3 Merge of Team Radio and Telemetry Data

The reference variable used to perform an outer join on both datasets is the **date**. If a team radio entry exists for the same date as the telemetry data, it is assigned a True (1) value. If not on the exact date, then the next attempt is to match it with a previous date, and if that fails, then the last resort is to match it with a subsequent date. Telemetry dates that do not find a corresponding team radio entry are assigned a False value (0). This alignment of dates allows us to pair all team radio entries with telemetry data.



```

Date
2023-08-27 12:11:03.647
2023-08-27 12:59:13.118
2023-08-27 13:05:59.354
2023-08-27 13:22:20.078
2023-08-27 13:47:44.176
2023-08-27 13:48:53.956
2023-08-27 13:57:52.931
2023-08-27 14:20:25.016
2023-08-27 14:22:07.530
2023-08-27 14:27:13.478
2023-08-27 14:27:15.946
2023-08-27 15:12:40.023
2023-08-27 15:13:46.022
2023-08-27 15:17:10.846
2023-08-27 15:28:58.302

```

Figure 3.2: Team Radio Dates Data

Date	X	Y	Z	RPM	Brake	DRS	Throttle	Speed	nGear
2023-08-27 13:05:58.817	1252.023502	1626.517718	510.112007	5833	False	1	5	66	2
2023-08-27 13:05:58.975	1238.000000	1601.000000	511.000000	5947	False	1	6	66	2
2023-08-27 13:05:59.097	1229.173203	1578.831505	511.377195	6062	False	1	7	67	2
2023-08-27 13:05:59.297	1218.397340	1540.160026	511.710055	6027	False	1	10	68	2
2023-08-27 13:05:59.354	1216.000000	1530.000000	512.000000	6186	False	1	10	68	2

Figure 3.3: Telemetry Data

Date	Team Radio	X	Y	Z	RPM	Brake	DRS	Throttle	Speed	nGear
2023-08-27 13:05:58.817	0.0	1252.023502	1626.517718	510.112007	5833.0	False	1.0	5.0	66.0	2.0
2023-08-27 13:05:58.975	0.0	1238.000000	1601.000000	511.000000	5947.0	False	1.0	6.0	66.0	2.0
2023-08-27 13:05:59.097	1.0	1229.173203	1578.831505	511.377195	6062.0	False	1.0	7.0	67.0	2.0
2023-08-27 13:05:59.297	0.0	1218.397340	1540.160026	511.710055	6027.0	False	1.0	10.0	68.0	2.0
2023-08-27 13:05:59.354	0.0	1216.000000	1530.000000	512.000000	6186.0	False	1.0	10.0	68.0	2.0

Figure 3.4: Merged data frame

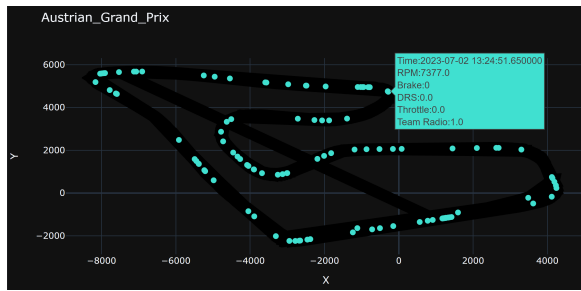
Figure 3.5: Outer joining telemetry and team radio data, by date feature.

3.2 Track Visualization

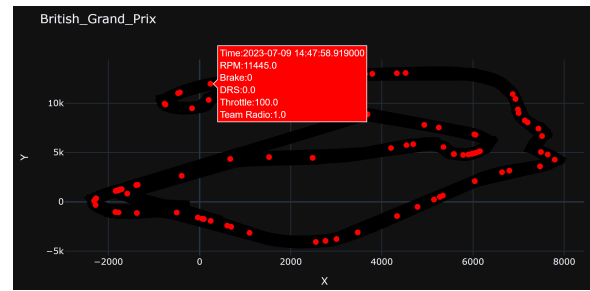
The coordinate features in our now merged data frame, grants us to visualize tracks. We conduct this process using Plotly[9], as it creates intractable graphs. All laps of all pilot in a track during a race is visualized with data points highlighted, where radio conversations occur. When hovered over these data points, it is possible to see telemetry data of the car during that position and time.

This visualization gives us a comprehensive view of the track with data points, such that we can observe the intensity of these points in certain positions and can derive meaningful insights from our observation, like the increase in communication during curves on the track. Nonetheless, it is important to be aware of the high computation of these graphs, as each of them contain over $727,871 \times 8 = 5,822,968$ data.

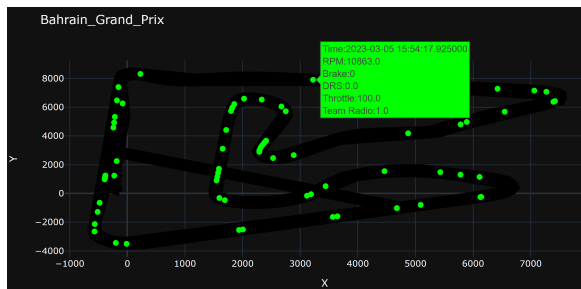
To see all track visualizations, see **Detailed Track Visualizations**A.1



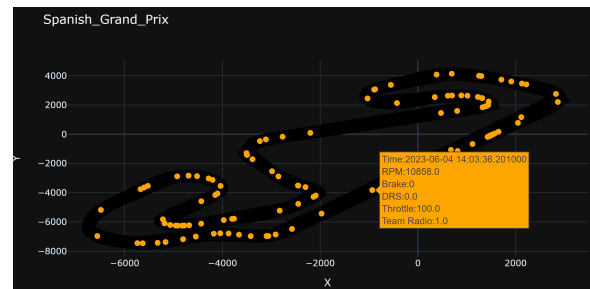
(a) Austrian Grand Prix



(b) British Grand Prix



(c) Bahrain Grand Prix



(d) Spanish Grand Prix

Figure 3.6: Some of the track visualizations.

3.3 Detecting Cause of Team Radio: Classification

Is there an exact cause that triggers the pilot to communicate with his team during a race? If so, which setting triggers this act more; track conditions or car conditions? We can test our speculation by setting up two distinct classification models: one using X-Y-Z coordinate data as input, and the other utilizing telemetry data, including all drivers, but one track only.

The biggest obstacle we face during this stage is our imbalance of team radio data. Approximately 99.98% of our position-telemetry data has no team radio (for one Grand Prix), therefore making it difficult to train the classification models in the positive region.

As mentioned in the methodology sub-chapter **Data Preprocessing and Quality Assurance Techniques** 2.2, we utilized certain techniques to handle imbalanced data, and applied standardization, converting timestamp to UTC to make it integer. Spite of that, the Logistic Regression and XGBoost models did not deliver victorious results, no matter the metric **ROC-AUC**, **F1**, **accuracy**, **precision**, **recall** we evaluated them. This points to two estimated conclusions: either this problem of Formula 1 requires more complex models such as neural networks, or there is not a regular pattern between team radio conversations occurring and their cause.

3.4 Comparing Drivers with Hierarchical Clustering

In our final experiment, we compare Formula 1 drivers in the same races, both within a single season and across the 2022 and 2023 seasons. We measure their similarities using cosine similarity computations and convert it into a distance matrix by subtracting from one. So, similar drivers, having a cosine similarity closer to 1, will have a distance closer to 0.

Hierarchical clustering is then performed to cluster drivers based on their similarities. We visualize these clusters with the help of dendrograms.

Analysis

4.1 Team Radio Classification

For our team radio classification problem, we wanted to determine which model would perform better, a model trained on track conditions (X-Y-Z coordinates) or telemetry data. By this goal, we worked on Logistic Regression and XGBoost algorithms, and evaluated their scores by ROC-AUC, F1, accuracy, precision, and recall metrics.

4.1.1 Logistic Regression

Neither the track model and the telemetry model performed well, indicating an accuracy score of almost 100%. Upon inspecting the confusion matrix, it is clear that while the models were able to identify non team radio calls perfectly, they fail at team radio calls immensely.

These results suggest the following:

- **Imbalanced Data:** It is no surprise that the model was able to identify non team radio calls the best. Since around 99.08% of our dataset consisted of this class, our method of stratified sampling was simply not enough. To move forward, new techniques like oversampling or undersampling should be applied. Additionally, data could always be increased, by including other seasons of Formula 1.
- **Non-Linearity:** Since logistic regression assumes a linear relationship between features and target, our problem may not be fit for this scenario.

4.1.2 Extreme Gradient Boosting

Since the XGBoost algorithm is known to handle larger amounts of data more effectively, we chose to apply it to the same problem. However, the fact that XGBoost, along with logistic regression, performed poorly suggests that the problem itself may be inherently challenging, or that our preprocessing steps need to be revised.

While team radio classification wasn't our primary focus for this project, but rather something we were curious about - whether there was a distinct method for detecting radio calls - we didn't delve into it extensively. However, by incorporating more data from various seasons and refining our neural networks, this project could be further developed.

Driver Number	Driver
1	Max Verstappen
2	Logan Sargeant
3	Daniel Ricciardo
4	Lando Norris
5	Sebastian Vettel
6	Nicholas Latifi
10	Pierre Gasly
11	Sergio Pérez
14	Fernando Alonso
16	Charles Leclerc
18	Lance Stroll
20	Kevin Magnussen
21	Nyck de Vries
22	Yuki Tsunoda
23	Alexander Albon
24	Zhou Guanyu
27	Nico Hülkenberg
31	Esteban Ocon
44	Lewis Hamilton
47	Mick Schumacher
55	Carlos Sainz Jr.
63	George Russell
77	Valtteri Bottas
81	Oscar Piastri

Table 4.1: F1 Driver Numbers and Drivers

4.2 Comparison of Formula 1 Drivers

It is time to analyze and delve deeper into the main aspect of our project: comparing Formula 1 drivers within a single season and extending our analysis across the 2022 and 2023 seasons. We conducted comparisons between drivers across 16 Grand Prix races within each season. This decision was based on our previous data pre-processing steps, which yielded 16 well-structured and robust datasets for both seasons.

4.2.1 Within-Season Comparison

When examining a single season, we will present dendrograms generated by hierarchical clustering using the average linkage method. Drivers will be clustered based on their performance data (telemetry, team radio, date) from races at the same track.

Bahrain Grand Prix

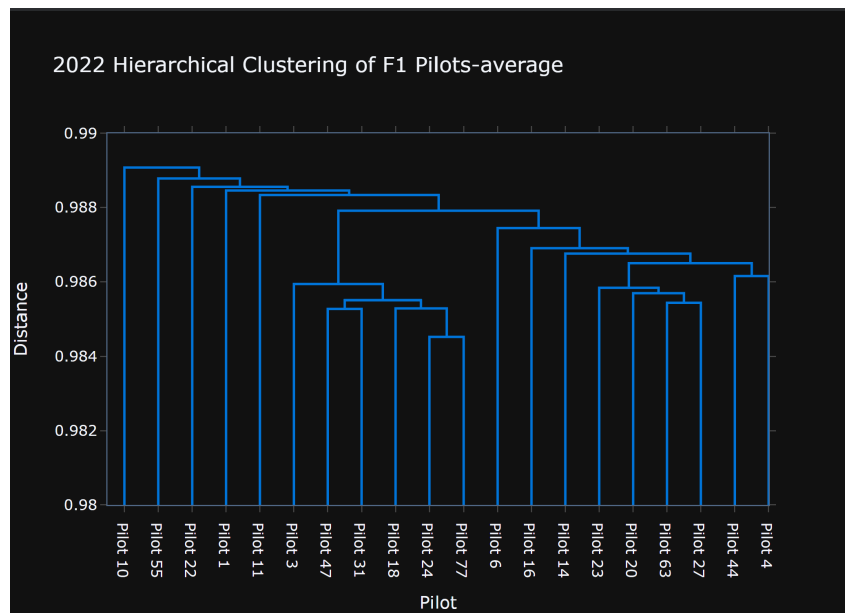


Figure 4.1: Pilot Comparison Dendrogram using average linkage method, 2022 Bahrain Grand Prix

Summary of Findings

Most Similar

Zhou Guanyu (24) and Valtteri Bottas (77) provided the most similarity. Their high similarity in the telemetry data suggests similarities in their driving styles and possibly similar car setups. This is a promising result as both pilots drive for the same team, Kick Sauber. In the 2022 Bahrain race, Bottas finished as 6th while Zhou finished as 10th, both gaining points by finishing at top 10.

Most Dissimilar

The most dissimilar pilots are Pierre Gasly (10) and Lewis Hamilton (55). The significant dissimilarity in the telemetry data between Gasly and Hamilton indicates differences in their driving styles and how they handle their respective cars. Since they drive for different teams, these differences may also reflect variations in car performance and setup. Pierre Gasly even DNF'ing (Did not finish) and not able to finish.

Additional Findings

Despite both Charles Leclerc (16) and Carlos Sainz (55) racing for Scuderia Ferrari and achieving 1st and 2nd place respectively, their different clustering suggests variations in how they utilize the car's capabilities. Leclerc's clustering with drivers who ended up in the top 5 implies exceptional utilization of the car's performance compared to Sainz.

The surprising similarity in clustering between the race winner (16) and loser (27) suggests unexpected similarities in how they utilized their cars' capabilities despite vastly different race outcomes. This may indicate that even though one driver won and the other lost, their driving styles or car setups were similar during the race.

The fact that drivers who did not finish the race are in far distances and different clusters in the telemetry data is reasonable. DNFs often result from issues related to the car's performance or accidents, leading to significantly different race performances observed from the telemetry data. Considering that two Red Bull drivers (1 and 11) are show no similarity despite driving for the same team, it suggests that their race performances, as observed from the telemetry data, were notably different. Possible interpretations could be driving style variations or car setup differences.

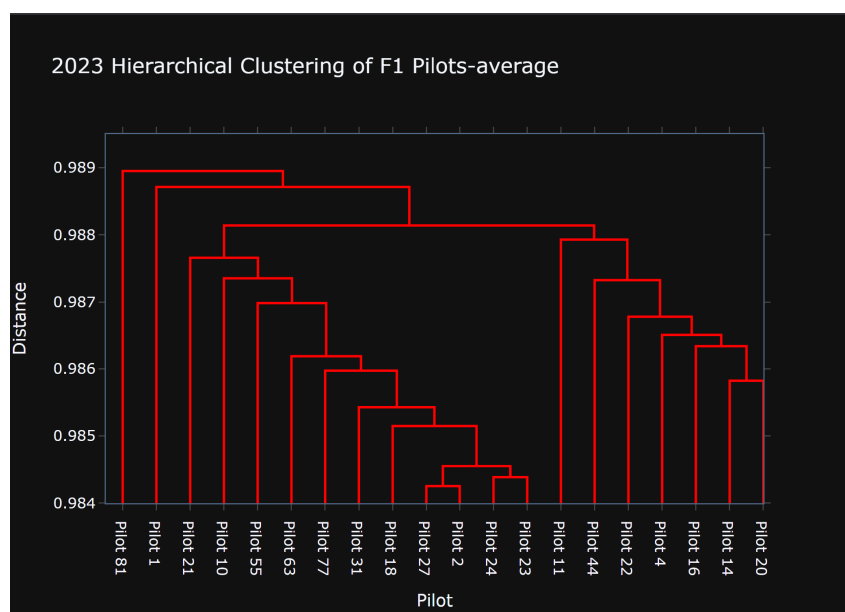


Figure 4.2: Pilot Comparison Dendrogram using average linkage method, 2023 Bahrain Grand Prix

Monaco Grand Prix

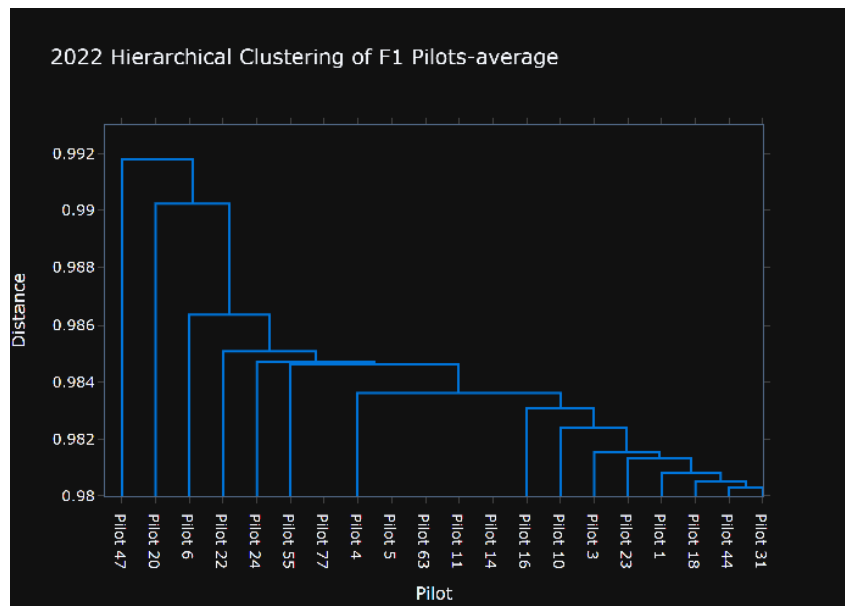


Figure 4.3: Pilot Comparison Dendrogram using average linkage method, 2022 Monaco Grand Prix

Most Similar

Lewis Hamilton (44) and Esteban Ocon (31) provided the most similarity, despite being in different teams and overall having different performances. But they respectively finished as 8th and 12th. Another similarity occurs with Lance Stroll (18). This similarity suggests that despite their different overall race outcomes, Hamilton and Ocon might share similar driving styles or strategies during the race.

Most Dissimilar

The most dissimilar pilots are Kevin Magnussen (20) and Mick Schumacher (47). Despite both being part of the Haas Ferrari team, their telemetry data indicates significant dissimilarity, which is not surprising given their shared DNF result. The dissimilarity in telemetry data underscores the impact of individual driving styles and circumstances during the race, even within the same team.

Additional Findings

In the last race both Red Bull drivers were in different clusters even though they performed just the same (DNF), in the Monaco race, both drivers are in the podium, gaining immense success, but they are still in different clusters. This arises the question of what contributes more to a driver's win? The car they are driving or their own performance. While the car's performance is undoubtedly crucial in Formula 1, a driver's skill, consistency, adaptability, and racecraft are equally important factors that contribute to their success.

The different clustering of Scuderia Ferrari drivers despite both finishing in the top 5 highlights the complexity of individual driver performance versus team success. Sainz's clustering with drivers who mostly did not earn points suggests variations in driving style or race strategy.

The proximity of the race winner (11) and loser (22) in telemetry clustering, despite different outcomes, may indicate similarities in driving performance or shared characteristics between Red Bull Racing and Scuderia AlphaTauri cars.

The significant distance between drivers who did not finish the race (23, 47, 20), despite two being on the same team, underscores the impact of individual circumstances such as accidents or technical failures on race outcomes, regardless of team affiliation.

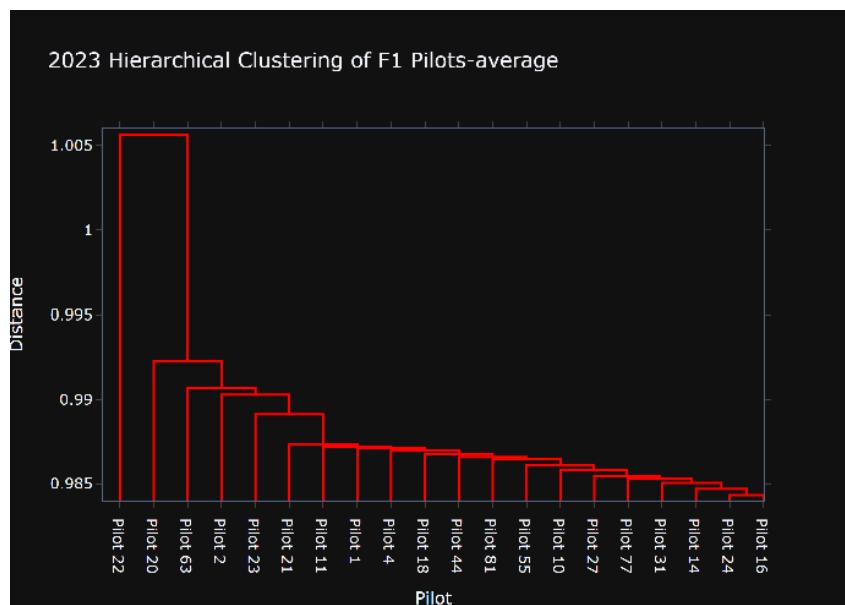


Figure 4.4: Pilot Comparison Dendrogram using average linkage method, 2023 Monaco Grand Prix

4.2.2 Across-Season Comparison

In this section, we extend our analysis across the 2022 and 2023 seasons. We will again present dendrograms comparing drivers across these two seasons, utilizing the same clustering method as in the within-season comparison. Additionally, we will introduce a tanglegram to visually compare how drivers have performed across the two seasons. It's worth noting that some drivers may differ between the two seasons.



Figure 4.5: Pilot Comparison Tanglegram, 2022 & 2023 Monaco Grand Prix

The general structure of clustering remains consistent, therefore the overall number and size of clusters, as well as the criteria for grouping drivers, have not changed significantly between the two seasons. This implies that the overall distribution of performance levels among drivers is relatively stable. Although the overall clustering structure is consistent, some of the drivers moved to different clusters based on changes in their performance.

Teammates Mick Schumacher (47) and Kevin Magnussen (20) remained at similar cluster levels, indicating stable performance. Given that Kevin Magnussen (20) did not finish both races, his performance stability is expected.

Driver Sergio Pérez (11) has shown a notable difference in 2023, dropping from 1st place in 2022 to 16th in 2023. This significant drop is accurately reflected in the tanglegram with a major shift in cluster placement.

Another cluster placement shift in Lewis Hamilton (44) shifts from 8th place in 2022 to 4th place in 2023. This improvement is correctly noted, showing a positive change in his cluster placement.

Charles Leclerc (16) and Fernando Alonso (14) are both in similar clusters in 2023 due to Alonso's improved ranking. This observation correctly captures how changes in individual performance can lead to different clustering in the following season.

Esteban Ocon (31) is now clustered with higher ranked drivers, indicating better performance.

Conclusion

As we conclude this project, we have carefully compared 20 Formula 1 racing drivers, examining their performance within individual seasons as well as across multiple seasons. Beginning this project we wanted to find the answer questions in Formula 1 races, in a fast-paced and not only physically but mentally challenging sport environment such as Formula 1, what exactly is they key to success?

Undoubtedly, Formula 1 is a team sport where drivers, alongside engineers, strategists, and mechanics, collaborate dynamically. From the car's inception to optimizing pit stops and the driver's performance on the track, it is a teamwork. Yet, when drivers exhibit varying performances in identical cars or when telemetry data reveals similarity in performance between point earners and non-scorers, or when races end prematurely due to DNFs, where should our focus lie? This sport, known for its exorbitant costs and lightning-fast pace, demands constant evolution—from data-driven car upgrades to mid-season driver replacements. Hence, it's necessary to address these questions: Are there areas for improvement? Are there untapped talents among drivers? What are the root causes of challenges? Delving deeper into these inquiries not only uncovers potential solutions but also opens the way for advancements in the sport.

Through our analytical process, we have uncovered intriguing diversities within top Formula 1 teams. Take Red Bull Racing, for example, whose drivers consistently vie for championships. Despite allegedly having identical cars, their racing performances based on telemetry often diverge, suggesting that driving skill plays a pivotal role beyond mere technical data.

Similar conclusions arise with Scuderia Ferrari, another impactful team. Even when both Carlos Sainz and Charles Leclerc achieve impressive rankings, their telemetry data present distinct patterns, hinting at individualized racing styles. This underscores the challenge for race engineers to tailor strategies that harmonize the unique strengths of each driver, whether the team follows a hierarchical 'first-driver' approach like Red Bull or opts for equality, as seen with Ferrari.

Moreover, we have observed that drivers in lower-budget teams can occasionally match the performance of race winners, raising questions about the relative impact of driver skill versus mechanical of the car. Indeed, many race retirements stem from mechanical issues, suggesting that the car's reliability often dictates outcomes alongside the driver's abilities.

In conclusion, while our analysis has primarily focused on telemetry and team radio data, it is important to acknowledge the multitude of factors that contribute to the performance of drivers and teams in Formula 1. Pit stop timings, race results, and historical achievements all play significant roles in shaping the narrative of success in this sport. However, despite the

limitations of the data we have accessed and processed, we have gained valuable insights into the key points of winning races at the highest level of competition.

References

- [1] Catapult. F1 data analysis: Transforming performance, 2024. Accessed: 2024-06-09.
- [2] Mohammed Z Al-Faiz, Ali A Ibrahim, and Sarmad M Hadi. The effect of z-score standardization (normalization) on binary input due the speed of learning in back-propagation neural network. *Iraqi Journal of Information and Communication Technology*, 1(3):42–48, 2018.
- [3] Haibo He and Edwardo A Garcia. Learning from imbalanced data. *IEEE Transactions on knowledge and data engineering*, 21(9):1263–1284, 2009.
- [4] Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.
- [5] Ian T Jolliffe. *Principal component analysis for special types of data*. Springer, 2002.
- [6] Markus Ringnér. What is principal component analysis? *Nature biotechnology*, 26(3):303–304, 2008.
- [7] Balaji Venkatachalam, Jim Apple, Katherine St. John, and Daniel Gusfield. Untangling tanglegrams: Comparing trees by their drawings. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 7(4):588–597, 2010.
- [8] Red Canary. Using telemetry collection, we will show how you can collect 1,000 data points worth of telemetry per 2-minute lap, March 2023.
- [9] Plotly. Plotly - the interactive graphing library for python, r, matlab, and javascript, 2024.

Appendix

A.1 Detailed Track Visualizations

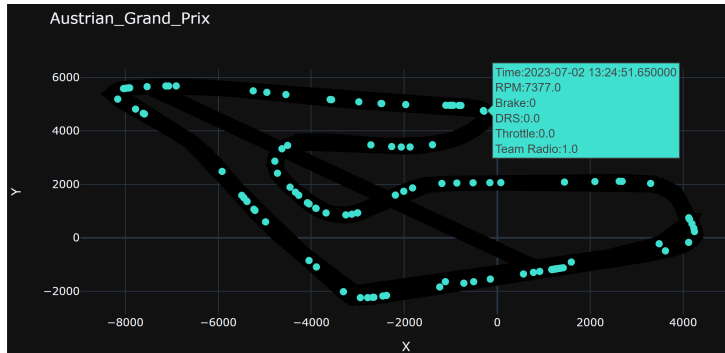


Figure A.1: Austrian Grand Prix

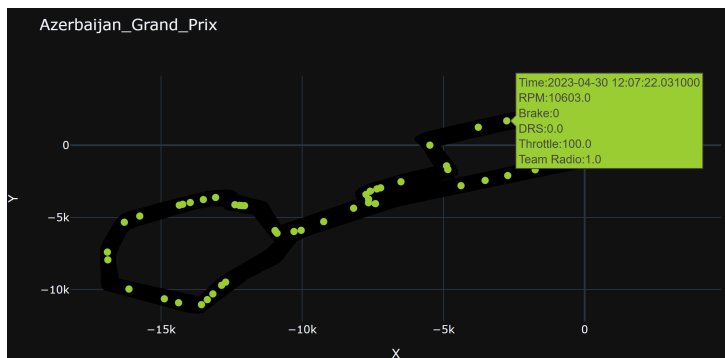


Figure A.2: Azerbaijan Grand Prix

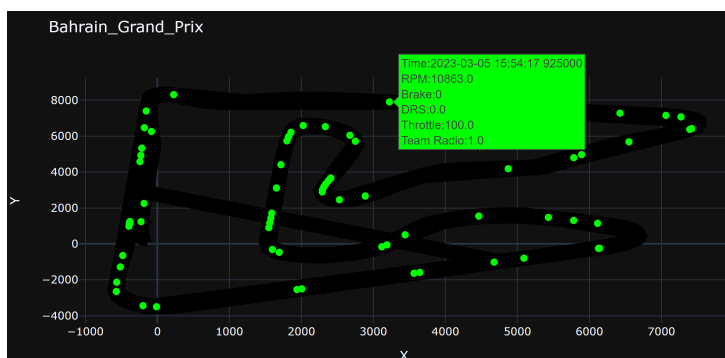


Figure A.3: Bahrain Grand Prix

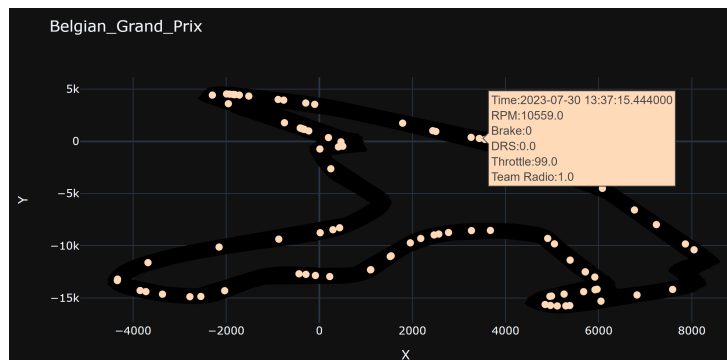


Figure A.4: Belgian Grand Prix

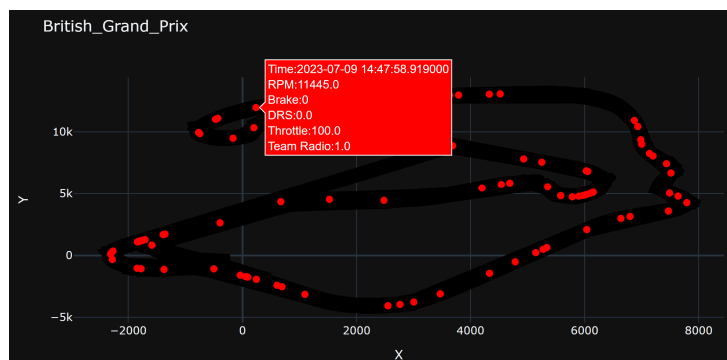


Figure A.5: British Grand Prix

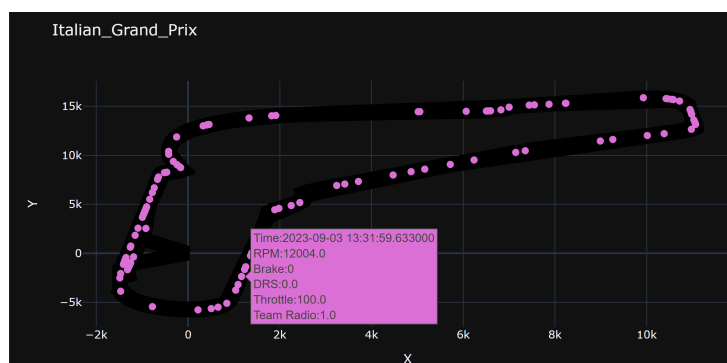


Figure A.6: Italian Grand Prix

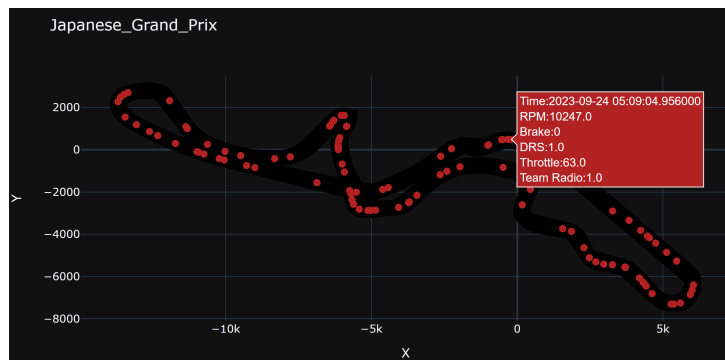


Figure A.7: Japanese Grand Prix

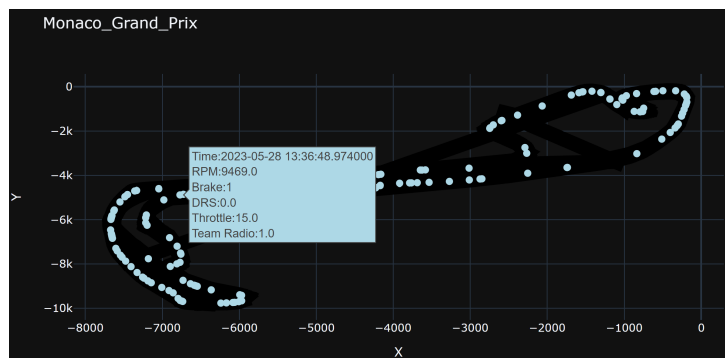


Figure A.8: Monaco Grand Prix

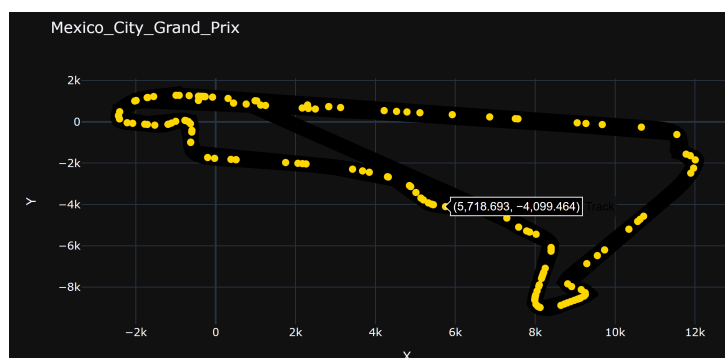


Figure A.9: Mexico City Grand Prix

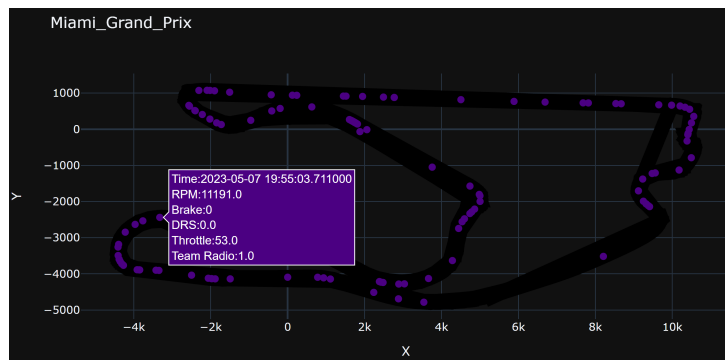


Figure A.10: Miami Grand Prix

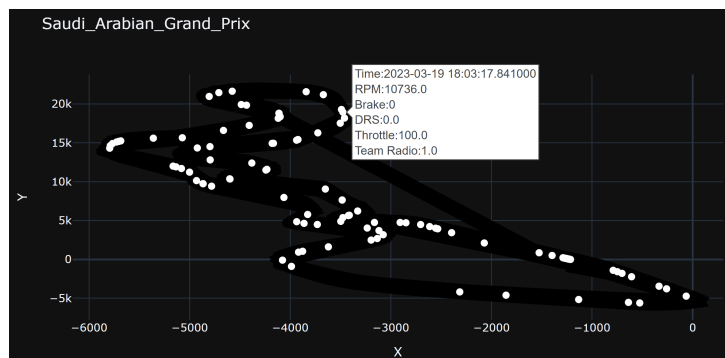


Figure A.11: Saudi Arabian Grand Prix

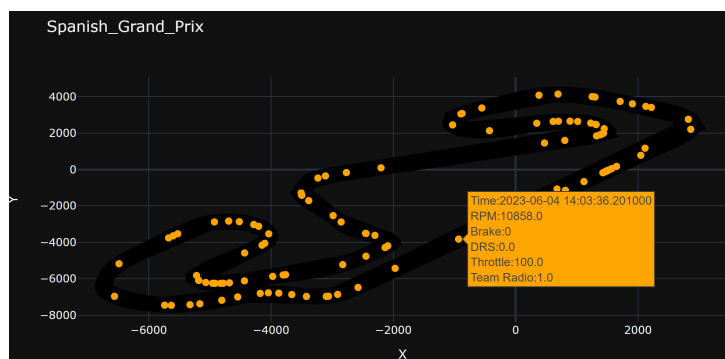


Figure A.12: Spanish Grand Prix