	FPT Education FPT UNIVERSITY	DSP391m Project

Formula 1 Championship Prediction

by

Dung Nguyen Van - SE173009
Hieu Trinh - SE173129
Anh Nguyen Quoc - SE171346

1. Introduction and Background

Formula 1 (F1) is one of the most prestigious and technically complex motorsport competitions in the world. The competition is fierce, with outcomes influenced by a wide range of factors, including driver skill, team strategies, car performance, weather conditions, and the specific characteristics of each racing circuit. Predicting the F1 World Champion is challenging due to the dynamic and unpredictable nature of these variables, but with the increasing availability of data and advancements in machine learning, there is an opportunity to analyze historical trends and use predictive models to forecast future champions.

Over recent years, data science has proven to be a powerful tool in sports analysis, offering insights that were previously impossible to capture. Machine learning, in particular, has been applied in various sports to predict outcomes, improve team strategies, and optimize performance. This project aims to apply machine learning techniques to the world of Formula 1, specifically to predict the World Champion based on historical data, driver and team performance, and other external factors.

The significance of such a prediction model lies not only in its potential accuracy but also in its ability to provide deeper insights into the factors that contribute to winning an F1 championship. From fan engagement to team strategy planning, such a model can have a wide range of applications in the sport, contributing to a more data-driven understanding of racing outcomes.

By using historical race data, driver performance metrics, and external factors such as weather and track conditions, this project will develop a machine learning model capable of predicting the F1 World Champion accurately. The project will serve as a case study for how advanced analytics can be used in high-stakes, complex environments like Formula 1 racing.

2. Pipeline

Week	Tasks	Assigned members	Status	Notes
Week 1	**Paper Review and Methodology Research:** Review research papers and methodologies related to the project to inform the design and approach.	DungNV, HieuT, AnhNQ	Complete	Start: Sep 7th End: Sep 14th
	Introduction Writing: write the introduction and background for the	DungNV, HieuT,	Completed	Start: Sep 15th End: Sep 18th

	project proposal.	AnhNQ		
	Dataset Search: Search for and compile relevant datasets for the project.	DungNV, HieuT, AnhNQ	Complete	Start: Sep 7th End: Sep 16th
"The meeting for the first week is on September 16th."				
Week 2	**Data crawling:** Crawl for more data	DungNV	Complete	Start: Sep 17th End: Sep 21th
	Data cleaning: Clean data	DungNV, HieuT, AnhNQ	Complete	Start: Sep 22th End: Sep 24th
	Data preprocessing: Standardize and encode categorical variables	DungNV, HieuT, AnhNQ	Complete	Start: Sep 22th End: Sep 24th
"The meeting for the second week is on September 23rd."				
Week 3	**EDA:** Visualize data patterns	DungNV, HieuT, AnhNQ	Complete	Start: Sep 24th End: Sep 27th
	Analyze feature correlations to identify the most predictive features.	DungNV, HieuT, AnhNQ	Complete	
	Identify potential outliers or unusual data patterns.	DungNV, HieuT, AnhNQ	Complete	
"The meeting for the third week is on September 27th."				
Week 4,5	**Model Selection & Initial Training:** Experiment with decision trees and other ML algorithms	DungNV, HieuT, AnhNQ	Complete	Start: Sep 28th End: Oct 11th
	Set up initial decision tree models with various depths, like max_depth=12, for preliminary insights.	DungNV, HieuT, AnhNQ		
	Evaluate initial model performance and tune hyperparameters.	DungNV, HieuT, AnhNQ		
"The meeting for the fourth week is on October 11th."				
Week	**Model Optimization & Feature**	DungNV,	Complete	Start: Sep 28th

6, 7	**Tuning Training:** Refine features based on initial model results (add/remove features if necessary)	HieuT, AnhNQ		End: Oct 11th
	Implement RandomForestRegressor and test different parameter combinations.			
	Conduct cross-validation to evaluate model consistency.			
"The meeting for the fifth week is on October 3rd."				
Week 8, 9	**Report:** combine results and submission	DungNV, HieuT, AnhNQ	Complete	Start: Oct 12th End: Oct 25th

3. Data Cleaning and Preprocessing

Use JSON to extract individual race information from Ergast API [1]. The API holds all information on races, results, drivers, qualifying, lap times, pit stops, constructors' and drivers' standings, and circuits from Formula 1's inception in 1983 to date.

The data acquired for each table was as follows:

Predictor data contain 6 datasets:

races	constructor_standings	driver_standings	results	qualifying	weather
season	season	season	season	grid_position	season
round	round	round	round	driver_name	round
circuit_id	constructor	driver	circuit_id	car	circuit_id
lat	constructor_points_after_race	driver_points_after_race	driver	season	weather
long	constructor_wins_after_race	driver_wins_after_race	date_of_birth	round	weather_warm
country	constructor_standings_pos_after_race	driver_standings_pos_after_race	nationality		weather_cold
date	constructor_points	driver_points	constructor		weather_dry
url	constructor_wins	driver_wins	grid		weather_wet
	constructor_standings_pos	driver_standings_pos	time		weather_cloudy
			status		
			points		
			podium		
			url		

This is the method we use to collect datasets:

- races: use the Ergast API [1] to collect data on Formula 1 races from 1950 to 2023 and store it in a Pandas DataFrame. For each year, an API request is sent, and the received JSON data is parsed to extract information such as season, round, circuit location, and race date. Missing data is handled by adding None values where needed.

- constructor_standings: Based on races dataset, collect all Formula 1 constructor standings from the Ergast API [1] for each season and race round. Iterates through seasons and rounds, making API requests to retrieve team ranking information, including points, number of wins, and ranking position. The data is then added to a Pandas DataFrame.

- results: Iterate through the unique seasons in the race and season data to create a rounds list, where each element contains a season and its corresponding rounds. Create a dictionary ‘results’ to store information about the season, round, circuit ID, driver, constructor, points, position, etc.

- driver_standings: loops through each season and its associated rounds stored in the rounds list. For each round, a request is made to the Ergast API [1] to fetch the driver standings data in JSON format. The JSON response is processed by iterating through each driver's standings, extracting relevant information using try-except blocks to handle potential missing data. After collecting the data, the driver_standings dictionary is converted into a Pandas DataFrame

- qualifying: An empty DataFrame called qualifying_results is created to store the qualifying times. For each year, a request is sent to the Formula 1 [2] results page to retrieve the HTML content for that specific year. The HTML content is parsed using BeautifulSoup to extract relevant links to individual race results. Then search for all links corresponding to race results for the given year, filtering them based on their URL structure. For each circuit link found, the code replaces the 'race-result.html' part of the URL with 'starting-grid.html' to access the starting grid page. The starting grid data is read into a DataFrame using pd.read_html(), which extracts the HTML table directly into a DataFrame. The year and round (based on the index of the loop) are added as new columns to the DataFrame. Any columns with "Unnamed" in their name are dropped from the DataFrame to remove unnecessary data. The results from each circuit are concatenated into a yearly DataFrame (year_df). Each yearly DataFrame is concatenated into the main qualifying_results DataFrame to compile all qualifying results from all years.

- weather: iterate through the Wikipedia links of each race appended in the races_df and scrape the weather forecast. Use Selenium to click on the Italian page for each link and append the missing weather data because most languages do not provide the common structure. Eventually, create a dictionary to categorise the weather forecasts and map results. After crawling data, each row of column ‘weather’ has a description of the weather type

    + We create a ‘weather_dict’ dictionary that defines categories for different types of weather conditions, mapping keywords to specific weather categories. Here’s what each key-value pair represents:

    + 'weather_warm': Represents warm or sunny weather. Keywords include: 'soleggiato' (Italian for sunny), 'clear', 'warm', 'hot', 'sunny', 'fine', 'mild', and 'sereno' (Italian for clear).
    + 'weather_cold': Represents cold weather. Keywords include: 'cold', 'fresh', 'chilly', and 'cool'.
    + 'weather_dry': Represents dry conditions. Keywords include: 'dry' and 'asciutto' (Italian for dry).
    + 'weather_wet': Represents wet or rainy weather. Keywords include: 'showers', 'wet', 'rain', 'pioggia' (Italian for rain), 'damp', 'thunderstorms', and 'rainy'.
    + 'weather_cloudy': Represents cloudy or overcast conditions. Keywords include: 'overcast', 'nuvoloso' (Italian for cloudy), 'clouds', 'cloudy', 'grey', and 'coperto' (Italian for overcast).
- Then we create a one-hot encoding for weather conditions, where each row in weather_df contains ‘1’ or ‘0’ to indicate the presence or absence of specific weather conditions for each race.

Final dataset (dataset used for train model):

- We merge 6 datasets (races.csv, results.csv, qualifying.csv, driver_standings.csv, constructor_standings, weather.csv) into final_df dataframe.
- Create drive_age column by using dateutil library on data-of-birth column.
- Drop all the unnecessary features
- Change all category numerics into boolean types.
- Get dummies to the columns ['circuit_id', 'nationality', 'constructor']. This converts each of these categorical columns into multiple binary (0 or 1) columns, where each new column represents the presence (1) or absence (0) of a specific category.
- After creating the dummy variables, check the sum of each column (i.e., the number of occurrences of a specific category) and drop columns that have too few occurrences (reasons: noise reduction, improve model efficiency, save computational cost).

- Final dataset shape is: (16279, 103)

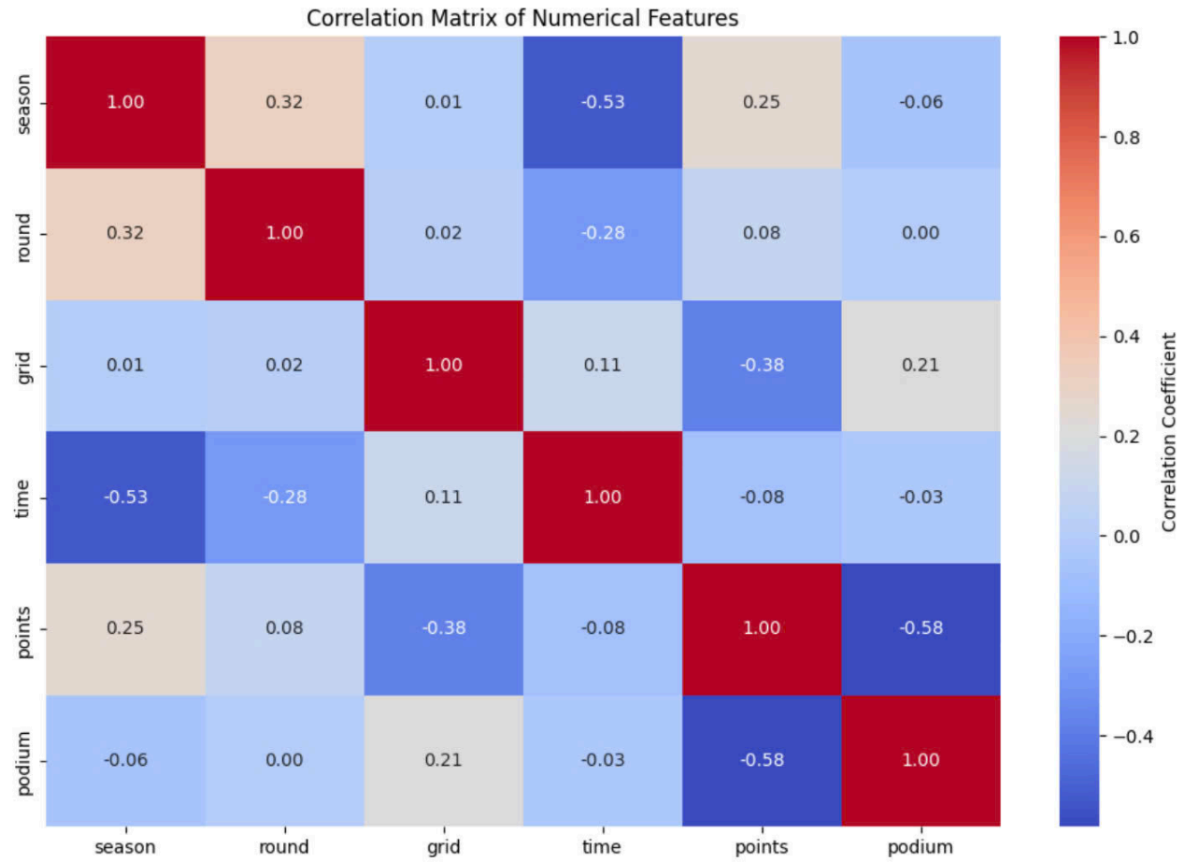
Additional information:

- Qualifying dataset:

- Because Ergast API [1] contains only qualifying races from 1983 onwards, therefore we will drop data from 1982 to before in the Result dataset.
- Because the Qualifying dataset we crawl at Formula 1[2] lacks information from 2019 to 2023. We use fastf1 [3], a special Python library to access and analyse F1 data. Thanks to fastf1 [3], we could obtain the data for each year from 2019 to 2023.

- After obtaining the data from 2019 to 2023, we take the minimum of all Q1, Q2, and Q3. Matching driver name and renaming all the columns to match the original Qualifying dataset, arrange time in ascending order to add grid position.
- Lastly, we standardize car names and combine them into the qualifying dataset.

4. Exploratory Data Analysis

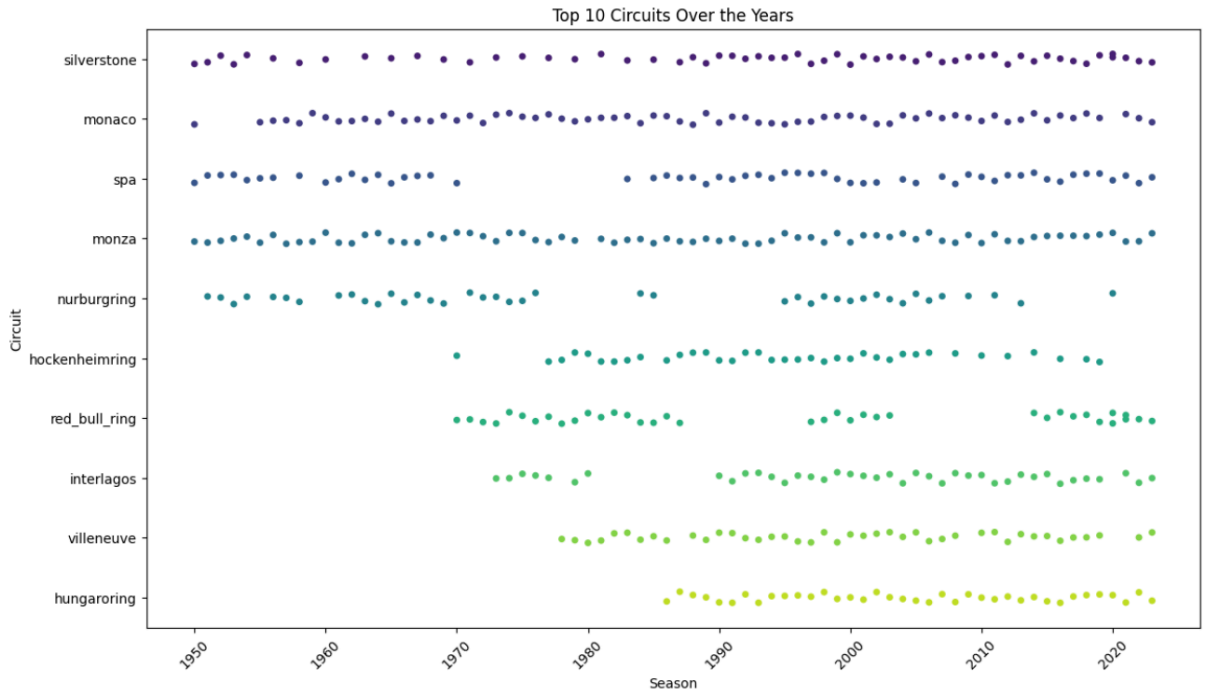


According to the heatmap, the correlation between points and podium position has a moderate negative relationship (-0.58). In Formula 1 context, this indicates the better the podium position, the better points they gain.

Surprisingly, the correlation between gird position and podium position shows a weak relationship (0.21). This means that there is a slight tendency for drivers who have better grid positions. However, many factors, such as driver skill, car performance, strategy, race conditions, etc., significantly influence the final results.

On the other hand, starting at a lower grid position (1st, 2nd, 3rd, etc.) is moderately associated with earning higher points in the race, as shown in the relationship between points and grid position (-0.38).

From 1970 to 2022, it seems that the driver's age didn’t affect much the performance. However, the average diver’s age in the recent year tends to be younger. This could suggest that training, technology, and access to racing opportunities have improved, allowing younger drivers to compete at the top lever more effectively.

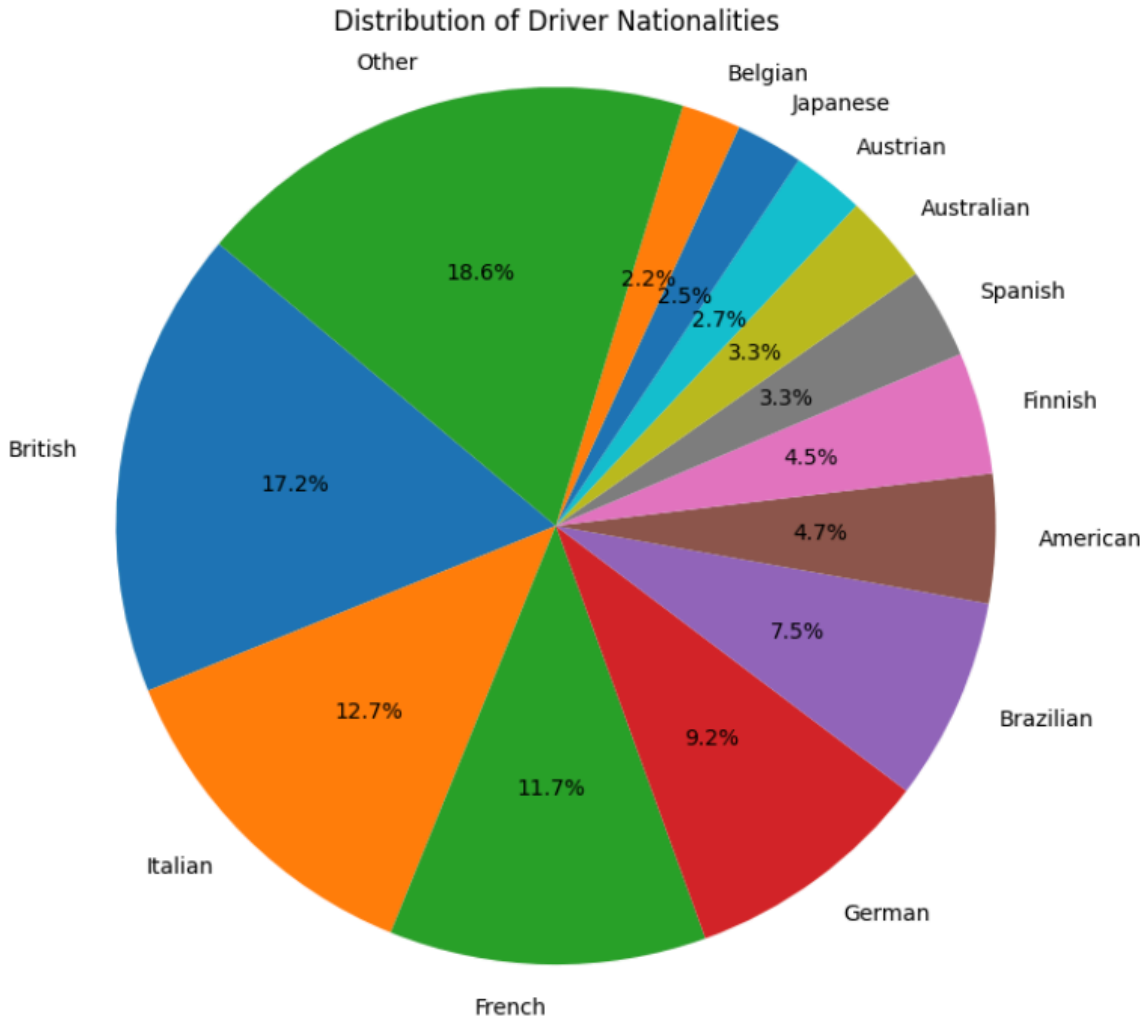


The graph displays the frequency of races at the top 10 Formula 1 tracks from 1950 until now. One season is symbolized by each dot, indicating when a specific circuit held a race. The vertical axis features the circuits, while the horizontal axis displays the timeline of seasons.

Silverstone, Monaco, Spa, and Monza have been part of the racing calendar consistently from the 1950s up to the 2020s. These circuits have established a significant presence in F1 history and are regularly included in the race schedule, demonstrating their significance and appeal.

Nürburgring, Hockenheimring, Red Bull Ring, Interlagos, and Villeneuve (possibly referring to Circuit Gilles Villeneuve in Canada) exhibit periods of discontinuity, with interruptions occurring in certain years. This trend could indicate shifts in F1 scheduling, safety rules, or regional hosting preferences throughout the years.

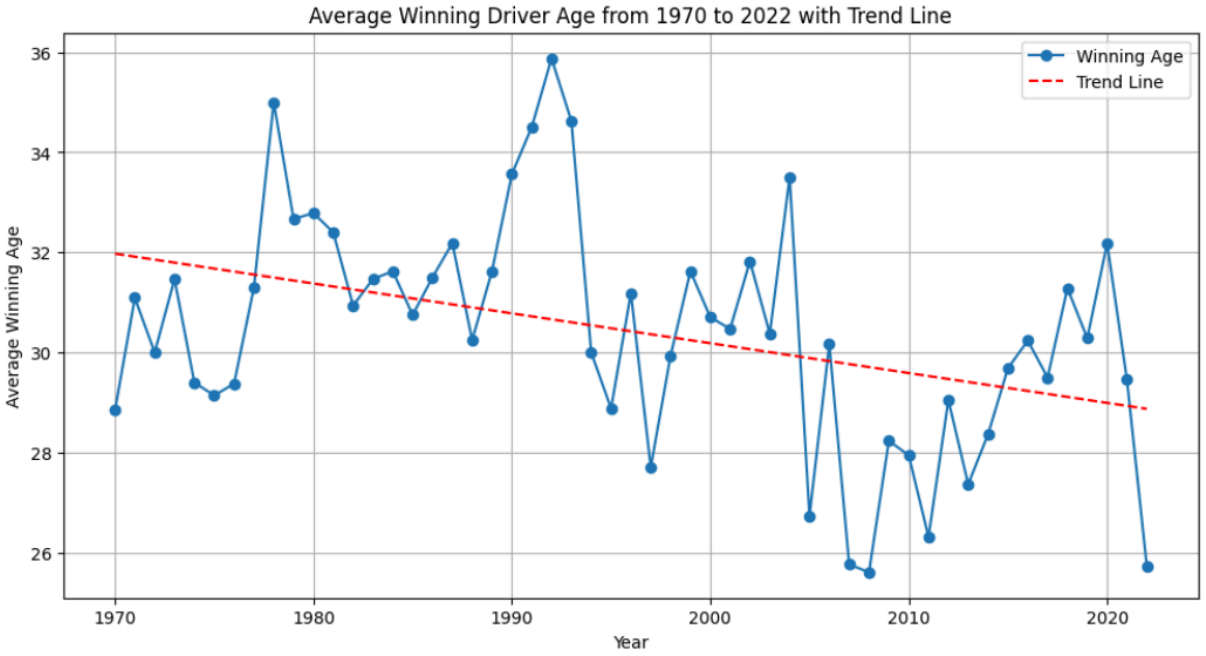
Hungaroring shows up in the later years, starting around the mid-1980s, which coincides with the expansion of F1 to Eastern Europe. This represents the sport’s growth into new markets over time.



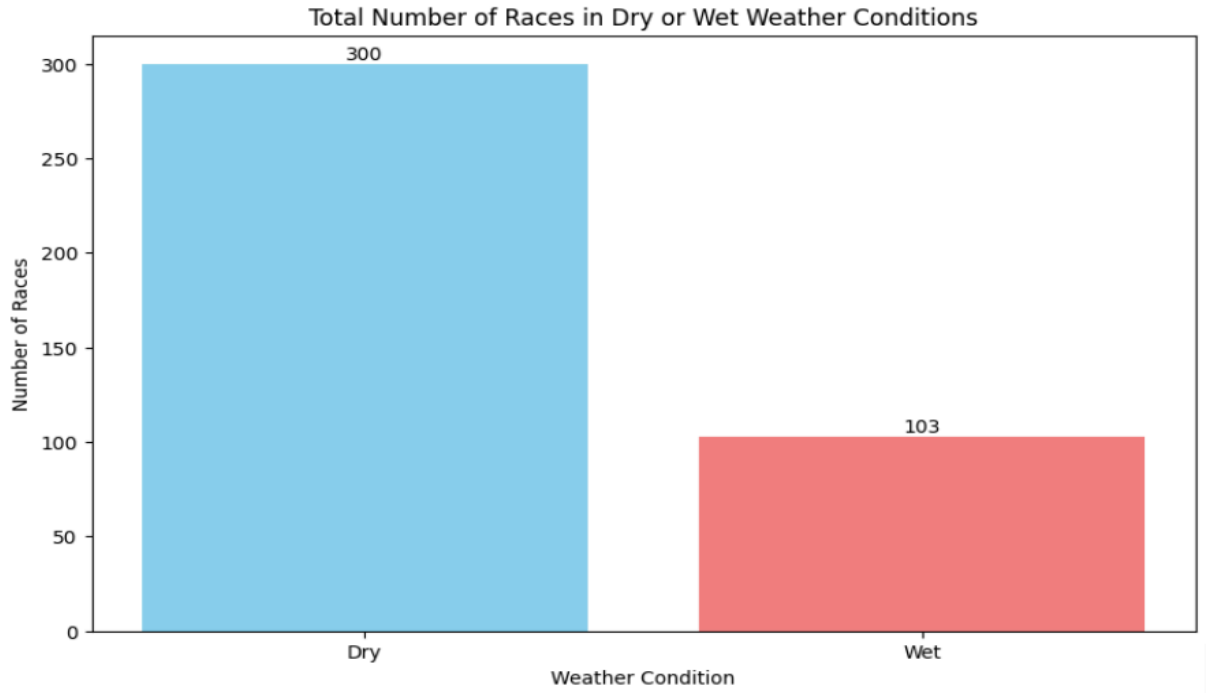
The pie chart illustrates the breakdown of F1 driver nationalities, showing British drivers at 17.2% and drivers from different countries at 18.6% leading the way. Italy, France, and Germany have substantial representations of drivers at 12.7%, 11.7%, and 9.2%, respectively, highlighting the impact of traditional motorsport pioneers from these nations. Brazilians, Americans, Finns, Spaniards, and others from different nationalities each comprise less than 5% of the overall total. The graph highlights the continued dominance of F1 in Europe and the increasing variety of driver nationalities, suggesting a more worldwide perspective for the sport.

The graph showcases how specific nationalities with a strong history in motorsport have a significant impact on F1. Countries like Britain, Italy, and France have a rich background in F1, contributing significantly to the growth and achievement of the sport.

The reason British has a high number of experienced F1 drivers due to its extensive and impressive racing past, including well-known circuits such as Silverstone, Brands Hatch, and Donington Park. The first F1 race was held at Silverstone in 1950, making it a legendary symbol of F1.



The chart shows a decrease in the average age of F1 drivers who win races from 1970 to 2022, indicating that younger drivers are enjoying more success in recent years. While there may be variations from year to year, the overall trend reveals a notable decrease in the age of champions, particularly in the last few years. This change is probably a result of shifts in training techniques, advancements in sports science, and team tactics that prioritize developing younger players. The results indicate that F1 teams are more inclined to choose younger drivers to adapt to the changing physical requirements and competitive nature of the sport.



The bar graph compares the number of races held in dry versus wet weather conditions, showing a clear difference: 300 races occurred in dry weather and only 103 in wet. It may be better to host races in dry weather to make conditions safer, reduce accident risk, and offer a more predictable environment for drivers and teams. Rainy weather can cause visibility to decrease, reduce tire traction, and make skidding more likely, all of which make managing races and safety measures more challenging. On the other hand, rainy weather is also an opportunity for some high-skill drivers, because when it rains, all cars are the same; only skill matters.

5. Methodology

5.1. Model Selection

To predict race results in this case, we chose to use Logistic Regression, Support Vector Machine, Random Forest, and Neuron Network.

Logistic Regression: Selected for its ease of use and interpretability, this technique enables us to examine the effects of variables such as starting grid position or past race outcomes on the likelihood of winning or placing third.

Support Vector Machine (SVM): Featured because it can model intricate patterns in high-dimensional feature spaces and can handle non-linear interactions. Given the interaction of track conditions, driver talent, and vehicle performance, this is pertinent.

Random Forest: Chosen for its important scores for feature selection, its capacity to handle missing values, and its resilience in dealing with both numerical and categorical data.

F1-score was used to assess each of the models due to class imbalance, as only a handful of drivers manage to win or stand on the podium at each race. This develops a metric that measures model performance that takes both precision and recall into account.

5.2. Data Splitting Strategy.

Our dataset provides information about all Formula 1 races from season 1983 to season 2023. We train data before 2023, and our target is to predict the winner for each Grand Prix in 2023.

6. Model

6.1. Model Architecture

Logistic Regression: A basic linear model used for binary classification tasks. It calculates the likelihood of an instance being part of a particular class (e.g. if a driver will finish in the top three).

Support Vector Machine (SVM): SVM creates a perfect hyperplane for class division. To handle non-linear relationships, we employed the Radial Basis Function (RBF) kernel to transform features into a higher-dimensional space.

Random Forest: A collection of 100 decision trees. Every tree is taught with a random selection of data and characteristics, with forecasts made through a collective decision from all trees. This aids in decreasing overfitting and enhancing generalization.

Neural Network: A feedforward neural network with:

+ Hidden Layers: 4 layers with 80, 20, 40, and 5 neurons
+ Activation Function: ReLU.
+ Solver: Adam.
+ Alpha: 0.1

6.2. Training Procedure

First, we import the necessary library and pass the dataset into the project. The podium variable is transformed into a binary format, with 1 representing a podium finish and 0 representing anything else. The information is divided according to the time of year. Data from previous years before 2023 is utilised for training purposes, with the 2023 season specifically set aside for testing. Next, input features are standardized using a StandardScaler to ensure consistent scaling for models such as Logistic Regression and SVM, with a mean of 0 and a variance of 1.

The models are assessed using a custom function called score_classification (model) that predicts the 2023 season's data. Preprocessed training data is used to train models such as Logistic Regression, SVM, Random Forest, or Neural Network. After all, the model is assessed on the 2023 season by utilizing the score_classification() function.

7. Model Evaluation

7.1. Evaluation Metrics

F1 metrics were utilized to accurately assess the models' performance, particularly due to the risk of class imbalance. It combines precision and recall in a harmonic mean. It strikes a balance between false positives and false negatives, which makes it a dependable measure for datasets with imbalances.

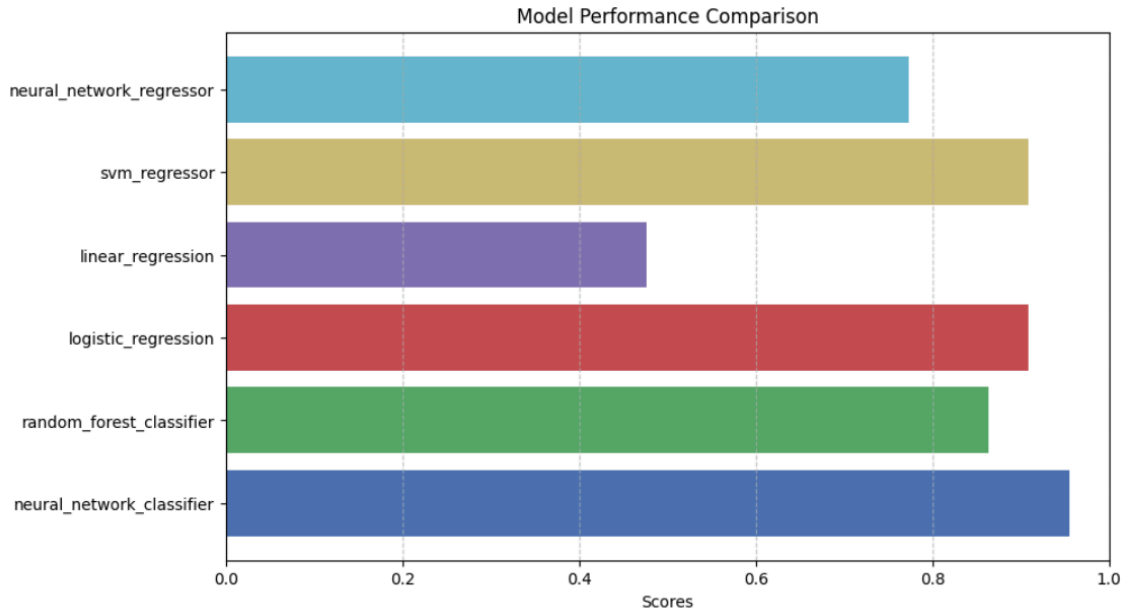
7.2.Hyperparameter Tuning

During model development, several models were trained using different hyperparameters and configurations. Each model was adjusted with varying parameters and diverse kernels. The goal was to find the top-performing setup by looking at the highest F1 score for every model.

8. Results Interpretation and Visualization

Each model was evaluated on the 2023 season data, with the following F1 scores recorded:

neural_network_classifier	0.954545
random_forest_classifier	0.863636
logistic_regression	0.909091
linear_regression	0.47619
svm_regressor	0.909091
neural_network_regressor	0.772727



Neural Network Classifier as the Top Performer

- The Neural Network Classifier has the highest F1-score (0.9545), demonstrating its strong ability to accurately predict podium finishes with a favourable balance of precision and recall.

SVM and Logistic Regression Models Perform Reliably

- SVM Regressor and Logistic Regression showed strong performance, indicating their ability to accurately represent linear connections in the dataset.

Linear Regression Underperforms

- The subpar results of Linear Regression (F1-score: 0.4762) indicate that linear models are not appropriate for this task, possibly because of the non-linear characteristics of race outcomes.

Feature Insights from Random Forest

- Grid position and weather played crucial roles, highlighting the significance of qualifying performance and weather tactics in races.

Implications for Teams and Drivers

- Teams can leverage these models to:
    + Give priority to qualifying sessions to ensure improved starting positions.
    + Alter tactics according to weather predictions to increase chances of success on the podium.

12. Conclusion

Multiple machine learning models were used in this project to forecast race results for the upcoming 2023 season, including Neural Network Classifier, Random Forest, Logistic Regression, SVM, and Linear Regression models. The main goal was to predict podium placements with a balance between precision and recall to minimize false predictions.

The Neural Network Classifier demonstrated its capacity to capture intricate, non-linear data relationships by achieving an F1-score of 0.9545, making it the top-performing model. The Random Forest Classifier and Logistic Regression also did a good job, with F1-scores of 0.8636 and 0.9091, respectively. Nevertheless, Linear Regression faced challenges with the assignment, highlighting the constraints of linear models in forecasting categorical results.

The critical factors that influence race outcomes include grid position and weather conditions, highlighting the significance of qualifying sessions and race strategies that are aware of the weather conditions.

Combining models such as Random Forest and Neural Networks has shown greater effectiveness compared to basic algorithms, emphasizing the importance of advanced techniques in forecasting competitive and uncertain events like races.

These results have practical implications for race teams and drivers, allowing them to concentrate on enhancing qualifying results and creating adaptable tactics. In the future, it would be beneficial to examine other models like XGBoost and improve prediction accuracy by including extra contextual information such as pit-stop strategies, tire choices, and mid-race weather changes.

In general, this research shows the importance of machine learning in sports analysis, offering practical insights and enhancing decision-making to maximize race results.

References

[1] Ergast Developer API – A public open source Formula One API

[2] F1 - The Official Home of Formula 1® Racing

[3] FastF1 3.4.3

[4]willgeorge93/Formula1 (github.com)

[5]veronicanigro/Formula_1