University of Brighton

CI603 Data mining coursework

Formula 1 predictions

Stine Letrud - 22835002

25.05.2025

Executive Summary

This report presents a data mining and machine learning project aimed at predicting Formula 1 race winners using historical data. By merging and analysing multiple structured datasets, the project explores factors that influence race outcomes. Exploratory Data Analysis (EDA) highlighted the impact of pole position, home advantage team reliability, and grid position on winning likelihood.

The modelling task was tackled as a binary classification problem, where each driver-race entry is either labelled a win or a not-win. Due to the inherent class imbalance, weighted training was used, and the XGBoost algorithm was used. Feature engineering and preprocessing steps were taken and were essential for improving the predictive performance. (Guyon and De, 2003).

Although predicting a single race winner is difficult due to the dynamic nature of Formula 1. The model demonstrated reasonable performance. The findings offer insight into the sport and shows how predictive modelling can be applied in real-world scenarios.

Table of Contents

1. Introduction

Predicting modelling in motorsport has emerged as an exiting application of data science combining rich datasets with complex, real-world dynamics. In this project, we focus on the task of predicting Formula 1 race winners using historical data. Formula 1 is not only one of the most technologically advanced sports in the world, but also a domain where performance outcomes are influenced by a range of interrelated factors, including driver skill, team performance, qualifying position, and race-day conditions (Atharva Urdhwareshe, 2025). The availability of extensive historical data makes F1 an ideal candidate for exploratory data analysis and machine learning applications.

The primary objective of this project is to develop a supervised learning model that can predict the winning driver of a race based on information available before/at the start of the race. By transforming this challenge into a binary classification problem, identifying whether a driver wins or does not win a race, we aim to uncover patterns and features that are most predictive of victory.

The project consisted of a few phases. The first phase was Exploratory Data Analysis (EDA) for understanding of trends and correlations in the data. Then, data preprocessing and feature transformation. Third, handling of class imbalance using weighted training. And lastly, model training and evaluation using the appropriate metrics like precision, recall and F1-score.

This report documents the full process of data mining and machine learning tasks performed, while explaining the theoretical background behind decisions and actions, and discussing the performance and limitations of the resulting model.

1 Dataset Overview

The dataset used for this project was sourced from Kaggle's Formula 1 World Championship Dataset (1950-2024). It consists of 14 interrelated CSV files that comprehensively document Formula 1 races over 7 decades. The data includes detailed information on races, circuits, drivers, constructors, qualifying and race results, pit-stops and more. These datasets provide a rich foundation for both exploratory data analysis and predictive modelling.

1.1 Summary of Included CSV files

File name	Description
circuits.csv	Contains names and locations of F1 circuits
constructor_results.csv	Team-level race results
constructor_standings.csv	Constructor points after each race
constructors.csv	Information about the teams
driver_standings.csv	Driver points after each race
drivers.csv	Biographical data on each driver
lap_times.csv	Timing data for each lap by each driver
pit_tops.csv	Pit stop timing per driver per race
qualifying.csv	Records qualifying performance per driver per race
races.csv	Contains information about each race, including date, location and circuit.
results.csv	Core dataset with finishing positions, grid positions, and status for each driver.
seasons.csv	Overview of each f1 season, including year and URLs
sprint_results.csv	Sprint race data
status.csv	Explains special circumstances as reasons for retirement and penalties.

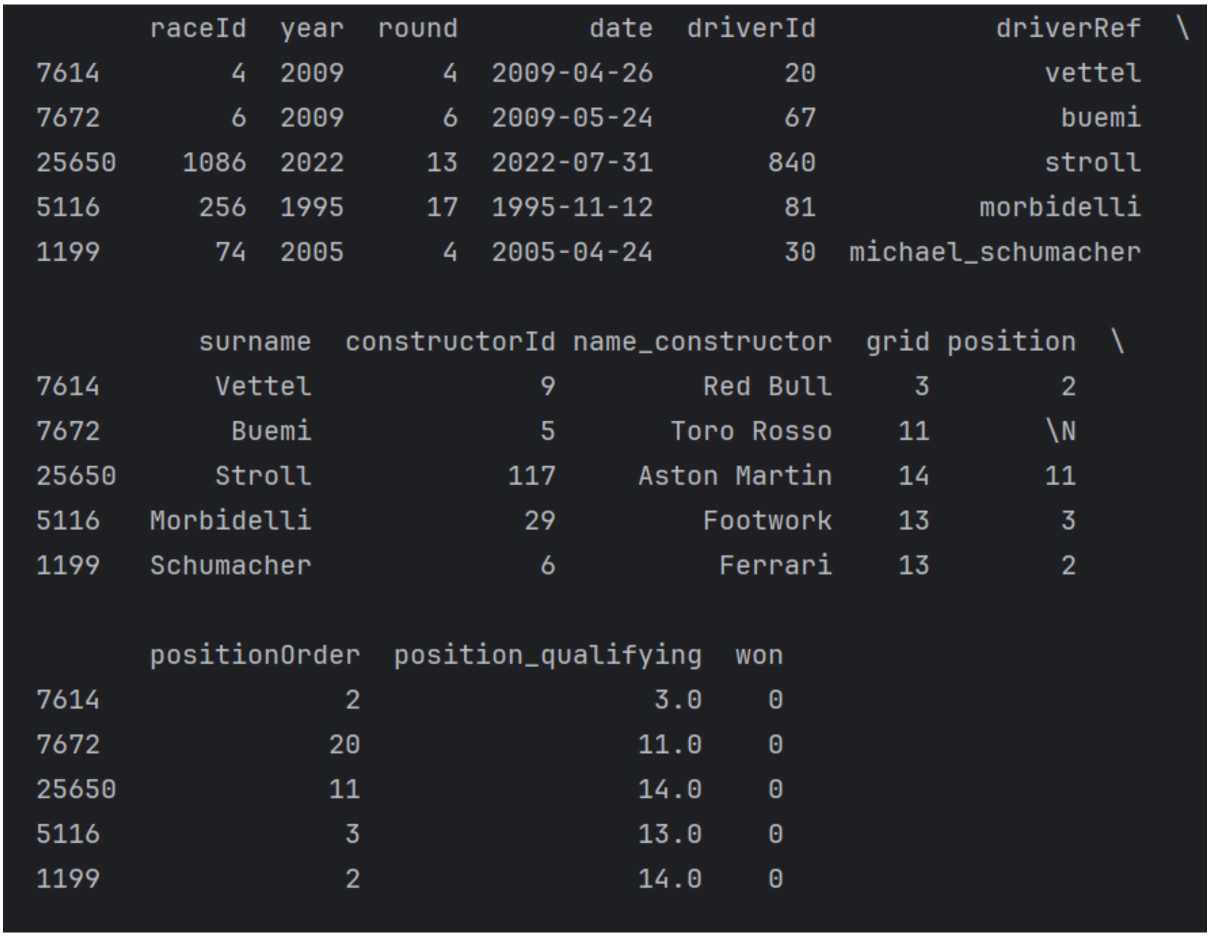
1.2 Data preparation process

Not all datasets were equally relevant for the task of predicting race winners. For modelling purposes, the following files were prioritized and merged.

- Results..csv provided the target variable, grid start and race outcome

- Qualifying.csv offered additional context on pre-race performance.
- Races.csv added temporal metadata such as year and round
- Drivers.csv and Constructors.csv enriched the data with categorical and biographical variables
- Circuits.csv was merged for contextual EDA

Here is a sample of the merged dataset:



```
       raceId  year  round        date  driverId          driverRef  \
7614        4  2009      4  2009-04-26        20             vettel   
7672        6  2009      6  2009-05-24        67              buemi   
25650    1086  2022     13  2022-07-31       840             stroll   
5116      256  1995     17  1995-11-12        81         morbidelli   
1199       74  2005      4  2005-04-24        30  michael_schumacher   

          surname  constructorId name_constructor  grid position  \
7614       Vettel              9         Red Bull     3        2   
7672        Buemi              5       Toro Rosso    11       \N   
25650      Stroll            117     Aston Martin    14       11   
5116   Morbidelli             29         Footwork    13        3   
1199   Schumacher              6          Ferrari    13        2   

       positionOrder  position_qualifying  won  
7614               2                  3.0    0  
7672              20                 11.0    0  
25650             11                 14.0    0  
5116               3                 13.0    0  
1199               2                 14.0    0  
```

The data was filtered to exclude rows with missing key variables such as qualifying position and grid start, and further restricted to modern seasons to reflect current racing dynamics.

By integrating these datasets, constructed a structured and feature-rich DataFrame suitable for binary classification, enabling us to train a machine learning model on race outcomes.

2 Exploratory Data Analysis (EDA)

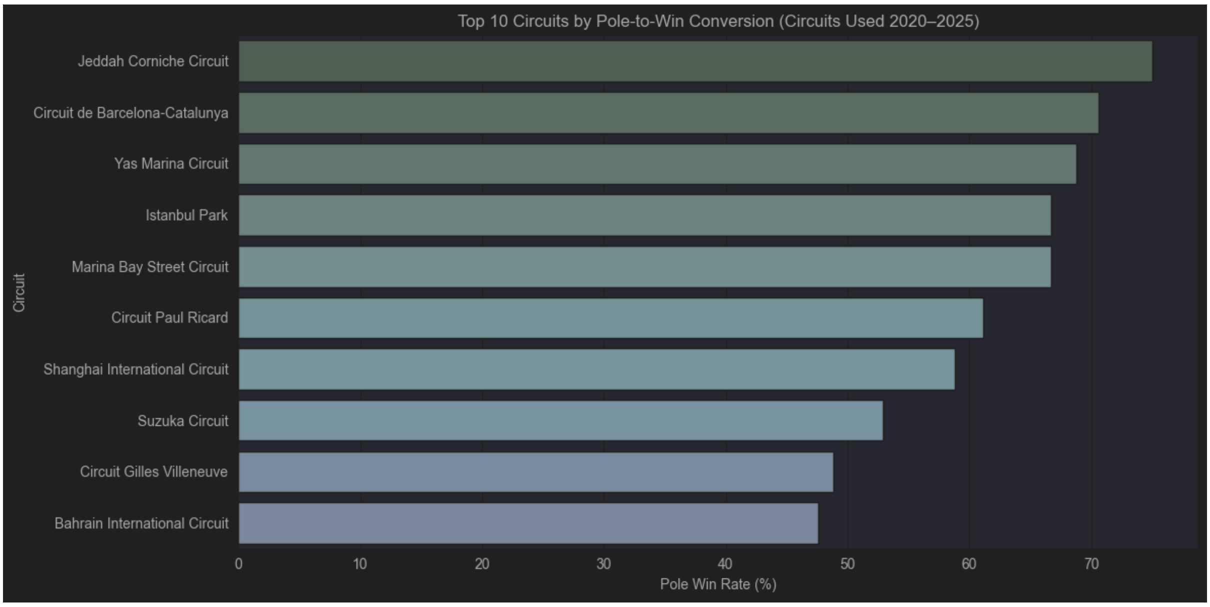
Before developing the predictive model, an in-depth exploratory data analysis (EDA) was conducted to uncover key patterns, relationships, and anomalies in the dataset. The goal was to better understand the dynamics of Formula 1 races, especially the outcomes that influence race outcomes, such as driver performance, team reliability, circuit characteristics, and starting positions. This section highlights the most insightful findings that also helped inform the feature selection for the prediction model.

2.1 Pole Position Importance per Track

One of the most widely debated factors in motorsports is the influence of pole position (starting first on the grid) on race outcomes. In this analysis, it is examined whether starting from pole position significantly increases the likelihood of winning across different Formula 1 circuits.

To test how important pole position is on different tracks the database was filtered to include only races where both qualifying and race results were available. Then the pole win rate was calculated per circuit as the percentage of times the driver starting from first position went on to win the race.

Here is the resulting graph of the top ten circuits by pole position win rate:



The data reveals the importance o starting from pole position in determining the race outcome varies significantly across circuits. Jeddah tops the list with a win rate of 75%. However, the sample size is quite small, with only 4 races. Still the high win rates from pole signifies a significant advantage. Barcelona (70,6%), Yas Marina (68,7%), Istanbul (66,7%) and Marina Bay (66,7%) are well-established circuits with consistently high pole win rates, indicating that track position plays a critical role. This is likely due to the technical layouts of the tracks that make overtaking difficult.

On the lower end with the tracks with the lowest pole win rates we find circuits like Montreal (48,8%), Bahrain (47,6%), Zandvoort (47,1%) and surprisingly Monaco (45,7%) as it is now commonly known as the most important race to get pole, because overtakes are famously very difficult. In earlier decades, overtaking was more common there due to narrower cars, different tire/fuel strategies, and fewer aero limitations. Plus, mechanical failures were more frequent, so leading from pole didn't guarantee a win.

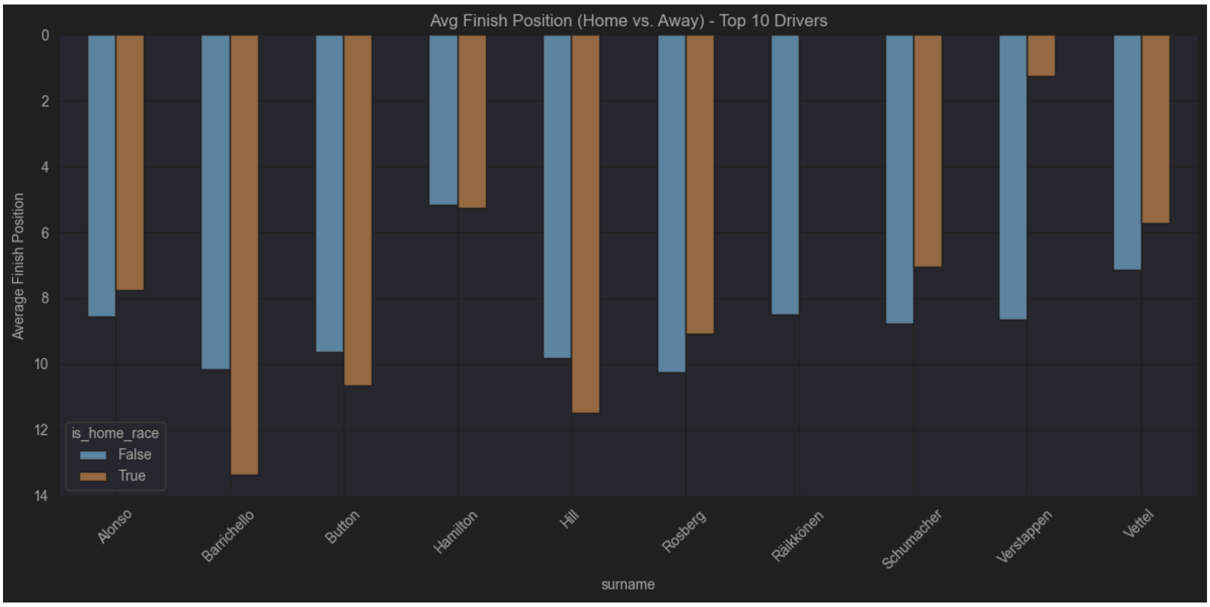
Key takeaways from this EDA are that circuit layout and overtaking difficulty significantly influence race predictability from grid position. Including circuit information and l\pole position as features in a predictive model is justified, as their interaction clearly impacts outcomes. A potential area for further analysis is correlating pole win rate with number of overtaking opportunities, track width, or weather patterns.

2.2 Impact of Racing in Home Country

A interesting question in motorsport analytics is whether drivers perform better when racing in their home country. This idea of a "home advantage" is common in many sports, and in Formula 1, it might translate to better familiarity with the track, increased motivation, or greater crowd support. In this EDA, it is examined whether F1 drivers are more likely to win when competing at a home race.

To analyse this I merged the drivers, races, and circuits datasets to determine each driver's nationality and the country of each circuit. Then filtered for races where the driver's nationality matches the circuit's country. Then the win rates for each driver is calculated for when racing in their home country.

The results are visualised for drivers racing at home versus away, and looked at home race wins across all drivers with at least one victory in their home country.



Insights we can get from this is that home wins are rare. Most drivers do not win more often at home compared to other races. Only a few elite drivers, such as Lewis Hamilton, show consistent success in home races. However, many drivers only race once or a few times in their m\home country, which make statistical conclusions difficult.

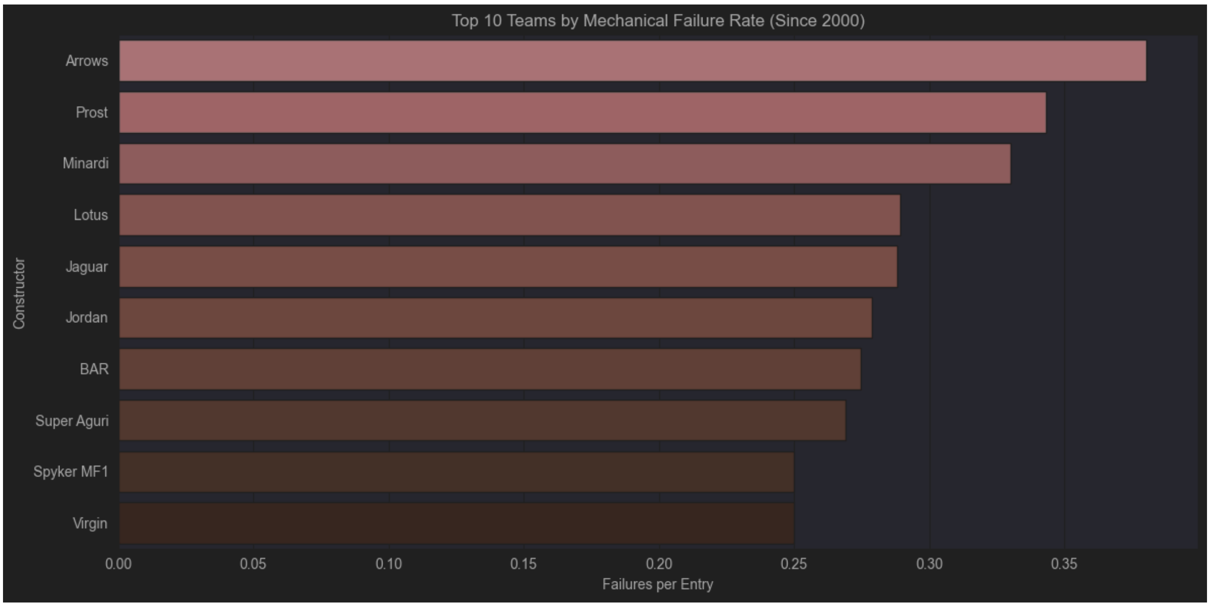
Including “home race” as a binary feature in the predictive model could be experimented with, but based on the analysis, it’s unlikely to add significant predictive power.

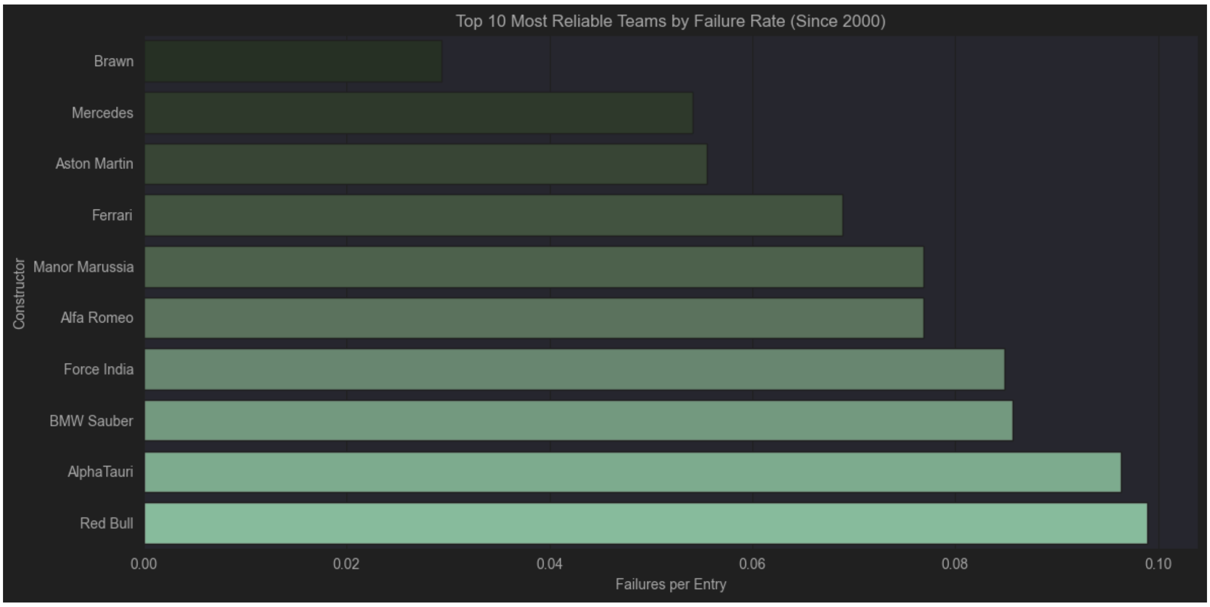
2.3 Car failures by Team

Mechanical reliability plays a crucial role in Formula 1 success. Even the most talented drivers and dominant cars can be undone by technical failures. In this analysis, we explore how often each constructor experiences race-endings mechanical issues, shedding light on the importance of reliability in determining race outcomes.

To examine this the results, status and constructors dataset were merged. The focus of this EDA is on non-finishes caused by mechanical issues, excluding accidents or disqualifications. Identified races where the status code indicated a technical failure, such as engine, gearbox, hydraulic, brakes, suspension, and other related reasons. Then the number of technical failures by teams were aggregated across all seasons.

To visualize the EDA two horizontal bar charts shows the total number of mechanical DNFs (Did Not Finish) per constructor. The chart highlights both teams with historically poor reliability (high failure counts), and more reliable teams that have maintained a consistent finish rate.

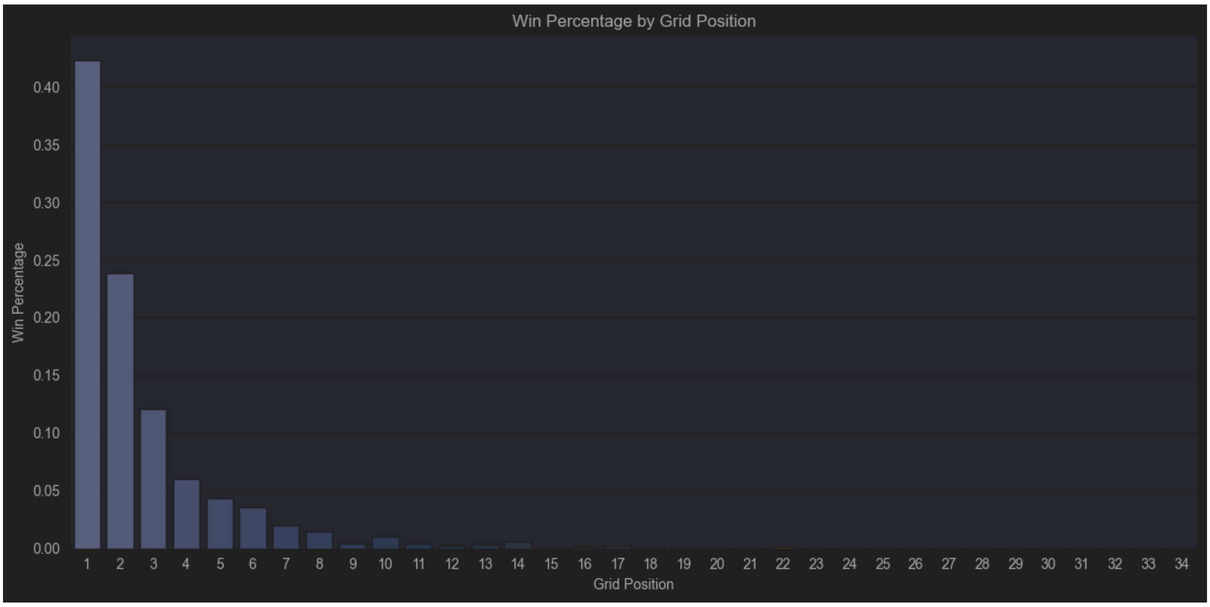




What we can see from this EDA is that early era teams like Minardi, Prost and Arrows, recorded significantly more mechanical failures, likely due to limited budgets and older technology. Newer top teams like Mercedes and Red Bull show much lower failure counts, reflecting the development in F1 where mechanical failures are rarer. Including team reliability metrics (e.g., average mechanical DNFs per season) as a feature could improve the model's ability to predict non-finish outcomes, however for a model predicting only winners, these failures are often already reflected in past performance data.

2.4 Grid Position vs. Win Probability

Starting position plays a crucial role in determining a driver's likelihood of winning a Formula 1 race. To quantify this impact, a bar plot was created showing the win percentage for each grid position across all races in the dataset.



The results reveal a strong advantage for the front runners. Drivers starting from pole position, win approximately 40% of the time. Beyond the top three grid slots, the win probability sharply decreases, and after tenth position the chance of winning becomes negligible.

This pattern confirms the predictive value of grid position and justifies its inclusion in the feature set used for the classification model. Additionally, a binary variable (top_3_grid) was created to capture whether a driver started in the top three, helping the model generalize the relationship.

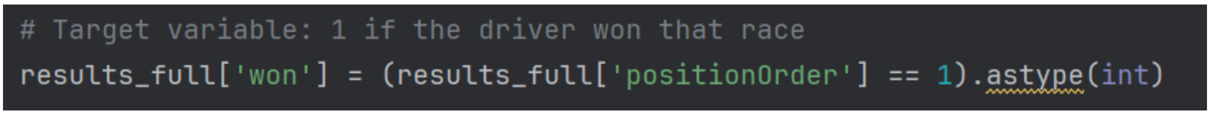
3 Predictive Modelling

Predictive modelling is the mathematical process used to predict future events or outcomes by analysing patterns in a given set of data (Lawton, 2022).

The objective of this predictive modelling was to build a machine learning model to predict the winner of a Formula 1 race based on information available prior to the race, including qualifying performance, grid position, constructor (team) and driver identity.

The task was formulated as a supervised binary classification problem where each race entry (a driver in a particular race) is classified as either Class 1 (the driver won the race) or as Class 0 (the driver did not win the race). This binary framing is suitable as each race only has one winner, and all the other drivers are considered non-winners. (Karabiber, n.d.).

Formally, the target variable won was defined as follows:

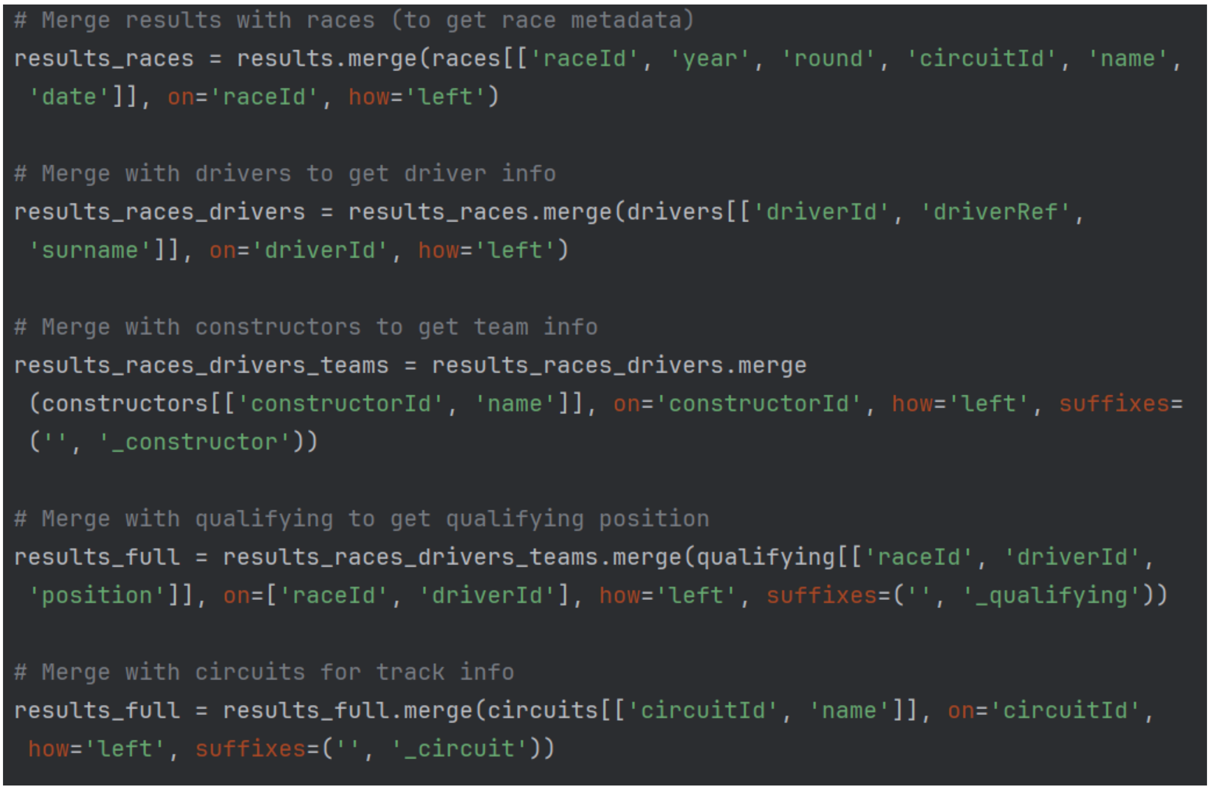


```
# Target variable: 1 if the driver won that race
results_full['won'] = (results_full['positionOrder'] == 1).astype(int)
```

Due to the imbalance of F1 races, where only one out of twenty participants win per race, the classification problem presents challenges in terms of model accuracy and evaluation metrics. In classification problems where one class is significantly underrepresented, standard machine learning algorithms tend to bias toward the majority class, achieving high accuracy by predicting all the examples as negative (Brownlee, 2019). However, this is useless for the goal of the model.

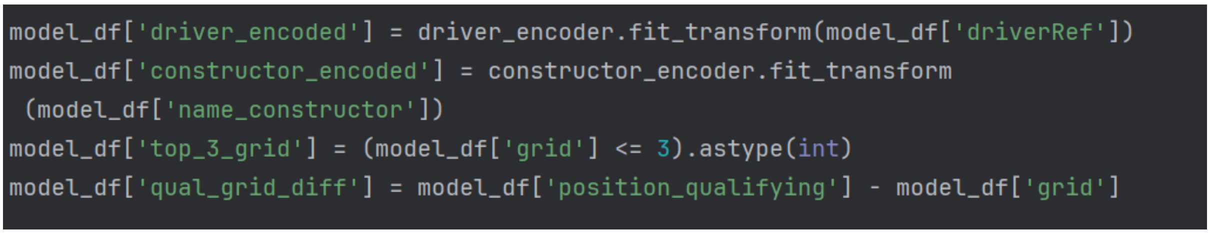
3.1 Data preparation

To prepare the data from the dataset. The relevant CSV files were merged to form a consolidated dataset for modelling. The information gathered was for: final race positions, race metadata, driver identity, constructor identity, qualifying performance, and track information.



```
# Merge results with races (to get race metadata)
results_races = results.merge(races[['raceId', 'year', 'round', 'circuitId', 'name',
 'date']], on='raceId', how='left')

# Merge with drivers to get driver info
results_races_drivers = results_races.merge(drivers[['driverId', 'driverRef',
 'surname']], on='driverId', how='left')

# Merge with constructors to get team info
results_races_drivers_teams = results_races_drivers.merge
 (constructors[['constructorId', 'name']], on='constructorId', how='left', suffixes=
 ('', '_constructor'))

# Merge with qualifying to get qualifying position
results_full = results_races_drivers_teams.merge(qualifying[['raceId', 'driverId',
 'position']], on=['raceId', 'driverId'], how='left', suffixes=('', '_qualifying'))

# Merge with circuits for track info
results_full = results_full.merge(circuits[['circuitId', 'name']], on='circuitId',
 how='left', suffixes=('', '_circuit'))
```

After preparing the base dataframe, some derived features were created to improve the prediction. They are driver_encoded, constructor_encoded, top_3_grid, and qual_grid_diff.



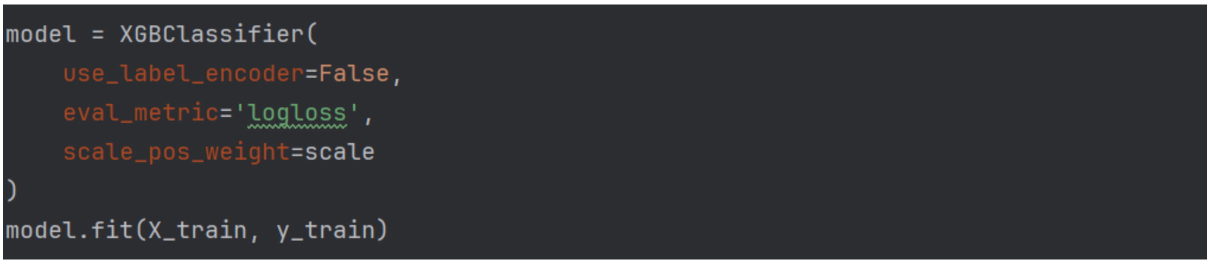
```
model_df['driver_encoded'] = driver_encoder.fit_transform(model_df['driverRef'])
model_df['constructor_encoded'] = constructor_encoder.fit_transform
 (model_df['name_constructor'])
model_df['top_3_grid'] = (model_df['grid'] <= 3).astype(int)
model_df['qual_grid_diff'] = model_df['position_qualifying'] - model_df['grid']
```

3.2 Model Selection & Training

XGBoost (Extreme Gradient Boosting) was selected for its robustness and performance in structured/tabular data (Chen and Guestrin, 2016). XGBoost is a scalable implementation of gradient boosting decision trees, a technique that builds an ensemble of weak learners (shallow trees) in a stage-wise fashion. Gradient boosting combines multiple decision trees by optimizing a loss function iteratively. At each stage the current model's residuals are computed, a new decision tree is fit to these residuals, the trees' predictions are scaled and added to the existing model, and the overall loss is minimized. The mathematical iteration is $F_t(x) = F_{t-1}(x) + \gamma h_t(x)$.

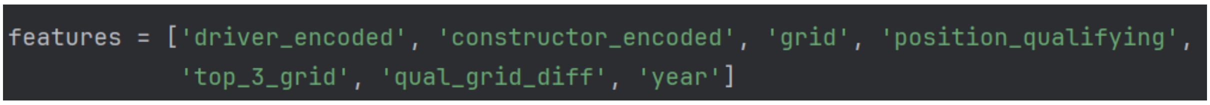
Where $F_t(x)$ is the current ensemble, $h_t(x)$ is the new base learner (the decision tree), and γ is the learning rate. XGBoost improves this with regularization, tree pruning, parallelization and handling missing values. These enhancements make it faster, more accurate, and less prone to overfitting than traditional boosting methods.

The XGBoost model was trained like this:



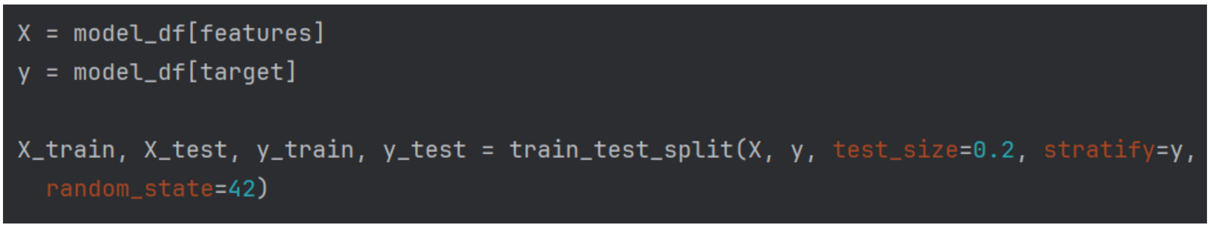
```
model = XGBClassifier(
    use_label_encoder=False,
    eval_metric='logloss',
    scale_pos_weight=scale
)
model.fit(X_train, y_train)
```

And used these following features:



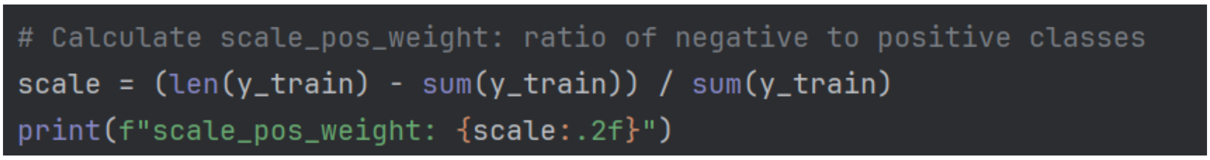
```
features = ['driver_encoded', 'constructor_encoded', 'grid', 'position_qualifying',
            'top_3_grid', 'qual_grid_diff', 'year']
```

And split the dataset:



```
X = model_df[features]
y = model_df[target]

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, stratify=y,
   random_state=42)
```

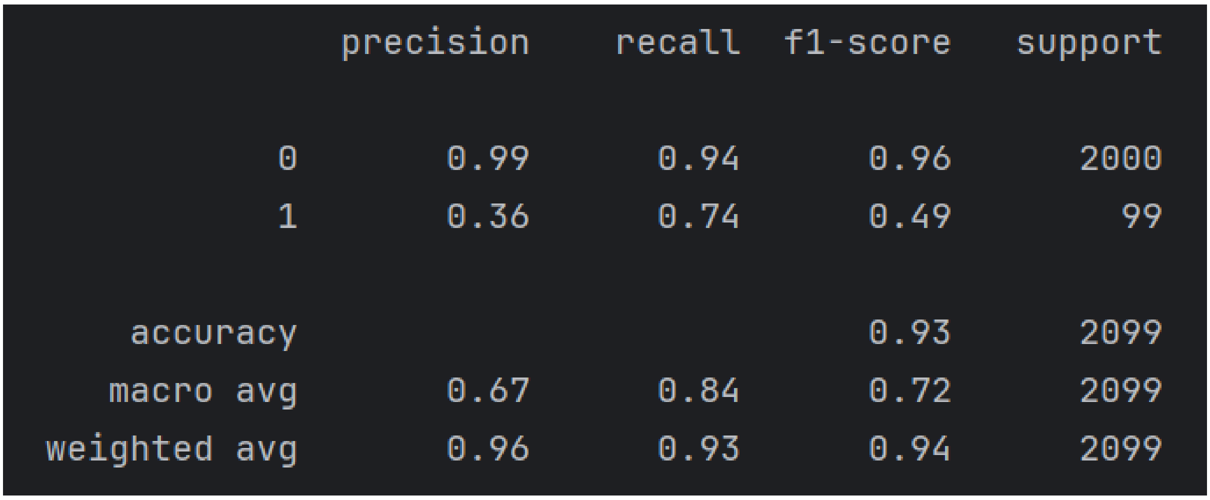
Because the dataset has a strong class imbalance (only 99 out of 2099 examples in the test are winners), scale_pos_weight was added to penalize the false nagatives and make the model more sensitive to predicting winners. This is a example of class weighting, a method used of assigning more importance to the minority class during training. The scale_pos_weight parameter scales the gradient for positive examples during loss computation (xgboost.readthedocs.io, 2022). This caused a significant improved recall at the cost of some precision, which was deemed acceptable in this use case.



```
# Calculate scale_pos_weight: ratio of negative to positive classes
scale = (len(y_train) - sum(y_train)) / sum(y_train)
print(f"scale_pos_weight: {scale:.2f}")
```

3.3 Results

To assess the model, a classification report was used. And these are the final results:



```
              precision    recall  f1-score   support

           0       0.99      0.94      0.96      2000
           1       0.36      0.74      0.49        99

    accuracy                           0.93      2099
   macro avg       0.67      0.84      0.72      2099
weighted avg       0.96      0.93      0.94      2099
```

The accuracy of the model is 93%, with a recall for winners at 74%, and precision for Winners is 36%.

The high recall indicates that the model is good at catching most of the winners, though many non-winners are occasionally predicted as winners (lower precision).

3.4 Reflection and Future work

Predicting winners in F1 is challenging due to the extreme class imbalance. This model achieved high accuracy for the majority class (0 for non winners), but underperformed in predicting the actual winners. This suggests that further feature engineering or an alternative model approach (e.g., ranking or regression) could improve results.

The model shows promising performance given the imbalance and unpredictability of F1 races. However, further improvements could include: Incorporating water, lap time trends, or driver standings, adding more granular time-series data, exploring ensemble techniques or stacking with other classifiers, and conducting feature importance analysis to better understand which variables influences the model.

This predictive model demonstrates how structured race data can be used to make intelligent predictions, and could serve as a prototype for race strategy tools, betting systems, or for commentary.

4 Conclusions

This project demonstrates how historical Formula 1 data can be used to build a predictive model capable of identifying likely race winners using pre-race features such as grid position, qualifying results, and team affiliation. Despite challenges like class imbalance and the complexity of race dynamics, the model achieved reasonable performance by leveraging structured data and the XGBoost algorithm.

Beyond motorsport, the methods used here, data integration, feature engineering, and imbalanced classification, have broad applicability across domains where prediction is based on heterogeneous and temporal data. This includes areas like finance, healthcare, and logistics, where small outcome classes (e.g. defaults, diagnoses, delivery failures) must be predicted from complex structured datasets. As such, the project not only provides insight into F1 racing but also reinforces the power of data science in solving real-world classification problems.

5 References

Atharva Urdhwareshe (2025). The Use of Machine Learning in Predicting Formula 1 Race Outcomes. [online] doi: https://doi.org/10.20944/preprints202504.1471.v1.

Brownlee, J. (2019). A Gentle Introduction to Imbalanced Classification. [online] Machine Learning Mastery. Available at: https://machinelearningmastery.com/what-is-imbalanced-classification/.

Brownlee, J. (2020). SMOTE for Imbalanced Classification with Python. [online] Machine Learning Mastery. Available at: https://machinelearningmastery.com/smote-oversampling-for-imbalanced-classification/.

Bunker, R.P. and Thabtah, F. (2019). A machine learning framework for sport result prediction. Applied Computing and Informatics, [online] 15(1), pp.27–33. doi: https://doi.org/10.1016/j.aci.2017.09.005.

Chen, T. and Guestrin, C. (2016). XGBoost: a Scalable Tree Boosting System. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16, 1(1), pp.785–794. doi: https://doi.org/10.1145/2939672.2939785.

Guyon, I. and De, A. (2003). An Introduction to Variable and Feature Selection André Elisseeff. Journal of Machine Learning Research, [online] 3, pp.1157–1182. Available at: https://www.jmlr.org/papers/volume3/guyon03a/guyon03a.pdf.

Karabiber, F. (n.d.). Binary Classification. [online] www.learndatasci.com. Available at: https://www.learndatasci.com/glossary/binary-classification/.

Lawton, G. (2022). What is Predictive Modeling? [online] SearchEnterpriseAI. Available at: https://www.techtarget.com/searchenterpriseai/definition/predictive-modeling.

www.kaggle.com. (n.d.). Formula 1 World Championship (1950 - 2022). [online] Available at: https://www.kaggle.com/datasets/rohanrao/formula-1-world-championship-1950-2020?resource=download.

xgboost.readthedocs.io. (2022). XGBoost Parameters — xgboost 1.5.2 documentation. [online] Available at: https://xgboost.readthedocs.io/en/stable/parameter.html.