Data Premier League

Formula 1 Driver Performance Prediction

CONTENTS

DATA VISUALIZATION AND PREDICTION CHALLENGE

STATEMENT OBJECTIVE

ABOUT FORMULA 1:

Formula 1 is the highest class of single-seater auto racing sanctioned by the Fédération Internationale de l'Automobile (FIA) and owned by the Formula One Group. The FIA Formula One World Championship has been one of the premier forms of racing around the world since its inaugural season in 1950. The word "formula" in the name refers to the set of rules to which all participants' cars must conform. A Formula One season consists of a series of races, known as Grands Prix, which take place worldwide on purpose-built circuits and on public roads.

The dataset consists of all information on the Formula 1 races, drivers, constructors, qualifying, circuits, lap times, pit stops, championships from 1950 till the latest 2024 season.

"Races are won at the track. Championships are won at the factory." - Mercedes (2019)

With the amount of data being captured, analyzed and used to design, build and drive the Formula 1 cars is astounding. It is a global sport being followed by millions of people worldwide and it is very fascinating to see drivers pushing their limit in these vehicles to become the fastest racers in the world!

STATEMENT MOTIVE:

"Predict the finishing positions of Formula 1 drivers in an upcoming race using historical data and advanced analytics. The sponsors need clarity, and our strategy depends on it.

DEPLOY LINK : Time Series App · Streamlit

TOOLS AND FRAMEWORK

PROGRAMMING LANGUAGE

- PYTHON
- SQL

TOOLS

- GOOGLE COLAB
- VSCODE
- POWERBI (with DAX query)

FRAMEWORK

- STREAMLIT

- WEBBROWER

LIBRARIES

- PANDAS
- NUMPY
- TENSORFLOW
- SCIKIT-LEARN
- Y-DATA-PROFILING
- MATPLOTLIB
- SEABORN
- PLOTY EXPRESS
- PLOTLY GRAPH OBJECTS
- NETWORKX
- PICKLE
- JOBLIB
- KERAS
- IMBLEARN

DATA PREPROCESSING

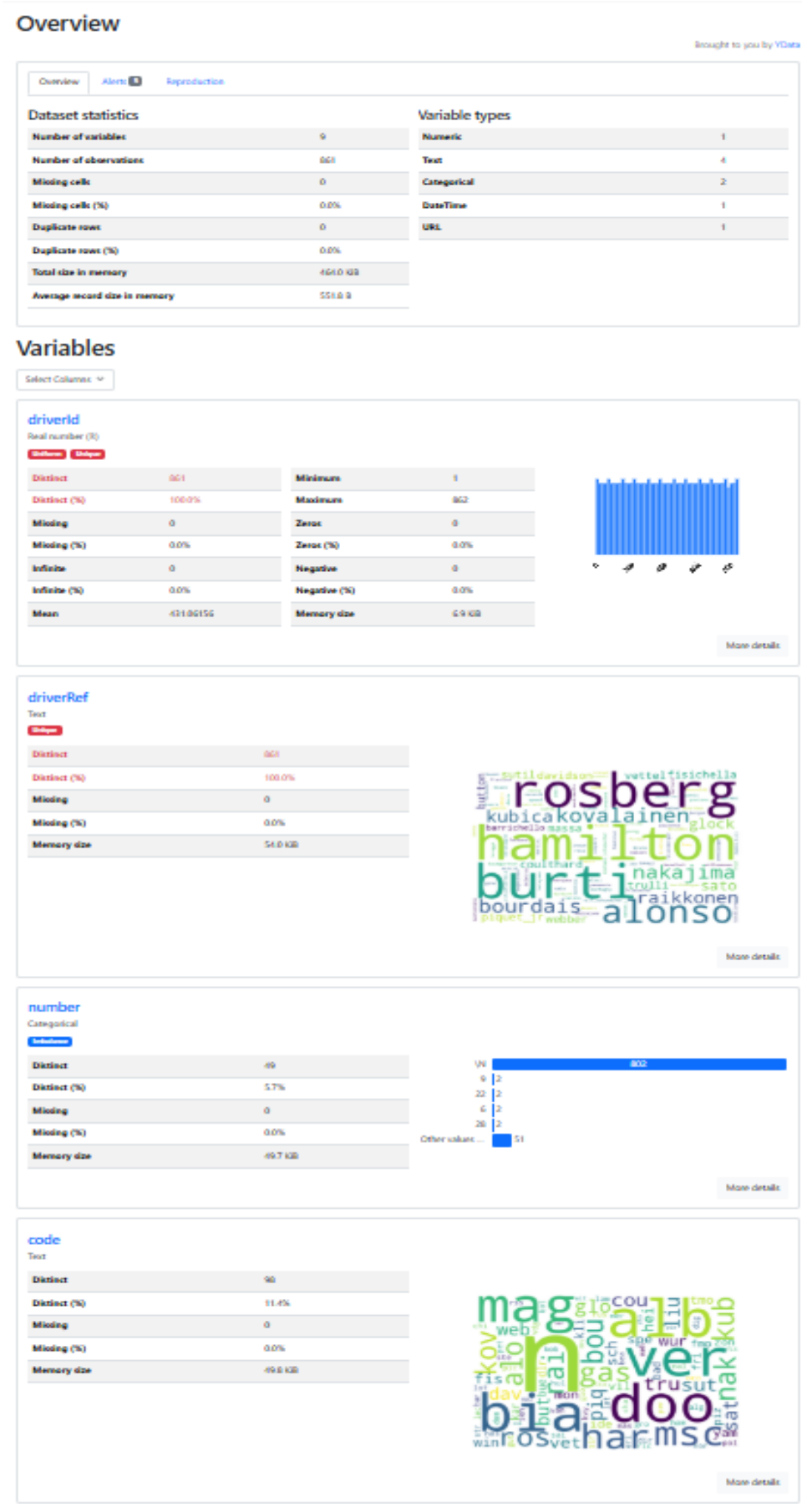
The data files have been preprocessed by handling missing values in both numerical and categorical columns, removing duplicate records, and generating detailed profiling reports using **ydata-profiling**.

Key Features of ydata-profiling:

- **Automated EDA:** Generates a comprehensive summary of the dataset.
- **Missing Values Analysis:** Identifies and visualizes missing data patterns.
- **Duplicate Detection:** Highlights duplicate records for potential removal.
- **Statistics:** Provides insights into numerical, categorical, and text data distributions.

- **Correlation Analysis:** Detects relationships between features using Pearson, Spearman, and other metrics.

SAMPLE



FEATURE LEARNING

- **Driver Consistency** – Evaluates the stability of a driver's performance over a season.
  Average Finishing Position: Mean of the driver's finishing positions.

$$\text{Avg Finish} = \frac{\sum \text{Finishing Positions}}{\text{Total Races}}$$

  Average Qualifying Position: Mean of the driver's qualifying positions.

$$\text{Avg Qualifying} = \frac{\sum \text{Qualifying Positions}}{\text{Total Qualifying Sessions}}$$

- **Team Strength** – Assesses the overall performance and reliability of constructors.
  Total Constructor Points: Sum of all points scored by team drivers.

$$\text{Constructor Points} = \sum \text{Driver Points}$$

  Reliability Rate: Percentage of races finished without mechanical failure.

$$\text{Reliability Rate} = \left(\frac{\text{Races Finished}}{\text{Total Races}} \right) \times 100$$

- **Track Complexity** – Reflects the overtaking difficulty of circuits.
  Overtaking Opportunities: Average overtakes per race on a circuit.

$$\text{Avg Overtakes} = \frac{\sum \text{Overtakes Per Race}}{\text{Total Races on Circuit}}$$

.

- **Win Ratio** – Measures the driver's winning efficiency.

$$\text{Win Ratio} = \frac{\text{Total Wins}}{\text{Total Races}} \times 100$$

- **Podium Percentage** – Percentage of podium finishes relative to total wins.
  Podium: Positions 1 to 3 are considered podium finishes.

$$\text{Podium Percentage} = \frac{\text{Total Podiums}}{\text{Total Wins}} \times 100$$

- **Career Length Years**- Duration of a driver's career.

$$\text{Career Length} = \text{End Year} - \text{Start Year}$$

- Status Finished-the flag that represents whether the driver finishes the race
- Is Winner-the flag that represents is the driver finished the race first in the race

ANSWERING QUERIES

1.DRIVER AND CONSTRUCTOR PERFORMANCE

- Combines race results, driver details, constructor details, and race metadata into a unified dataset.
- Calculates for constructors and drivers
    - Total races, wins, podium finishes, and total points.
    - Win ratio and podium percentage to measure success
- For Career
    - Career start and end years, total races, and career length for each driver.

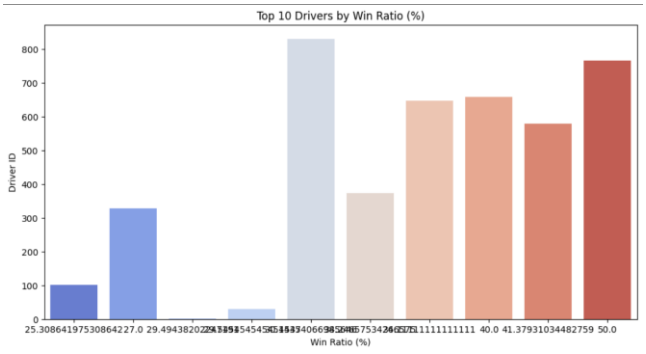
KEY INSIGHTS:

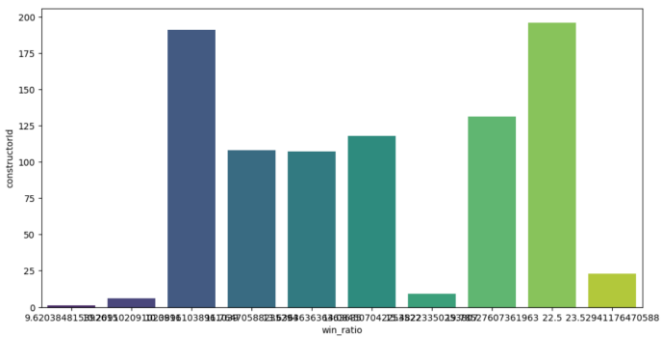
Driver & Constructor Performance

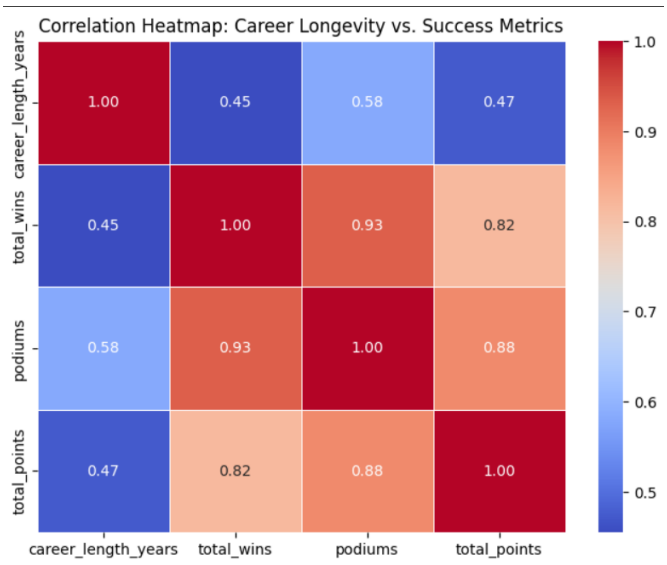
- Identify **dominant drivers and constructors** based on win ratios and podium finishes.
- Compare **success metrics** like **wins, podiums, and points** across different career spans.

Career Longevity vs. Performance

- Analyze whether a **longer career correlates with more wins**.
- Determines how **career duration impacts race performance**.





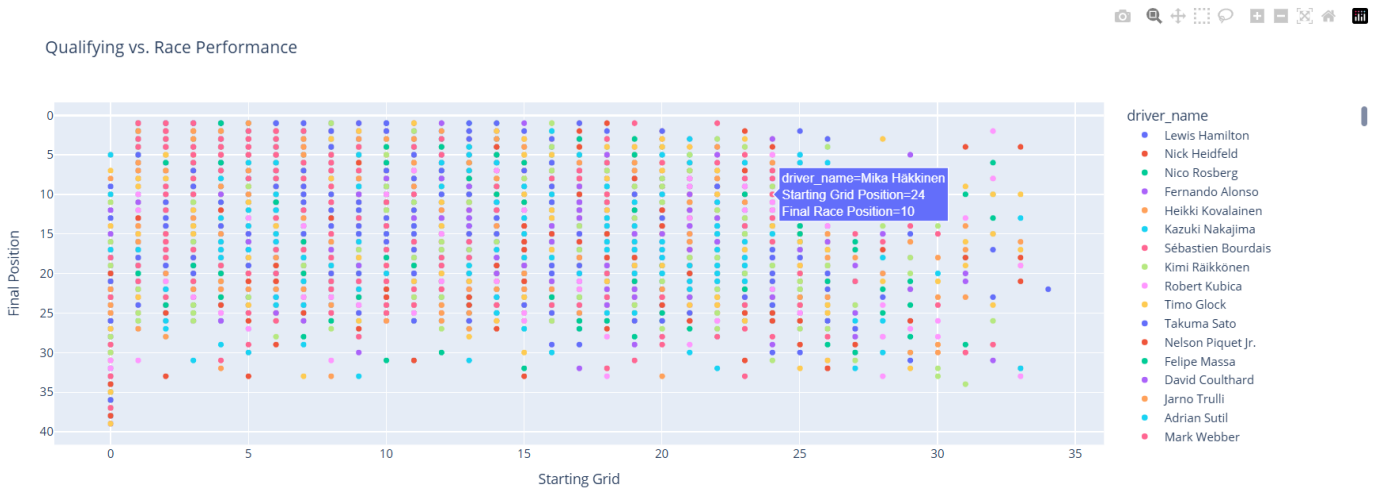


2.QUALIFYING VS RACE PERFORMANCE

- Convert positionOrder column to **numeric** to handle any non-numeric values.
- Calculate position change → **position_change = grid - positionOrder** (positive values indicate a driver gained positions).
- Merge driver names into the **results dataset** for better visualization.
- Scatter plot:
    - X-axis: Starting grid position
    - Y-axis: Final race position

KEY INSIGHTS:

- Do drivers who start in the front row typically maintain their position and finish at the top?
- Which drivers regularly improve or drop positions throughout a race?
- Are there specific teams or drivers that consistently perform better in race conditions compared to their qualifying results?



3.PIT STOP STRATEGIES

HYPERPARAMETER:

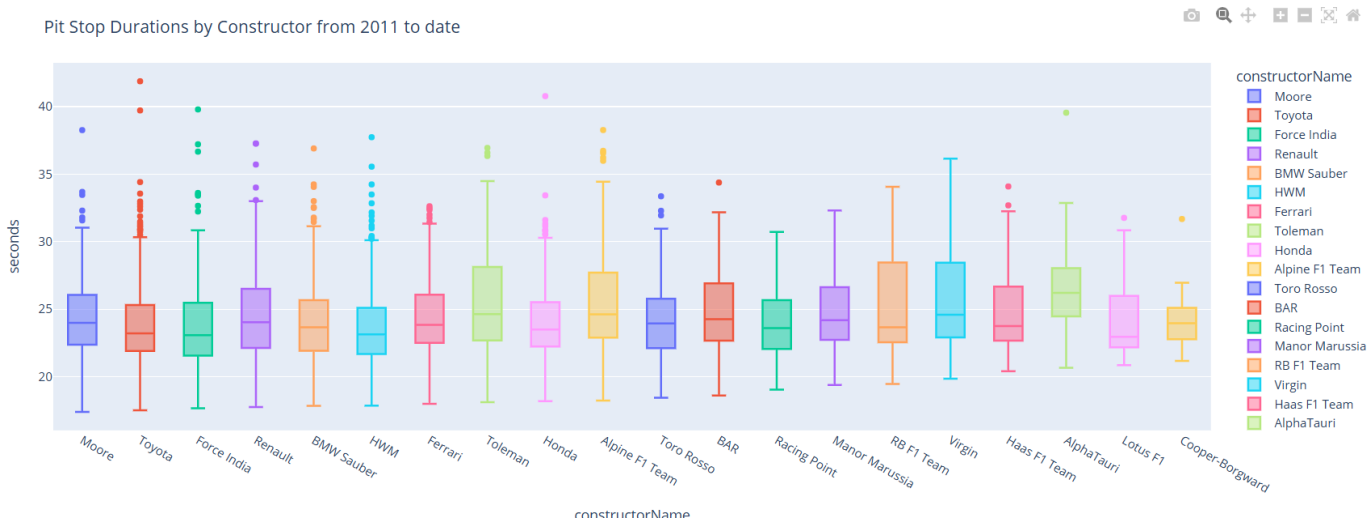
Pit Stop Threshold: Pit stops over **50 seconds** are excluded to avoid outliers

- The analysis evaluates the relationship between pit stop timing, efficiency, and race performance by merging race, driver, constructor, and pit stop data.
- Merge **race results** with circuits, constructors, and drivers to get complete race details.
- Merge **pit stop data** with races and results to link each pit stop to a specific race, driver, and constructor.
- **Remove extreme pit stops** (over **50 seconds**) to exclude outliers.
- **Group data by constructor and year** to compute mean pit stop times.

- **Group by race and constructor** to compare average pit stop duration in specific races.

KEY INSIGHTS:

- Evaluate how **pit stop strategies impact final race positions**.
- Identify **teams that consistently have faster or slower pit stops**.
- Compare **yearly trends** to see if pit stop times are improving over time.

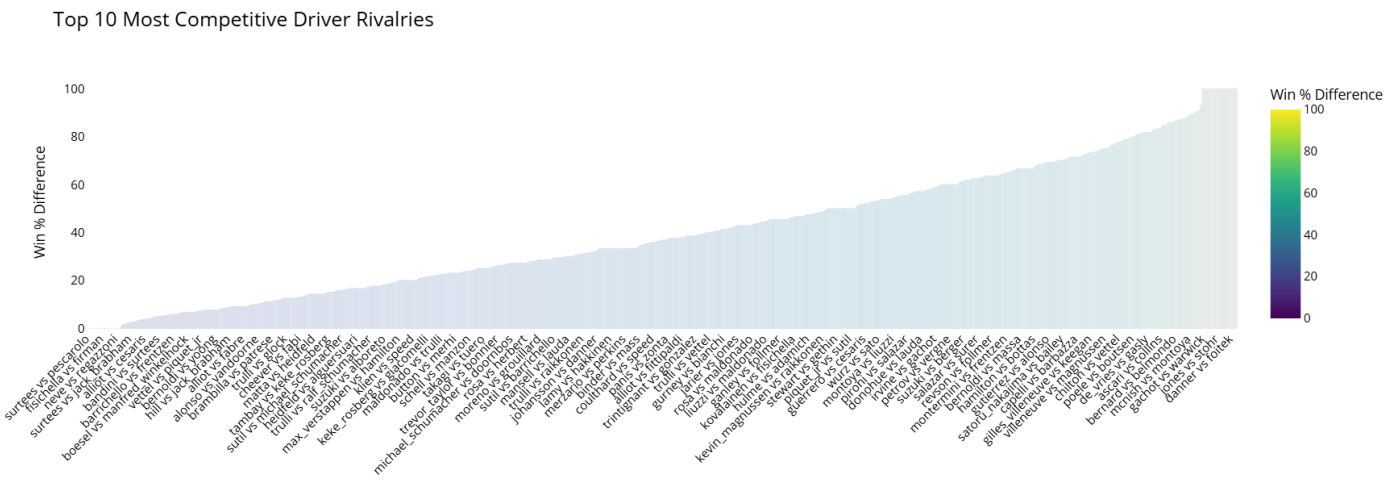


4. HEAD-TO-HEAD DRIVER ANALYSIS

- Combine race results with driver details and race year to create a dataset tracking driver performance per race.
- Perform a self-join on raceId to compare each driver against every other driver in the same race.
- Identify race wins in head-to-head matchups based on finishing positions.
- Group driver pairs to count the total races they competed in and the number of times Driver 1 finished ahead.
- Determine Driver 2 wins by subtracting from the total race count.
- Compute win percentages for both drivers.
- Filter rivalries to include only those with more than 10 races for meaningful analysis.
- Calculate the win percentage difference between the two drivers.
- Rank rivalries by the smallest win percentage difference to highlight the most competitive matchups.

KEY INSIGHTS:

- The closest rivalries feature **drivers with nearly equal win percentages**, indicating tight on-track battles.
- Some rivalries emerge from **teammates in the same car**, while others span **different teams** over several seasons.
- Analyzing **win percentages over time** can show shifts in dominance between two drivers.

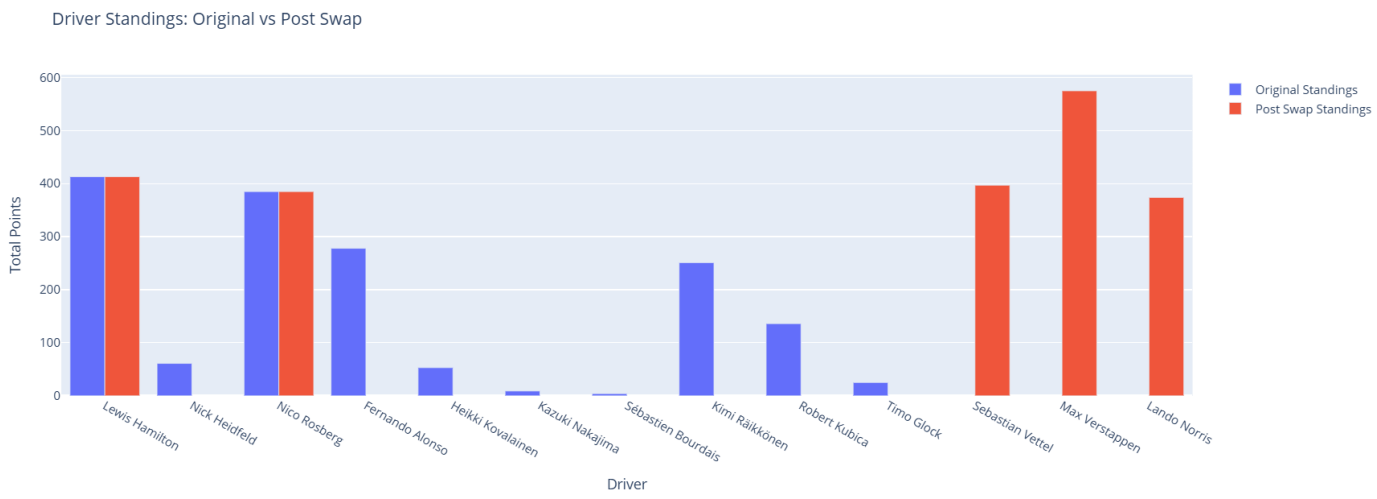


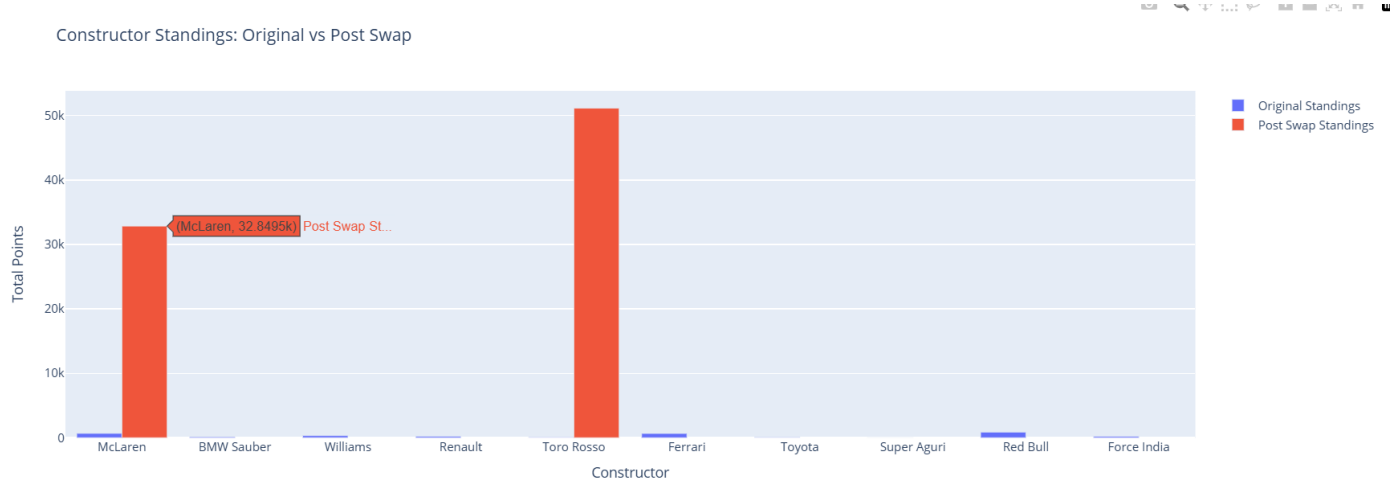
5. HYPOTHETICAL DRIVER SWAPS

- Identify two drivers to swap and retrieve their team assignments.
- Update the dataset to swap the drivers between their respective teams.
- Recalculate the total points for each team after the swap.
- Compare original and post-swap driver standings using bar charts.
- Compare original and post-swap constructor standings to analyze team performance shifts.
- Assess how the swap impacts individual driver rankings and overall team points.
- Analyze whether a weaker team benefits from gaining a top driver or if a strong team weakens due to the change.
- Evaluate how competitive balance and race strategies might shift due to the swap.

KEY INSIGHTS:

- Swapping drivers can significantly impact both individual standings and team performance.
- A top-performing driver moving to a weaker team can boost that team's total points but may lower their own ranking.
- A strong team losing a top driver might see a decline in constructor points, affecting championship outcomes.



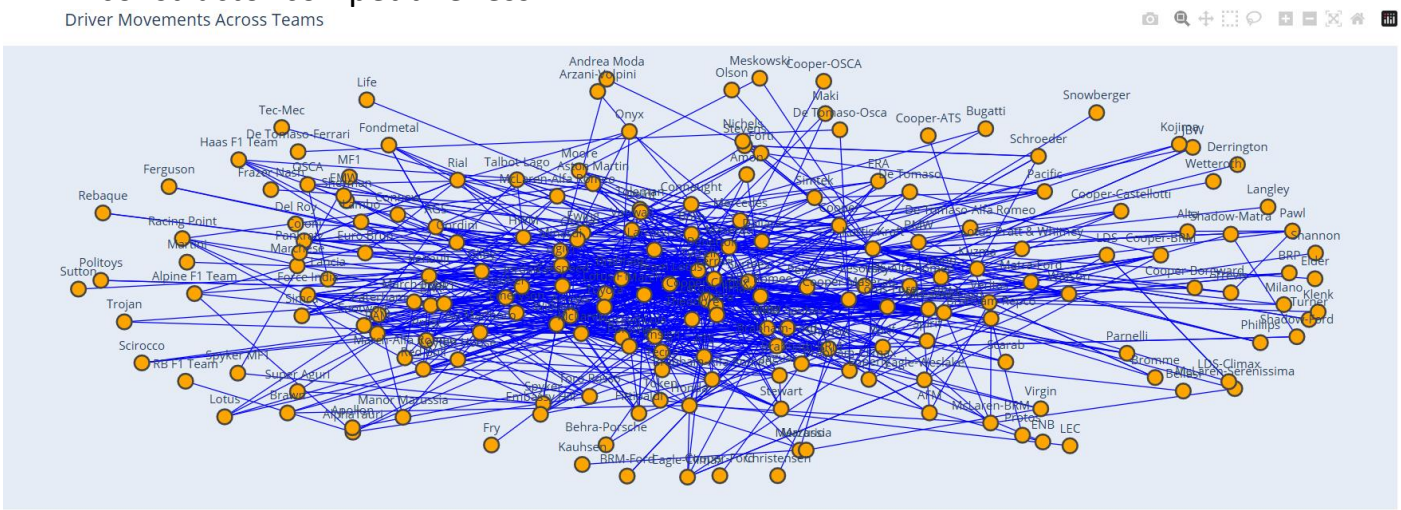


6. DRIVER MOVEMENTS AND TEAM NETWORKS

- Merge race results with driver and constructor data to associate drivers with teams for each season.
- Create a dataset of unique driver-team-season combinations and sort them by driver and year.
- Identify transitions by detecting changes in constructor names for each driver.
- Construct a directed graph where nodes represent teams and edges represent driver movements.
- Use a spring layout algorithm to optimize node positioning.
- Generate an interactive visualization using Plotly, where edges display driver transitions and nodes represent constructors.

KEY INSIGHTS:

- Frequent transitions indicate instability or strategic moves by drivers and teams.
- Stable drivers with fewer transitions highlight strong team retention.
- Top teams attract more drivers, showing dominance in the sport.
- The network structure reveals key talent shifts influencing team performance.
- Visual representation helps analyze historical trends in driver movements and constructor competitiveness.

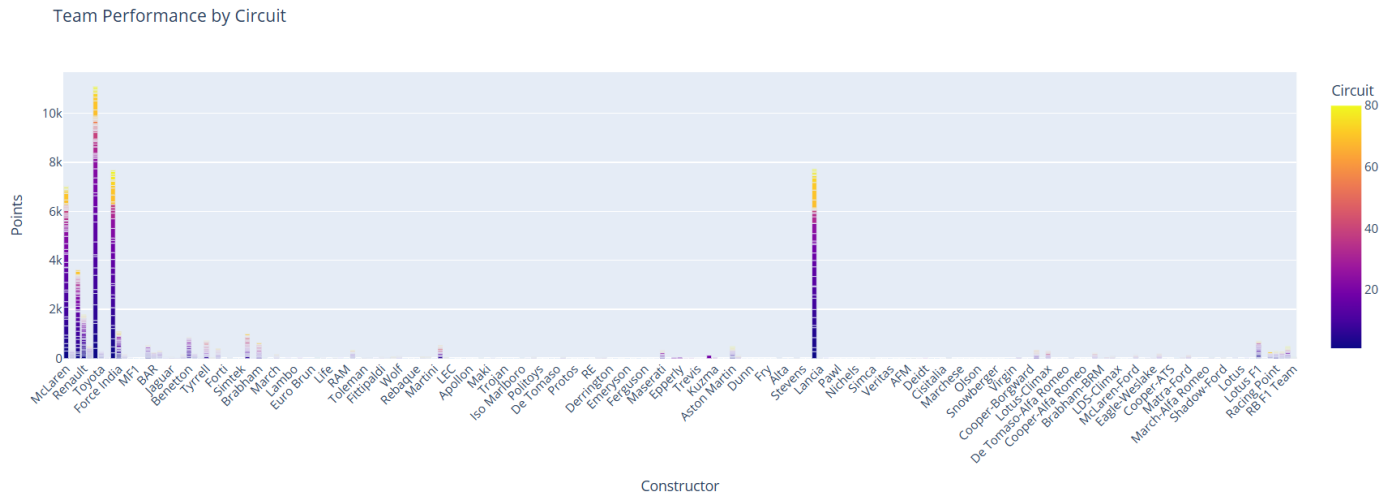


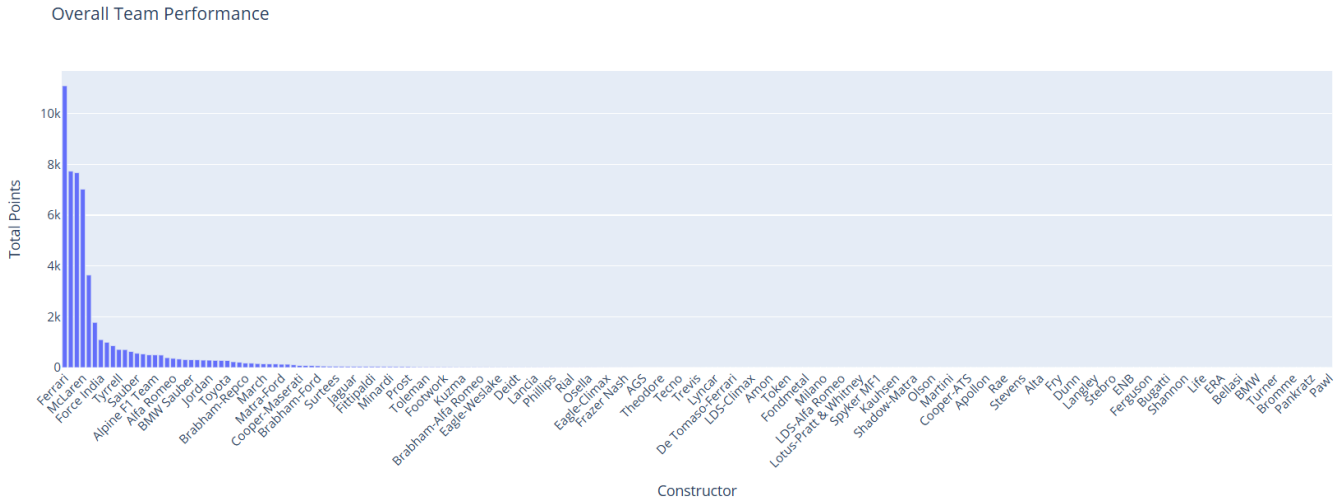
7. TEAM PERFORMANCE COMPARISON

- Aggregate total points per team to determine the best-performing constructor.
- Identify the top team based on accumulated points.
- Merge with constructor names for better readability.
- Merge race results with circuit data to analyze circuit-wise team performance.
- Group data by constructor and circuit to measure performance across different tracks.
- Visualize circuit-based team performance using a bar chart with circuits as a factor.
- Compare overall team performance using a separate bar chart without circuit influence.

KEY INSIGHTS:

- The best team is determined by total accumulated points over the seasons.
- Certain teams perform better on specific circuits, indicating track specialization.
- The overall performance comparison highlights dominant teams across all circuits.
- Circuit-specific analysis reveals teams' strengths and weaknesses in different environments.
- Visualizations provide clear insights into team dominance and consistency.





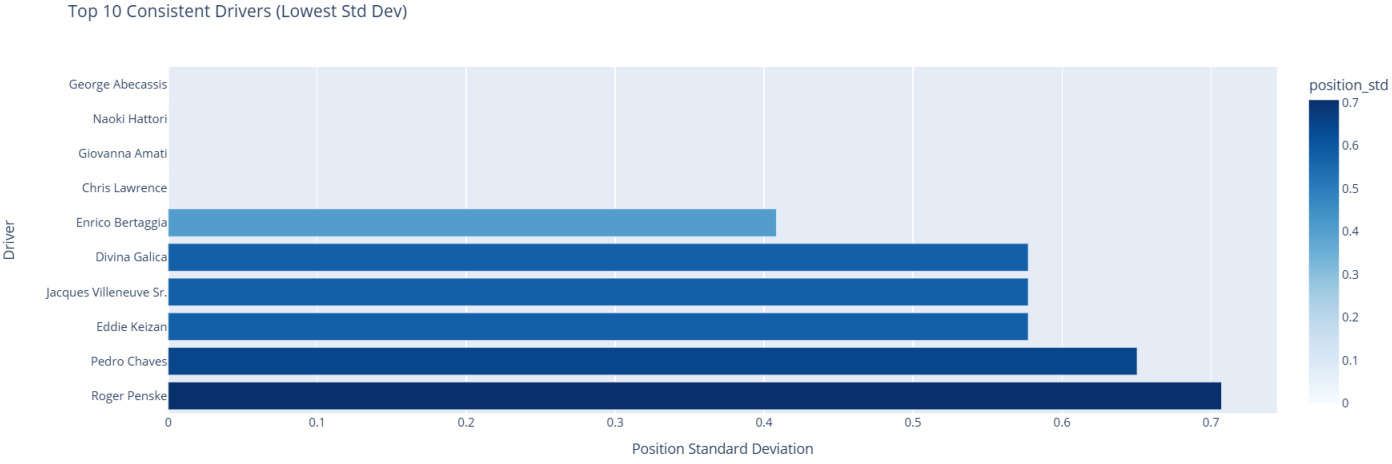
8. DRIVE CONSISTENCY IN RACE PERFORMANCE

- Gather comprehensive datasets, including race results, driver details, constructor information, and circuit specifics.

- Merge these datasets on common keys (e.g., raceId, driverId, constructorId) to create a unified DataFrame for analysis.
- Develop new features such as driver_name by combining first and last names, and calculate metrics like total points per constructor.
- Construct graphs to depict relationships and transitions, like driver movements between teams, using network analysis tools.

KEY INSIGHTS:

- Identified top-performing teams by aggregating total points, highlighting constructors with consistent success.
- Assessed drivers' average finishing positions and variability, pinpointing those with stable performances versus those with fluctuations.
- Highlighted drivers with the highest percentages of top-5 finishes, underscoring elite performers in the field.
- Mapped driver movements between teams over seasons, uncovering patterns in career progressions and team changes.
- Conducted direct comparisons between drivers or teams, providing a clear view of competitive standings.

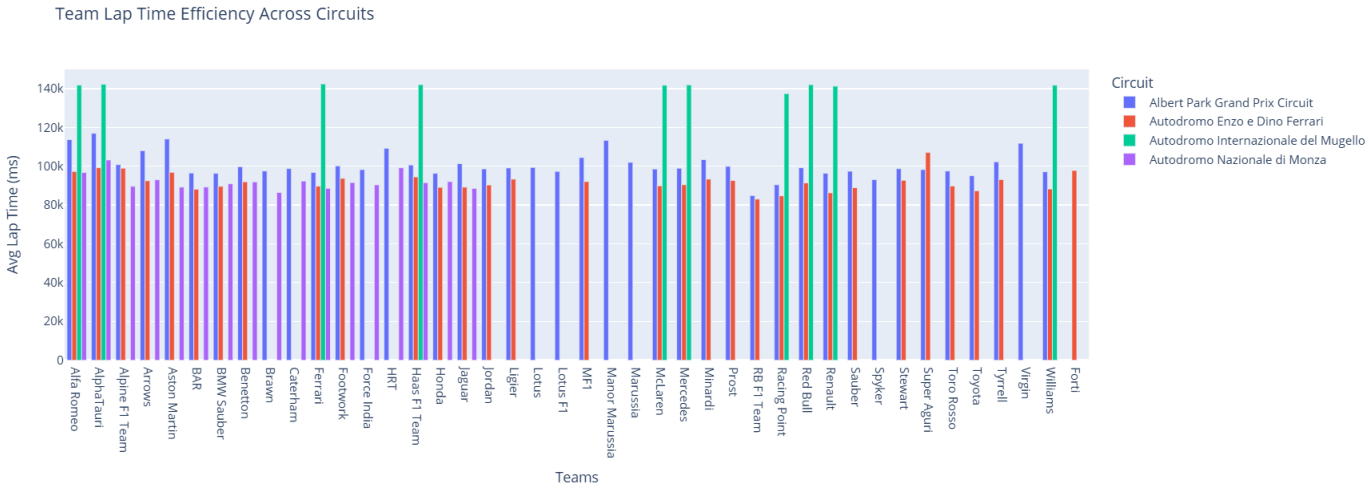


9. LAP EFFICIENCY

- Combine lap_times_data with results_data on raceId and driverId to obtain lap_results_df.
- Merge lap_results_df with constructors_data on constructorId, and with race_data on raceId to include circuitId. Then, merge with circuit_data on circuitId to append circuit names.
- Group the data by circuit name and constructor, computing the mean lap time in milliseconds for each group.

KEY INSIGHTS:

- Teams exhibit different average lap times depending on the circuit, highlighting strengths on specific track types.
- Analyzing average lap times helps teams pinpoint circuits where they excel or need improvement, informing strategic decisions.
- Understanding lap time efficiency assists teams in tailoring car setups and strategies to maximize performance on various circuits.

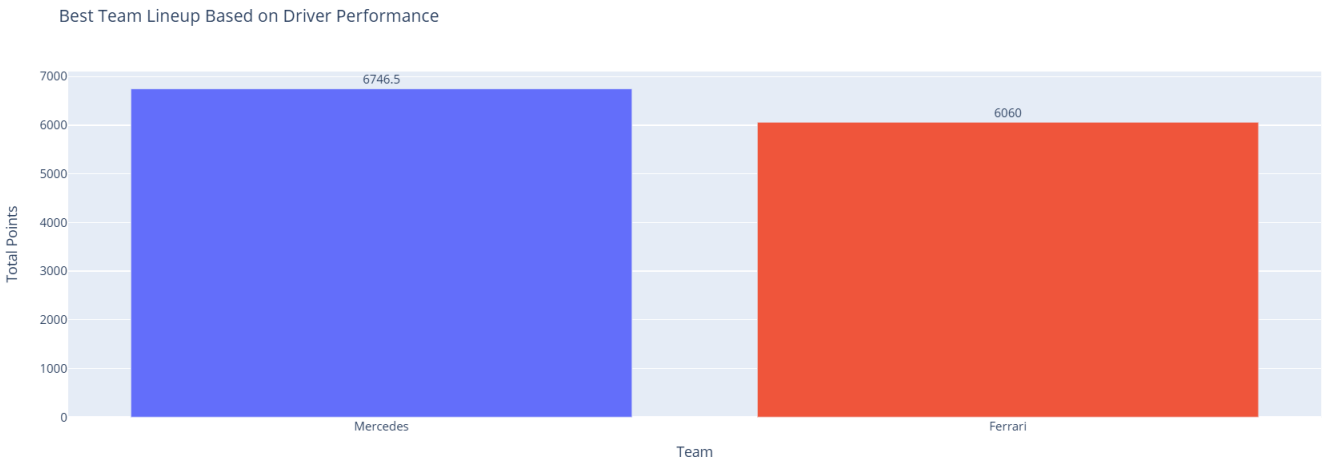


9. BEST TEAM LINE UP

- Merge lap_times_data with results_data to associate lap times with race results.
- Integrate constructors_data, race_data, and circuit_data to include team names and circuit details.
- Compute the average lap time per constructor for each circuit to assess team performance.

KEY INSIGHTS:

- Analyzing average lap times reveals which teams excel on specific circuits, highlighting strengths and weaknesses.
- Understanding lap time efficiency assists teams in tailoring strategies for different tracks, potentially improving overall performance.
- Identifying circuits where a team underperforms can guide targeted improvements in car setup and driver training.

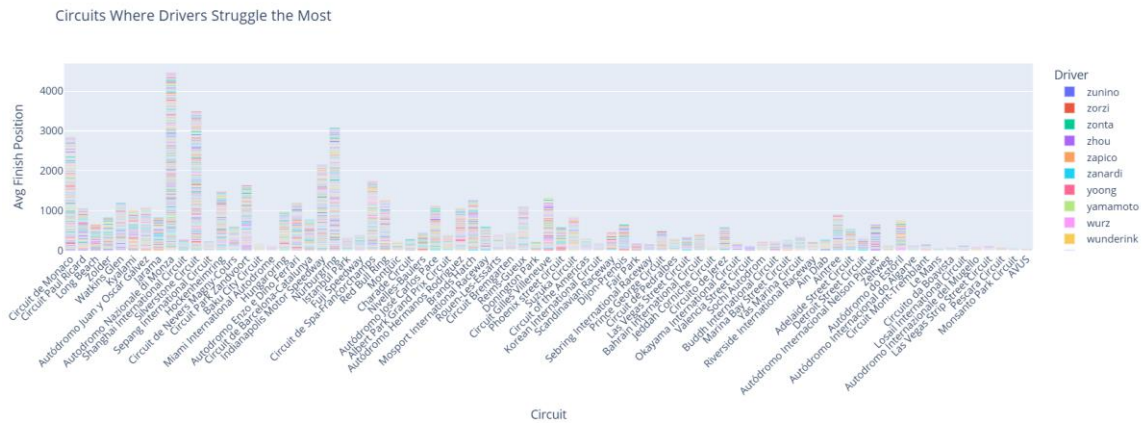


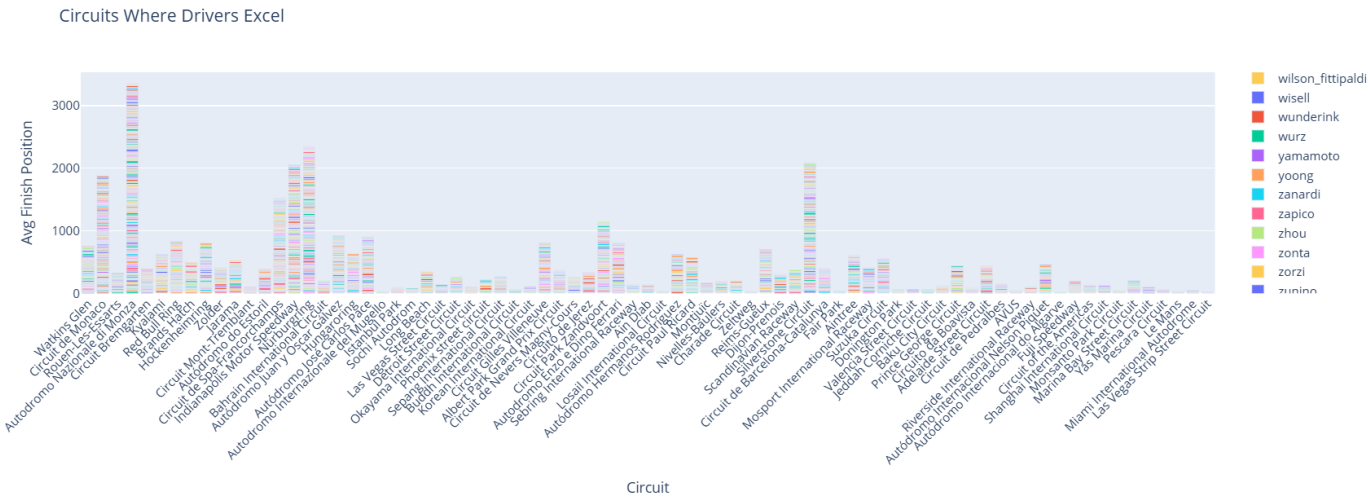
13. DRIVER SPECIFIC TRACK STRUGGLES

- Merge results_data with race_data to associate each race with its corresponding circuit.
- Merge the combined data with circuit_data to obtain circuit names.
- Merge with drivers_data to include driver references.
- Group the data by driver and circuit, computing the mean finishing position (positionOrder) for each combination.
- For each driver, determine the top 5 circuits where they have the lowest average finishing positions (indicating strong performance).
- Similarly, identify the top 5 circuits where they have the highest average finishing positions (indicating struggles).

KEY INSIGHTS:

- Certain drivers consistently perform better at specific circuits, showcasing their proficiency and comfort with particular track characteristics.
- Recognizing circuits where drivers struggle can inform teams' strategies and preparations, allowing for targeted improvements.
- Understanding these performance patterns enables teams to optimize race strategies, allocate resources effectively, and tailor training programs to enhance driver performance on less favorable tracks.





15. CHAMPION AGE TRENDS

- Merged datasets including race results, driver details, constructor information, and circuit data to create a comprehensive analysis framework.
- Analyzed driver movements between teams over seasons, constructing a network graph to visualize these transitions.
- Aggregated total points for each constructor to identify top-performing teams, both overall and on specific circuits.
- Calculated average finishing positions and standard deviations for drivers to assess performance consistency, highlighting those with stable results versus fluctuating ones.
- Evaluated average lap times of teams across different circuits to determine which teams maximize efficiency on specific tracks.
- Identified top drivers based on point accumulation and analyzed their contributions to constructors, determining the most effective team compositions.
- Assessed individual driver performances on various circuits to pinpoint tracks where they excel or struggle consistently.
- Examined the ages of championship-winning drivers over the years to identify trends and prevalent age ranges for peak performance.

KEY INSIGHTS:

- Visualizing driver transitions between teams reveals patterns in career moves, with certain teams acting as common stepping stones.
- Specific constructors consistently accumulate higher points, indicating sustained success and effective team strategies.
- Drivers with lower variability in finishing positions often contribute more reliably to team success, underscoring the value of consistent performance.
- Teams demonstrate varying lap time efficiencies across circuits, suggesting specialized strengths on particular track types.

MODEL BUILDING AND EVALUATION

(11. Predictions for 2025 Season and 12. Struggling Teams Analysis)

MULTI LINEAR REGRESSION:

- Encoded categorical data using **LabelEncoder** for both drivers (driver_encoded) and constructors (constructor_encoded).
- Merged **results_data** with **race_data** to extract **raceId** and **year**, and combined with **driver_data** to obtain **driverId** and **year**.
- Selected **features (X)** as **avg_finish** (average finishing position) and **num_races** (total races), and **target (Y)** as **points**.
- Split data into **80% training** and **20% testing** sets.**(HYPERPARAMETER)**
- Trained a **Multi-Linear Regression** model to predict driver performance.
- Predicted the **2025 Champion Driver** and the corresponding **Champion Constructor** based on the trained model.

```
Driver Points Prediction MSE: 1415.98
```

```
Predicted Drivers' Champion for 2025: Driver ID 830.0 with 77.3 points
```

```
Predicted Constructors' Champion for 2025: Constructor ID 1.0 with 34118.2 points
```

Drawbacks:

- Assumes **independence among features**, leading to potential **multicollinearity** issues.
- A potentially high **Mean Squared Error (MSE)** may indicate **underfitting** or the inability of the linear model to capture complex relationships.
- Fails to consider **temporal dependencies** and **seasonal trends**, which are critical in motorsport data; **time-series models** (e.g., LSTM) would handle this better.

XGBOOST:

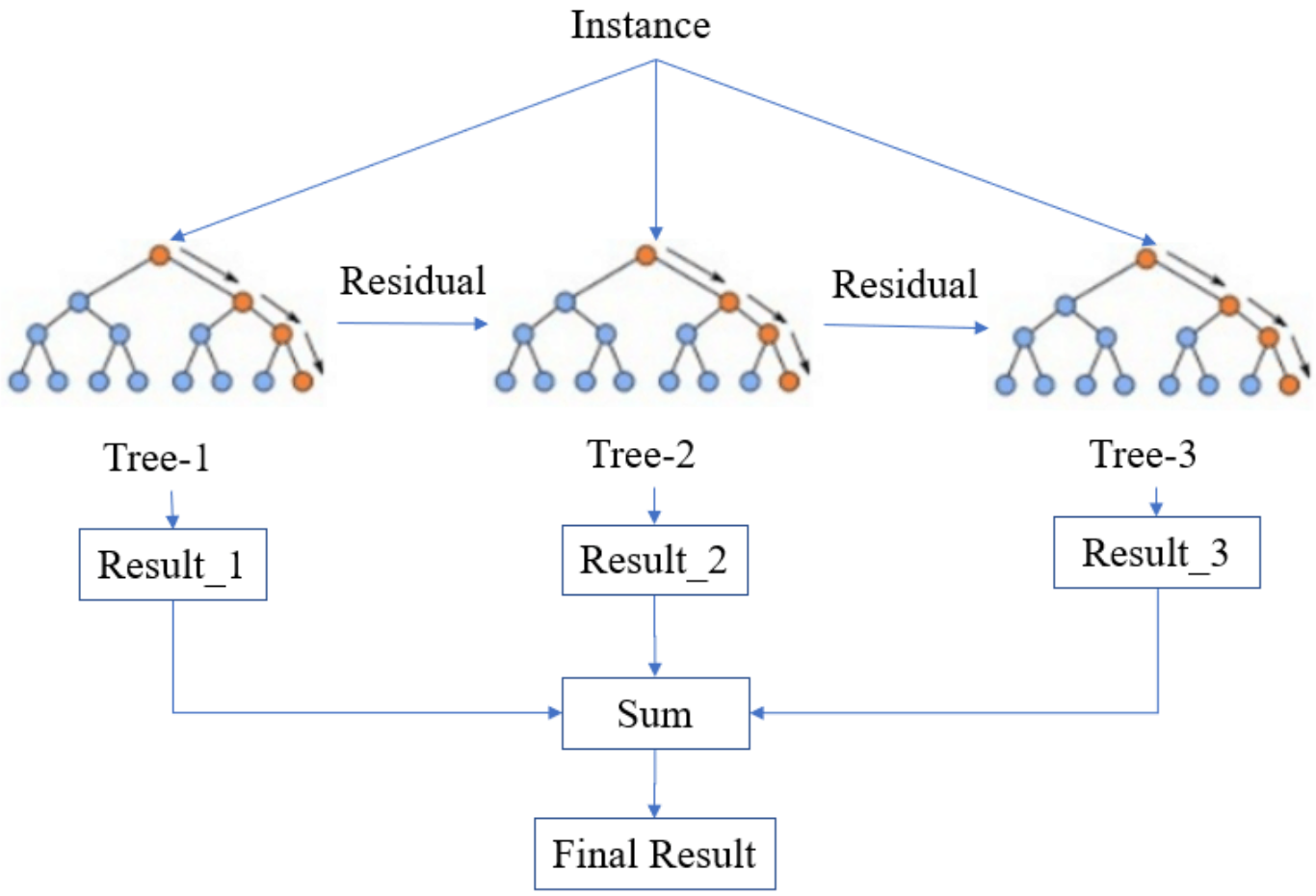
ABOUT:

BOOSTING : Boosting in machine learning is a technique that combines multiple weak learners into a single strong learner.

XGBOOST PROCEDURE EXPLANATION:

It builds decision trees sequentially with each tree attempting to correct the mistakes made by the previous one. The process can be broken down as follows:

1. **Start with a base learner**: The first model decision tree is trained on the data. In regression tasks this base model simply predict the average of the target variable.
2. **Calculate the errors**: After training the first tree the errors between the predicted and actual values are calculated.
3. **Train the next tree**: The next tree is trained on the errors of the previous tree. This step attempts to correct the errors made by the first tree.
4. **Repeat the process**: This process continues with each new tree trying to correct the errors of the previous trees until a stopping criterion is met.
5. **Combine the predictions**: The final prediction is the sum of the predictions from all the trees.



EXPLANATION ON CODE:

- Merged results_data with race_data, drivers_data, constructors_data, and status_data to create a comprehensive dataset (merged_results) containing all relevant features.
- Performed feature engineering by creating new features like is_podium (flag if driver finished in top 3), Status_Finished (flag indicating if the driver completed the race), and is_winner (flag for race winners).
- Built a machine learning pipeline that included a scaler for feature normalization, an over_sampler to handle class imbalance, and the XGBoost model for predictions.

- Defined a hyperparameter grid for tuning the XGBoost model, which included variations for max_depth, n_estimators, learning_rate, subsample, and colsample_bytree.
- Used GridSearchCV to perform hyperparameter tuning, selecting the best model based on cross-validation accuracy. The best parameters identified were max_depth: 3, n_estimators: 100, learning_rate: 0.01, subsample: 0.8, and colsample_bytree: 0.8.
- Achieved a Best CV Score of 1.0, which indicated a perfect score on the training data, raising concerns about potential overfitting.
- Used the tuned model to predict the 2025 Driver and Constructor Championships and identified teams that were likely to underperform based on the prediction results.

DRAWBACK:

- It fails to capture temporal patterns or long-term trends in driver and team performance, leading to potentially less accurate future predictions.
- These limitations led to the transition to LSTM, which better models sequential data and captures long-term dependencies.

LSTM (TIME SERIES FORECASTING):

- Aggregated the data by grouping prepared_data based on year and driverId and summed the points to get each driver's total points per year.
- Transformed the grouped data into a pivot table where year was the index, driverId became the columns, and the corresponding points filled the table.
- Created sequences for time-series modeling by defining a create_sequences function that generated input sequences (X) and their corresponding targets (y) based on a sequence length of 3.**(HYPERPARAMETER)**
- Reshaped the input data X into the shape expected by LSTM: (samples, timesteps, features), where samples are the number of sequences, timesteps is the sequence length, and features are the number of drivers.

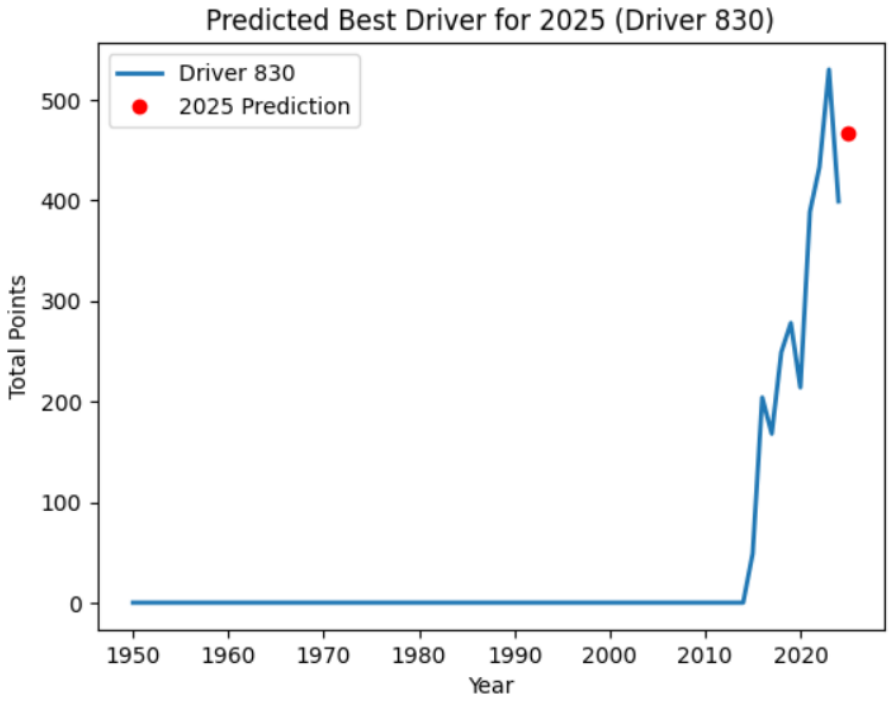
LAYERS OF LSTM

- Built an LSTM model using **Keras**, starting with an LSTM layer of 64 units with ReLU activation, followed by a Dense layer that outputs predictions for all drivers. The model was compiled using the Adam optimizer and **mean squared error** (MSE) loss.
- Trained the LSTM model on the prepared sequences (X and y) for **100** epochs with a batch size of **8**.**(HYPERPARAMETER)**

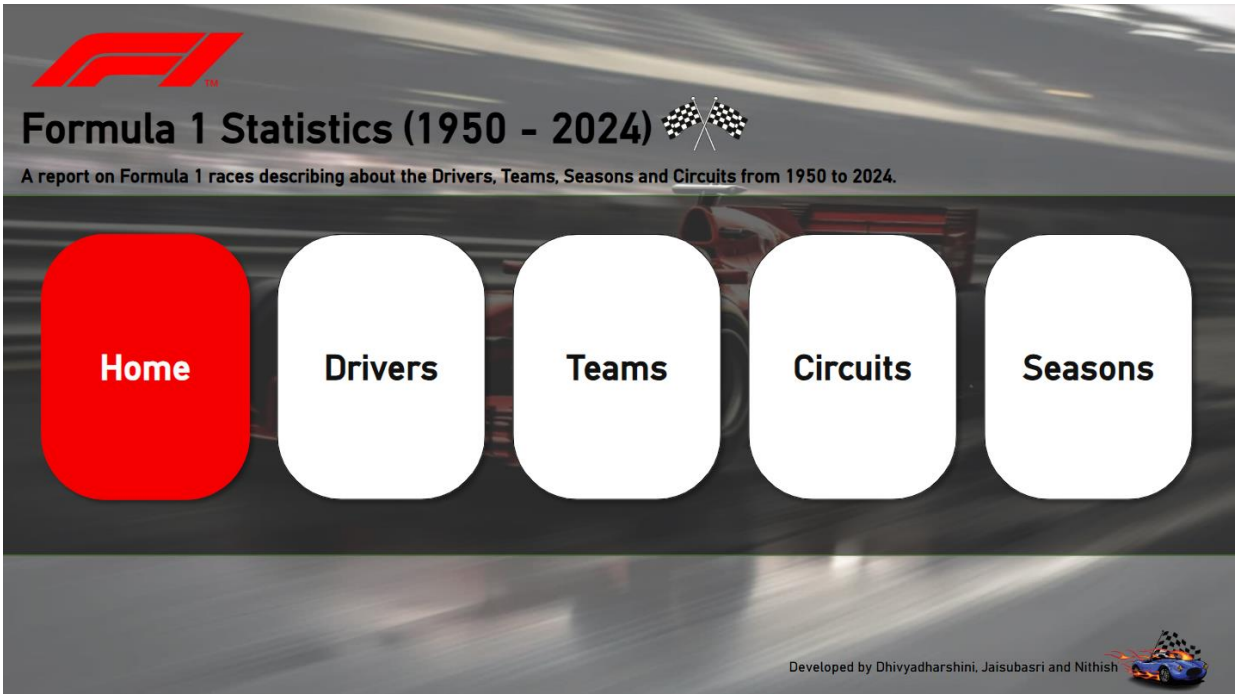
PREDICTION

- Prepared the last sequence from the scaled data to predict the next year (2025) by reshaping it into the required LSTM input format.

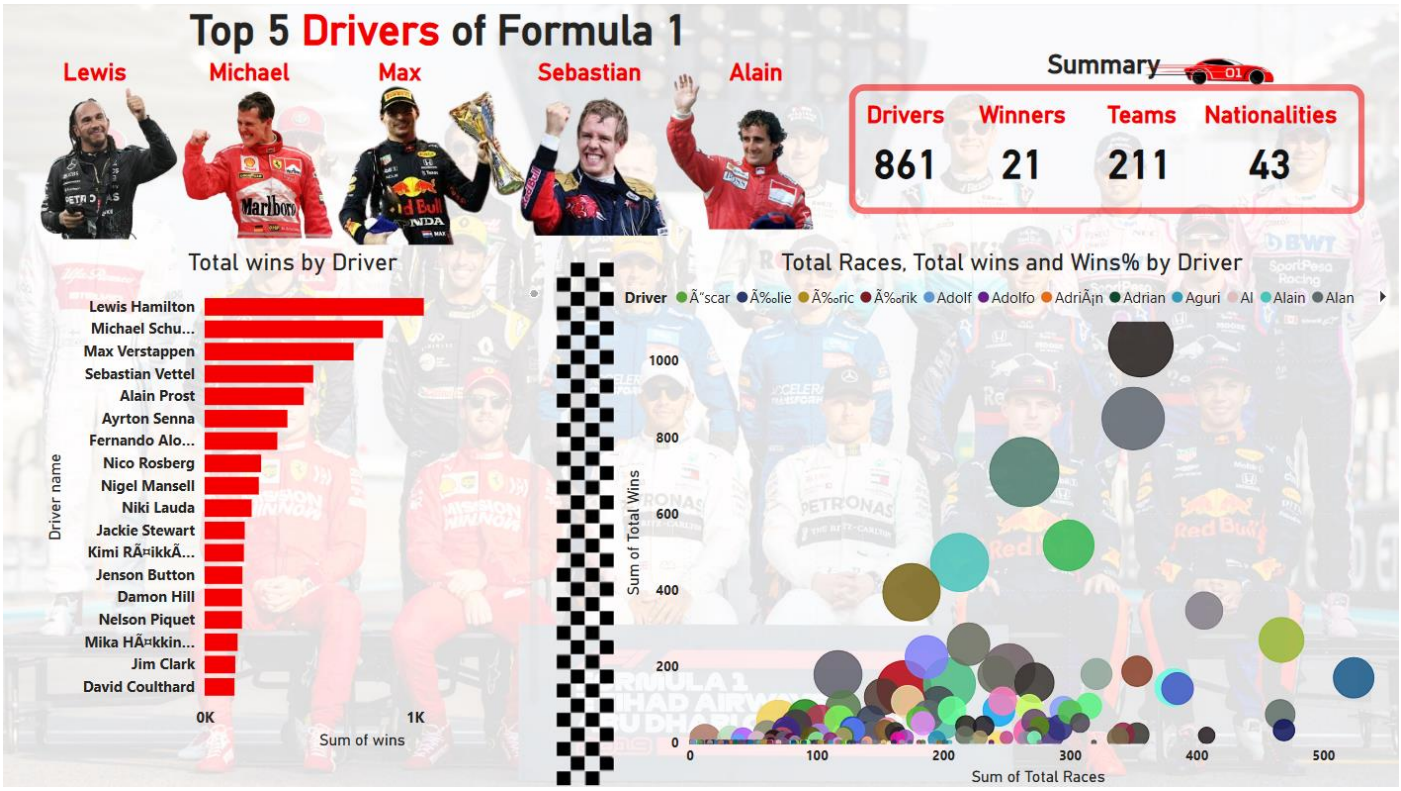
- Used the trained model to predict the scaled points for 2025, then inverse transformed these scaled predictions back to the original scale using the fitted MinMaxScaler.
- Mapped the predicted points to corresponding driver IDs and sorted the drivers in descending order of predicted points to get the projected standings for 2025.
- Identified the driver with the highest predicted points as the potential 2025 champion.
- Visualized the historical points of the predicted top driver using a line plot and added the 2025 prediction as a red dot to show the forecasted result.



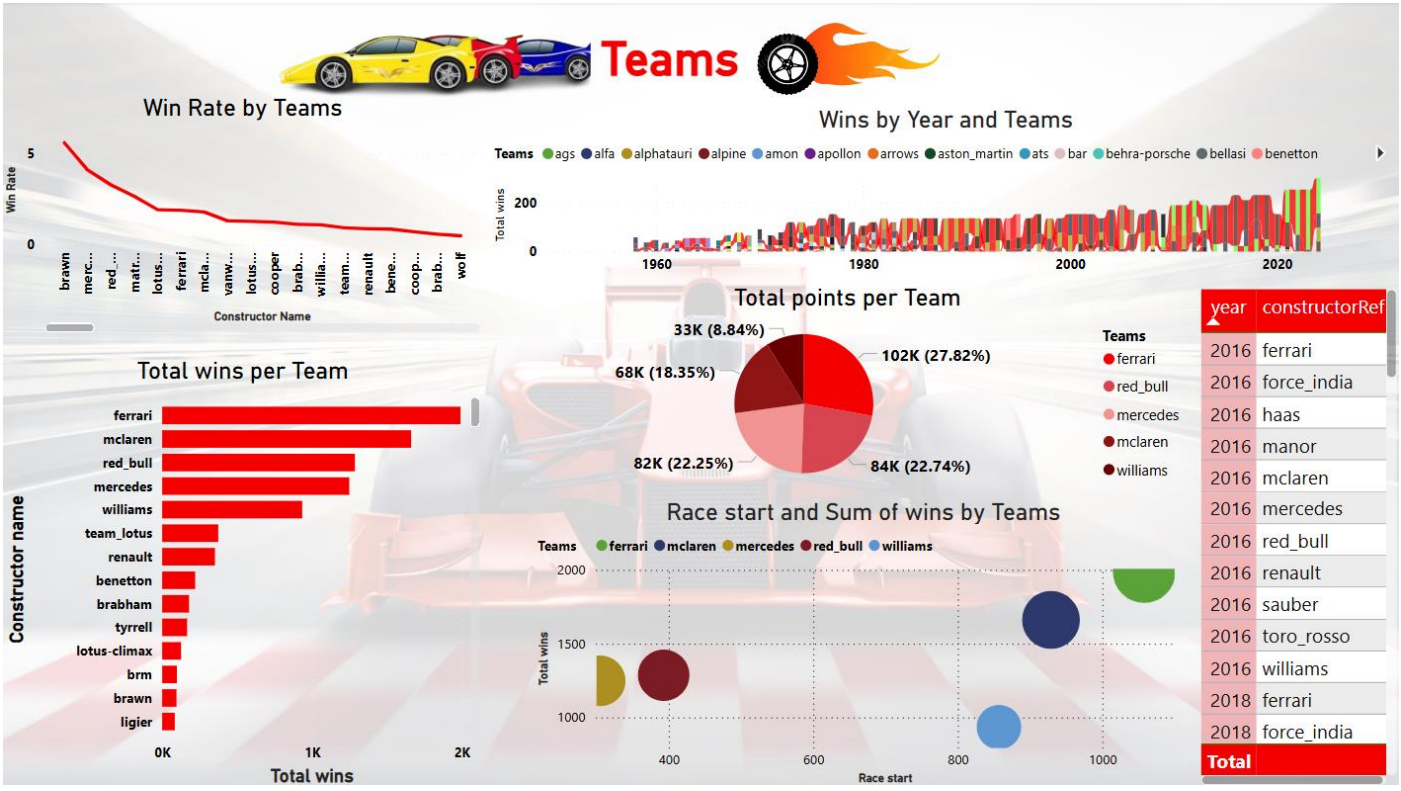
DATA VISUALIZATION



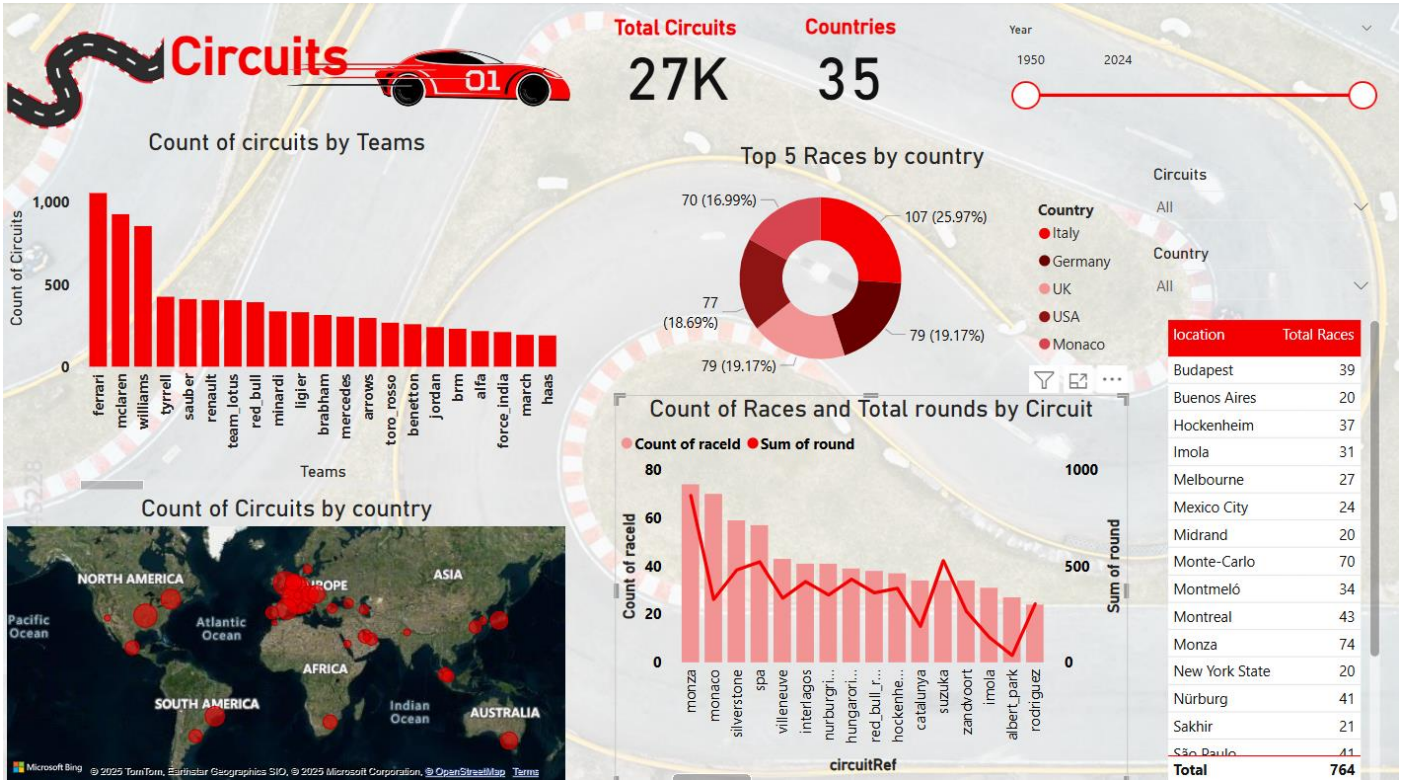
PAGE 1 – FRONT DASHBOARD



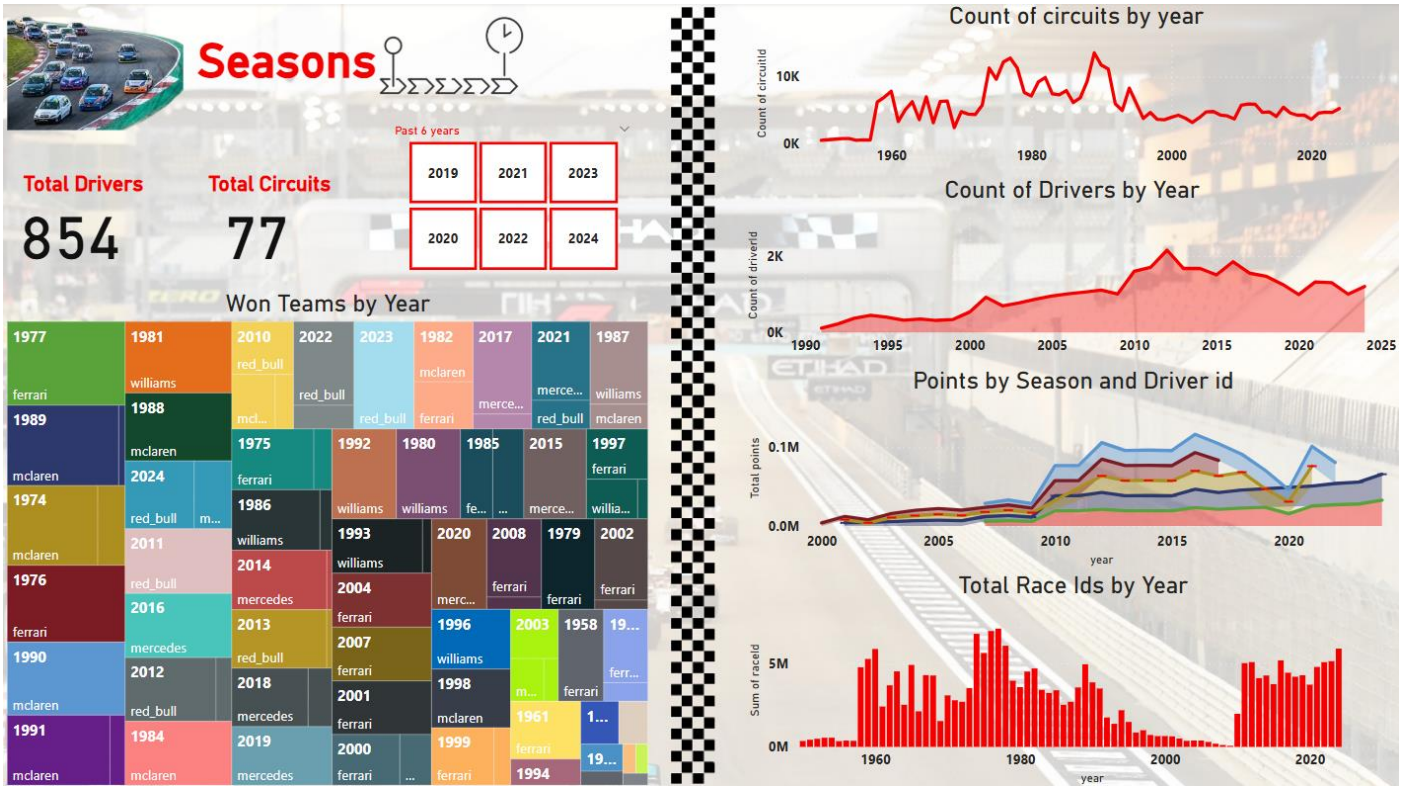
PAGE 2 – DRIVER DASHBOARD



PAGE 3 – TEAMS DASHBOARD

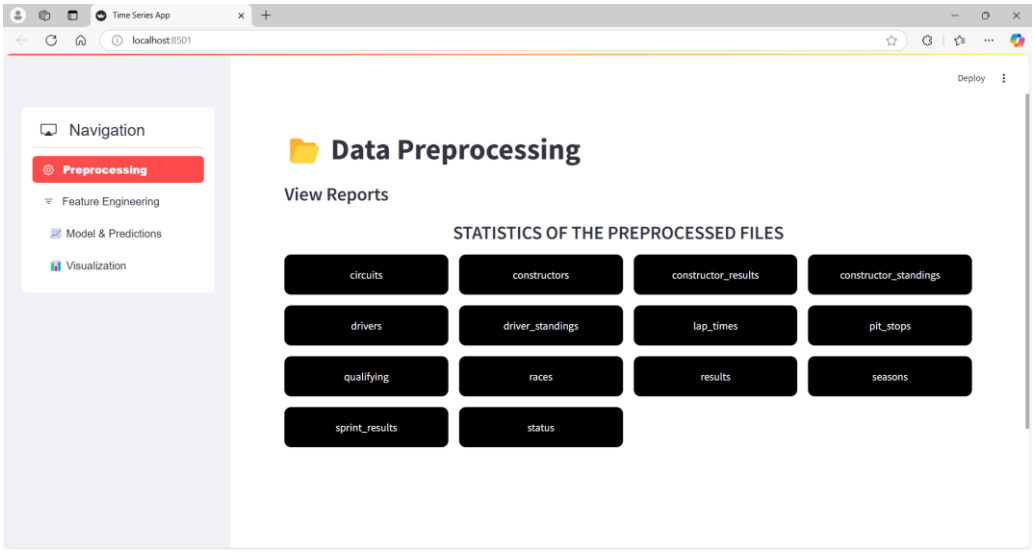


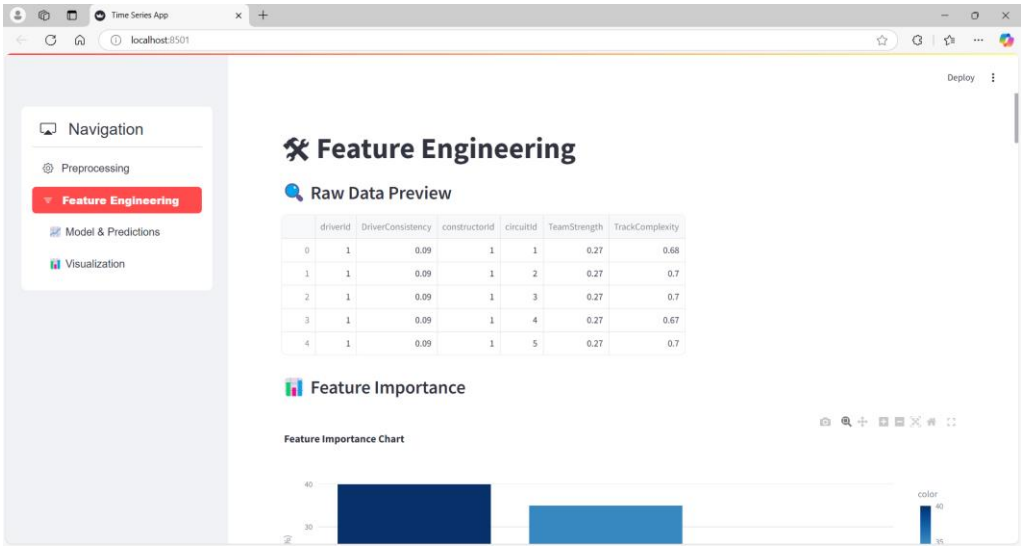
PAGE 4 – CIRCUITS DASHBOARD

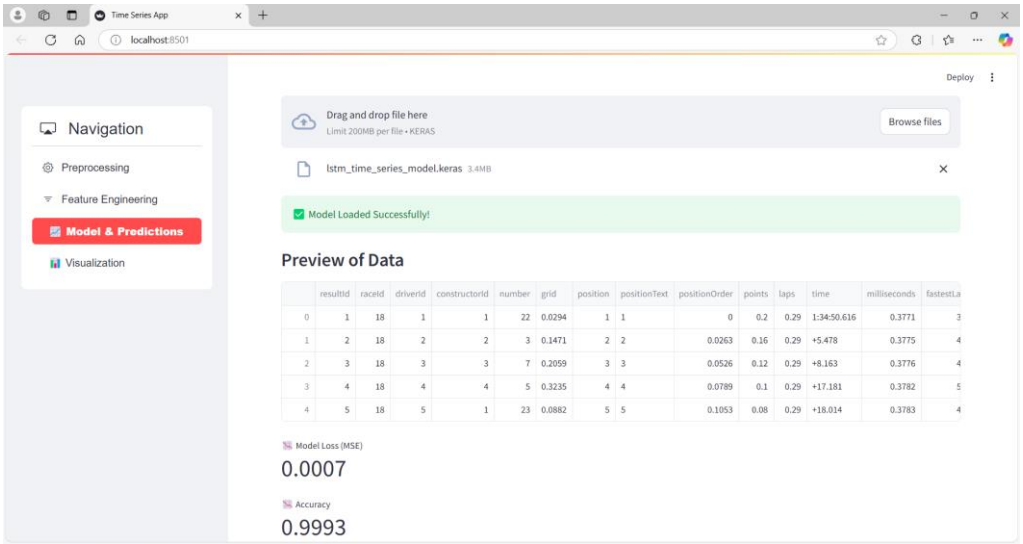


PAGE 5 – SEASONS DASHBOARD

FRONTEND


Time Series App
localhost:8501
Deploy
Navigation
Preprocessing
Feature Engineering
Model & Predictions
Visualization
Data Preprocessing
View Reports
STATISTICS OF THE PREPROCESSED FILES
circuits
constructors
constructor_results
constructor_standings
drivers
driver_standings
lap_times
pit_stops
qualifying
races
results
seasons
sprint_results
status


Time Series App
localhost:8501
Deploy
Navigation
Preprocessing
Feature Engineering
Model & Predictions
Visualization
Feature Engineering
Raw Data Preview
driverId DriverConsistency constructorId circuitId TeamStrength TrackComplexity
0 1 0.09 1 1 0.27 0.68
1 1 0.09 1 2 0.27 0.7
2 1 0.09 1 3 0.27 0.7
3 1 0.09 1 4 0.27 0.67
4 1 0.09 1 5 0.27 0.7
Feature Importance
Feature Importance Chart
color


Time Series App
localhost:8501
Deploy
Navigation
Preprocessing
Feature Engineering
Model & Predictions
Visualization
Drag and drop file here
Limit 200MB per file • KERAS
Browse files
lstm_time_series_model.keras 3.4MB
Model Loaded Successfully!
Preview of Data
resultId raceId driverId constructorId number grid position positionText positionOrder points laps time milliseconds fastestLa
0 1 18 1 1 22 0.0294 1 1 0 0.2 0.29 1:34:50.616 0.3771 3
1 2 18 2 2 3 0.1471 2 2 0.0263 0.16 0.29 +5.478 0.3775 4
2 3 18 3 3 7 0.2059 3 3 0.0526 0.12 0.29 +8.163 0.3776 4
3 4 18 4 4 5 0.3235 4 4 0.0789 0.1 0.29 +17.181 0.3782 5
4 5 18 5 1 23 0.0882 5 5 0.1053 0.08 0.29 +18.014 0.3783 4
Model Loss (MSE)
0.0007
Accuracy
0.9993

TEAM MEMBERS

TEAM NAME : HACKI DYNAMOS

TEAM MEMBERS NAME

- DHIVYADHARSHINI B (21pd06)
- JAISUBASRI K (21pd14)
- NITHISH R (21pd23)

REGISTERED EMAIL ID – 21pd14@psgtech.ac.in

CONTACT NUMBERS:

- DHIVYADHARSHINI B - +91 63691 69462
- JAISUBASRI K - +91 93641 34555
- NITHISH R - +91 89257 27111

****