**Predicting F1 Race Winners at the Monaco Grand Prix**

**1. Introduction**

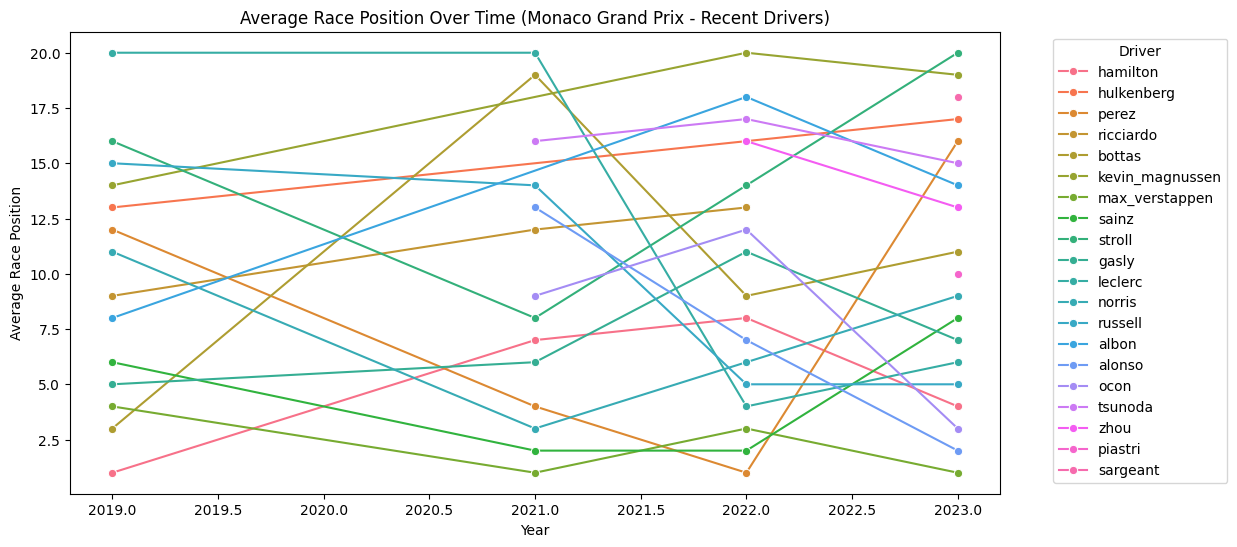
This project aimed to predict the winner of a specific Formula 1 race, focusing on the Monaco Grand Prix. The dataset included information on drivers, constructors, qualifying sessions, races, and results from 2019 to 2024. Two models were employed: Logistic Regression, known for its simplicity and interpretability, and Random Forest Classifier, chosen for its ability to handle non-linear relationships and capture feature interactions. By leveraging key performance metrics such as qualifying positions, race outcomes, and constructor performance, the goal was to identify which driver had the highest probability of winning an upcoming race.

**2. Data Exploration & Preprocessing**

The dataset underwent comprehensive exploration and cleaning to ensure quality and relevance:

* **Data Scope**: Data from 1950 to 2024 was available, but only races from 2019 to 2024 were retained to maintain relevance to current performance trends.
* **Data Cleaning**: Irrelevant columns such as URLs, exact race times, and nationality fields were removed. Drivers' birthdates were replaced with their ages for simplicity.
* **Handling Missing Data**: Features with missing values were imputed using mean imputation to ensure completeness.
* **Feature Engineering**: Several features were derived to enhance predictive capabilities:
  + **Driver Performance Metrics**: Average qualifying position, race position, consistency (position standard deviation), and position improvement from qualifying to race.
  + **Constructor Metrics**: Average points and positions of constructors across races.
* **Visualizations**:

**Trends in driver performance:** Trends in driver performance represent a combination of individual skill, adaptability, team dynamics, and external factors like track conditions or car improvements. These trends are crucial for building a predictive model, as they provide context and historical benchmarks that inform future race outcomes. By analyzing these trends, we can better understand which drivers are likely to excel at a given track, such as the Monaco Grand Prix.



**Correlations between features:** The correlations between features reveals how variables interact and help refine the model by identifying redundant features and emphasizing key drivers of race performance. In this case:

* Strong correlations between constructor metrics and driver race outcomes highlighted the importance of constructor performance.
* Overlap between driver metrics like avg\_race\_position and avgRacePositionAll led to feature selection decisions.
* Understanding these relationships ensured that the model used the most predictive and non-redundant features for better accuracy and generalizability.

**A screenshot of a graph

Description automatically generated**

**A graph of blue and white bars

Description automatically generated with medium confidence**

**Differences in qualifying positions for winners vs. non-winners**: The differences in qualifying positions between winners and non-winners underline the importance of starting position in Formula 1 races. While winners often start in better positions, non-winners show greater variability, emphasizing the role of race-day factors for the latter group.

**A graph of a group of blue squares

Description automatically generated with medium confidence**

**Average Race Position at Monaco Grand Prix graph:** This graph visualizes the average race positions for drivers over the last five years at the Monaco Grand Prix.The average position graph provides a clear picture of which drivers consistently perform well at Monaco and which are unlikely to contend for victory. It emphasizes the importance of race-day consistency, constructor influence, and track-specific performance trends.

**A graph of a graph

Description automatically generated with medium confidence**

**3. Model Selection & Training**

Two models were selected to address the binary classification problem of predicting race winners:

1. **Logistic Regression**:
   * Simplicity and Interpretability: Logistic Regression is straightforward to implement and interpret. The coefficients directly indicate the influence of each feature on the probability of winning, providing valuable insights.
   * Baseline Comparison: Logistic Regression serves as a baseline model to compare with more complex methods, ensuring the added complexity of models like Random Forest is justified.
   * Efficient for Small Datasets: Logistic Regression works well with smaller datasets, making it a practical choice for this task.
2. **Random Forest Classifier**:
   * Powerful for Non-Linear Relationships: Random Forest is capable of capturing complex, non-linear relationships in the data, making it well-suited for problems like predicting race winners where interactions between features can be intricate.
   * Robust to Overfitting: By averaging the results of multiple decision trees, Random Forest reduces overfitting and provides reliable predictions.
   * Feature Importance: It offers insights into which features are most influential in determining the outcome, helping to interpret the model’s decisions.
   * Hyperparameters (max\_depth, min\_samples\_split, min\_samples\_leaf) were tuned via grid search to improve generalization and avoid overfitting.

The models were trained using the following features: avg\_qualifying\_position avg\_race\_position position\_std\_dev avg\_position\_improvement avgRacePositionAll constructor\_avg\_position constructor\_avg\_points

The target variable (winner) was binary: 1 for winners and 0 for non-winners.

**4. Performance Evaluation**

The models were evaluated using accuracy and classification reports:

* **Logistic Regression**:
  + Accuracy: 90% — Indicates the model correctly predicts 90% of test cases.
  + Precision for Class 1 (Winners): 1.0 — Every predicted winner was correct.
  + Recall for Class 1 (Winners): 0.50 — Only half of the true winners were identified.
  + Most Important Features:
    - constructor\_avg\_points (Coefficient: 0.646): Teams with higher average points are more likely to win.
    - avg\_position\_improvement (Coefficient: 0.332): Drivers who improve their position from qualifying to race day have a higher chance of winning.
  + Negative Features:
    - avgRacePositionAll and constructor\_avg\_position have negative coefficients, indicating that better (lower) values in these metrics increase winning probability.
  + Least Important Feature:
    - position\_std\_dev (0.075): Consistency in position has a smaller impact compared to performance metrics.
* **Random Forest Classifier**:
  + Accuracy: 0.90 — The model is correctly predicting 90% of the samples in the test set.
  + Precision for Class 1 (Winner): 1.00 — When the model predicts a winner, it is always correct.
  + Recall for Class 1 (Winner): 0.50 — Out of the actual winners, the model correctly identifies only half of them.
  + Feature Importance:
    - constructor\_avg\_points: 30.99% importance — The average points scored by the driver's constructor strongly influence predictions, aligning with domain expectations since strong constructors often field winning cars.
    - constructor\_avg\_position: 24.55% importance — Reflects how well the constructor's drivers perform on average in races.
    - avgRacePositionAll: 22.80% importance — Average race position for all races. This suggests consistent race performance significantly impacts predictions.
    - avg\_qualifying\_position: 10.01% importance — Consistency in qualifying still has influence, though less than race-related metrics.
    - position\_std\_dev: 9.62% importance — Consistency in race positions impacts predictions, but less than raw performance metrics.
    - avg\_position\_improvement: 2.02% importance — Shows minimal influence, possibly because other metrics already capture its effect.

**Comparison**:

While both models achieved similar accuracy and precision-recall metrics, Random Forest Classifier performed better overall due to its ability to handle non-linear relationships and capture feature interactions. Its reliance on ensemble methods made it more robust to variations in the data and better equipped for future expansion of features.

However, Logistic Regression excelled in interpretability, making it a valuable choice for understanding the factors driving predictions. This interpretability makes it a strong baseline model and a complementary tool for providing insights into the dataset.

If the goal is to optimize prediction accuracy and capture complex relationships, Random Forest Classifier is the better-performing model. However, for cases where understanding feature contributions is critical, Logistic Regression provides unmatched clarity and should be used in conjunction with Random Forest for a balanced approach.

**5. Challenges & Future Work**

**Challenges**

1. **Feature Selection and Redundancy**:
   * **Challenge**: Overlap between features such as avg\_race\_position and avgRacePositionAll.
   * **Solution**: Retained avgRacePositionAll, which captured broader trends.
2. **Overfitting in Random Forest**:
   * **Challenge**: Unrestricted tree depth caused overfitting.
   * **Solution**: Hyperparameters like max\_depth and min\_samples\_split were tuned to improve generalization.
3. **Handling Missing Data**:
   * **Challenge**: Missing values in prediction datasets due to incomplete race data.
   * **Solution**: Used mean imputation to fill gaps.

**Future Work**

1. Feature Engineering:
   * Incorporate dynamic features such as weather conditions.
   * Add track-specific features, as circuits like Monaco have unique characteristics influencing race outcomes.
2. Experimenting with Advanced Models:
   * Test Gradient Boosting Models like XGBoost or LightGBM, which handle non-linear relationships effectively and often outperform Random Forest on tabular data.
3. Enhancing Interpretability:
   * Visualize feature importance across models (e.g., bar charts of coefficients for Logistic Regression and impurity-based importance for Random Forest).
4. Evaluating Alternative Targets:
   * Instead of predicting a single winner, rank drivers by winning probability to provide richer insights into race competitiveness.
   * Explore multi-class classification to predict podium finishes (top 3) rather than only the winner.
5. Expanding Dataset Scope:
   * Include additional seasons or a broader range of races to capture more trends and improve model robustness.
   * Incorporate constructor-level data, such as upgrades or reliability metrics, for more granular predictions.

**Conclusion**

This project successfully predicted F1 race winners with high accuracy using Logistic Regression and Random Forest Classifier. The findings emphasized the importance of constructor performance and race consistency in predicting outcomes. While the models performed well, future work could enhance recall for winners, incorporate additional features, and explore alternative predictive targets like podium finishes. This approach provides a robust foundation for further exploration of machine learning in motorsport analytics.