Formula 1 Race Outcome Prediction Using Pre-Race Features

Team 1:
Rucha Bhale
Aiden Insall

Table of Contents

Introduction and Motivation

Formula 1 racing represents one of the most data-rich sports in the world, with thousands of races spanning over seven decades providing a wealth of information for predictive modeling. Understanding which factors influence race outcomes has both practical and analytical value.

This project addresses two fundamental research questions:

1. Which pre-race factors are most predictive of a Formula 1 driver's race outcome?
2. Can regression models accurately predict driver finishing positions using only pre-race data, and which model performs best for this task?

These questions are particularly relevant because they require distinguishing between factors truly available before a race starts (qualifying position, team performance, driver form) versus information that becomes known only during or after the race (lap times, pit stop strategies, mechanical failures, other telemetry details).

We utilized the Formula 1 World Championship dataset from Kaggle, originally compiled from the Ergast Developer API, spanning 1950 to 2024. This dataset encompasses 74 seasons, approximately 1,100 Grand Prix events, and over 25,000 driver-race combinations across 14 CSV files. The dataset is curated by the F1 analytics community, and it's clear separation of pre-race and post-race information made it ideal for building a prediction model without data leakage concerns. Our analysis focused on understanding what influences success rather than achieving perfect prediction accuracy, recognizing that Formula 1's inherent unpredictability crashes, weather changes, mechanical failures set a natural ceiling on model performance.

Methods

1. Data Preparation and Merging

We merged 9 of the 14 available CSV files to construct our analytical dataset:

- results.csv,
- races.csv,
- drivers.csv,
- constructors.csv,
- qualifying.csv,
- circuits.csv,
- driver_standings.csv,
- constructor_standings.csv,
- status.csv.

Files containing in-race data such as lap_times.csv and pit_stops.csv were explicitly excluded to prevent data leakage. The merging process used appropriate keys like raceId, driverId, constructorId to match records across tables, resulting in 10,494 complete records after removing entries with missing qualifying data.

The merged dataset contained approximately 36 columns, but only a subset was relevant for pre-race prediction. We removed records with missing grid positions, which resulted in a dataset spanning 1994-2024 left with 10,494 records. Earlier F1 eras lacked consistent qualifying data and were naturally excluded through this process.

2. Feature Engineering

Rather than using raw categorical variables (driver ID, constructor ID, circuit ID), we engineered six performance-based features that capture historical patterns while remaining applicable to new drivers and changing team lineups:

1. **grid_position**: The driver's starting position from qualifying, used directly without transformation. This represents the single most important pre-race variable, as qualifying performance demonstrates both driver skill and car pace.
2. **avg_finish_last_3**: Rolling average of finishing positions from the driver's most recent three races. Calculated to ensure only past races inform predictions. This captures very recent form and momentum.
3. **avg_finish_last_5**: Rolling average over five races, providing a slightly smoothed indicator of recent performance less susceptible to single-race anomalies.
4. **avg_finish_at_circuit**: Driver's historical average finishing position at the specific circuit, computed as an expanding mean. This captures circuit-specific expertise. Some drivers excel at Monaco's tight corners while others perform better at some other circuit.
5. **constructor_avg_finish**: The team's average finishing position for the current season but not including the current race. This quantifies car quality and team performance, which fundamentally constrains driver outcomes regardless of individual skill.
6. **points_before_race**: Cumulative championship points earned before the race. While this correlates with performance, it provides minimal additional predictive value beyond the other features.

Missing values were created naturally for drivers' first races or when racing at new circuits. We imputed these using median values: avg_finish_last_3 and avg_finish_last_5 filled with 11.33 and 11.40 respectively avg_finish_at_circuit with 10.21, and constructor_avg_finish with 11.64. This approach provides reasonable default expectations for drivers without historical data.

```
# Handle Missing Values
print("\nHandling missing values")

feature_cols = ['avg_finish_last_3', 'avg_finish_last_5', 'avg_finish_at_circuit',
                'constructor_avg_finish', 'points_before_race']

print("Missing values before imputation:")
print(df[feature_cols].isnull().sum())

for col in feature_cols:
    if df[col].isnull().any():
        median_val = df[col].median()
        df[col] = df[col].fillna(median_val)
        print(f"Filled {col} with median: {median_val:.2f}")

print("\nMissing values after imputation:")
print(df[feature_cols].isnull().sum())
```

3. Temporal Validation Strategy

We employed a temporal train-validation-test split to mimic real-world forecasting conditions and prevent data leakage across time periods. Training data included all races from 1994-2015 (6663 records) validation data covered 2016-2018 (1275 records) and test data comprised 2019-2024 (2556 records). This chronological split ensures models learn from historical patterns and are evaluated on genuinely future races, keeping in view the temporal nature of Formula 1 data where regulations, technology, and driver/team compositions keep evolving over time.

Model Development

We tested three regression models representing different approaches to the prediction task:

1. **Linear Regression** serves as our baseline, assuming linear relationships between features and finishing position. Despite its simplicity, linear models are interpretable and perform well when feature relationships are approximately linear.

2. **Random Forest Regressor** is an ensemble of decision trees that can capture non-linear interactions between features. We configured it with default parameters initially: 100 estimators, maximum depth of 10, and minimum samples per split of 20.

```python
# Model 2: Random Forest

print("MODEL 2: RANDOM FOREST\n")

rf = RandomForestRegressor(
    n_estimators=100,
    max_depth=10,
    min_samples_split=20,
    min_samples_leaf=10,
    random_state=42,
    n_jobs=-1
)

rf.fit(X_train, y_train)
y_val_pred_rf = rf.predict(X_val)

mae_rf = mean_absolute_error(y_val, y_val_pred_rf)
rmse_rf = np.sqrt(mean_squared_error(y_val, y_val_pred_rf))
r2_rf = r2_score(y_val, y_val_pred_rf)

print(f"MAE: {mae_rf:.3f}")
print(f"RMSE: {rmse_rf:.3f}")
print(f"R²: {r2_rf:.3f}")
```

3. **XGBoost Regressor** implements gradient boosting with efficient handling of missing values and regularization. Initial configuration used 100 estimators, maximum depth of 6, and learning rate of 0.1.

```python
# Model 3: XGBoost

print("MODEL 3: XGBOOST\n")

xgb = XGBRegressor(
    n_estimators=100,
    max_depth=6,
    learning_rate=0.1,
    subsample=0.8,
    colsample_bytree=0.8,
    random_state=42,
    n_jobs=-1
)

xgb.fit(X_train, y_train)
y_val_pred_xgb = xgb.predict(X_val)

mae_xgb = mean_absolute_error(y_val, y_val_pred_xgb)
rmse_xgb = np.sqrt(mean_squared_error(y_val, y_val_pred_xgb))
r2_xgb = r2_score(y_val, y_val_pred_xgb)

print(f"MAE: {mae_xgb:.3f}")
print(f"RMSE: {rmse_xgb:.3f}")
print(f"R²: {r2_xgb:.3f}")
```

For Random Forest and XGBoost, we performed hyperparameter tuning using GridSearchCV with 3-fold cross-validation on the training set. For Random Forest, we tested **n_estimators** in [100, 200], **max_depth** in [15, 20], and **min_samples_split** in [5, 10]. For XGBoost, we explored **n_estimators** in [100, 200], **max_depth** in [6, 9], and **learning_rate** in [0.05, 0.1].

Additionally, we implemented a **Stacking Regressor** combining Linear Regression, Random Forest, and XGBoost as base models with Linear Regression as the level 1 model. This ensemble approach leverages the strengths of different model types, potentially improving predictions when base models make complementary errors.

Results

1. Exploratory Data Analysis

Our exploratory analysis of 10494 F1 race records ranging from 1994-2024 revealed that :

- Grid position is the dominant predictor of race outcomes, with a correlation of 0.58 to finishing position.
- Drivers starting from pole position win 50% of races, but this drops sharply to 24% for P2 starters, demonstrating both the advantage of qualifying well and the unpredictability of race execution.
- The average position change of -0.02 shows most drivers finish near where they start, though the distribution has long tails with some gaining or losing 10+ positions due to incidents and strategy.
- The relationship between grid and finish position follows a predominantly linear pattern across the field, which informed our choice of Linear Regression as the primary modeling approach.

These findings confirm that while qualifying performance sets baseline expectations, substantial race-day variance remains, with only 36% of finishing position variance explained by grid position alone.

2. Baseline Model Performance

1. **Grid position only Linear regression:** We established a grid-position-only baseline to quantify how much additional features improve predictions. Using only **grid_position** as input, Linear Regression achieved validation MAE of 3.865 and R^2 of 0.363. This demonstrates that starting position alone explains 36% of variance in race outcomes which is a strong foundation but also leaves a substantial room for improvement.
2. **Linear Regression Full features :** This model used all six features achieved validation MAE of 3.622 and R^2 of 0.411, representing a 0.243 position improvement i.e. a 6.3% reduction in error and 0.047 increase in explained variance. This confirms our engineered features add meaningful predictive value beyond only grid position.

3. Model Comparison

Performance on the validation set results:

Model	MAE	R^2
Linear Regression with only Grid Position	3.865	0.363
Linear Regression with all features	**3.622**	**0.411**
Random Forest (tuned)	3.793	0.364
XGBoost (tuned)	3.724	0.386
Stacking Ensemble	**3.661**	**0.408**

Test Set Evaluation

Final evaluation on the held-out test set from 2019-2024 confirmed Linear Regression's superiority:

Model	Test MAE	Test R^2
Linear Regression	**3.320**	**0.453**
Random Forest (tuned)	3.486	0.411
XGBoost (tuned)	3.406	0.435
Stacking Ensemble	3.354	0.452

Interpretation of Model Performance

Logistic regression came out to be the best model with an MAE of 3.320 positions which means predictions are, on average, within 3-4 positions of actual results. For a 20-driver field, this represents reasonable but not exceptional accuracy. R^2 of 0.45 means your model explains 45% of why finishing positions vary, while 55% remains unpredictable due to race-day events.

Interestingly, the Stacking Ensemble by combining Linear Regression, Random Forest, and XGBoost came very close with an MAE of 3.661 with only 0.04 positions worse. While this slight difference wasn't enough to justify the added complexity of ensemble methods for our purposes, it suggests that combining different model approaches captures complementary patterns in the data. Cross-validation consistency analysis showed Linear Regression achieved mean MAE of 4.210 across 5 folds on the training set, while Stacking achieved 4.224. The marginally lower standard deviation for Stacking (0.649 vs 0.656) suggests slightly more consistent performance, but the difference is negligible. For this project, we chose Linear Regression as our final model due to its **superior performance, simplicity, and interpretability.**

This aligns with F1's reputation as an unpredictable sport. While qualifying pace and team quality set baseline expectations, race-day execution matters enormously. Our model captures the predictable component while acknowledging inherent limits.

Feature Importance Insights

1. **Grid position** (35%) and **constructor performance** (28%) matter most in this perdition model.
2. The rolling averages from recent races **avg_finish_last_3** and **avg_finish_last_5** contribute about 13% combined. This helps but isn't huge. If you start P1 in a fast car, you'll probably finish well even if your last few races were bad. And if you're at the back in a slow car, good recent form won't save you.

3. Circuit-specific history **avg_finish_at_circuit** barely matters at 5%, which is surprising since commentators talk about track knowledge a lot. Past performance at a specific circuit doesn't add much beyond being a good driver overall.
4. Championship points **points_before_race** showed almost no impact, which confirms it's redundant. It just reflects how well you've been doing recently, which we already captured with the rolling averages. In retrospect, we could have dropped this feature without losing anything.

Discussion

Why Linear Regression Outperformed Complex Models

While tree-based models like Random Forest and XGBoost are often assumed to be superior for complex prediction tasks, Linear Regression outperformed them in our case. However, three factors explain this outcome:

1. **The relationships in our data are predominantly linear:** As shown in our EDA, the scatter plot of grid position vs finish position follows a roughly diagonal pattern, and the average finish by grid position plot demonstrates a consistent linear trend. A driver starting one position worse tends to finish roughly one position worse, without complex interaction effects. Tree-based models search for non-linear patterns that simply don't exist in this domain.

2. **Dataset size :** Our dataset size with 6,663 training samples is modest for tree-based methods. Random Forest and XGBoost require substantial data to learn robust decision rules and interactions. With samples spread across 21 years and multiple evolving regulations, these models may have overfit noise rather than capture true patterns.

3. **Formula 1's high variance nature favors simpler models:** Races contain irreducible randomness like sudden rainstorm, a safety car deployment, a mechanical failure that no model can predict from pre-race data. Complex models may overfit these random events in training data, while Linear Regression's simplicity provides robustness.

Limitations and Future Directions

Our analysis has several limitations.

1. We excluded telemetry and in-race data, which limits predictions to pre-race information only. Incorporating real-time data like current lap times, tire degradation, fuel loads could substantially improve accuracy for in-race predictions.
2. Our temporal split i.e. training on 1994-2015, testing on 2019-2024 spans multiple regulation eras with different car designs, scoring systems, and race formats. This heterogeneity may reduce model accuracy, as patterns from the 1990s may not fully be applicable to modern F1. Limiting analysis to the hybrid era (2014-present) might yield better performance at the cost of less training data.
3. Our regression model treats all positions equivalently, meaning an error predicting the winner (P1 vs P3) is weighted the same as an error predicting backmarkers (P15 vs P17). In reality, podium positions matter more.
4. Additionally, our model predicts each driver independently, not accounting for how one driver's crash promotes others. These simplifications prioritize model interpretability and computational efficiency over perfect accuracy

Future work could explore additional features like weather forecasts, tire compound choices, engine penalty grid drops, and driver-specific circuit telemetry data.

Conclusion

This project successfully addressed both research questions. Regarding which pre-race factors are most predictive: grid position emerged as the dominant predictor, explaining 36% of outcome variance alone. Constructor performance i.e. the team/car quality contributes nearly as much, while recent driver form provides moderate additional value. Circuit-specific experience and championship points add minimal predictive information beyond these core factors.

Regarding model performance: Linear Regression achieved the best results with test MAE of 3.320 positions and R^2 of 0.45, slightly outperforming Random Forest, XGBoost, and ensemble methods. This stems from the predominantly linear relationships in F1 data and the robustness of simpler models against high-variance race outcomes. The improvement over a grid-position-only linear regression confirms that engineered features capturing team performance and driver form add meaningful value.

Our analysis demonstrates that Formula 1 race outcomes can be predicted to only a certain extent from pre-race information, with grid position and constructor performance accounting for the majority of this predictability. The remaining variance reflects the sport's inherent unpredictability that includes mechanical failures, weather changes, strategic decisions, and driver mistakes that cannot be anticipated from historical data alone.

END