Potential Models
https://www.journalijar.com/uploads/2022/08/632449eb530fa_IJAR-40641.pdf
https://www.frontiersin.org/journals/artificial-intelligence/articles/10.3389/frai.2025.1673148/full
https://pmc.ncbi.nlm.nih.gov/articles/PMC10660124/pdf/jqas-19-4-jqas-2022-0021.pdf

1. **Regression Model: Predicting Final Race Position based on Qualifying & Other Features**
    a. Linear regression where the dependent variable is the final race position (could transform into something like finishing rank in points or improvement from qualifying). Independent variables include obviously qualifying position, circuit characteristics (overtaking difficulty), weather, tire strategy, number of pit stops, etc.
    b. Improvements:
        i. Original paper shows averages and differences; can estimate effect sizes (qualifying 1 grid spot ahead gives you X positions better on average)
        ii. Include more features as original was limited to small set of drivers and only 3 seasons
        iii. Cross-validation, goodness of fit (chi-square), for capturing driver/team random effects
    c. Extensions:
        i. Transform the target (finishing <= 10, podium = yes or no)
        ii. Use ranking/regression techniques handling ordinal outcomes
2. **Classification Model: Predicting Podium, top 5, top 10, based on Qualifying and Features**
    a. Classification tasks such as: features from qualifying (qualifying position, sector times, driver/team, circuit features, weather) predict if the driver will finish on podium, top 5, or top 10 (yes/no). Using a classification algorithm (logistics regression, random forest, gradient boosting)
    b. Improvements:
        i. Uses more granular data (sector times, tire used/pit stops, tire degradation per lap, etc.)
        ii. Performance metrics (accuracy, precision/recall, AUC) rather than descriptive counts
        iii. Feature importance (SHAP) to see how qualifying position is compared to other features
        iv. Class imbalance will be an issues due to few podiums relative to all races
        v. Certain features like tire strategy could require cleaning
3. **Time Series/Sequential Model: Predicting Race Progression or Overtake Likelihood**
    a. Using time/telemetry data to build a sequential model (recurrent neural network) or Markov model, to predict the position changes over the race (did driver move up or down from first to last lap). Features include starting grid position, lap times, pit stop lap numbers, tie compounds/strategy, weather/track state, overtaking difficulty of circuit

b. Improvements:
    i. Linking qualifying to progression to final, instead of just qualifying position to final position
    ii. Using more telemetry features
    iii. Could potentially simulate what would happen if a driver starts from p10 instead of p5, or if they pitted earlier in the race, which enables scenario analysis
c. Challenges
    i. Data volume and complexity since telemetry can be large
    ii. Define clearly what it is that is being predicted or simulated (finishing position, number of overtakes, positions gained or lost)
    iii. Deep learning required

4. **Causal Inference Model: Estimating the Causal Effect of Qualifying Position on Final Outcome**
    a. Use causal inference techniques (matching, instrumental variables, regression with confounder controls) to estimate how much improvement in qualifying causes better finishing position, rather than just being correlated.
        i. Treating pole position (qualifying first) vs two grid spots behind as a treatment, control for driver/team/circuit difficulty
    b. Steps
        i. Defining treatment (qualifying position = top 3 vs > 3)
        ii. Collect covariates: driver ability, team ability, circuit overtaking difficulty index, weather, previous driver/team form
        iii. Use matching or regression with fixed)effects (driver, team) to control
    c. Benefits
        i. More rigorous and can answer strategy questions as well (Would improving qualifying position by one spot be likely improve outcomes by X)

Multinomial logistic model
Probability
Dependent Variable:
Class 1**:** 1-10 → y = 1
Class 2**:** 11-20 → y = 0

Predictors:

Tier 1: Directly from qualifying and race results

- Qualifying position (will affect probability quite a bit but is important)
- Grid Penalty (yes or no) - affects qualifying start position
- Race Position - ending race position
- Circuit - extra detail, can use to find overtaking/passing difficulty index

- Year/season - every couple years regulations change so cars are built differently

Tier 2: Race-level variables (laps and weather)

- Avg lap time
- Fastest lap time
- Number of pit stops
- First pit lap
- Tire compounds
- Weather conditions
- Safety car/virtual safety car occurrences
- DNF indicator

Tier 3: Derived predictors

- Positions gained/lost
- Team strength index (TeamPointsPrevRace) - team points or constructor rank prior to race,
- Driver form (AvgFinishPrev3) - mean finish in last 3 races
- Circuit overtaking index (OvertakeScore) - externally assigned metric
- Grid position (Front/Mid/Back) grouping of qualifying position
- Race Length (NumLaps) - number of laps
- Sprint Weekend - if it is a sprint weekend or not
- Rain Start (WetStart) Binary - was race wet at start, helps judge overtaking and luck factor

Overall some most important predictors include

- Qualifying position
- Driver and their current form
- team/constructor and their current form/strength
- Circuit overtaking index
- Number PitStops
- Tire compounds used
- Weather, wet races can cause more safety cars causing a lot of teams' and drivers' strategy to change mid race where decisions have to be made within seconds, adding a luck factor especially in rainy and wet weather
- Safety car laps/ race interruptions

Evaluation Metrics

- Accuracy
- AUC/ROC curve - class separation ability
- Precision/Recall
- R^2 - explanatory power
- Odds Ratio - Ex: one position higher in qualifying increases odds of finishing in top 10 by x%

$$\text{logit}(P(\text{Top10} = 1)) = \beta_0 + \beta_1(\text{QualiPos}) + \beta_2(\text{DriverForm}) + \beta_3(\text{TeamStrength}) + \beta_4(\text{OvertakeIndex}) + \beta_5(\text{PitStops}$$

The model estimates how different predictors (qualifying position, driver form, etc.) change the log-odds of finishing in the top 10

Interpretation
Coefficient (β): Direction and strength of each effect.
p-value: Whether that effect is statistically significant.
exp(β): Odds ratio (how much a one-unit change multiplies the odds of finishing top 10).
Pseudo-R^2 / AUC: Model fit and predictive performance.

Fit the logistic model based on all predictors

GOAL
The goal of this model is to predict whether a Formula 1 driver will finish inside the **top 10 positions** (Class 1) or outside the top 10 (Class 0) in a given race

The current version of the code fits a logistic regression model on all available data to estimate coefficients for each predictor

Binary logistic regression model trained on full-season data (e.g., 2024) pulled directly from the **FastF1 API**, which provides official timing, weather, and event data.

1. **Data Acquisition**
    a. The script first enables FastF1's local cache and iterates over all Grand Prix rounds in a selected season
    b. For each race it loads
        i. Results - driver name, team, grid position, finishing position, and points
        ii. Lap data - to compute average and fastest lap times, number of pit stops, and first pit lap
        iii. Weather data - average air temperature and humidity
2. **Derived Features**
    a. TeamStrength - rolling average of constructor points before the current race (team competitiveness)
    b. DriverFrom - 3-race rolling average of previous finishing positions (driver momentum)
    c. PosGainLoss - change between qualifying and final positions, indicating overtaking or race loss
    d. GridGroup - binned grid position (Front 1-5, Mid 6-13, Back 14-20).
    e. Top10 - binary target variable: 1 = finished P1–P10, 0 = P11–P20

Placeholder variables such as `OvertakeIndex`, `TyreCompounds`, and `WetStart` were included for future integration of external metrics

3. **Cleaning and Preprocessing**
    a. Dropping records with incomplete lap data
    b. Filling empty pit stop counts or weather fields with zero where appropriate
    c. Predictors are divided into
        i. Numeric Features - e.g. `QualifyingPosition`, `AvgLapTime`, `TeamStrength`, etc.
        ii. Binary Flags - e.g. `DNF`, `SprintWeekend` etc
        iii. Categorical Features - `Driver`, `Team`, `Circuit`, etc., which are one-hot encoded

A **Logistic Regression** classifier is wrapped inside a scikit-learn **Pipeline**

- The `class_weight="balanced"` option corrects for possible imbalance between top-10 and non-top-10 outcomes
- The pipeline allows fully automated preprocessing and fitting in one call (`model.fit(X, y)`).

Output Interpretation

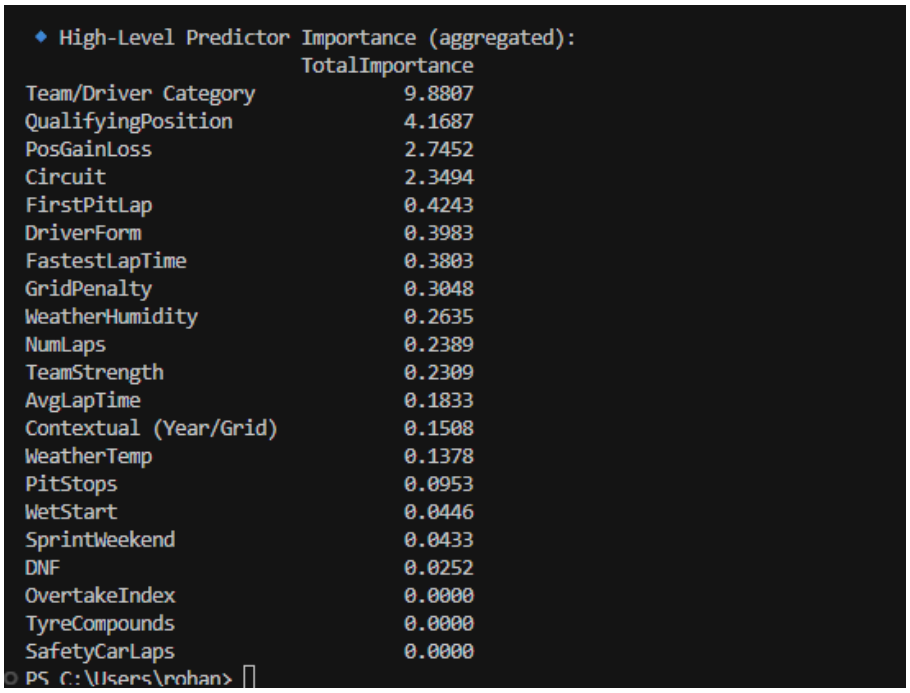
1. **Fine-Grained Coefficients**

```
✅ Logistic regression model fitted successfully!

Top 15 Most Influential Predictors:
 QualifyingPosition   -4.168727
PosGainLoss           2.745227
Driver_STR           -0.782467
Driver_TSU            0.679935
Driver_NOR           -0.628801
Team_Ferrari          0.603899
Team_Alpine          -0.574904
Team_McLaren         -0.572953
Team_Kick Sauber      0.468924
Driver_BOT            0.457527
FirstPitLap           0.424308
Driver_LEC            0.412141
Team_Aston Martin     0.402878
DriverForm            0.398321
Driver_RUS            0.386915
dtype: float64
```

1. **QualifyingPosition:** -4.17, Strong negative impact — starting further back sharply lowers odds of top-10.
2. **PosGainLoss:** +2.75, Drivers gaining positions during race strongly increase top-10 likelihood.
3. **Driver_STR, Team_Ferrari:** ±0.6–0.8, Encoded effects of specific drivers/teams (e.g. strong performers)
4. **DriverForm**: +0.39, Consistently good recent results increase probability.
5. **FirstPitLap:** +0.42, Later first stops correlate slightly with success, likely due to better tyre management

Negative coefficients indicate that increasing the variable reduces the chance of finishing in the top 10 (e.g., higher grid number = worse), while positive coefficients improve it

2. Aggregated Feature Importance



The **aggregated importance** table shows each variable's overall contribution:

1. **Team/Driver Category:** 9.88, Combined effect of who drives which car — largest overall influence
2. **QualifyingPosition:** 4.17, Starting grid position strongly predicts race result.
3. **PosGainLoss:** 2.75, Captures real race-day performance.
4. **Circuit:** 2.35, Track-specific characteristics affect outcomes.
5. **DriverForm:** 0.40, Rolling momentum has measurable predictive power
6. **TeamStrength:** 0.24, Team's prior competitiveness contributes meaningfully
7. **Weather & Strategy:** <0.3, Minor influence given small variance across races

These magnitudes are relative measures of model weight — **higher numbers mean stronger overall influence**, not literal probabilities.

Notes

- **Team/Driver category** is when the model encodes "Driver" and "Team", creating dummy variables like: Driver_VER, Driver_NOR, Driver_LEC, Team_Ferrari, Team_Red Bull, …
    - Captures fixed effects — the consistent advantage or disadvantage associated with a particular driver/team, averaged over the dataset
    - Driver_VER = 1 → strong positive coefficient because Verstappen almost always finishes top 10

    - Team_Red Bull = 1 → positive coefficient because Red Bull cars are dominant
    - Team_Haas = -1 → negative coefficient because Haas rarely scores points
- The aggregated value ("Team/Driver Category = 9.88") is large because it's summing the influence of *all* those one-hot column
- **Team Strength** and **Driver Form** are numeric derived variables
    - Team Strength: Quantifies **how well the team has been performing up to this race**, without assuming identity dominance. If a historically weak team (like Williams) starts improving across races, their `TeamStrength` rises
    - Driver Form: Represents **driver consistency and recent performance trend**

Seems redundant but, **team/driver category** captures the *baseline level* of performance - Verstappen/Red Bull will always be strong, Williams always weaker (Static) - Analogy: Who they are

TeamStrength captures the *current competitive state* of the car – form of the constructor (team) in recent races (dynamic) - Analogy: How fast the car is right now

DriverForm captures the *recent consistency* of the driver – how they've been performing in the last few races (Dynamic) - Analogy: Momentum or confidence

If you only use **Team/Driver Category**, the model assumes every season is identical — it can't adapt to form shifts.
If you only use **TeamStrength/DriverForm**, the model loses the fundamental *hierarchy* (e.g., Red Bull > Alpine, Verstappen > Sargeant)

Why Some Predictors Are Low

- `OvertakeIndex` and `TyreCompounds` currently show near 0 importance because placeholder zeros were used.
  Once real overtaking difficulty and tyre-strategy data are integrated, these will become more influential.
- Weather and sprint indicators contribute modestly due to limited variability or sample size in one season.

Summary:

1. Driver and Team Quality/Strength: Dominates overall as engineering plus talent adds a lot in terms of overall finishing race position
2. Qualifying performance: Is the biggest determinant of finishing position
3. Race Execution: position gains/losses, pit stop timing, overall strategy adds secondary predictive power
4. Form and Strength Metrics: Successfully quantify momentum and team pace from past couple races
5. Environmental and Strategic Variables: Play smaller roles but will matter more once richer data (overtaking indices, tyre choices, rain events) are added

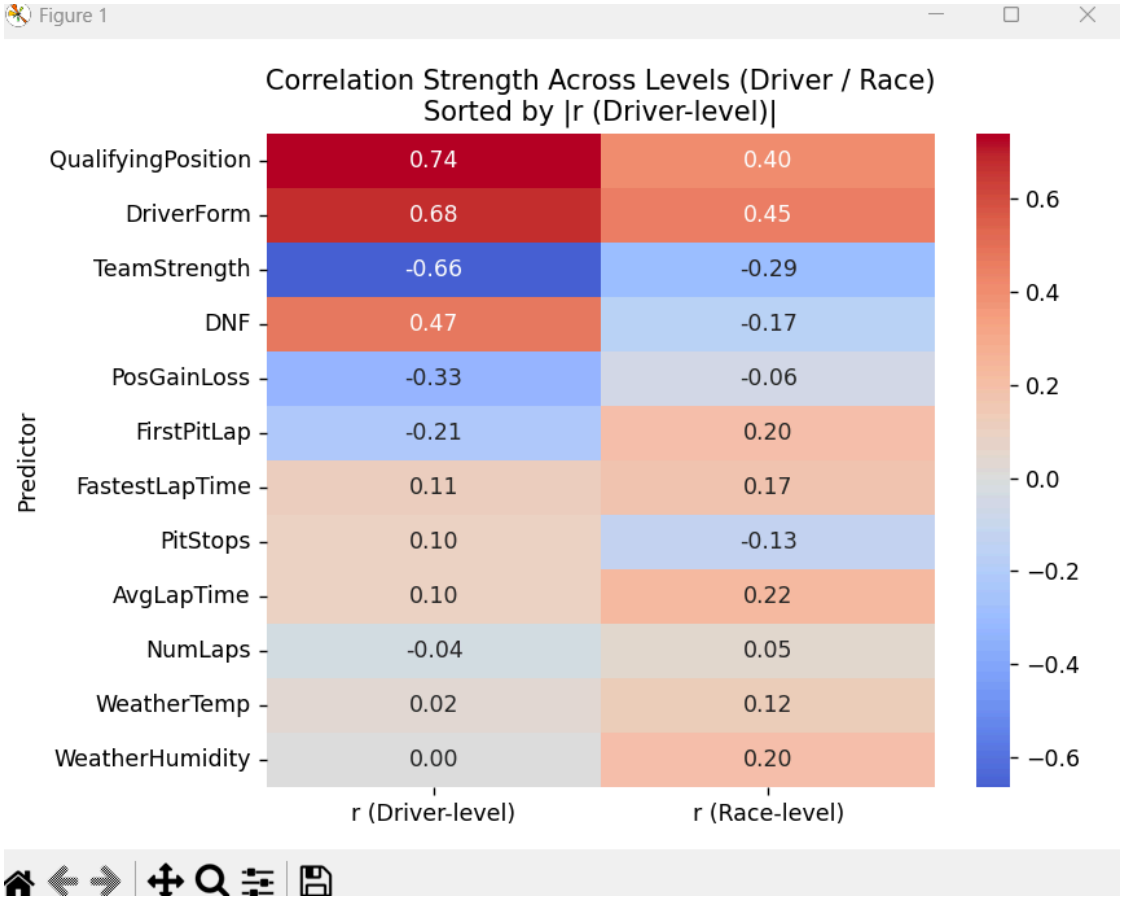
Put data in a way we can fit too it

Race Level (Level 1) vs. Driver-Race Level (Level 2) R and p predictor coefficients

✅ Total driver-race records loaded: 359
✅ Cleaned dataset shape: (359, 24)
📊 Circuits loaded: 18

=== 📊 Correlation Comparison: Driver-level vs Race-level ===

Predictor	r (Driver-level)	p (Driver-level)	r (Race-level)	p (Race-level)
QualifyingPosition	0.741	0.000	0.404	0.096
DriverForm	0.677	0.000	0.450	0.070
TeamStrength	-0.663	0.000	-0.290	0.243
DNF	0.474	0.000	-0.169	0.503
PosGainLoss	-0.332	0.000	-0.059	0.817
FirstPitLap	-0.211	0.000	0.196	0.436
FastestLapTime	0.105	0.058	0.173	0.493
PitStops	0.101	0.070	-0.127	0.616
AvgLapTime	0.097	0.081	0.225	0.370
NumLaps	-0.036	0.518	0.049	0.846
WeatherTemp	0.024	0.663	0.124	0.624
WeatherHumidity	0.001	0.991	0.197	0.434



FastF1 retrieves official timing data from the **FIA Formula 1 Timing API**, which sometimes omits certain races due to one of the following:

- No timing data published
- Corrupted or partial session data
    - Some Grand Prix (Miami, China, Qatar, Austin, São Paulo, Austria, Belgium), occasionally throw internal FastF1 errors — missing telemetry or mismatched laps cause them to be skipped.

In 2024, out of 24 scheduled races, **only 18 had full session data** accessible in the FastF1 timing archives, which is interesting as for my data dashboard that I did in the summer I checked and was able to pull up these “missing” races so that’s what I am working on now.

<u>Level Explanation:</u>

Driver Level

- Unit of observation is one driver’s race entry
- Each row = 1 driver in one race
- Insight example: How does a driver’s qualifying position or form relate to their finishing position?
- keeps individual driver variation intact → many observations (~350 for 18 races × ~20 drivers)

Race-Level

- Unit of observation is one Grand Prix average
- Each row = 1 race (averaged across all drivers)
- Insight example: Across different races how do overall conditions or averages influence finishing outcomes
- Aggregates to only 1 observation per race so only 18 total data points

Overall Difference between the levels:

- **Sample Size (n)**
  - Driver Level: Large (hundreds of driver-race pairs)
  - Race-Level: Small (18 races)
- **Variance**
  - Driver Level: Captures within-race differences (how drivers differ within a GP)
  - Race-Level: Captures between-race differences (how GPs differ overall)
- **Interpretation of r-value**
  - Driver Level: Within a race season, drivers who start higher or have stronger form tend to finish higher
  - Race-Level: Across all races, events with certain averages (e.g., lower humidity, fewer pit stops) tend to produce different finishing outcomes.
- **Susceptible to noise from**
  - Driver Level: Driver skill differences, team strategies, random DNFs
  - Race-Level: Weather, track layout, aggregate randomness (fewer points → more unstable r-values)
- **Correlation Strength**
  - Driver Level: Usually higher (more data + direct relationships)
  - Race-Level: Usually lower (averaging dilutes differences)

Results:

- **Qualifying Position**
  - Driver Level ($r = 0.741$, $p = 0.000$) Race Level ($r = 0.404$, $p = 0.096$)
  - Highly significant at driver-level: starting grid strongly predicts finishing outcome. Moderate and weaker significance at race-level ($p \approx 0.10$), since averaging across all drivers per GP dilutes within-race effects.
- **Driver Form**
  - Driver Level ($r = 0.677$, $p = 0.000$) Race Level ($r = 0.450$, $p = 0.078$)
  - Strong and significant at driver-level — consistent recent form improves finishing position. At race-level, the effect is smaller and only marginally significant ($p \approx 0.08$).
- **Team Strength**
  - Driver Level ($r = -0.656$, $p = 0.000$) Race Level ($r = -0.290$, $p = 0.243$)
  - Strong negative and highly significant at driver-level — stronger teams (higher prior points) tend to finish better (lower positions). At race-level, it weakens and becomes statistically non-significant ($p > 0.05$)
- **DNF**

    - Driver Level (r = 0.474, p= 0.000) Race Level (r= -0.169, p= 0.503)
    - Moderate, because DNFs are rare and not simply “out of points.
- **PosGainLoss**
    - Driver Level (r = –0.332, p= 0.000) Race Level (r= –0.059, p= 0.817)
    - Significant at driver-level — drivers gaining more places relative to qualifying finish higher. No significance at race-level.
- **FirstPitLap**
    - Driver Level (r = –0.211, p= 0.000) Race Level (r= 0.196, p= 0.436)
    - Weak but significant at driver-level — earlier pit stops often indicate worse races (shorter stints). At race-level, no significance
- **FastestLapTime**
    - Driver Level (r = 0.215, p= 0.000) Race Level (r= 0.173, p= 0.493)
    - Weak but significant at driver-level — slightly faster drivers tend to finish higher, though not a major factor. Not significant at race-level
- **PitStops**
    - Driver Level (r =0.101, p= 0.070) Race Level (r= –0.127, p= 0.616)
    - Marginal significance at driver-level (p≈0.07). Drivers making more pit stops tend to perform slightly worse, but the relationship is weak and vanishes at race-level
- **AvgLapTime**
    - Driver Level (r =0.097, p= 0.081) Race Level (r= 0.225.127, p= 0.370)
    - Weak and marginal (p≈0.08) at driver-level — faster overall pace correlates with slightly better results, but not statistically strong. No race-level significance
- **NumLaps**
    - Driver Level (r = –0.036, p= 0.518) Race Level (r= 0.049, p= 0.846)
    - Non-significant in both — race distance does not explain finishing order
- **WeatherTemp**
    - Driver Level (r =0.024, p= 0.663) Race Level (r= 0.124, p= 0.624)
    - Insignificant at both levels — minor temperature variation across GPs doesn’t affect final outcomes
- **WeatherHumidity**
    - Driver Level (r =0.001, p= 0.991) Race Level (r= 0.197, p= 0.434)
    - No significance; environmental humidity does not predict results

Summary:

Driver-Level Analysis

- **r-values** are stronger (typically |r| > 0.3) and **p-values** < 0.001 for most performance metrics.
- Indicates **within-race differences** (like qualifying, team strength, or driver form) **strongly predict individual finishing positions**.

- The large sample (359 driver-race records) gives more statistical power and stability.

Race-Level Analysis

- **r-values** are weaker and most **p-values > 0.05**, meaning fewer significant relationships.
- Aggregating to race averages **smooths out individual variation**, masking key performance relationships.
- Essentially measures how different races themselves (as events) differ, not how drivers differ *within* them.

Driver-race level matrix

=== Driver-Race Feature Matrix (with Descriptive Columns) ===

Driver	Team	RacePosition	QualifyingPosition	PosGainLoss	AvgLapTime	FastestLapTime	PitStops	FirstPitLap	WeatherTemp	WeatherHumidity	TeamStrength	DriverForm	NumLaps	DNF
VER	Red Bull Racing	1.0	1.0	0.0	96.574421	92.608	2.0	18.0	18.227389	48.821656	26.0	NaN	57	0
PER	Red Bull Racing	2.0	5.0	3.0	96.968404	94.364	2.0	13.0	18.227389	48.821656	26.0	NaN	57	0
SAI	Ferrari	3.0	4.0	1.0	97.014947	94.507	2.0	15.0	18.227389	48.821656	15.0	NaN	57	0
LEC	Ferrari	4.0	2.0	-2.0	97.270368	94.090	2.0	12.0	18.227389	48.821656	15.0	NaN	57	0
RUS	Mercedes	5.0	3.0	-2.0	97.395263	95.065	2.0	12.0	18.227389	48.821656	10.0	NaN	57	0
NOR	McLaren	6.0	7.0	1.0	97.424561	94.476	2.0	14.0	18.227389	48.821656	8.0	NaN	57	0
HAM	Mercedes	7.0	9.0	2.0	97.457298	94.722	2.0	13.0	18.227389	48.821656	10.0	NaN	57	0
PIA	McLaren	8.0	8.0	0.0	97.558316	94.774	2.0	13.0	18.227389	48.821656	8.0	NaN	57	0
ALO	Aston Martin	9.0	6.0	-3.0	97.888228	94.199	2.0	16.0	18.227389	48.821656	2.0	NaN	57	0
STR	Aston Martin	10.0	12.0	2.0	98.209789	95.632	2.0	10.0	18.227389	48.821656	2.0	NaN	57	0
ZHO	Kick Sauber	11.0	17.0	6.0	98.419661	95.458	2.0	10.0	18.227389	48.821656	0.0	NaN	57	0
MAG	Haas F1 Team	12.0	15.0	3.0	98.447464	95.570	2.0	12.0	18.227389	48.821656	0.0	NaN	57	0
RIC	RB	13.0	14.0	1.0	98.458929	95.163	2.0	14.0	18.227389	48.821656	0.0	NaN	57	0
TSU	RB	14.0	11.0	-3.0	98.468286	95.833	2.0	15.0	18.227389	48.821656	0.0	NaN	57	0
ALB	Williams	15.0	13.0	-2.0	98.511214	95.723	2.0	16.0	18.227389	48.821656	0.0	NaN	57	0
HUL	Haas F1 Team	16.0	10.0	-6.0	98.613821	94.834	3.0	2.0	18.227389	48.821656	0.0	NaN	57	0
OCO	Alpine	17.0	19.0	2.0	98.860571	96.226	2.0	11.0	18.227389	48.821656	0.0	NaN	57	0
GAS	Alpine	18.0	20.0	2.0	98.877839	94.805	3.0	13.0	18.227389	48.821656	0.0	NaN	57	0
BOT	Kick Sauber	19.0	16.0	-3.0	98.503745	96.202	2.0	13.0	18.227389	48.821656	0.0	NaN	57	0
SAR	Williams	20.0	18.0	-2.0	99.327352	94.735	3.0	11.0	18.227389	48.821656	0.0	NaN	57	0

<u>Description</u>
Each **row = one driver's performance in a single race**
Each **column = a feature (predictor variable)** summarizing that driver's weekend

Certain Predictors Missing Currently

- **OvertakeIndex:** Metric of net overtakes during the race
    - Needs per-lap position deltas for each driver — can be computed by comparing each lap's position with previous lap's position (`laps["Position"]` differences).
- **SafetyCarLaps:** Laps run under Safety Car conditions
    - Requires parsing `TrackStatus` from session telemetry (`TrackStatus == 4` or `5`)
- **VirtualSafetyCarLaps:** Laps run under VSC
    - Same source — `TrackStatus == 6` or similar codes
- **WetStart:** Whether race started on wet tires
    - Only appears in weather or compound data (`laps["Compound"]`) — if first stint starts on `Intermediate` or `Wet`
- **GridPenalty:** Starting grid penalty from quali
    - FastF1 does not expose this directly — you'd have to cross-check official FIA start grid vs. quali results (external source)
- **TyreStrategy/Aggressiveness:** Aggressiveness of pit/tire strategy
    - Derived from compound changes per stint — needs lap-level tire compound data
- **OvercutEffectiveness/UndercutLoss:** Timing impact around pit stop
    - Requires sector timing data and per-lap position deltas — advanced-level metric
- **WeatherVariance:** Change in weather during race
    - Derived from session.weather_data over time (max–min difference)

1. **Logistic Model Building and Training**

Data Collection:

- **DriverForm** (rolling 3-race average finishing position, seeded with last 3 races of 2023)
- **TeamStrength** (average points per race from 2023, then updated per race in 2024)
- **DriverDNF_Rate** and **TeamDNF_Rate** (rolling reliability rates)
- **OvertakeIndex** (numerical measure of overtaking difficulty by circuit)
- Weather metrics (mean, variance of air temperature and humidity)
- **QualifyingPosition**, **NumLaps**, and **SprintWeekend** flags

Only Pre-race predictors, not using post-race predictors

Binary Classification

$$Top10 = \begin{cases} 1 & \text{if RacePosition} \leq 10 \\ 0 & \text{if RacePosition} > 10 \end{cases}$$

Predicting probability of finishing in top 10

Data Splitting:

- **75% of the data** (races and drivers) was used to *train* the model.
- **25% of the data** was used to *test and evaluate* it.
- `stratify=y` ensures both sets have the same ratio of top-10 and non-top-10 finishes

Although there are 24 races in 2024, the cleaned dataset contains **359 rows instead of 480** because FastF1 occasionally **omits drivers who DNFed early, did not start, or had missing telemetry/weather data**. Additionally, a few circuits such as Las Vegas and Qatar have incomplete records in the FastF1 database.

A 75/25 train-test split gives us about 90–120 samples in the test set, which is sufficient for evaluating a logistic regression model.
However, for better reliability, we could also use **5-fold cross-validation** or a **chronological split by race rounds** to simulate predicting unseen future races

Model Fitting

```
logit = LogisticRegression(max_iter=1000)
logit.fit(X_train, y_train)
```

This learns how each predictor (qualifying position, driver form, etc.) influences the *log-odds* of finishing in the top 10

2. Model Evaluation

Classification Metrics

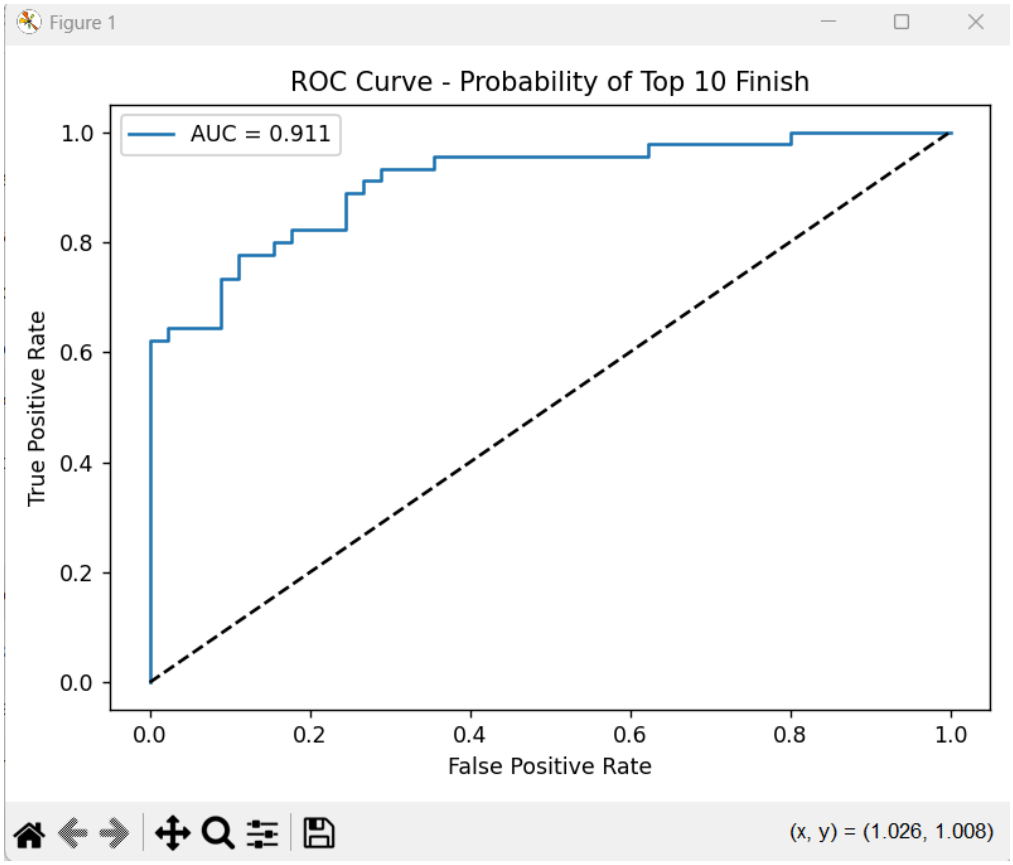
```
✅ Loaded previous season (2023) context for seeding.
✅ Added reliability features: DriverDNF_Rate, TeamDNF_Rate.
✅ Cleaned dataset shape: (359, 24)

=== 🏁 Logistic Model Performance: Top 10 Finish ===
              precision    recall  f1-score   support

           0       0.77      0.91      0.84        45
           1       0.89      0.73      0.80        45

    accuracy                           0.82        90
   macro avg       0.83      0.82      0.82        90
weighted avg       0.83      0.82      0.82        90

Confusion Matrix:
 [[41  4]
 [12 33]]
ROC AUC: 0.911
```

Metric	Meaning
Precision (Top10=1)	Of all drivers predicted to finish top 10, 89% actually did.
Recall (Top10=1)	Of all actual top 10 finishes, 73% were correctly identified.
F1-score (Top10=1)	Harmonic mean of precision and recall (balances both)
Accuracy (0.82)	Model correctly classified 82% of all drivers
ROC AUC (0.911)	Excellent discrimination ability — near-perfect ranking between classes

Confusion Matrix

Term	Value	Meaning
True Negatives (TN)	41	The model correctly predicted 41 drivers *would not* finish in the top 10, and they didn't.
False Positives (FP)	4	The model incorrectly predicted 4 drivers *would* finish top 10, but they didn't.
False Negatives (FN)	12	The model predicted 12 drivers *would not* finish top 10, but they actually did.
True Positives (TP)	33	The model correctly predicted 33 drivers *would* finish in the top 10, and they did.

ROC Curve



- True Positive Rate (recall) vs. False Positive Rate.
- The dashed diagonal = random guessing baseline.
- The blue curve = model performance

AUC = 0.911 means the model correctly ranks a random "Top 10" driver higher than a "Non–Top 10" driver 91% of the time

Interpretation

- The model **does a great job** identifying top-10 finishers (few false positives, high precision).
- It sometimes **misses** a few top-10 finishers (higher false negatives) — meaning some mid-tier drivers that sneak into the top 10 weren't predicted to.
- The **recall of 0.73** suggests it's slightly conservative: it's better at saying *"this driver won't finish top 10"* than *"this driver will"*.
- **Overall accuracy = 82%** and **ROC AUC = 0.911**

Coefficients + Race-Level Predictions

```
=== Logistic Regression Coefficients ===
         Predictor  Coefficient
      TeamStrength       0.8774
      TeamDNF_Rate       0.1740
    DriverDNF_Rate       0.1293
WeatherHumidityVar       0.0573
   WeatherHumidity       0.0054
     SprintWeekend       0.0000
           NumLaps      -0.0278
     OvertakeIndex      -0.0625
       WeatherTemp      -0.1684
    WeatherTempVar      -0.1840
        DriverForm      -0.5480
QualifyingPosition      -1.4609

Sample predicted probabilities (Circuit: Bahrain Grand Prix) — sorted by finishing position:
 RacePosition Driver           Team  QualifyingPosition  Predicted_Prob_Top10  Top10
        1.000    VER Red Bull Racing               1.000                 0.996      1
        2.000    PER Red Bull Racing               5.000                 0.971      1
        3.000    SAI         Ferrari               4.000                 0.948      1
        4.000    LEC         Ferrari               2.000                 0.974      1
        5.000    RUS        Mercedes               3.000                 0.935      1
        6.000    NOR         McLaren               7.000                 0.746      1
        7.000    HAM        Mercedes               9.000                 0.837      1
        8.000    PIA         McLaren               8.000                 0.807      1
        9.000    ALO    Aston Martin               6.000                 0.772      1
       10.000    STR    Aston Martin              12.000                 0.310      1
       11.000    ZHO     Kick Sauber              17.000                 0.052      0
       12.000    MAG    Haas F1 Team              15.000                 0.067      0
       13.000    RIC              RB              14.000                 0.184      0
       14.000    TSU              RB              11.000                 0.254      0
       15.000    ALB        Williams              13.000                 0.204      0
       16.000    HUL    Haas F1 Team              10.000                 0.245      0
       17.000    OCO          Alpine              19.000                 0.066      0
       18.000    GAS          Alpine              20.000                 0.055      0
       19.000    BOT     Kick Sauber              16.000                 0.055      0
       20.000    SAR        Williams              18.000                 0.038      0
```

These show the direction and strength of influence of each predictor on the probability of finishing in the top 10

- The model ranked probabilities nearly perfectly by finishing order.
- Top teams and good qualifiers have predicted probabilities near 1.0.
- Backmarkers like Williams, Kick Sauber, and Haas have probabilities below 0.25

Predictor/Coefficient	**Meaning**

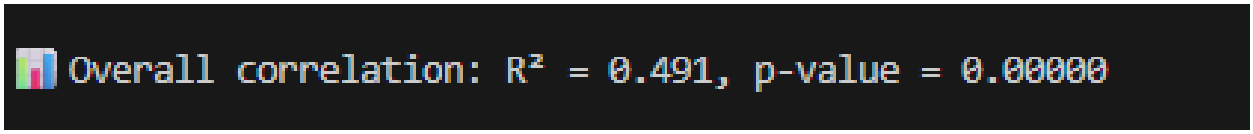
QualifyingPosition (-1.4069)	Each additional grid position farther back multiplies odds of a top-10 finish by 0.245 (≈75% lower odds).
TeamStrength (+0.8774)	For every 1-point increase in average team strength, odds of finishing top-10 increase by 2.40 (exp(0.8774)).
DriverForm (-0.5480)	A worse average finishing position in past races reduces odds of a top-10 by 0.58× (≈42% lower)
OvertakeIndex (-0.0625)	A worse average finishing position in past races reduces odds of a top-10 by 0.58× (≈42% lower)
WeatherTemp (-0.1684)	Hotter conditions slightly reduce odds (≈15% lower odds per °C, depending on scale).
Team/DriverDNF_Rate (+)	Higher reliability increases odds of finishing top-10.
WeatherHumidityVar (+0.0573)	Greater humidity variance slightly increases odds (≈6%

Dummy variable for predictor qualifying position:
For 20 positions, create 19 dummy columns

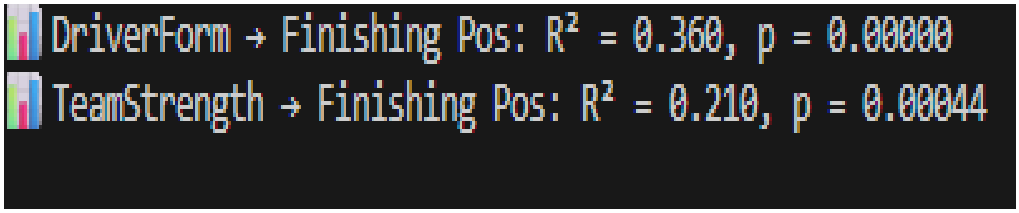
```python
# Convert qualifying position to categorical
model_df["QualifyingPosition"] = model_df["QualifyingPosition"].astype("category")

# Create dummy variables, dropping position 1 as the baseline
qual_dummies = pd.get_dummies(model_df["QualifyingPosition"], prefix="QualPos", drop_first=True)

# Combine with the rest of your predictors
X = pd.concat([model_df[other_features], qual_dummies], axis=1)
y = model_df["Top10"]
```

Each dummy variable shows the **marginal change in log-odds** of finishing top 10 *compared to the baseline*

R^2 and P values for qual vs race pos:



R^2 and P values for driverform and teamstength vs race pos:



Graph Analysis:

HTML Files include all 24 graphs for all races in order. Includes a hover feature as well to see the driver/team and extra details depending on the graph. Also includes a baseline line that shows perfect prediction where points above and below underperformance or overperformance respectively.

A) Qualifying Position vs. Finishing Position

Baseline line:

a) If Qualifying position perfectly predicts performance, the highest qualifying driver finishes P1, next P2, next P3, etc

Deviations mean:

- **Above the line** → **underperformed** (potentially good qualifying but dropped places in the race causing them to finish worse than expected i.e. qualified 1 but finished 4)
- **Below the line** → **overperformed** (potentially not as good qualifying, but gained places in the race causing them to finish better than expected, i.e. qualified 4 but finished 1)

B) **Driver Form vs Finishing Position**: average points from previous 3 races

Baseline line:

a) If DriverForm perfectly predicts performance, the highest form driver finishes P1, next P2, next P3

So deviations mean:

- **Above the line** → **underperformed** (high form but finished *worse* than expected, i.e. higher position number)

- **Below the line** → **overperformed** (low form but finished *better* than expected, i.e. lower position number).

C) **<u>Team Strength vs Finishing Position</u>**: rolling average team points form season so far

Baseline line:

b) If TeamStrength predicts race results perfectly, the strongest team finishes P1, next strongest P2, etc

Deviations mean:

- **Above the line** → **underperformance** (team strong but finished poorly)
- **Below the line** → **overperformance** (team weak but finished better than expected)