F1 Race Outcome Prediction — Project Proposal

Author: Suhas Bajjuri

Course/Context: BTech CSE (AI) — ML Project

Date: 31 August 2025

---

1) Problem Statement

Predict key outcomes of Formula 1 races using historical and pre-race data. The primary target is **probabilistic race finishing positions** (e.g., probability a driver finishes P1/P2/P3/Top-10/Finishes), with secondary targets such as **podium probability**, **points probability**, and **DNF probability**. The model will produce **calibrated probabilities** and **rankings** for each race weekend.

2) Motivation & Impact

- F1 is an evolving, data-rich sport with changing regulations, tracks, and weather. Accurate forecasts highlight how data and ML interact with real-world strategy.
- Educational value: end-to-end ML pipeline (data ingestion → feature engineering → modeling → evaluation → reporting).
- Potential users: student analysts, motorsport clubs, commentators, and fan analytics communities.

3) Research Questions

1. Given data available **before lights-out** (FP/Qualifying/Weather/Starting grid), how accurately can we predict finishing order and podium/points probabilities?
2. Which features matter most (driver form, team pace, track characteristics, tyre strategy, weather)?
3. Does **time-aware validation** improve generalization to new seasons?

4. How well can we **calibrate** probabilities (Brier score/log loss vs. naive baselines)?

4) Scope & Assumptions

- Focus on **race-day predictions** using data *known before the race starts* to avoid leakage.
- Seasons targeted: **2014 → 2024** (hybrid era) with option to back-fill earlier years for robustness.
- Qualifying-only or mid-race live predictions are stretch goals.

5) Data Sources (to be finalized)

- **Historical results & schedules:** Ergast Developer API; official race calendars.
- **Timing & session data:** Open-source tooling like *FastF1* for session and lap timing where permitted; selected Kaggle F1 datasets for convenience.
- **Weather:** Race-weekend weather summaries (historical) at the venue level.
- **Track metadata:** Circuit layout, length, altitude; overtaking difficulty proxies (e.g., DRS zones).
- **Tyres & strategy (optional):** Compound allocations, pit windows (derived).

Data governance: Keep a clear separation between pre-race and post-race variables. Any feature created from post-race information is prohibited for training to prevent leakage.

6) Target Variables

- **Primary:** Driver finishing position ranking per race (modeled as probabilities over ranks or as a permutation-aware objective).
- **Secondary:** Podium (Top-3), Top-10/points, finish vs. DNF, positions gained/lost from grid.

7) Features (initial design)

Driver/Team form

- Rolling averages over last *k* races for: qualifying delta to teammate, race pace proxies, points scored, DNFs.
- Team constructor form: average finishing pos., reliability rate, pit-stop performance.

Weekend context

- Track-specific factors: circuit type (street/permanent), length, downforce level proxy, historical safety-car rate.
- Weather forecasts: rain probability, temperature range.
- Starting grid & penalties (positions drop).
- Tyre info (if available pre-race): allocated compounds, expected stint lengths.

Session signals (if available)

- FP long-run pace estimates; single-lap pace; degradation indicators (from lap deltas); consistency (variance).

Encoded effects

- Driver, team, and track embeddings or one-hot with target encoding (time-aware).
- Season phase indicator (early/mid/late), sprint weekend flag.

8) Methodology

This project will compare multiple modeling approaches and converge on the best validated option. The plan emphasizes **simplicity → performance → interpretability**.

8.1 Baselines

- **Naive Grid Baseline:** Predict finish order = starting grid.
- **Persistence Form Baseline:** Rank by rolling points/finishing position.
- **Simple Classifier:** Logistic regression for Top-10 vs. not.

8.2 Candidate Models (to be down-selected empirically)

- **Tree Ensembles:** Random Forest, Gradient Boosting, XGBoost/LightGBM/CatBoost for classification (Top-k) and regression (finishing pos.).
- **Learning-to-Rank:** LambdaMART (pairwise) to predict finishing order as a ranking problem.
- **Probabilistic Ranking:** Plackett-Luce / Bradley-Terry models for finishing order likelihoods.
- **Neural Nets:** MLP for tabular features; optional LSTM/Temporal CNN for session/lap sequences (stretch).
- **Reliability (DNF) Submodel:** Binary classifier whose output informs race outcome mixture.

8.3 Probability Calibration

- Platt scaling or isotonic regression on validation sets; assess with **Brier score** and **calibration curves**.

8.4 Validation Strategy (time-aware)

- **Walk-forward validation by race weekend**: Train on seasons up to $t-1$, validate on season t.
- Avoids temporal leakage and mimics deployment on future races.

9) Evaluation Metrics

Ranking quality

- Spearman's ρ / Kendall's τ between predicted and actual finishing orders.
- **NDCG@k** for Top-k accuracy ($k \in \{1,3,10\}$).

Classification quality

- ROC-AUC / PR-AUC for Top-10 and podium; F1 for points class.

Probabilistic quality

- Log Loss, **Brier Score**, Expected Calibration Error; Reliability curves.

Baselines vs. Models

- Report deltas vs. grid baseline and persistence baseline.

10) Experimental Plan

1. **Data Ingestion & Cleaning**
    - Pull seasons; normalize driver/team names across seasons; handle mid-season substitutions; map constructor rebrands.
    - Create event-level dataset per race with strict pre-race feature cutoff.
2. **EDA**
    - Missingness analysis; leakage checks; season drift charts; feature distributions.
3. **Baseline Implementation**
    - Grid and persistence; establish reference metrics.
4. **Model Sweep**
    - Train/evaluate candidate models under walk-forward splits; retain top-2.
5. **Calibration & Ensembling**
    - Calibrate probabilities; optional stacking of ranker + classifier.
6. **Ablation Studies**
    - Remove feature groups (form, grid, weather) to quantify importance.
7. **Interpretability**
    - SHAP/feature importance; per-track and per-driver error analysis.
8. **Packaging**
    - Reproducible pipeline (Makefile/CLI), model card, and final report.

11) Deliverables

- **Code repo** with reproducible environment and CLI (train/predict/evaluate).
- **Processed dataset** (documented schema + data dictionary).
- **Experiments log** with configuration files and metrics.
- **Model card** summarizing intended use, limitations, and fairness considerations.
- **Final report & slide deck** with results, insights, and errors.
- **(Optional) Demo notebook** for a chosen race weekend.

12) Timeline (6 weeks, adjustable)

Week 1: Data sourcing, schema, and pre-race cutoff rules.

Week 2: Feature engineering v1; baselines; initial EDA.

Week 3: Model sweep (tree ensembles, ranker); pick top-2.

Week 4: Calibration, ensembling, and ablations.

Week 5: Interpretability, write-up, and packaging.

Week 6: Polishing, slides, and (optional) demo.

13) Risks & Mitigations

- **Data leakage:** Use strict time cutoffs; automated tests that fail if post-race fields appear.
- **Small sample size per season:** Aggregate across seasons; prefer simple models first; regularization.
- **Regulation changes (concept drift):** Season indicators; walk-forward training; re-train per year.
- **Noisy/biased labels (DNF randomness):** Separate reliability submodel; predict distributions, not single points.

14) Ethical & Responsible AI Considerations

- **Fair use & licensing:** Respect data licenses; credit sources; share only derived features if raw data licensing restricts redistribution.
- **Transparency:** Publish model card; avoid implying certainty; communicate uncertainty.
- **Reproducibility:** Fixed seeds, pinned dependencies, and experiment tracking.

15) Resources Needed

- **Compute:** Laptop with CPU/GPU sufficient for tree ensembles; optional cloud for experiments.
- **Libraries:** Python 3.11+, pandas, numpy, scikit-learn, xgboost/lightgbm/catboost, shap, fastf1 (optional), matplotlib.
- **Tracking:** Weights & Biases or MLflow (optional), or CSV logs.

16) Success Criteria

- Beat **grid baseline** by ≥ **X%** on NDCG@10 and reduce Brier score vs. naive baseline on Top-10/Podium predictions across seasons.
- Deliver calibrated probability outputs and clear interpretability visualizations.
- End-to-end pipeline reproducible from a fresh clone.

17) Stretch Goals (nice-to-have)

- **Live Qualifying Predictor:** Q1→Q3 advancement probabilities.
- **Monte Carlo Championship Simulator:** Aggregate race probabilities across season to forecast WDC/WCC odds.
- **Interactive dashboard:** Per-race probability explorer and what-if scenarios (weather/SC).
