F1 RACE STRATEGY OPTIMIZATION

Data Science Module — Project Proposal

University of Twente

Recommended Topics:

Data Mining (DM) + **Feature Extraction from Time Series (TS)**

February 2026

Table of Contents

Table of Contents 2
1. Executive Summary 4
2. Data Sources & Availability Audit 5
  2.1 Primary Source: FastF1 (Python Library) 5
    Lap-Level Timing Data (session.laps) 5
    Weather Data (session.weather_data) 6
    Track Status & Race Control (session.track_status, session.race_control_messages) 6
    Car Telemetry (session.car_data / lap.get_telemetry()) 7
  2.2 Secondary Source: OpenF1 API 7
  2.3 Tertiary Source: Jolpica/Ergast API (Historical Data) 7
  2.4 Supplementary Data (Manual/Scraped) 8
    Asphalt Abrasiveness & Circuit Characteristics 8
    Fuel Effect Model 8
3. Strategy Factor to Data Mapping 9
4. Topic Selection & Justification 10
  4.1 Topic DM: Data Mining 10
    Regression 10
    Classification 10
    Clustering 10
  4.2 Topic TS: Feature Extraction from Time Series 11
    Filtering & Smoothing 11
    Peak/Event Detection 11
    Dynamic Time Warping (DTW) 11
    Prediction Models 11
  4.3 Why This Combination Works 11
5. System Architecture & Pipeline 12
  5.1 Data Ingestion Layer 12
  5.2 Data Preparation & Feature Engineering Layer 12
  5.3 Modeling Layer 13
  5.4 Strategy Simulation Engine 13
6. Detailed Modeling Approach for Key Factors 14
  6.1 Tyre Degradation Model 14
  6.2 Fuel Correction Model 14
  6.3 Incident Response Model 14

6.4 Weather Transition Handling 14
6.5 Driver Form & Team Strength 15
7. Project Deliverables 16
7.1 Data Pipeline & Processed Dataset 16
7.2 Analytical Models 16
7.3 Strategy Simulator 16
7.4 Visualization & Reporting 16
7.5 Documentation 16
8. Proposed Work Plan 17
9. Technical Stack 18
10. Risks & Mitigations 19
11. References & Resources 19

1. Executive Summary

This project proposes a comprehensive Formula 1 race strategy optimization system that combines real-time telemetry, historical race data, and predictive modelling to recommend optimal pit stop timing, tyre compound selection, and strategic responses to in-race events. The system will ingest data from multiple open-source APIs (FastF1, OpenF1, Jolpica/Ergast) and supplementary datasets (Pirelli circuit characteristics, fuel models) to build a multi-factor simulation engine capable of modelling the complex, interdependent variables that govern F1 race outcomes.

The project directly exercises two Data Science topics from the module: Data Mining (DM) for building classification, regression, and clustering models that predict tyre degradation, lap time evolution, and strategic outcomes; and Feature Extraction from Time Series (TS) for processing high-frequency telemetry signals, detecting safety car events via signal analysis, and extracting degradation curves from noisy lap time sequences.

2. Data Sources & Availability Audit

A critical component of this proposal is grounding every modelled variable in actually available data. The following audit maps each strategic factor to concrete data endpoints, fields, and any gaps that need to be bridged with derived or supplementary data.

2.1 Primary Source: FastF1 (Python Library)

FastF1 is the primary data source, providing structured access to the official F1 timing feed. Data is available from 2018 onwards and is returned as Pandas DataFrames. The library exposes the following key data objects relevant to strategy optimization:

Lap-Level Timing Data (session.laps)

Each lap record contains the following fields directly useful for strategy:

Field	Type	Strategy Relevance	Available
LapTime	Timedelta	Core metric: total lap duration including degradation and fuel effects	✓
Sector1/2/3Time	Timedelta	Sector-level granularity for identifying where time is lost/gained	✓
Compound	String	Tyre compound: SOFT, MEDIUM, HARD, INTERMEDIATE, WET	✓
TyreLife	Float	Number of laps driven on current tyre set (includes previous sessions for used sets)	✓
FreshTyre	Bool	Whether TyreLife was 0 at stint start (new vs. used set)	✓
Stint	Float	Stint number: directly maps to pit stop count per driver	✓
PitInTime / PitOutTime	Timedelta	Exact session timestamps of pit entry/exit; delta = total pit stop duration	✓
SpeedI1/I2/FL/ST	Float	Speed traps at 4 track positions (km/h); proxies for car performance and setup	✓
LapNumber	Float	Sequential lap count; essential for fuel-weight modeling	✓
Position	Float	Driver position at end of lap; for race-state modeling	✓
TrackStatus	String	Concatenated status codes active during the lap (see Incident section)	✓
IsAccurate	Bool	Timing accuracy flag; excludes SC/VSC-affected first laps	✓
Deleted / DeletedReason	Bool/String	Track limits violations; useful for filtering valid representative laps	✓

Weather Data (session.weather_data)

Weather data is sampled approximately once per minute throughout each session and provides:

Field	Type/Unit	Strategy Relevance	Available
AirTemp	°C (float)	Ambient temperature; affects engine cooling and aerodynamic density	✓
TrackTemp	°C (float)	Surface temperature; critical factor for tyre thermal degradation and grip	✓
Humidity	% (float)	Relative humidity; correlates with grip changes and rain probability	✓
Rainfall	Boolean	Binary rain flag; triggers wet-tyre strategy decisions	✓
Pressure	mbar (float)	Barometric pressure; affects downforce and engine performance	✓
WindDirection	Degrees (int)	0°–359°; sector-specific drag/downforce effects	✓
WindSpeed	m/s (float)	Wind magnitude; affects straight-line speed and cornering stability	✓

Track Status & Race Control (session.track_status, session.race_control_messages)

These two data streams provide complete incident coverage:

Code	Status	Strategy Implication
1	Track Clear	Normal racing conditions; green flag
2	Yellow Flag	Localized caution; may affect sector times but not global strategy
4	Safety Car (SC)	Free pit stop window; field bunches up; massive strategic opportunity
5	Red Flag	Session stopped; free tyre change; effectively a reset
6	VSC Deployed	Reduced-speed regime; pit stop time loss reduced by ~10–15 seconds
7	VSC Ending	Transition to green; timing pit stop for VSC ending is critical

Race control messages additionally provide DRS enable/disable status, specific driver investigations, penalties, and the Flag and Scope fields that indicate whether incidents are track-wide, sector-specific, or driver-specific.

Car Telemetry (session.car_data / lap.get_telemetry())

High-frequency telemetry (sampled at approximately 3–4 Hz) is available per car and includes:

- **Speed** (float, km/h): Car speed at each sample point
- **RPM** (int): Engine revolutions; correlated with engine mode and fuel consumption
- **nGear** (int): Current gear; gear shift patterns indicate driver inputs
- **Throttle** (float, 0–100%): Throttle application percentage
- **Brake** (bool): Whether brakes are applied
- **DRS** (int): DRS activation status indicator
- **X, Y, Z** (float): Car position coordinates on track; enables gap calculations

This telemetry is essential for computing derived features such as tyre-slip indicators, cornering speed degradation over a stint, and throttle-application smoothness as a proxy for driver confidence in grip levels.

2.2 Secondary Source: OpenF1 API

OpenF1 provides a REST API with 18 endpoints, available from 2023 onwards. Its key additional data beyond FastF1 includes:

- **Stints endpoint:** Provides compound, tyre_age_at_start, lap_start, and lap_end per stint per driver. This is a cleaner representation of stint boundaries than computing them from FastF1 lap data.
- **Intervals endpoint:** Gap to leader and gap to car ahead, updated frequently during the race. Essential for determining undercut/overcut windows.
- **Pit stops endpoint:** Precise pit stop durations per driver.
- **Location endpoint:** Higher-frequency car position data (X, Y coordinates) at up to ~3.7 Hz.

2.3 Tertiary Source: Jolpica/Ergast API (Historical Data)

The Jolpica API (successor to Ergast) provides historical F1 data from 1950 to present. While it lacks telemetry, it offers structured results data useful for computing historical driver/team performance features:

- **Race results:** Grid position, finish position, points, status (finished, retired, etc.), fastest lap.
- **Qualifying results:** Q1/Q2/Q3 times per driver; proxy for raw pace.
- **Constructor standings:** Season-long team performance for strength modeling.
- **Pit stops:** Lap number and duration per stop per race (available from 2012).
- **Lap times:** Individual lap times per driver per race (available from 1996).

2.4 Supplementary Data (Manual/Scraped)

Some strategy-critical factors are not directly available from any API and must be collected from supplementary sources:

Asphalt Abrasiveness & Circuit Characteristics

Pirelli publishes circuit characteristic infographics before each Grand Prix, rating tracks on a 1–5 scale across multiple dimensions: asphalt abrasiveness, asphalt grip, traction demand, braking severity, lateral forces, tyre stress, downforce level, and track evolution. These ratings are published as images/infographics and are not available as structured data. They must be manually transcribed into a lookup table (approximately 24 circuits per season), which is entirely feasible for a single season and can be extended to multiple seasons with moderate effort.

Additionally, the compound allocation per circuit (which of C1–C6 are nominated as Hard/Medium/Soft) serves as an indirect proxy for expected degradation severity: softer nominations indicate lower-severity circuits, harder nominations indicate more aggressive surfaces.

Fuel Effect Model

F1 cars start with up to 110 kg of fuel and burn approximately 1.8–2.0 kg per lap (circuit-dependent). The fuel-weight effect on lap time is widely estimated at approximately 0.03–0.035 seconds per kg per lap. Since fuel load is not directly reported by any API, it must be modeled as a linear function of lap number. This is standard practice in F1 analytics: starting fuel mass minus (lap number multiplied by estimated burn rate per lap) gives the estimated fuel mass at any point in the race. The resulting lap time improvement from fuel burn can then be subtracted from raw lap times to isolate tyre degradation.

3. Strategy Factor to Data Mapping

The following matrix maps each of the stated project requirements to available data fields, their sources, and the type of processing needed.

Strategy Factor	Data Fields	Source	Processing
Weather Changes	AirTemp, TrackTemp, Humidity, Rainfall, WindSpeed, WindDirection, Pressure	FastF1 weather_data (1-min intervals)	Time-series analysis; rain onset detection; trend features
Asphalt Abrasiveness	Abrasiveness rating (1–5), Grip rating, Tyre stress, Compound allocation	Pirelli infographics (manual transcription); compound = proxy	Static lookup table per circuit; one-hot encode or ordinal
Fuel Effect	LapNumber, TotalLaps, circuit length	Derived: 110kg - (lap × burn rate); burn rate from circuit data	Linear model; subtract fuel correction from raw lap times
Lap Time / Tyre Deg	LapTime, Sector times, TyreLife, Compound, FreshTyre, Speed traps	FastF1 laps data	Fuel-corrected degradation curves; regression per compound
Incidents (SC/VSC/Flags)	TrackStatus codes (1–7), race_control_messages (Flag, Scope, Sector, Message)	FastF1 track_status + race_control_messages	Event detection; pit window probability; time saved under SC/VSC
Driver Form	Qualifying times, race results, fastest laps, historical lap times	Jolpica API (multi-season); FastF1 (current weekend)	Rolling averages; qualifying delta to pole; consistency metrics
Team Strength (Car Delta)	Constructor standings, qualifying gaps, race pace (fuel-corrected), speed traps	Jolpica + FastF1	Rolling team-pace delta vs. field; cluster teams by tier

4. Topic Selection & Justification

4.1 Topic DM: Data Mining

Data Mining is central to this project. The three core DM techniques map directly onto strategy optimization tasks:

Regression

- Tyre degradation modeling: Predicting lap time as a function of tyre age, compound, fuel load, track temperature, and circuit abrasiveness. The regression target is fuel-corrected lap time, and features include TyreLife, Compound, TrackTemp, and circuit-level attributes.
- Fuel-corrected pace prediction: Building a regression model that separates fuel effect, tyre degradation, and true car pace from raw lap time data.
- Pit stop loss estimation: Regressing total time lost in a pit stop (pit lane transit plus stationary time) as a function of circuit layout and team.

Classification

- Optimal strategy classification: Given pre-race conditions (compound allocation, weather forecast, grid position, team strength tier), classify the likely optimal strategy archetype (e.g., one-stop S–H, two-stop S–H–M, wet start with tyre switch).
- Safety car probability classification: Given weather, lap number, circuit characteristics, and historical incident rates, classify the probability of an SC/VSC event in upcoming laps.

Clustering

- Circuit clustering by degradation profile: Group circuits with similar tyre degradation characteristics to enable transfer learning from one circuit to another with limited data.
- Driver behavior clustering: Group drivers by tyre management style (aggressive vs. conservative) based on degradation curves and stint length patterns.

4.2 Topic TS: Feature Extraction from Time Series

Time series analysis is the backbone of extracting meaningful features from the raw temporal data streams:

Filtering & Smoothing

- Lap time smoothing: Raw lap times contain noise from traffic, blue flags, and minor incidents. Applying filtering (e.g., Savitzky–Golay or moving median) to extract the underlying degradation trend.
- Weather signal processing: Smoothing temperature and humidity readings to detect meaningful trends vs. sensor noise.

Peak/Event Detection

- Safety car event detection: Identifying abrupt lap time increases across all drivers simultaneously as signatures of SC/VSC deployment, corroborated with track status codes.
- Tyre cliff detection: Detecting the point at which degradation becomes non-linear (the performance cliff) using change-point detection algorithms.
- Rain onset detection: Identifying the transition from dry to wet conditions using the Rainfall boolean, humidity trends, and lap time step changes.

Dynamic Time Warping (DTW)

- Stint similarity matching: Using DTW to compare the degradation curve of a driver's current stint against historical stints on the same compound/circuit combination to predict remaining tyre life.
- Cross-circuit comparison: Aligning degradation profiles from circuits with different lap counts but similar abrasiveness characteristics.

Prediction Models

- Short-horizon lap time prediction: Given the time series of fuel-corrected lap times in the current stint, predict the next N laps to estimate when the driver should pit.
- Weather evolution prediction: Extrapolating temperature and humidity trends over the remaining race distance.

4.3 Why This Combination Works

These two topics are complementary rather than overlapping. TS provides the feature engineering layer: extracting clean degradation curves, detecting events, and creating time-series features. DM then consumes those features to build the predictive and prescriptive models that drive strategy recommendations. The pipeline flows naturally from raw time series (TS domain) through feature extraction to model training and inference (DM domain).

5. System Architecture & Pipeline

5.1 Data Ingestion Layer

1. FastF1 bulk extraction: Script to iterate over all races in selected seasons (2020–2025), load session data (laps, telemetry, weather, messages), and store as structured Parquet files.
2. OpenF1 API client: REST client to fetch stints, intervals, and pit stop data per session, stored as supplementary Parquet tables.
3. Jolpica historical pull: Extract race results, qualifying results, and constructor standings for 2018–2025.
4. Pirelli circuit table: Manually curated CSV of circuit characteristics (abrasiveness, grip, compound nominations) per year.

5.2 Data Preparation & Feature Engineering Layer

1. Data cleaning: Filter inaccurate laps (IsAccurate = False), remove in/out laps, handle missing sector times.
2. Fuel correction: Subtract estimated fuel-weight lap time benefit from raw lap times using the linear model (burn rate per circuit).
3. Stint segmentation: Parse pit stops to define stint boundaries per driver per race; compute stint length, compound used, and tyre age at stint start.
4. Degradation curve extraction: For each stint, compute the fuel-corrected lap time trend using TS filtering techniques; extract slope (linear deg rate) and curvature (non-linearity / cliff proximity).
5. Incident encoding: Parse track status and race control messages to create binary time-series features (SC active, VSC active, red flag, yellow flag per sector).
6. Weather feature engineering: Compute rolling averages, deltas, and trend indicators from weather time series; flag rain onset events.
7. Driver/team performance features: Compute rolling 5-race average qualifying delta to pole, race finish delta to winner, and consistency (standard deviation of finishing positions).

5.3 Modeling Layer

The system comprises several interconnected models:

Model	Technique	Topic	Purpose
Tyre Degradation	Gradient Boosted Regression / State-Space	DM (Regression) + TS	Predict lap time given tyre age, compound, weather, circuit
Pit Window Optimizer	Simulation + Dynamic Programming	DM (Regression)	Find optimal pit lap minimizing total race time
SC/VSC Probability	Random Forest Classification	DM (Classification)	Estimate incident probability for reactive strategy
Strategy Archetype Classifier	Decision Tree / k-NN Classification	DM (Classification)	Pre-race strategy selection given conditions
Circuit Clusterer	K-Means / Hierarchical Clustering	DM (Clustering)	Group circuits for transfer learning
Stint Similarity Matcher	Dynamic Time Warping	TS (DTW)	Compare current stint to historical stints
Degradation Trend Extractor	Filtering, Peak Detection	TS (Filtering)	Clean lap time series; detect tyre cliff

5.4 Strategy Simulation Engine

The core deliverable is a race strategy simulator that, given a set of pre-race inputs (circuit, weather forecast, grid positions, team/driver assignments, compound allocations), simulates the race lap by lap. At each lap, the simulator:

1. Predicts the lap time for each driver given their current tyre age, compound, fuel load, and weather conditions using the trained regression model.
2. Evaluates whether pitting on this lap is beneficial by comparing the projected time cost of degradation over remaining laps against the time cost of a pit stop plus the projected gain from fresh tyres.
3. Incorporates a stochastic SC/VSC probability that, if triggered, recalculates optimal strategy (since a SC effectively offers a cheap pit stop).
4. Outputs the recommended strategy (pit laps and compound sequence) that minimizes projected total race time.

6. Detailed Modeling Approach for Key Factors

6.1 Tyre Degradation Model

Tyre degradation is the most complex factor and the core of the strategy optimizer. The approach models fuel-corrected lap time as:

```
LapTime_corrected = BasePace + f(TyreLife, Compound, TrackTemp,
                         Abrasiveness, DriverSkill)
```

Where BasePace is the car's inherent speed on fresh tyres, and f() is a learned degradation function. The model is trained per-compound using Gradient Boosted Regression (XGBoost or LightGBM) with features including: tyre age (laps), fresh/used indicator, track temperature, air temperature, circuit abrasiveness rating, circuit lateral force rating, and driver/team encoding. The target variable is fuel-corrected lap time. Non-linearity (the tyre cliff) is captured naturally by tree-based methods. For interpretability, a state-space model (as described in recent academic literature) can be used as a complementary approach.

6.2 Fuel Correction Model

The fuel correction follows the widely-used linear approximation:

```
FuelMass(lap) = StartFuel - (lap × BurnRate)
```

```
FuelEffect(lap) = FuelMass(lap) × 0.033 seconds/kg
```

The burn rate is circuit-specific and estimated from the ratio of circuit length to total laps (shorter circuits have lower fuel consumption per lap). The 0.033 s/kg coefficient is a consensus estimate from published analyses and can be refined per circuit using opening-lap data where tyre degradation is minimal but fuel effect is maximal. Safety car laps are excluded from fuel burn estimation since fuel consumption is significantly lower under caution.

6.3 Incident Response Model

Safety cars and VSCs fundamentally alter optimal strategy because they reduce the time cost of a pit stop. Under normal conditions, a pit stop costs approximately 22–25 seconds (pit lane transit + stationary time). Under a full safety car, the effective cost drops to approximately 10–15 seconds because the field is running slowly anyway. Under a VSC, the cost drops to approximately 15–18 seconds.

The SC/VSC probability model is trained on historical data with features including: lap number (incidents are more common in opening laps and restarts), circuit characteristics (street circuits have higher incident rates), weather conditions (wet races have dramatically higher SC rates), and number of cars remaining. This classification model outputs a per-lap probability that can be integrated into the simulation as a stochastic element.

6.4 Weather Transition Handling

Weather transitions (dry to wet or wet to dry) are the highest-variance strategic events in F1. The model handles these through:

- Monitoring the Rainfall boolean and humidity trend from weather data to detect rain onset.
- When rain probability exceeds a threshold, the simulator evaluates the cost/benefit of a preemptive switch to intermediate tyres versus waiting.
- Historical analysis of races where weather changed mid-race to calibrate transition detection thresholds and optimal switching points.

6.5 Driver Form & Team Strength

Driver and team performance are encoded as relative pace features rather than absolute metrics:

- **Qualifying delta:** The driver's average qualifying gap to pole position over the last 5 races, expressed in seconds. This is the strongest pre-race predictor of race pace.
- **Race pace delta:** The driver's average fuel-corrected race pace gap to the race winner over the last 5 races.
- **Consistency metric:** Standard deviation of qualifying positions over the last 10 races; lower values indicate more reliable performance.
- **Team strength tier:** Clusters teams into tiers (front-runner, midfield, backmarker) based on constructor standings trajectory; used for gap modeling between positions.

7. Project Deliverables

7.1 Data Pipeline & Processed Dataset

- Automated ETL pipeline that extracts data from FastF1, OpenF1, and Jolpica APIs for configurable season ranges.
- Cleaned, fuel-corrected, and feature-enriched dataset stored as Parquet files with approximately 200,000+ lap records across 5 seasons.
- Manually curated Pirelli circuit characteristics lookup table (CSV).

7.2 Analytical Models

- Trained tyre degradation regression model with documented performance metrics (MAE, RMSE, R^2).
- SC/VSC probability classification model with precision/recall evaluation.
- Strategy archetype classifier with confusion matrix analysis.
- Circuit clustering analysis with dendrogram and cluster profile visualizations.
- DTW-based stint similarity matching with example comparisons.

7.3 Strategy Simulator

- Python-based race simulation engine that accepts pre-race inputs and outputs optimal strategy recommendations.
- Ability to compare multiple strategy options (e.g., one-stop vs. two-stop) with projected total race times.
- Monte Carlo simulation mode that incorporates SC/VSC probability for robust strategy evaluation under uncertainty.

7.4 Visualization & Reporting

- Tyre degradation curves per compound per circuit, with confidence intervals.
- Strategy comparison visualizations showing lap-by-lap projected positions under different strategies.
- Fuel-corrected pace analysis dashboards.
- Weather impact visualizations showing lap time response to changing conditions.

7.5 Documentation

- Comprehensive project reflection report (40% of grade) documenting the full data science process.
- Project presentation (20% of grade) with key findings and live demonstration.

8. Proposed Work Plan

Week	Phase	Tasks
1–2	Data Ingestion	Build FastF1 extraction pipeline; collect 2020–2025 race data; set up OpenF1 and Jolpica clients; create Pirelli circuit lookup table
3–4	Data Preparation (TS)	Clean laps data; implement fuel correction model; extract degradation curves using TS filtering; parse incidents from track status; engineer weather features; build stint segmentation logic
5–6	Modeling (DM)	Train tyre degradation regression model; build SC probability classifier; cluster circuits; implement DTW stint matching; strategy archetype classifier
7–8	Simulation Engine	Build lap-by-lap race simulator; integrate all models; implement Monte Carlo SC mode; backtest against historical races
9	Validation & Refinement	Validate simulator predictions against actual race outcomes; tune hyperparameters; conduct sensitivity analysis
10	Deliverables	Finalize visualizations; write reflection report; prepare presentation; code documentation

9. Technical Stack

Component	Technology
Language	Python 3.10+
Data Access	FastF1 3.x, requests (OpenF1 REST), Jolpica API client
Data Processing	Pandas, NumPy, PyArrow (Parquet I/O)
Time Series (TS)	SciPy (filtering, FFT), tslearn (DTW), ruptures (change-point detection)
Machine Learning (DM)	scikit-learn, XGBoost / LightGBM, imbalanced-learn (for SC classification)
Visualization	Matplotlib (with FastF1 plotting integration), Seaborn, Plotly (interactive)
Simulation	Custom Python module; NumPy vectorized operations for speed
Version Control	Git + GitHub

10. Risks & Mitigations

Risk	Impact	Mitigation
API rate limits / downtime	Unable to collect sufficient historical data	FastF1 caching; bulk extraction in off-peak hours; local Parquet storage as checkpoint
Missing telemetry for some sessions	Incomplete feature set for certain races	Fall back to lap-level data (always available); flag incomplete records
Fuel burn rate estimation error	Incorrect fuel correction corrupts degradation analysis	Validate using opening-stint data where deg is minimal; sensitivity analysis on burn rate parameter
Non-linear tyre degradation	Linear regression underperforms near tyre cliff	Use tree-based models (inherently non-linear); change-point detection for cliff identification
SC/VSC inherently unpredictable	Low classification accuracy	Frame as probability estimation, not deterministic prediction; Monte Carlo integration in simulator
Scope creep	Attempting too many features reduces quality	Prioritize tyre deg + fuel correction as core MVP; layer on additional factors iteratively

11. References & Resources

- FastF1 documentation: https://docs.fastf1.dev/ (v3.7.0)
- OpenF1 API: https://openf1.org/ (2023 onwards)
- Jolpica/Ergast API: https://jolpi.ca/ (historical data from 1950)
- Pirelli F1 Tyres: https://www.pirelli.com/tyres/en-ww/motorsport/f1/tyres
- Pirelli Race Previews (circuit infographics): https://racingspot.pirelli.com/
- State-Space Tire Degradation Modeling (2024): arxiv.org/pdf/2512.00640
- F1 Tire Prediction — Cornell University: people.ece.cornell.edu/land/courses/ece5760/FinalProjects/f2020/
- TracingInsights F1 Data Archives: https://tracinginsights.com/