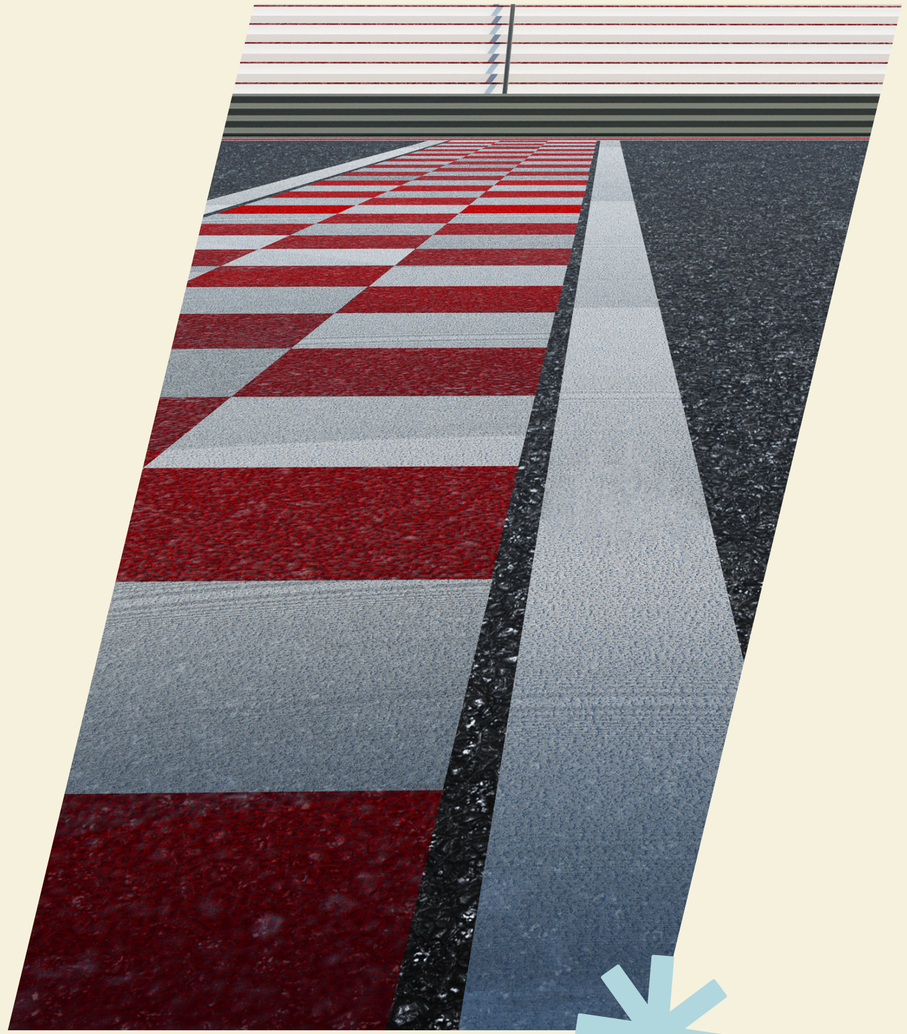
Team DOE

Formula 1

Race Prediction Dataset

ML-Ready F1 Dataset with Feature Engineering for Race Outcome Prediction

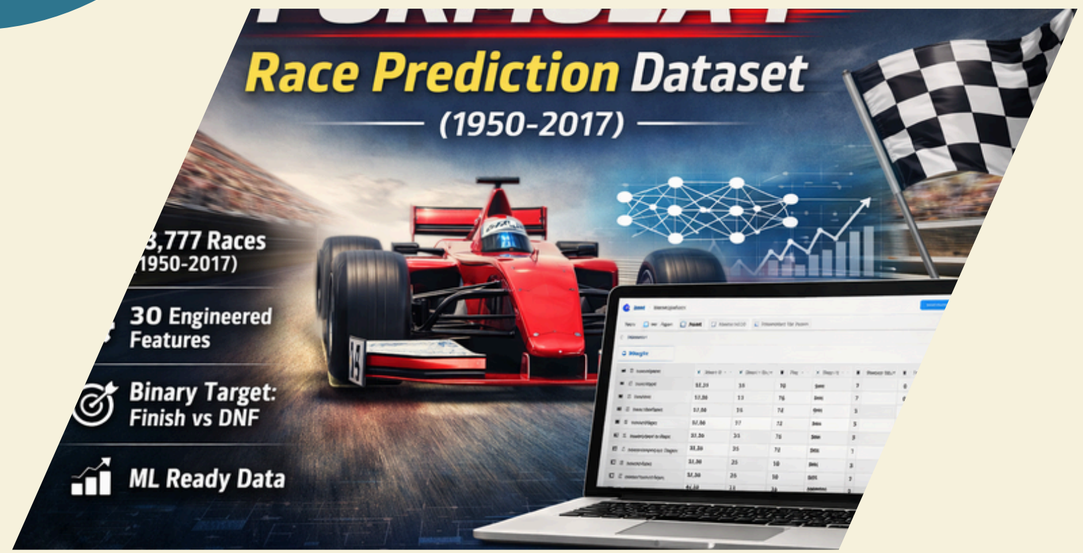


Why did we choose this...

Predict Formula 1 race outcomes using historical data (1950–2017).

We chose this dataset because it combines multiple Formula 1 data sources into a single, feature-engineered, machine learning–ready dataset. It includes historical driver performance, constructor reliability, circuit characteristics, qualifying data, and temporal features, allowing the model to learn meaningful patterns that influence race completion. Additionally, the binary target variable (Finished vs DNF) makes it well-suited for classification tasks and evaluating model generalization.

This project builds a binary classification model to predict whether a driver will finish a race (Finished vs DNF) based on historical race features.



Dataset Overview

Dataset Name: **Formula 1 Race Prediction Dataset (1950–2017)**
Type: **Structured tabular dataset**
Time span: **67 years of races**

Main processed dataset containing Formula 1 race entries from **1950–2017.**

Each row represents a driver's participation in a race.
Includes:

- **Race result features**
- **Driver and constructor statistics**
- **Circuit characteristics**
- **Qualifying performance**
- **Temporal features**

Target column: **finished (1 = Finished, 0 = DNF)**
Rows: **23,777**
Columns: **30**

This dataset combines multiple Formula 1 data sources to create a comprehensive, analysis-ready dataset for predicting race outcomes.

Data Exploration

What's unique about this dataset?

This data is **pre-processed** and **feature-engineered** specifically for machine learning tasks:

- ✅ **Integrated Data**: Merges results, qualifying, pit stops, drivers, constructors, and circuits data
- ✅ **Feature Engineering**: Calculated driver performance metrics, constructor stats, circuit characteristics
- ✅ **Clean & Ready**: No raw data - ready for training ML models immediately
- ✅ **Binary Classification Target**: finished column (1 = Finished race, 0 = DNF/Did Not Finish)

Use Cases

1. **Binary Classification**: Predict race finish vs DNF
2. **Feature Importance Analysis**: Which factors most influence race completion?
3. **Time Series Analysis**: Evolution of F1 reliability over decades
4. **Driver/Team Performance**: Compare historical performances
5. **Circuit Analysis**: Track-specific patterns
6. **Educational Projects**: Learn ML with real-world sports data

Data Quality

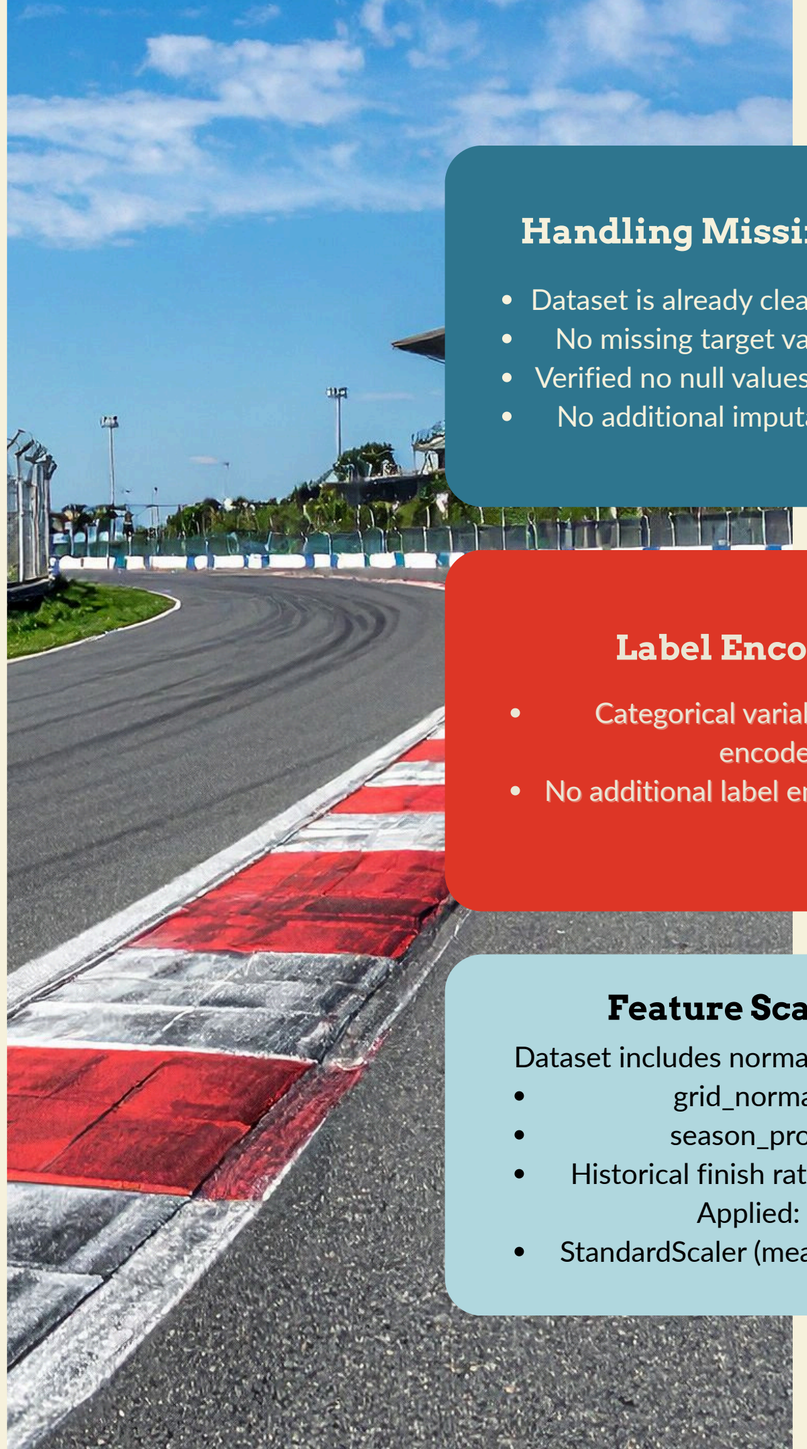
- **No missing target values**: All races have finish/DNF labels
- **Temporal integrity:** Features only use historical data (no data leakage)
- **Encoded categories**: All categorical variables numerically encoded
- **Normalized features**: Grid positions and times normalized for consistency

Dataset Statistics

- Total Entries: **23,777**
- Finished Races: **17,777 (74.8%)**
- DNF: **6,000 (25.2%)**
- Years Covered: **1950-2017**
- Features: **30**

The dataset was already clean and machine learning–ready, with encoded categorical variables and normalized features, so no additional missing value handling or label encoding was required. It was split into 85% training and 15% validation (≥15%) to evaluate model generalization and prevent overfitting.

Data Preprocessing



Handling Missing Values

- Dataset is already clean and ML-ready
- No missing target values (finished)
- Verified no null values before training
- No additional imputation required

Label Encoding

- Categorical variables already encoded
- No additional label encoding needed

Feature Scaling

Dataset includes normalized features:

- grid_normalized
- season_progress
- Historical finish rates (0–1 scale)

Applied:

- StandardScaler (mean = 0, std = 1)

Model Architecture (Keras)

- *Input Layer (number of features: ~20-50)*
- *Dense(64, activation='relu')*
- *Dropout(0.2)*
- *Dense(32, activation='relu')*
- *Dropout(0.2)*
- *Dense(16, activation='relu')*
- *Output Layer: Dense(1, activation='sigmoid')*

WHY?

Activation Functions

- ReLU (Hidden Layers) → learns complex patterns efficiently
- Sigmoid (Output Layer) → outputs probability (0–1)

Loss Function

- Binary Crossentropy - measures difference between predicted probability and actual label

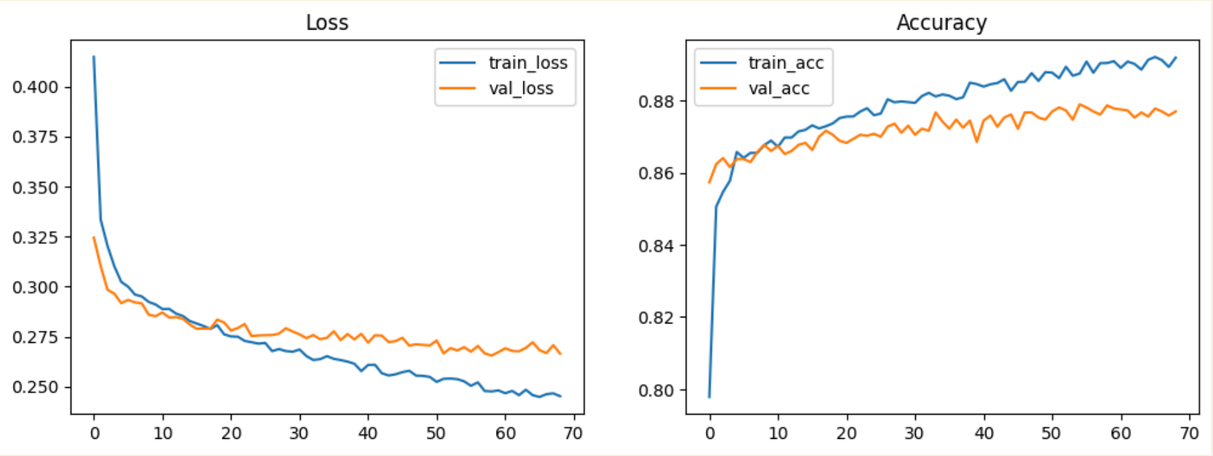
Optimizer

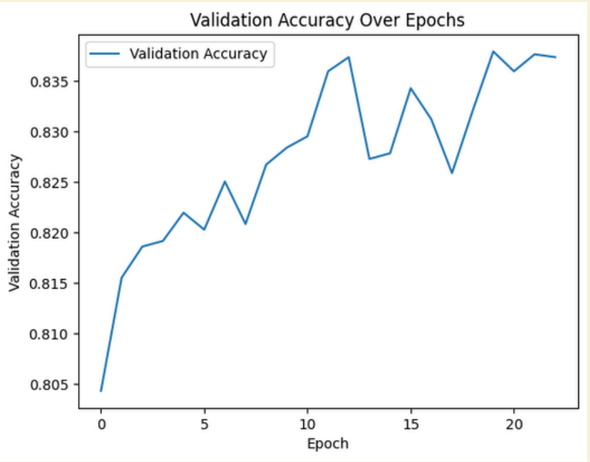
- Nadam - adaptive learning rate with momentum and Improves training stability and convergence

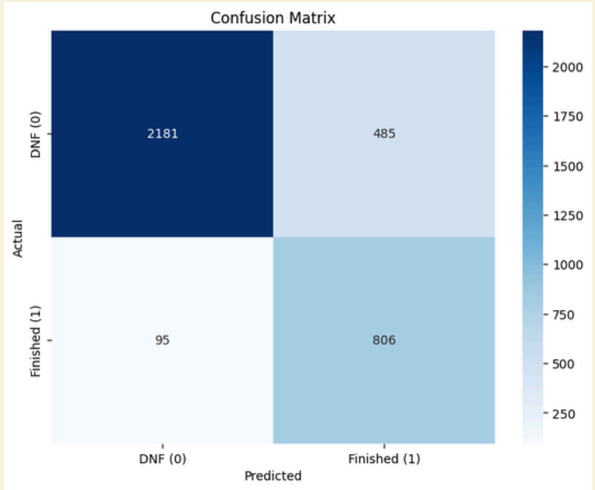
Training Configuration

- Epochs - 100 with early stopping
- Batch size - 64
- Optimizer - Nadam
- Loss function - binary_crossentropy
- Metrics - Accuracy

Results







Validation accuracy: 0.8786

Conclusion

Model Development

A binary classification model was built to predict race completion (Finished vs DNF) using historical driver, constructor, circuit, and qualifying features.

Performance & Training

Using StandardScaler, dropout regularization, and an 85%–15% train–validation split, the model achieved stable learning and good generalization with minimal overfitting.

Key Insight

Historical performance metrics and contextual race features are strong predictors of race completion, demonstrating the effectiveness of feature engineering in motorsport analytics.

Team DOE

Thank You

***

Dy, Zendy Mariel
Espina, Ruhmer Jairus
Ong, Lovely Shane