F1 Prediction Model

Overview

For my project I wanted to create a prediction model to predict F1 Qualifying and Race Results

I wanted to create this because I am passionate about F1 and Machine Learning and wanted to combine my interests in 1 project

Features

Features

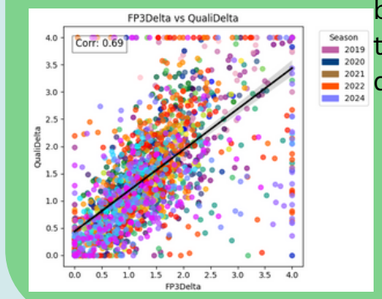
- Free Practice 2/3 timing
- Previous Qualifying timings
-

Transformation

- lap times converted to deltas (*see below*)
- Capped at +4s vs P1

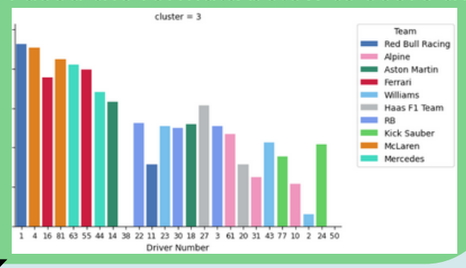
Missing Data

Regression models cannot handle missing data (crashes, driver replacements). FP3 delta used.



Previous analysis has shown that different drivers perform better at certain types of circuits. I used a KMeans model to group the circuits. I used the formula below to score the drivers.

$$e^{-0.5(T_D - T_B)}$$



Future Plans

Qualifying

- Track Profiles
  - My model is significantly weaker around more niche tracks with unique characteristics.
- Weather Data
  - Teams use Free Practice to optimise the car setup, however this is highly dependant on weather conditions, it could have a significant affect on performance
- XG-Boost
  - XG-Boost could improve my models by finding more complex patterns in the data that can't be found with just Lasso Regression

Race

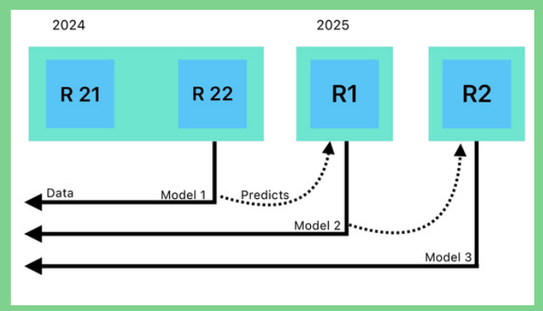
I plan to include my Qualifying prediction as a feature for a race prediction model.

A race prediction model is inherently more complicated, because it is a multi-lap affair

Model

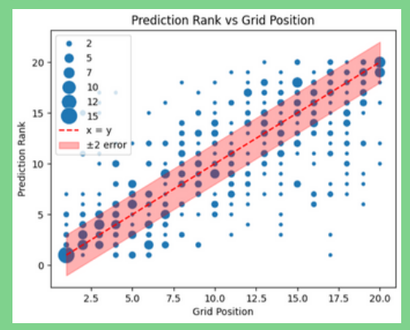
The model uses an ensemble consisting of a linear regression combined with random forest. Lasso regression is used because the features are inherently collinear; thus it penalises coefficients resulting in a more generalised model. Ensemble learning increases robustness as it reduces the risk of errors from a single model.

Linear models can't be used to predict discrete data such as qualifying position, so the model predicts the time difference from P1 (delta), these deltas are then ranked to give effective qualifying position. The model is fitted prior to each qualifying session using the data from 2019 to the previous race.

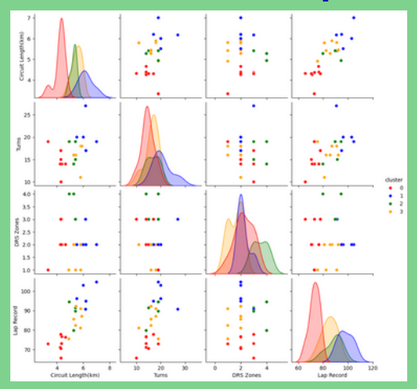


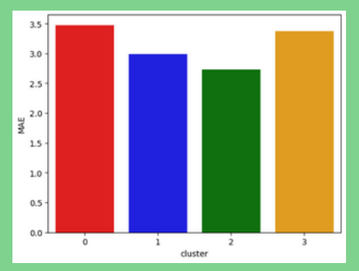
Performance

Mean Absolute Error (MAE) is used to score the model instead of Mean Squared Error (MSE) as the latter applies greater weight to outliers (arising from unpredictable events such as crashes or red flags). Overall the model achieves a 3.1 position MAE and a percentage within 2 places of 53%.



Performance Decomposition





The key to improving model performance is to understand the scenarios in which the model underperforms.

Github Repo: https://github.com/ZSH4RK/F1-Data-Analysis