Steven Jia And Quentin Phillips

5/10/2024

DATA 441

Predicting Formula 1 Results

1. Abstract

The purpose of this research project is to develop a model that can accurately predict the results of Formula 1 races based on input features known about the race before it begins. Using data from the last 5 years of Formula 1 we constructed linear and polynomial regression models, as well as 3 different neural networks to find the optimal algorithm for predicting race results. After comparing models, our final neural network utilizing a CORAL-Ordinal loss function scored a mean absolute error of ~3.4, meaning the model can generally predict a driver's placement within a few positions.

2. Introduction

Formula 1 is the top tier of open-wheel racing. This motorsport focuses on road courses, aerodynamics, and no-contact racing. Teams and their drivers travel around the world to race at different circuits. While there is a set of rules that all teams must follow called the “formula”, each team has a degree of control over their car’s design. Each race starts with a qualifying session where drivers try to finish one lap of the track as fast as possible. The rank of qualifying determines the starting order of the race itself. While each circuit’s lap length is different, varying lap counts ensure all races are around 190 miles long. This distance means that tires will not last an entire race, which forces teams to do a “pitstop” to swap the tires. These are just some of the factors that determine performance.

Performance at these races earn teams points. Currently there are 10 teams with 2 drivers each. The points total at the end of the season determine the winning team, a sum of the points of both team's drivers, and the winning driver. The goal of our project is to try to use historical data to predict the outcome of the races.

3. Description of Data

Data was found on Kaggle. It provides the race outcomes of every F1 race since 1950. This includes qualifying results, lap times, pit stop stats, and final results for each driver and their team. Driver, team, and circuit were recorded as numeric ID. Qualifying results and final results were both ordinal data. Pit stop stats included number of pit stops and duration of each in milliseconds.

While the data was comprehensive, we could not use all of it. The main goal of this project was to predict this season's race results, which makes older data less useful. Rule changes, driver retirements, and teams disbanding or being replaced makes a lot of historical data contain information on teams and drivers that no longer compete or no longer perform at the same level. Therefore, we decided to limit our data to the previous five years.

Another consideration was the availability of information. Certain things can only be determined during a race and therefore be unavailable when trying to predict outcomes. Therefore, we decided to omit the pit stop data from the model because any number of situations could occur during a race that would change this info, making historical trends irrelevant.

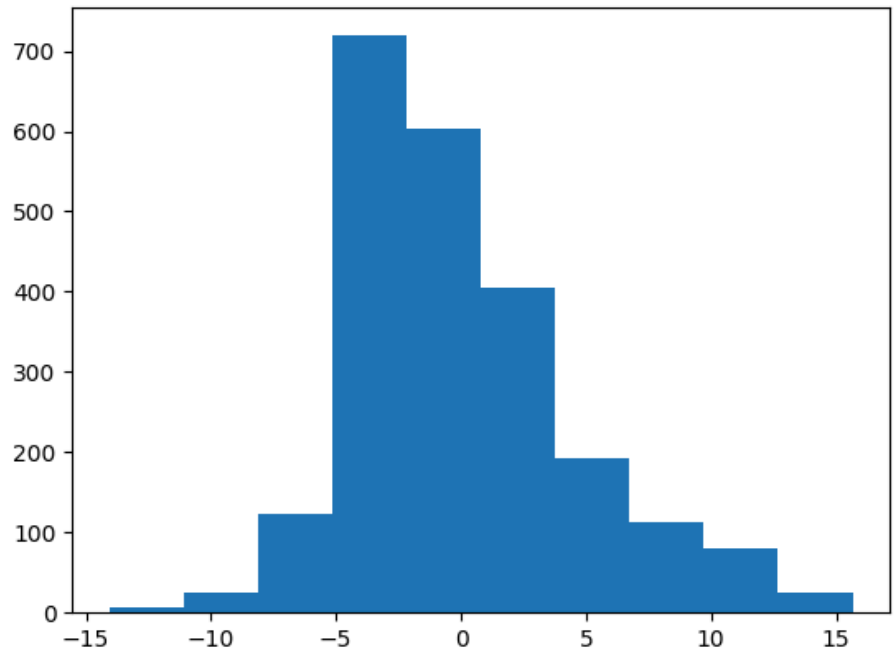
Finally, we felt like one key data point was missing: weather. Rain makes roads more slippery and often requires special "wet" tires designed to grip better on wet surfaces. This plus other factors like reduced visibility can change the nature of a race completely. Some drivers have a reputation of performing better in the rain while others have the opposite, so we believed

weather data must be included to try and model that intuition. The Kaggle data did not have weather in it so we manually added a binary variable for each race to represent whether the race had rain or not. After all considerations we ended up with circuit, driver, team, qualifying results, and weather as features to predict final position.

4. Methods

The preprocessing of the data consisted primarily of creating a DataFrame using the relational database found on Kaggle. The data was compiled into a .csv file so that all the necessary features were present and contained no null values. This file was read into Python using the Pandas library. The final change to the DataFrame was correcting data types and removing any remaining missing values.

The initial exploration of the data was done through creating a linear regression model. This model used circuit, driver, qualifying results, weather, and constructor as the features for prediction, and final position at the end of the race as the target. The Scikitlearn Linear Regression library was used to fit the model. This regression model returned a R squared value of 0.385. We then graphed the residuals in a histogram to determine whether they were close to normally distributed:



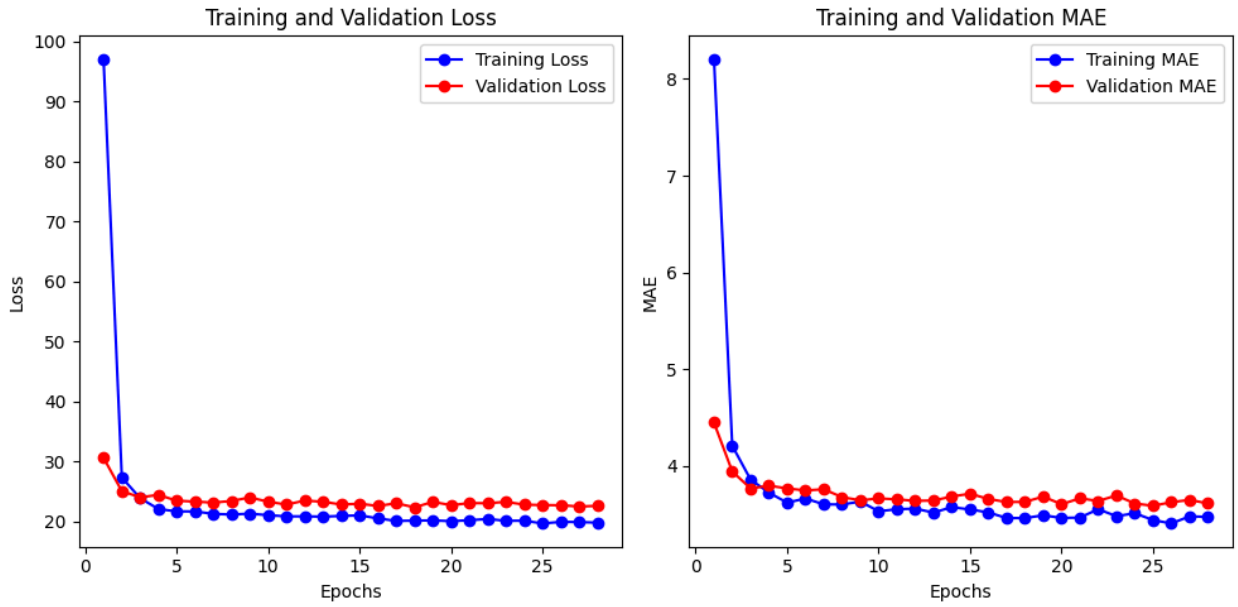
The residuals appear somewhat skewed, but mostly normally distributed.

In order to improve the accuracy of the regression model, we moved to a polynomial regression model with interaction terms. This model returned a R squared value of 0.403. This was a significant improvement in accuracy, and this model became the baseline for our statistical analysis using regression.

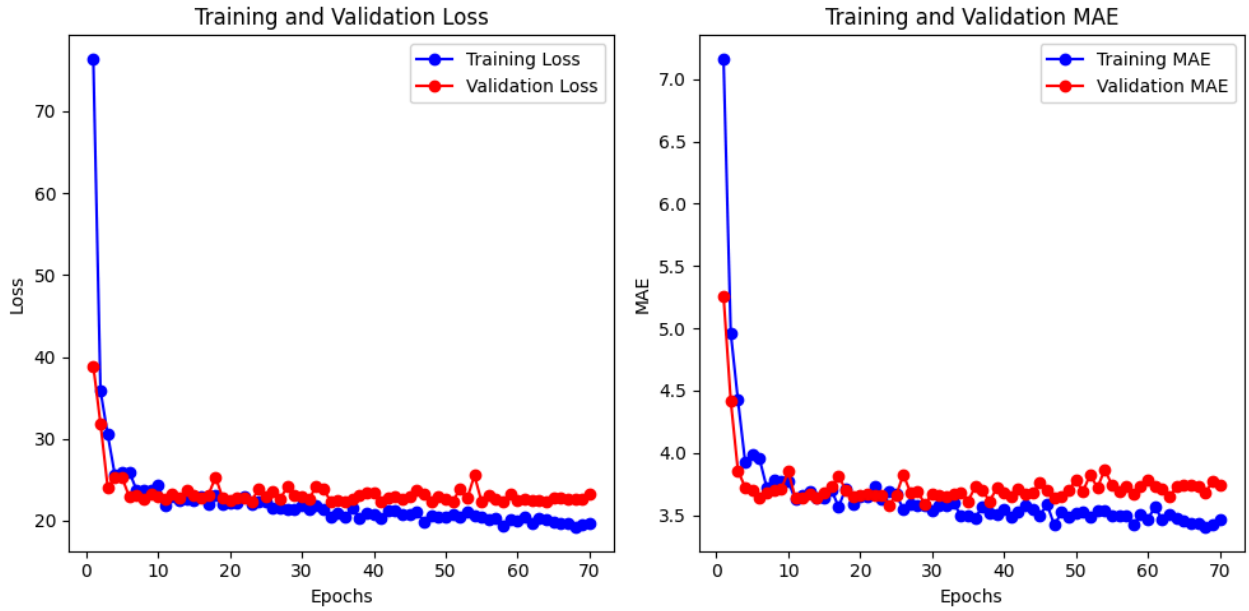
Neural Networks

Our initial neural network model attempted to use a classification algorithm to predict race results. This model was quickly scrapped after poor results that were not responsive to hyperparameter tuning. Our next model used mean squared error as the loss function and a linear activation function. The network consisted of 4 ReLU layers with between 4-256 neurons. The more complex layers had dropout values of 0.5. We utilized tensorflow's Adam optimizer with an initialization value of 0.001. The model used mean absolute error (MAE) to measure accuracy, returning a testing MAE of 3.605. The training and validation losses are plotted over

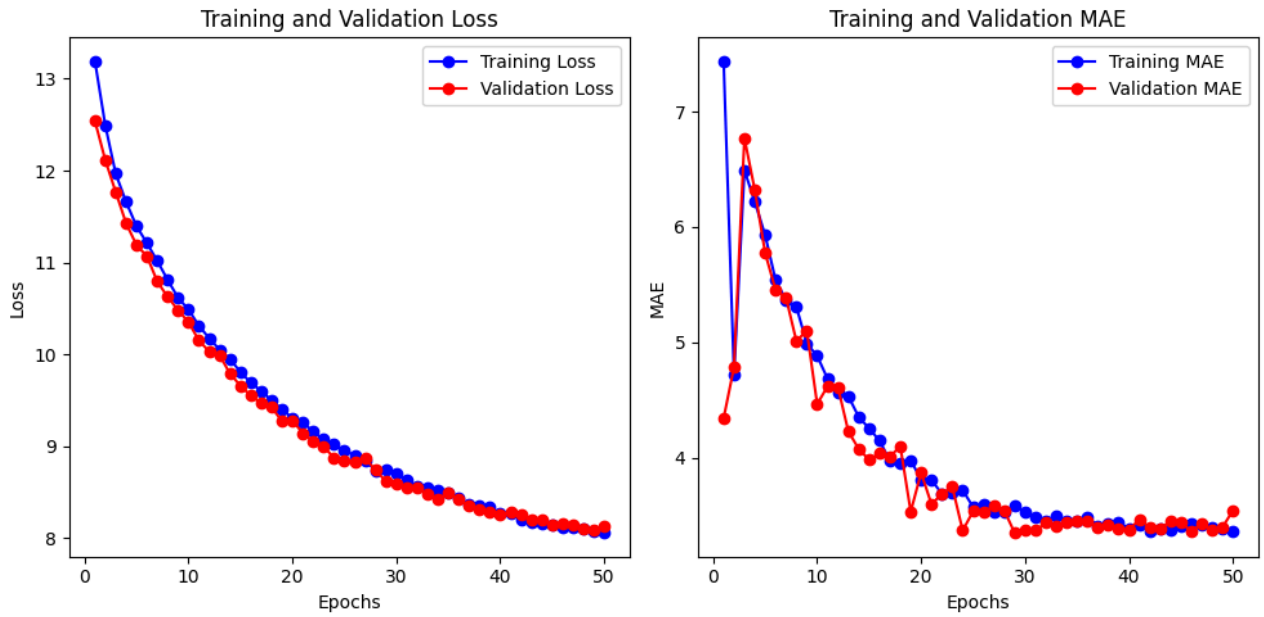
each epoch in the following figure:



The next attempted neural network model was a convolutional neural network (CNN) using 3 convolutional layers using filter levels of 32, 64, and 128, as well as 2 ReLU layers. This model returned slightly inferior results, with a final testing MAE of 3.762. The training and validation losses are plotted over each epoch in the following figure:



Our final neural network utilized the COnsistant Rank LOgits (CORAL) to better fit the ordinal data we are predicting. This model proposed by Cao, Mirjalili, and Rashka allows neural networks to categorize data while considering the rank of the categories. We used a package developed by Berkeley D-Lab called "coral-ordinal" which implemented the method proposed by Cao et al.. This neural network consisted of 5 "dense" layers using the "relu" activation function plus a final "CoralOrdinal" layer that would output ordinal classifications. The model also utilized coral-ordinal's custom loss and accuracy functions. The final iteration of this network after hyperparameter tuning resulted in a testing MAE of 3.408. This model had the highest accuracy, but performed similarly to the standard regression neural network. The loss and MAE of the CORAL model can be seen below:



5. Discussion and Inferences

The final neural network resulted in an MAE of 3.4 which outperformed initial expectations. Unsurprisingly, the CORAL loss function resulted in the best performing model

given its explicit design for ordinal dependent variables. All three of the neural networks outperformed the traditional linear and polynomial regression models. The linear and polynomial regressions were limited by their model complexity and the type of data. The linear model is not designed for nominal data like driver ID which also hurts its accuracy. The neural networks were more able to handle the data types in our data and model the complex relationships between our features and race results.

We may be reaching the limit of what is possible with the data available. F1 races have a degree of unpredictability due to the numerous variables that influence the race. Many of these variables only become known when a race happens, such as crashes and malfunctions. The expectation is that the best team and best driver will always win the race. This may happen often but it does not happen every time. Until every variable can be accounted for and modeled, the only true way to find the fastest person on the track will be to race.

Works Cited

1. Vopani. (2023). Formula 1 World Championship 1950-2023 (9). Retrieved from https://www.kaggle.com/datasets/rohanrao/formula-1-world-championship-1950-2020

2. Cao, W., Mirjalili, V., & Raschka, S. (2019). Rank-consistent ordinal regression for neural networks. arXiv preprint arXiv:1901.07884, 6.