Capstone Project Proposal

Avi Mukesh

March 2023

1 Domain Background

The Premier League (PL) is the most watched sports league in the world. 20 teams battle it out by playing two matches against every other team - one home and one away. 3 points are obtained for a win, 1 point for a draw, and no points are awarded for a loss. At the end of each season teams are relegated to the Championship if they finish in the bottom 3. Teams are also promoted from the Championship into the Premier League at the end of each season.

As with a lot of popular sporting events, fans and punters alike make bets on the outcome of games such as what events occur (e.g. the number of assists a certain player gets), and the actual outcome of the match. There is a lot of statistics that can be used to calculate accurate probabilities of certain events happening, or a certain team winning. Some of the big PL matches can easily attract over 2 million live viewers and therefore the football betting business is very big.

2 Problem Statement

One of the most popular bets to place is the outcome of a match - either a draw, the home team wins, or the away team wins. My goal is to use historical data to predict this outcome of a game based on factors such as the sides' respective positions in the table at the moment, their average player valuations, the number of clean sheets the teams have kept in their previous few games, etc.

3 Dataset

I will be using a dataset from Kaggle that has been scraped from the Transfermarkt website. The dataset is called "Football Data from Transfermarkt" [**?**]. It contains data about PL games dating back to 2012, including the goals, substitutions, scores, player valuations at different times of the year, the referees, attendance and more.

4 Solution Statement

I aim to train a machine learning model on the dataset mentioned. The input to the model will be the clubs that are playing, in addition to other relevant information about the match. The output should correspond to either a win for the home team, a draw, or a win for the away team.

5 Benchmark Model

There is Medium article [**?**] where the author has completed a similar project, using different data and a different methodology. They obtain an accuracy of 92%. [**?**] uses a Random Forest Classifier, and they obtain an accuracy of 60%. I will use these results as benchmarks to compare my model to.

6 Evaluation Metrics

I will use accuracy and precision to evaluate my model's performance. I will use accuracy to get an idea of what proportion of predicted outcomes is the model getting correct. Accuracy is between 0 and 1. However, the accuracy metric misses some key information. Hence I also will use the precision metric which is the number of true positives divided by the total number of positives. Since there are 3 classes, an average precision has to be calculated.

7 Project Design

I will use SageMaker Studio to create a Python Notebook. Within this notebook, after creating data visualisations and performing exploratory data analysis, I will do initial model training. There won't be much data cleaning to do, since the data is readily updated and cleaned already and there isn't much missing data. Here are the steps I will perform for data preparation:

- change columns which contain numbers into a numeric type
- add any additional columns for attributes that I require (e.g. clean sheets)
- remove any unnecessary columns (e.g. stadium name)
- categorise certain columns so they are easier to deal with during training (e.g. use a referee number instead of name)

I am interested to see how many goals are scored per game, so I will visualise this using a histogram. I may also visualise how other factors affect goals and the outcomes of matches.

For model training, I will experiment with AutoGluon and the RandomForestClassifer provided by scikitlearn to see which gives better metrics. I will

also experiment by using different sets of predictors to see which predictors have big effects on accuracy and precision.

I will evaluate the model performance using the metrics mentioned previously, and tweak the hyperparameters to find the set of hyperparemeters that gives the highest accuracy and precision.