Data Pre-Processing

I accessed football event data using the StatsBomb API python package statsbombpy. The data had to be accessed on a match level, so I had to obtain the shot and previous events for each match individually, subsequently concatenating these into one dataset. My initial dataset contained 82,491 shots and over 150 columns, after dropping the columns which contained only NaNs my dataset contained over 100 columns. I expanded the shot freeze frame column which was a list of dictionaries containing information about each player on the pitch at the moment of the shot. From this I was able to obtain the location of each player at the moment of the shot, creating columns for both location of the player and distance from the shot location. The columns were separated between teammate and opposition players and numbered based on the distance from the shot with 1 being the closest teammate or opposition, except for the goalkeeper position and distance which was named separately. Analysis of missing values identified that the location and distance of players further away from the shot contained more missing values. Following this analysis, I chose to retain the 4 closest opposition players and the opposition goalkeeper, and the 2 closes teammates. After dropping missing values from these columns, my dataset contained 77,681 shots. I retained all columns which had information about the context of the shot, as well as the type and location of the previous action. I also retained the expected goals value determined by StatsBomb’s model to enable comparison with future models. A ‘goal’ column was created using the ‘shot\_outcome’ column with a True or False value for if the shot was a goal or not.

To prepare the data for input to XGBoost, I used one hot encoding on the categorical variables. Once this was done, I removed and stored the ‘goal’ column as y values and the ‘shot\_statsbomb\_xg’ column as comparison results. I also used sklearn’s Standard Scaler to scale the data to account for differences in distribution of values.

EDA

I investigated the distribution of my variables, for my variables which were x and y coordinates I plotted 2 dimensional histograms for each of these. For my categorical variables I plotted bar charts showing the frequency of each category in the dataset.

XGBoost

I implemented an XGBoost binary classifier to predict the probability that a given shot will result in a goal. XGBoost is a variant of gradient boosting which uses regularisation. I used both grid search and hyperopt optimisation, focusing on the hyperparameters ‘learning\_rate’ , ‘n\_estimators’ and ‘tree\_depth’. The n\_estimators hyperparameter focuses on how many rounds of boosting occur, meaning the number of trees which are created. Relatedly the learning\_rate hyperparameter governs how much the next tree is adjusted by the results of the previous tree. The tree\_depth hyperparameter, states the maximum depth of each tree, with higher depths associated with potential overfitting and lower depths potential underfitting.

Notes

Randomising player location number.

Using an embedding for the coordinates.

Using another decision tree approach – maybe CATBoost.