Data Access + Prep

I accessed data from the StatsBomb API using the statsbombpy package. This API contains free data from selected matches, with observations for each event (shots, passes, dribbles) in each match and features providing information on these events. I concatenated the data for all the matches available for each shot event and each event directly preceding each shot. This resulted in a dataframe containing 84,891 shot observations and 134 features, once all features containing entirely missing values were removed. Following this I converted the location column into separate x and y columns. I then processed the pivotal shot\_freeze\_frame column which contained a dictionary of all the player positions (both opposition and teammate) at the moment of the shot. Through this column I created new columns for each players x and y position as well as each players distance from the shot in the x and y direction. These player position columns were also numbered by their Euclidean distance from the shot separately for both opposition, teammate and a separate set of columns for the opposition goalkeeper. Following this I investigated the distribution of missing values for these player position columns. I found that the players further away from the shot had more missing values, with the opposition goalkeeper having the least missing values. Following this I retained the sets of columns for the 4 nearest opposition outfield players, 2 nearest teammates and the opposition goalkeeper. This only reduced the dataset by 7,274 observations, following this I also removed other columns with a large number of missing values resulting in a final dataset containing 77,617 shots and 42 columns.

Further preprocessing occurred to prepare the dataset for input into both my XGBoost and neural network models. I created a new Boolean ‘header’ column based on the ‘shot\_body\_part’ column with values being true or false if the body part used for the shot was the head, I also converted all my Boolean to integer data type. The shot\_statsbomb\_xg column (containing the StatsBomb model’s predicted probability of the shot resulting in a goal) was removed from the dataset but retained to compare the performance of each model to StatsBomb’s expected goals model. A Boolean column ‘goal’ was created using the shot\_outcome column which was True if the outcome was goal and false elsewise. This was then removed and with the values retained as the y input to my models. Following this my categorical variables were encoded using one hot encoding. This resulted in a dataframe containing 63 features and 77,617 observations. The final step in the data preprocessing for both XGBoost and neural network methods was to scale the data using Sci-Kit Learn’s standard scaler to account for differences in the distribution of features.