Thesis Documentation

The general workflow of the feature engineering is completed using a combination of ipynb files and regular python files.

This allowed for features to be generated in individual code blocks to reduce the complexity of each call to build and clean the dataset.

Data Builder – Player Ratings

The majority of the dataset building is done here.

Game data is read and stored for each year as well the entire set of all games.  
The player statistics are read in the same fashion

Build Y Value:  
As it the outcome will be a prediction model, the y model is set as the winner from the game.   
This was simply based on computing the difference between the home teams score and the away teams score.  
There was also the case of drawn games, although making up 1% of match results. The drawn games were left in to leave the dataset as complete as possible. Due to this the outcome was not binary, and instead was set to 0 for a loss, 1 for a draw and 2 for a win

Add Result of Each Teams Previous Game  
This was done using a default dict which stored the result of each teams previous game  
The home team and away team were extracted from each row.  
Their previous results were than extracted from the default dict and stored in the new field  
After this the results in the dict for each team were updated based on the current game

Travel Distance  
The functionality for travel distance was defined in the “team\_travel.py”

Previous Ladder Position  
Based on the finishing position of the previous year

Team Similarity  
Functionality was defined in “team\_changes.py”

Player Importance  
Based on “supercoach” and “fantasy” scores which are calculated after each game  
Functionality was defined in “player\_rankings.py”, “Supercoach\_scores.py” and “Fantasy\_scores.py”

One Hot Encoding

Used to convert venue data and team names to integer data