ECM 3420 Coursework

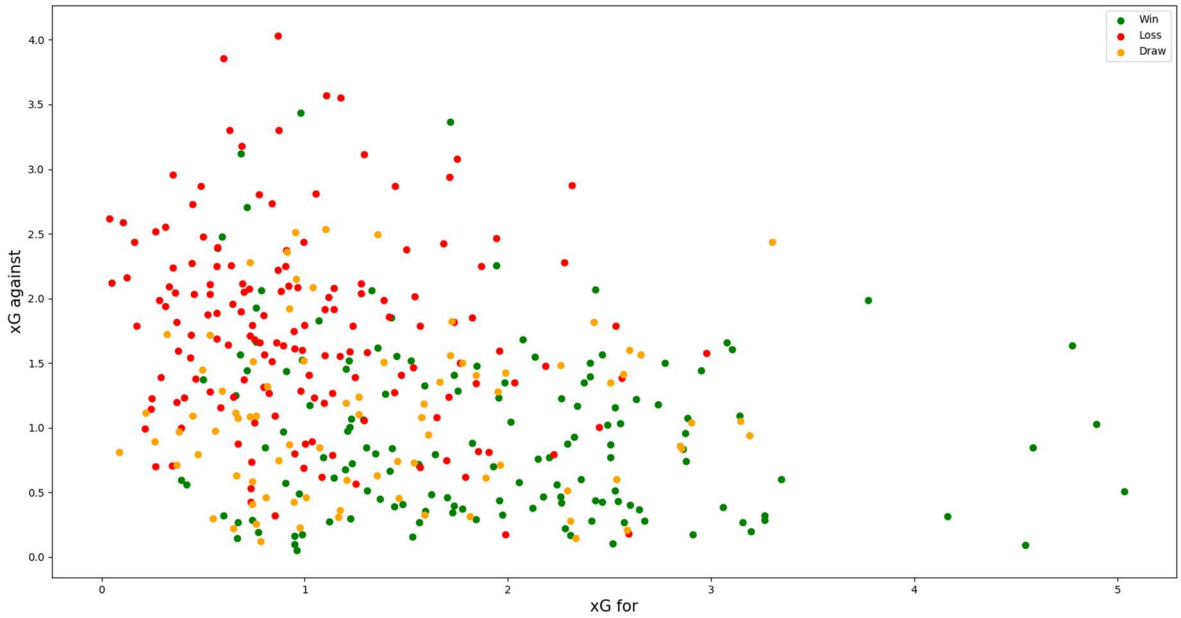
Introduction

The main objective of my analysis is to predict whether a football team will win, lose or draw a match given a set of data. The data I will be using will all be from the 2020-2021 English Premier Leagues season. The data has been sourced from understat.com via the following GitHub repository: https://github.com/vaastav/Fantasy-Premier-League/tree/master/data/2020-21/understat. To perform my prediction, I will have to import the data, which is in a csv format, then process it. My main issue is going to be deciding whether clustering or classification is going to be the most effective technique for this task. Another issue I will have to deal with is that the data set will have two copies of each data point. This is because the dataset includes the home and away matches of every team even include matches of teams who played each other. To solve this problem I removed all away matches from the dataset.

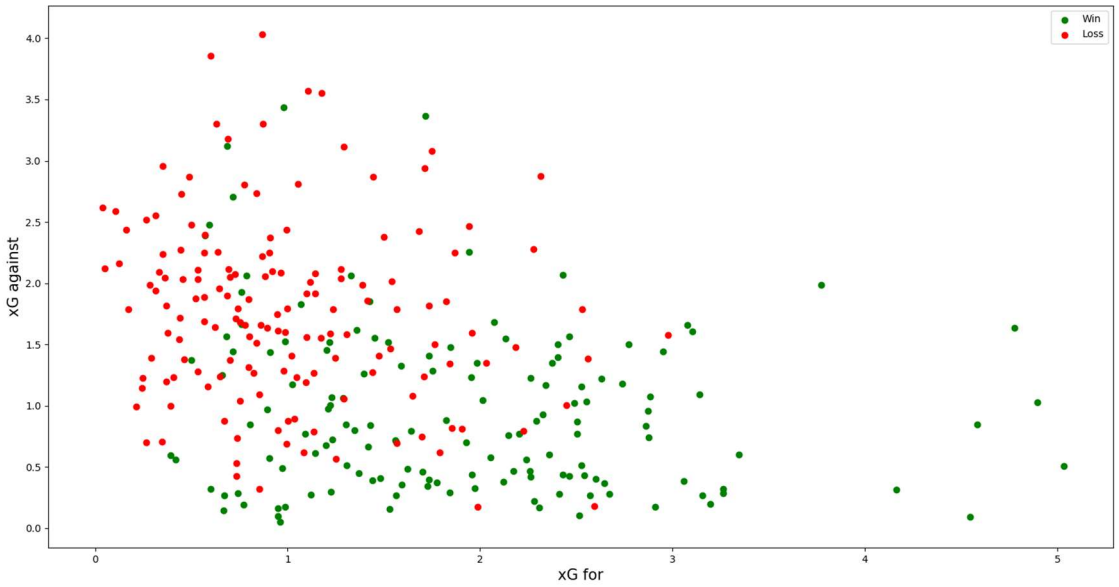
The Data

Data Dictionary

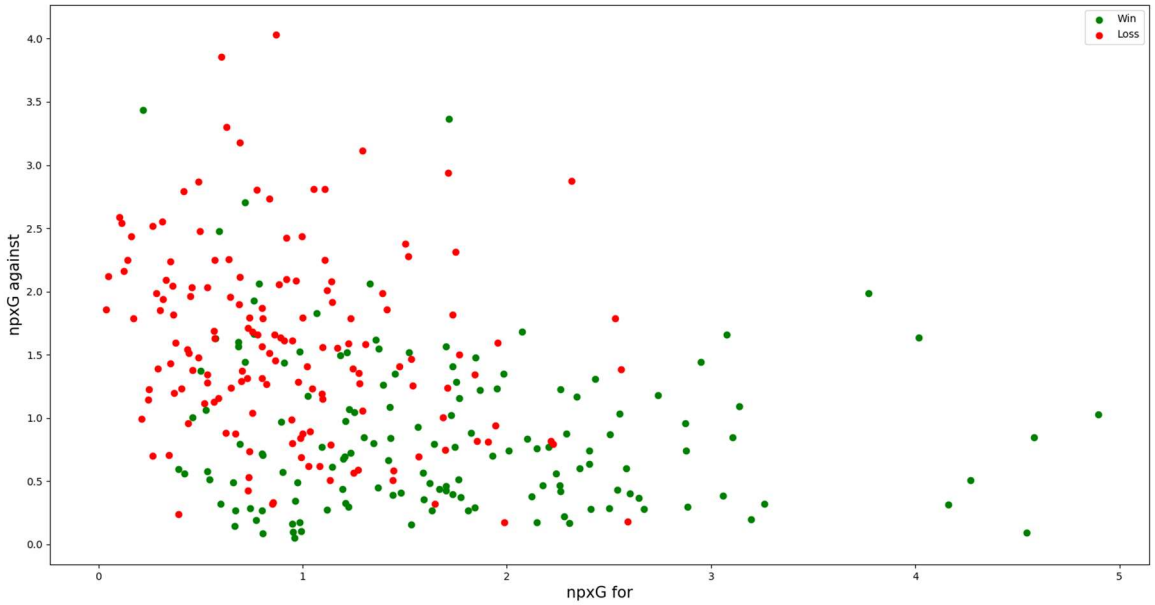
- xG – The expected goals of the team. Expected goals is calculated by using historical data from previous shots from the same position to calculate the probability of each shot resulting in a goal. xG is the sum for all shots using this metric.
- xGA – The expected goals against the team. This is calculated the same way as xG but for the away team.
- npXG – The xG excluding penalties for the home team
- npxGA – The non-penalty xG for the away team
- PPDA – The passes per defensive action for the home team
- PPDA_allowed – The passes per defensive action allowed for the away team
- deep – The number of plays in the opponent final third by the home team
- deep_allowed – The number of plays allowed in the team's final third by the away team



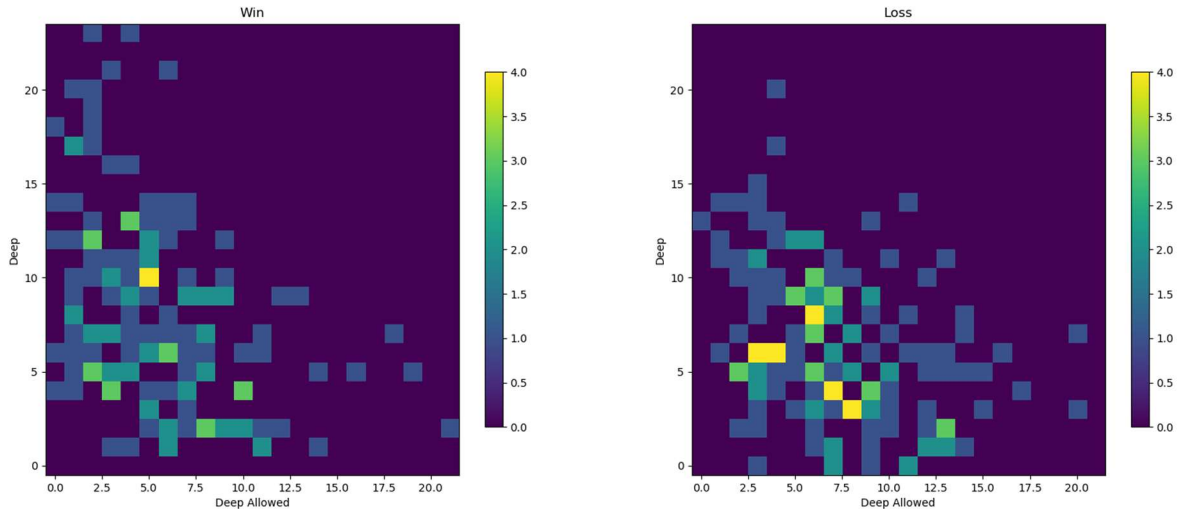
This is a graph of how xG and xGA affect the outcome of the match. After creating this graph, I realised that including draws would likely reduce the accuracy of the classifier significantly due to the lack of a clear separation as well as limiting the machine learning techniques I could use. Therefore, I removed draws.



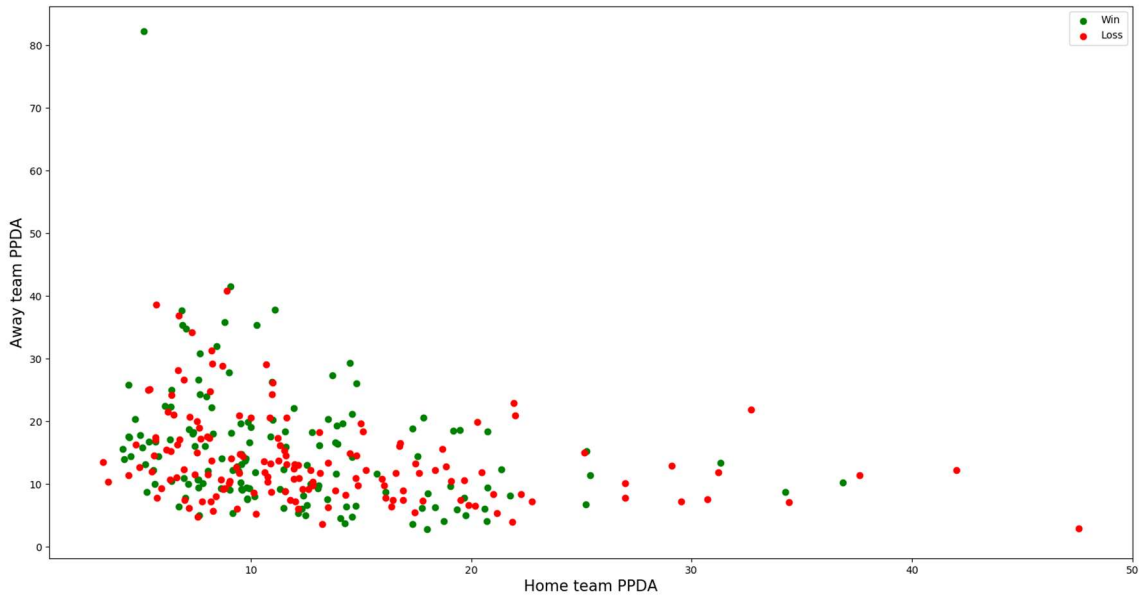
From this graph we can tell that as home team xG increases, the home team is more likely to win. However, as the xG against increases the team is more likely lose. However, there are some anomalies such as the match where the team had a xG of 0.98 and a xGA of 3.44. This is not surprising as there is an element of luck involved in football as one team could miss all their chances, even if each shot has a high probability of going in and your team gets a lucky penalty which is converted. The number of observations where the outcome was a victory is 144 and the amount where the outcome was a loss is 153.



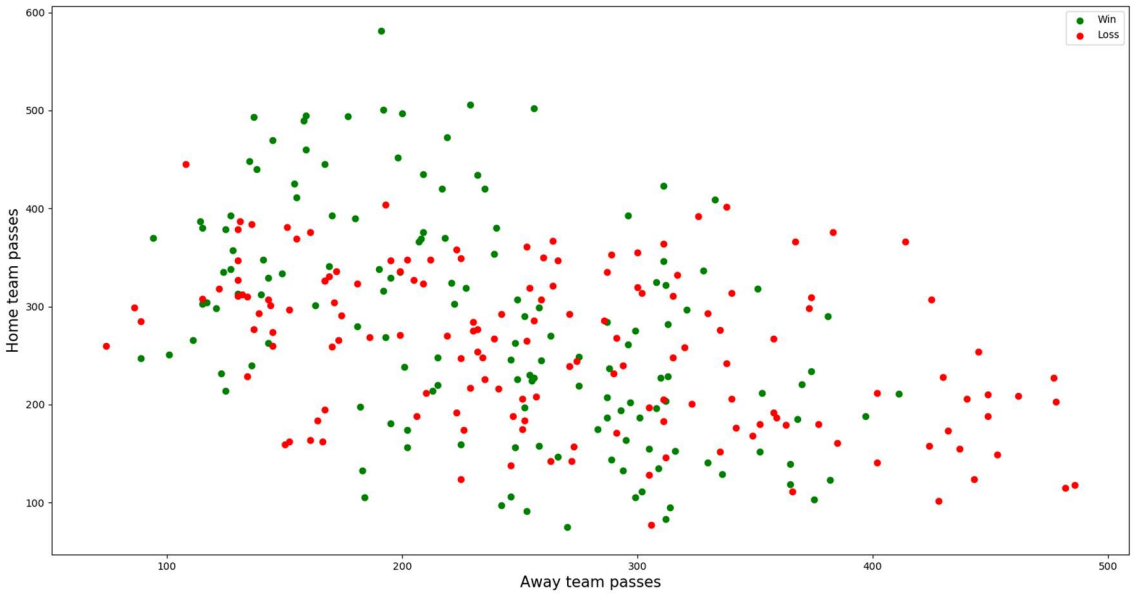
This set of data follows the same trend as the xG graph, which is to be expected as they rely on some of the same underlying data. Furthermore, there seems to be less of a grey area where data points are just as likely to be a loss as a win. This data set also seems to remove some of the seemingly anomalous points.



These heatmaps show the number of deep plays against the number of deep plays allowed with the intensity showing how many matches that happened in. For wins it is notable that there are far more deep plays and less deep plays allowed. Unsurprisingly Every victory required at least one deep play. For losses it seems like the results are much more clustered, with far more hot spots. These features were plotted as a heatmap as there was not enough of a disparity between the minimum and maximum values, and therefore lots of datapoints were in the same position. This is the cause for not being able to plot how a computer would classify each point with regard to deep plays.



PPDA is normally the measure of how intensely a team presses, interestingly it does not seem like PPDA has any impact on the outcome of the match. There are also some anomalous points in particular in the top left, where the home team hardly pressed, but still got a victory.



The final chart demonstrates the relationship between home and away team passes and how they impact the outcome of the match. From this chart one can derive that if a team holds a high proportion of the passes, they are more likely to win.

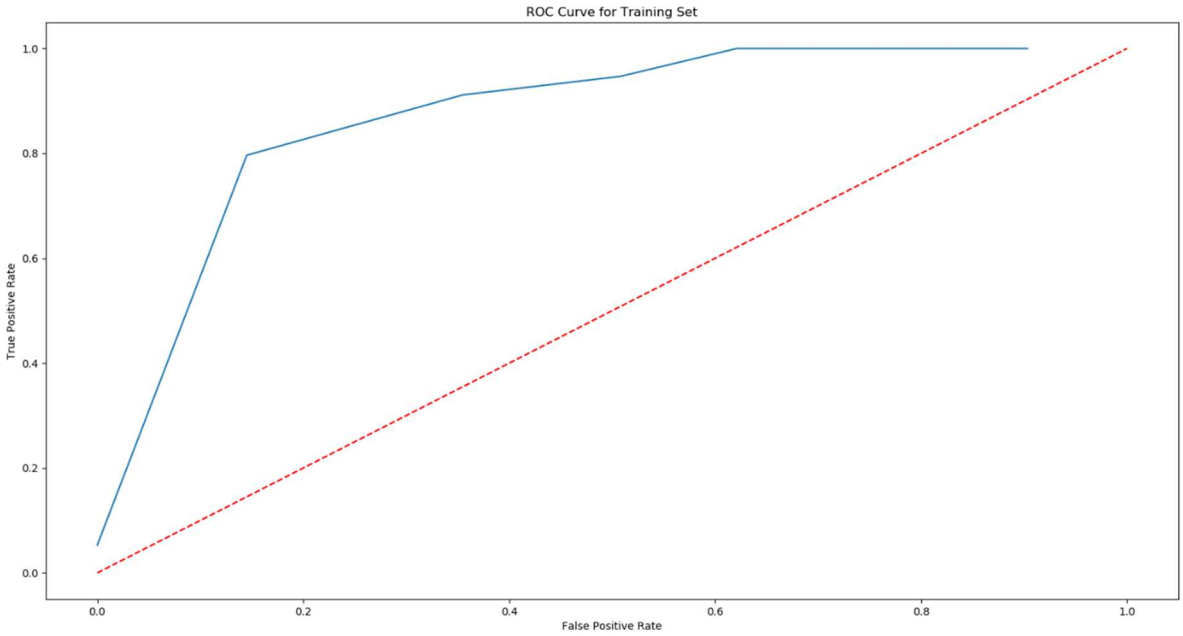
The Models

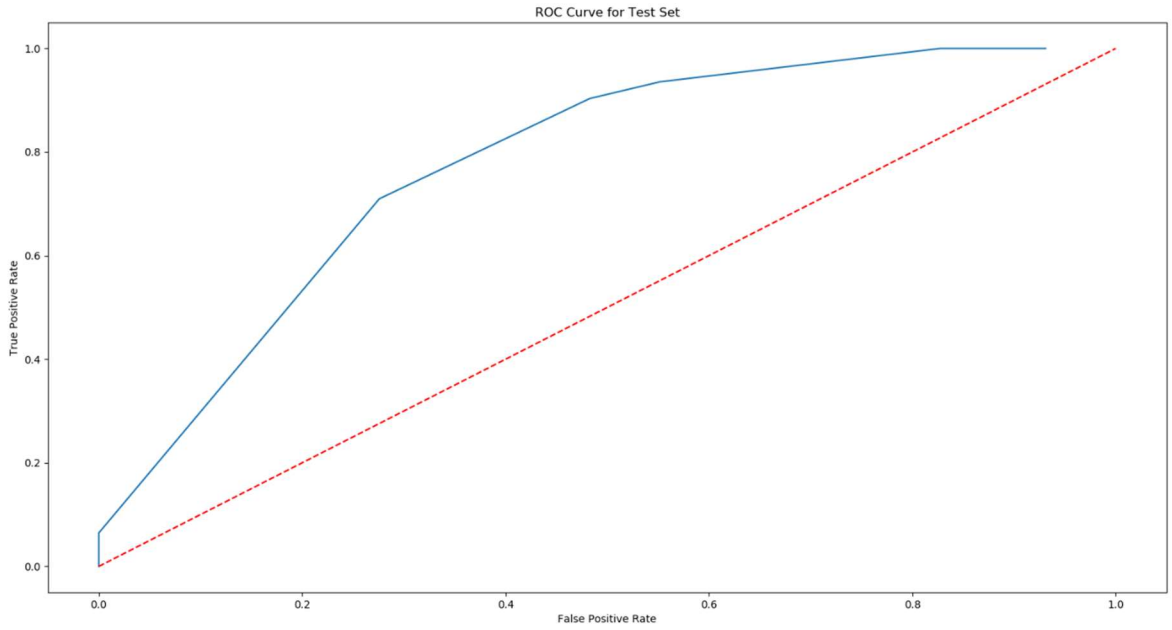
For my classification I have decided that I will use the following six features of the data. xG, xG against, deep plays for, deep plays allowed, home team passes, and away team passes. I have chosen to discard non penalty xG and non-penalty xG against as they I feel they are too closely linked to xG and xG against and therefore would not provide any additional information. Away team PPDA and home team PPDA have been discarded as it does not seem that they have any correlation to the outcome of the match. For the context of this study of a binary classification a win is classed as a positive and a loss is classed as a negative.

Pre-processing

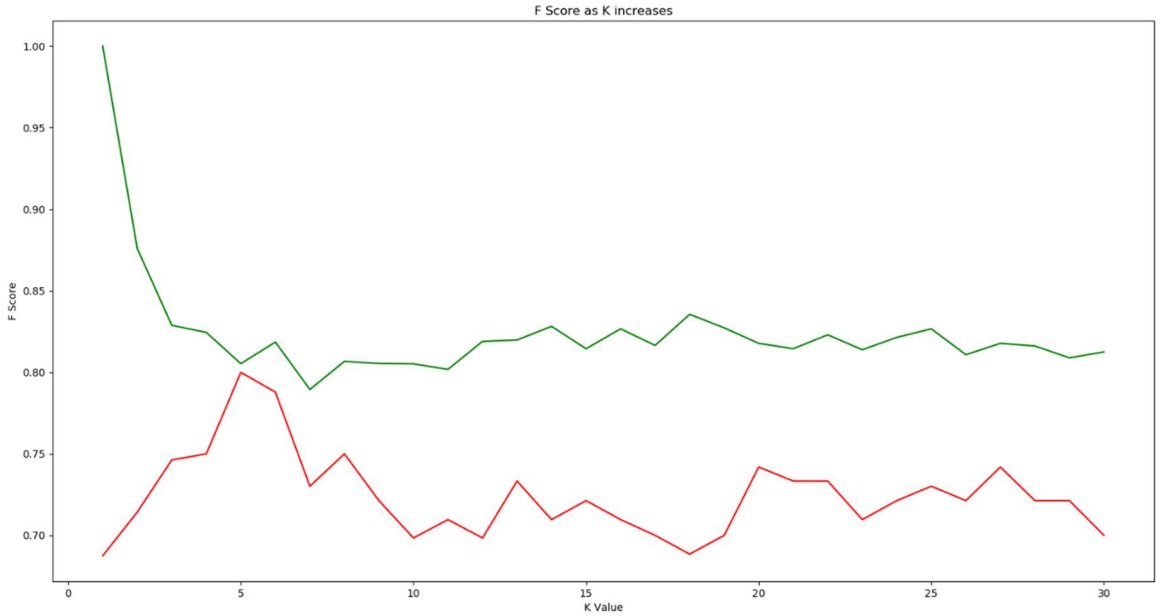
Before performing any training or processing I normalised the data so that each of the attributes would hold the same significance. This was especially important in KNN classification where the use of distance metrics may unnecessarily bias towards attributes with a large disparity between the maximum and minimum values. For example, the number of passes metric will have values in the order of 100s, whereas xG will likely be less than 5 and therefore the passes metric will hold more significance when calculating the distance unless normalised. The next step following this is to separate the training and validation set of the data. To separate the training and testing set 60 random observations were chosen.

K Nearest Neighbours

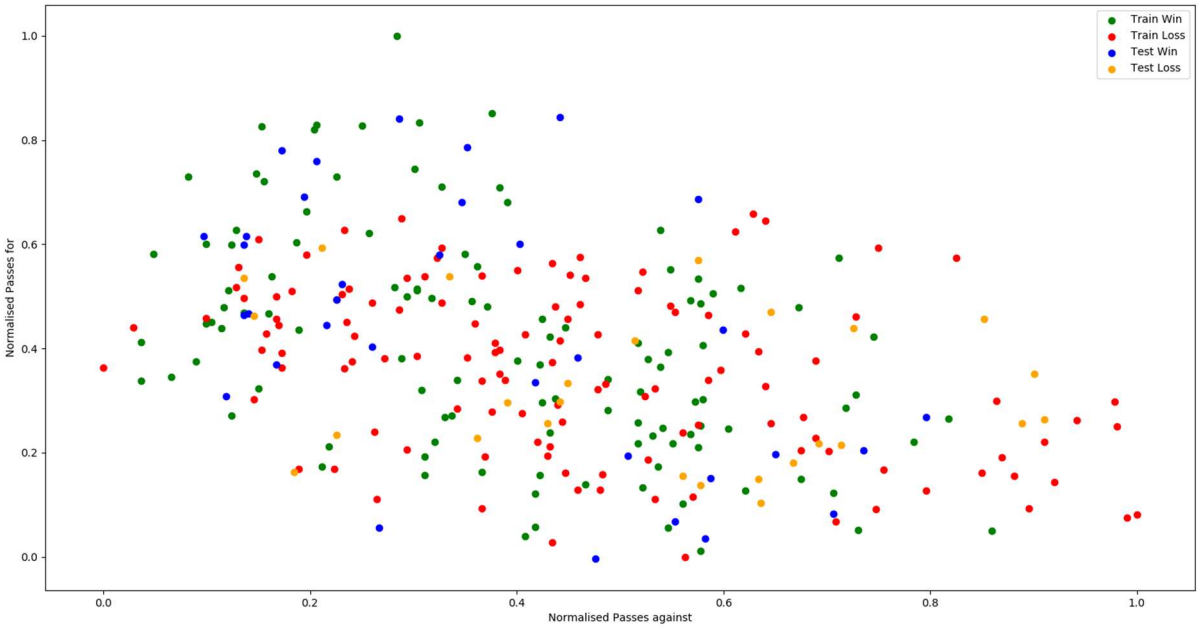


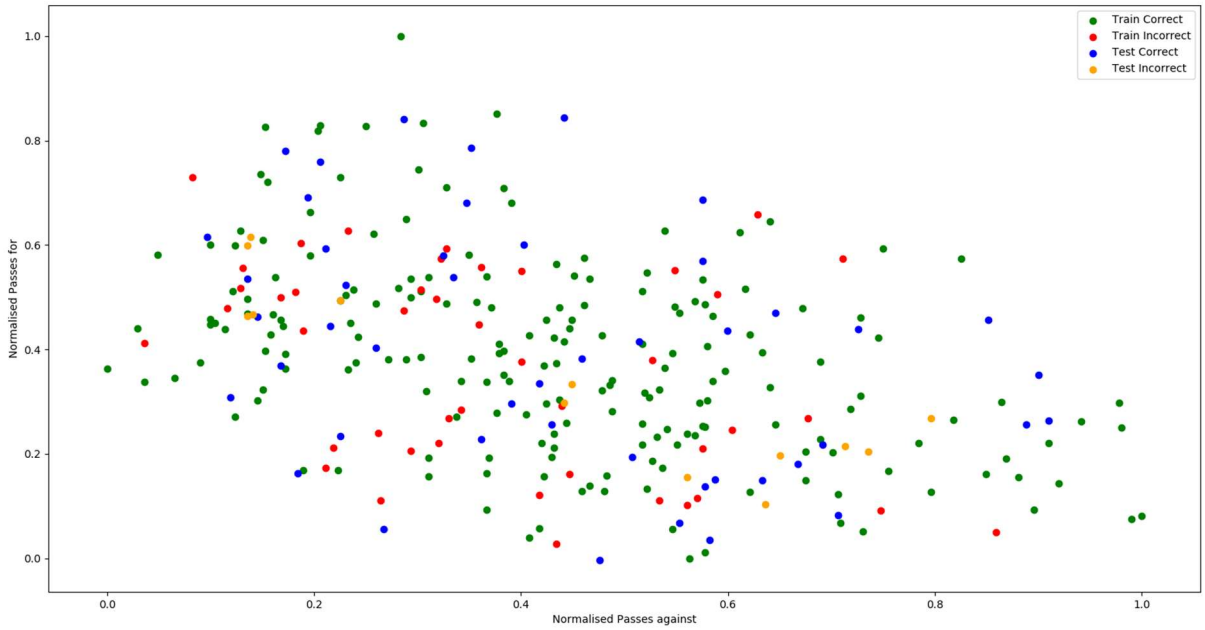


The ROC curve for both the training set and the test set of the data both outperform a random classifier. This indicates that KNN can successfully be used for identifying wins. The ROC curves were generated by using KNN with weighted votes. A probability of a win is then generated and then compared against the threshold for the ROC curves.

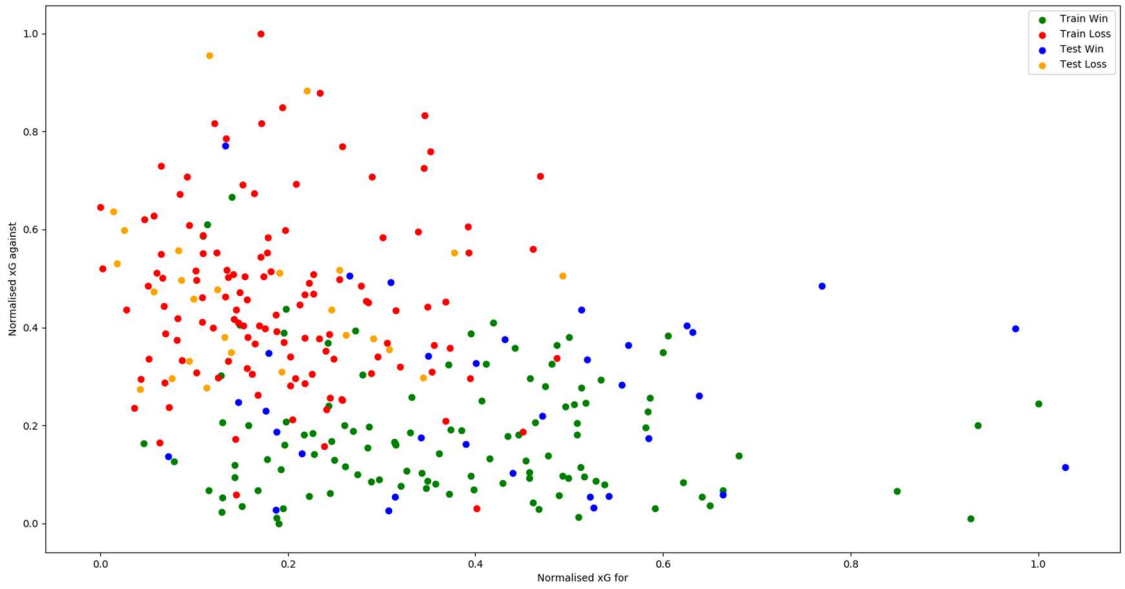


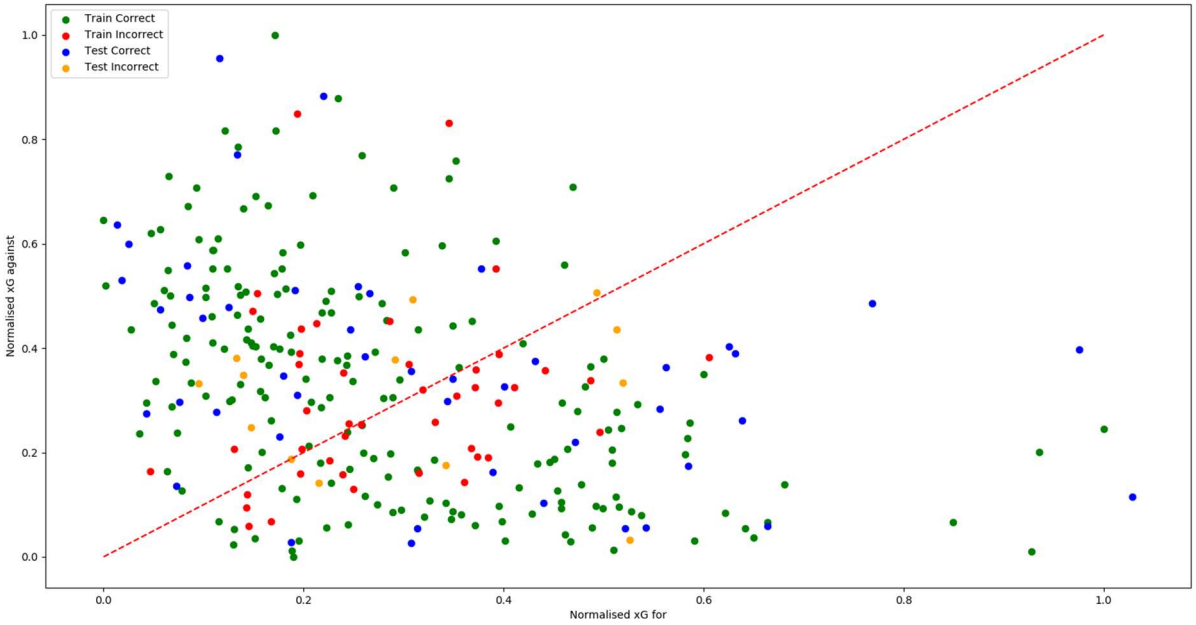
This chart shows that the ideal K value is around 5 as this is when the training and test F Scores are most similar. The training F1 Scores exponentially decreases as that beginning as using the training data to test at low K values will mean that the classification of a point is highly influenced by itself. Overall, the F1 Score for the training data mostly sits between 0.80 and 0.85 and for the test data sits between 0.70 and 0.75.





These two charts demonstrate how KNN classifies the number of passes for and against and the outcome. Most of the incorrectly classified points appear to be in the bottom right area of the chart with a normalised value of between 0.6 and 0.8 passes against for the training set. This indicates that the model is struggling to predict outcomes in these regions and may need more data for more accurate predictions. Overall, the distribution of miss classified points appears to be even across the entire graph. This indicates that it is likely that another feature is likely causing miss classifications.



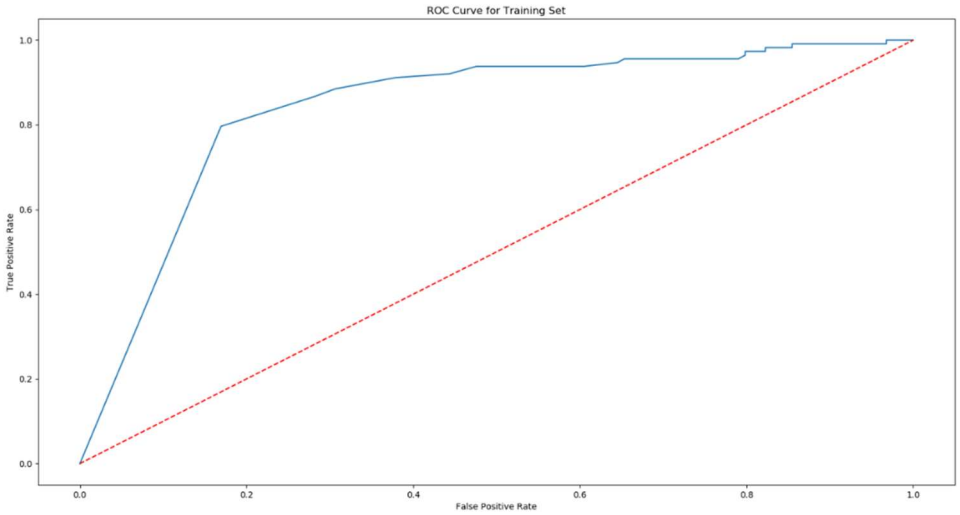


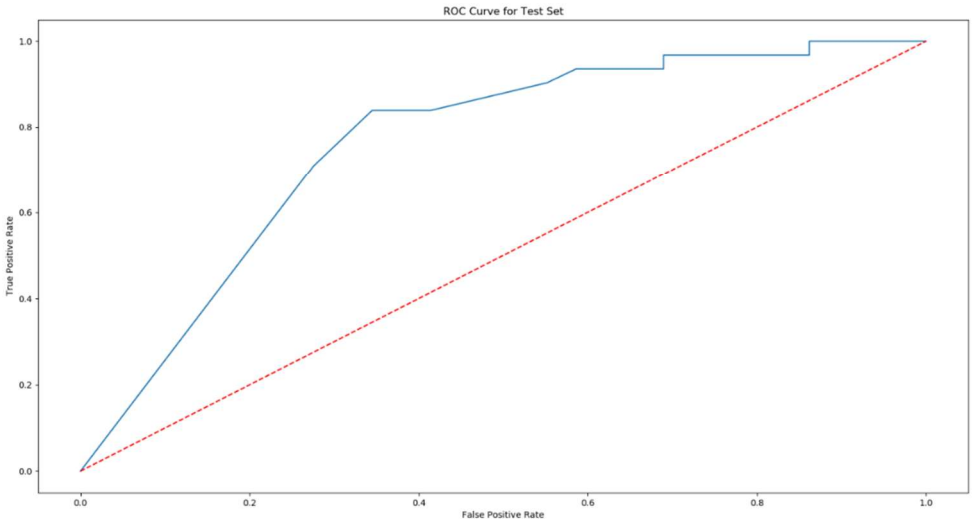
These two charts demonstrate how the KNN model classifies each graph and where the classifications are correct or not. Most of the classifications in the top left are classified as a loss and most of the results in the bottom right as a win. Most of the incorrectly classified points sit along the line X = Y. This is not surprising as this would indicate a very close match where a bit of luck decided the outcome.

Another point to note is that the model used was weighted KNN. Although weighted and non-weighted KNN produced identical results for this report, the use of weighted KNN makes the models more adaptable to changes in the data.

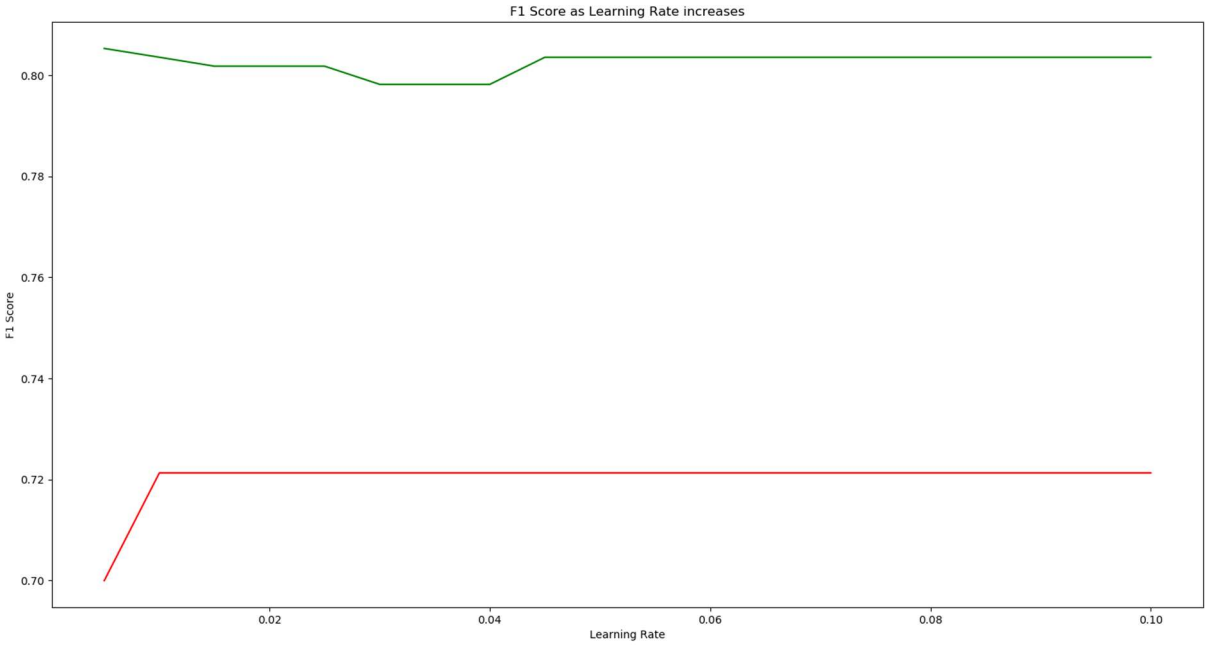
Logistic Regression

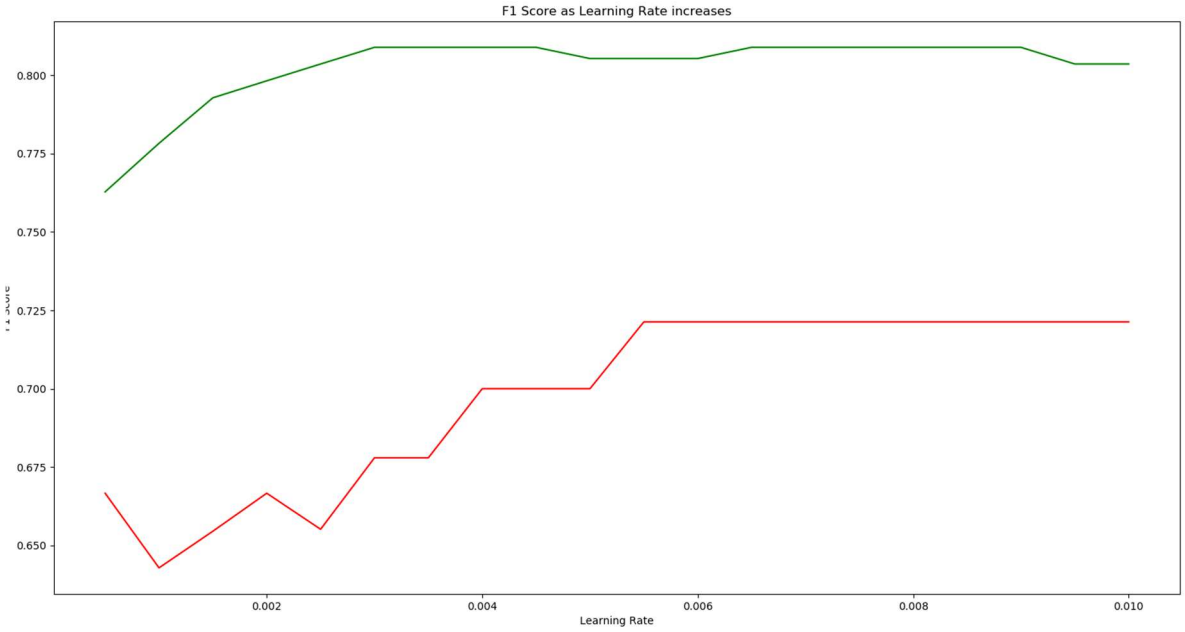
For logistic regression the loss function used was the mean squared error. This function was chosen as it was relatively simple to implement and provides a decent estimate of the accuracy of the model. It is possible that other loss functions may provide better results.



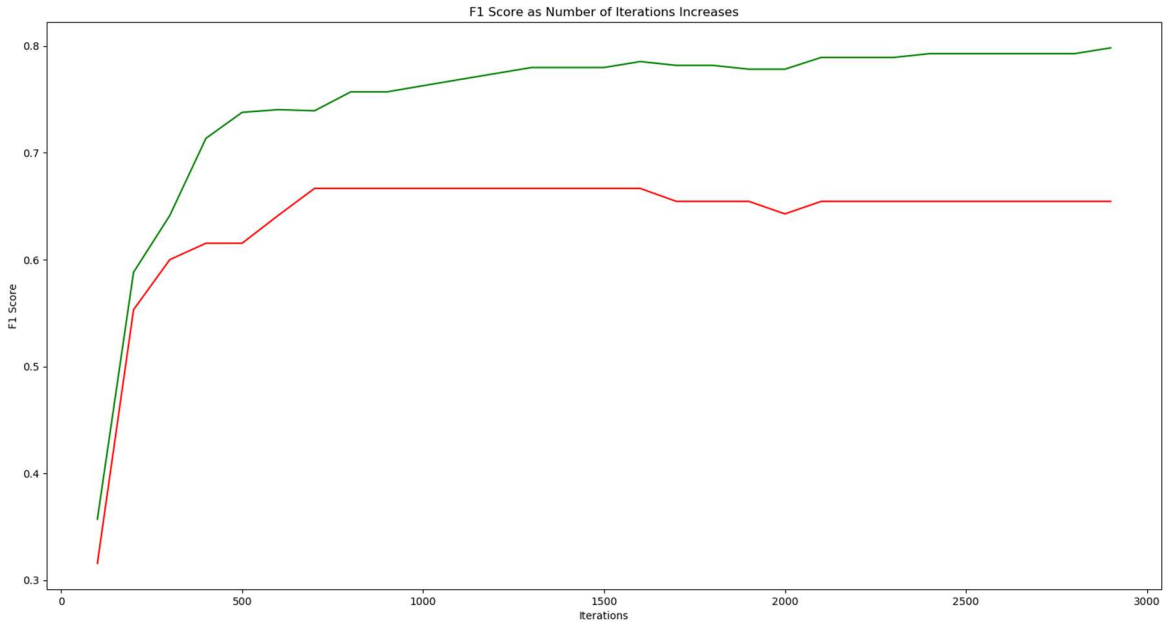


As the model ROC curve clearly deviates from the straight line from 0 to 1 which would resemble a random classifier, logistic regression is much better than just classifying randomly.



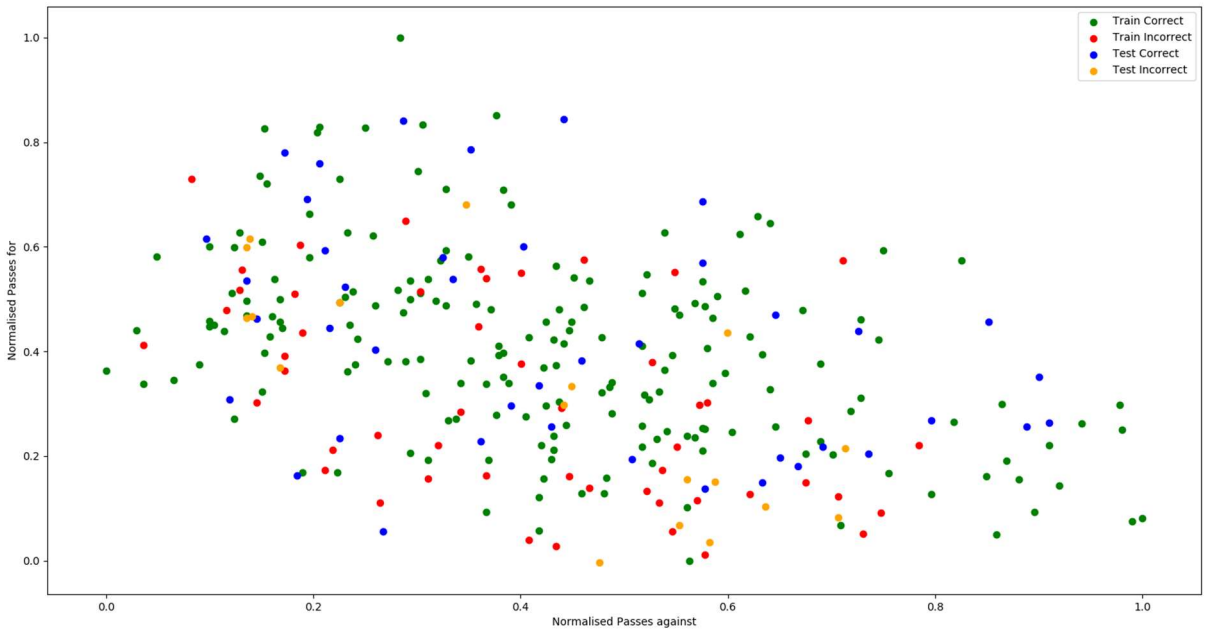


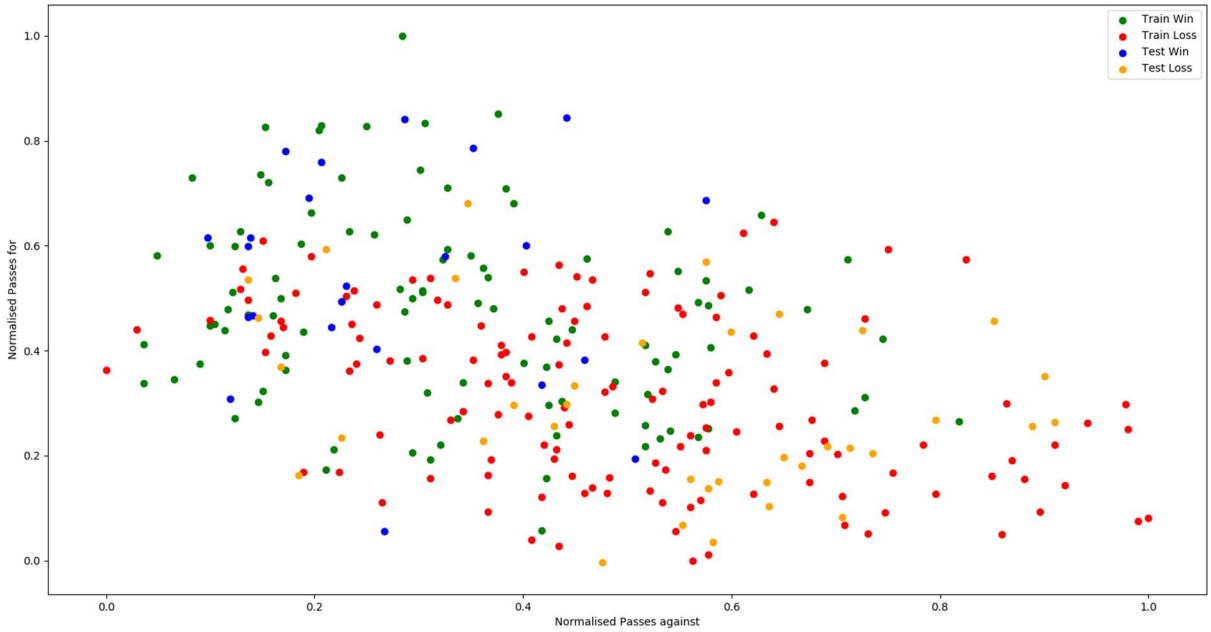
These 2 charts show how the f1 score increases on the test set as the learning rate increases. This indicates that lower learning rates will take a longer time to reach a solution. The number of iterations is kept constant in these graphs at 1000 iterations. From these graphs the ideal learning rate can be inferred to be around 0.006 as this is the value where the test set had the highest F1 Score, but is unlikely to have skipped an optima. Additionally, it is likely that this learning rate will provide good trade-offs between speed and final F1 Score.



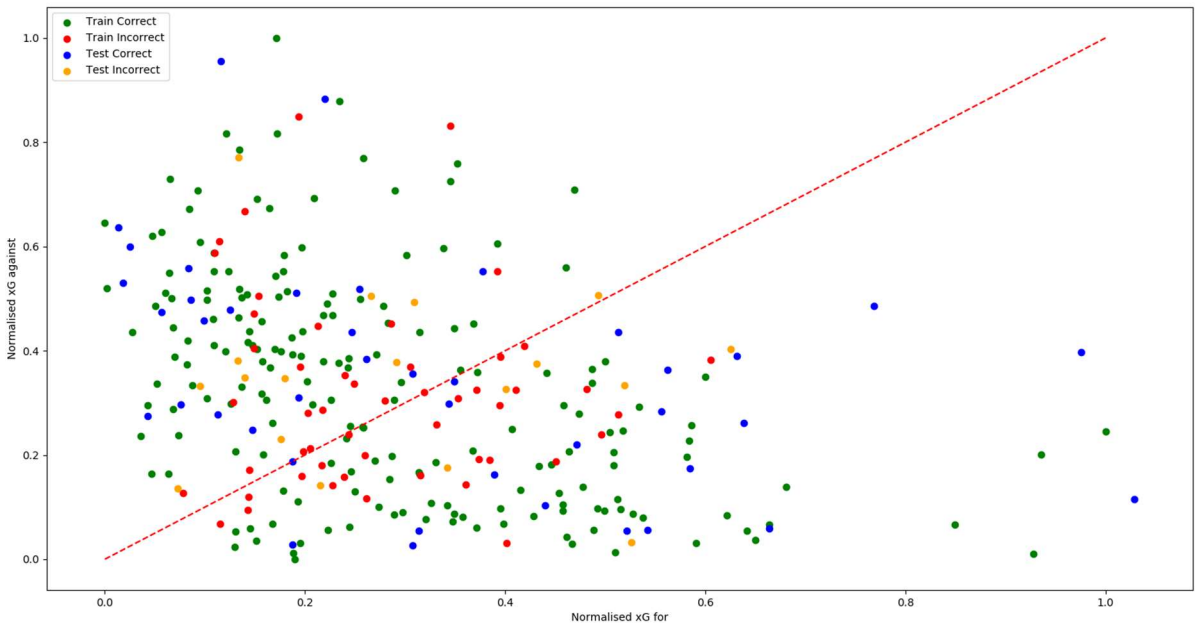
This chart demonstrates how the F1 Score changes as the number of iterations used to train the data increases. This is used to identify when the model begins to overfit to the training data. In this chart we can identify that the number of iterations to train the data should be around 1000. This is because after this the test set F1 Score (red) levels off. However, the training set F1 Score (green) continues to increase, indicating that the model is beginning to overfit, reducing the generalisability of the model. The learning rate is kept constant at 0.0005 during the production of this graph

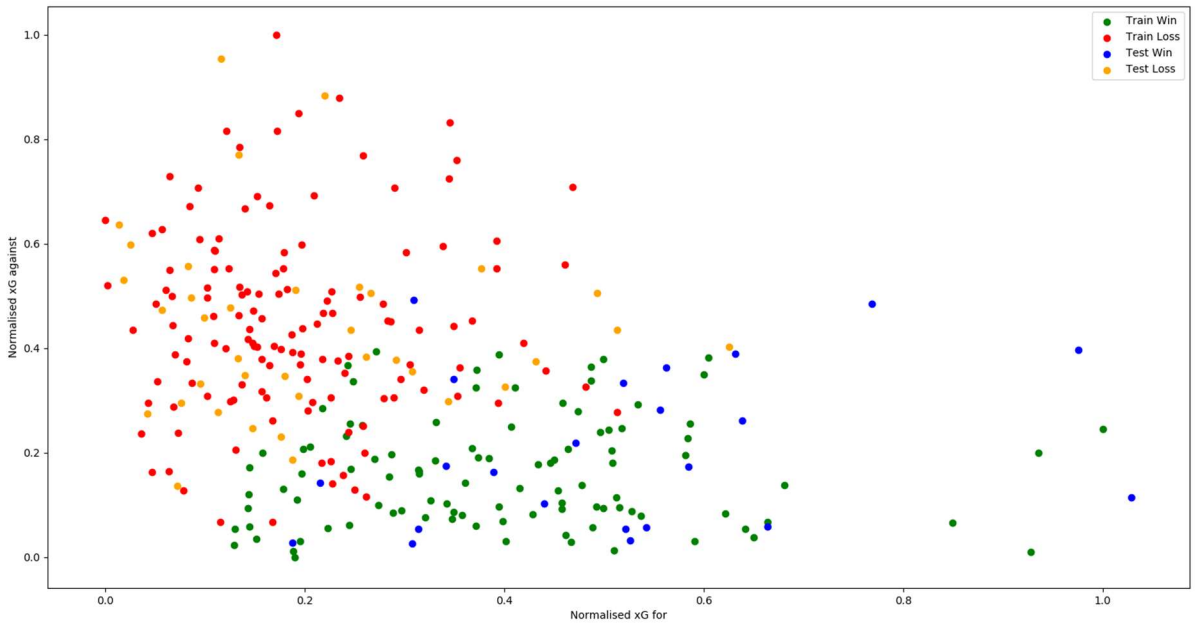
Overall, we can identify that the parameters for logistic regression are a learning rate of around 0.0005 and the number of iterations around 1000. However, the maximum F1 Score for the training set is around 0.80 and the maximum F1 score for the test set is around 0.65.





Once again, the passing distribution is similar with the top left area appears to be classified more as wins and the bottom right more as losses, like KNN. However, the errors are distributed mostly around the edges of the cluster that the data forms. This indicates that logistic regression struggles with values that are towards the extremes of what has already been seen.





Like with KNN, there is a visible cluster of incorrect classifications that lies around the line Y = X. However, unlike with KNN these errors seem to be more spread out. This indicates that the logistic regression model is not fully influenced by the difficulty of distinguish between wins when xG and xG against are very similar. It is likely that xG is the driving factor as xG for and against are given the largest weightings and therefore their value is most significant.

The weights that logistic regression applied were:

- xG for – 1.037
- xG against - -1.239
- deep plays for - 0.400
- deep plays against - -0.466
- passes for – 0.159
- passes against – 0.067

This shows that for all the features that are against the team lower values are more likely to result in a win. xG appears to be the most important feature, having been given the largest magnitude of weighting. Passes for and against have the lowest magnitude and therefore are least important in deciding a win.

Conclusion

Measure	KNN	Logistic Regression
Training Data		
Accuracy	0.814	0.784
Precision	0.805	0.804
Recall	0.805	0.726
False Positive Rate	0.177	0.161
F1 Score	0.805	0.763
Test Data		
Accuracy	0.783	0.700
Precision	0.765	0.782
Recall	0.839	0.581
False Positive Rate	0.276	0.172
F1 Score	0.800	0.667

These values show that both the models generalise well as both only have a slight dip in performance in all areas on the test set. Therefore, the training can be applied to unseen data effectively. It seems that the KNN model is more generalisable as there is less of a reduction in performance between the training and test set.

Overall, I believe that KNN is the more effective model. This is because KNN outperforms logistic regression in almost all metrics using both the training and test data. However, Logistic Regression had a lower false positive rate for both the training data and test data. This does not impact my evaluation as the impact of false positives/negatives is minimal in a real-world setting. If more data becomes available, it may be possible that logistic regression becomes a more effective technique as there are more parameters that can be fine tuned to generate better results. Additionally, different loss functions may also lead to better results from logistic regression which could improve performance.

In general, both models perform well, with KNN producing similar performance in both the training and test set, indicating that this model is very generalisable. In the future the performance of the models could be improved by gathering more data from other leagues as well as using more features to differentiate between wins and losses which were not provided in the data set such as distance run. The dataset also has the limitation that some major factors that can affect the outcome of a football match are not represented. For example, own goals which are not represented in the xG data, can have a huge effect on the outcome. Another limitation is that these models both assume that both teams go into the match with the same likelihood of winning, which is not true. Some teams can spend far more than others, meaning they have better players who are in turn more likely to be able to convert low xG chances and profit more from low numbers of deep plays. The best way to improve either model moving forward would be by obtaining more data. Future avenues of research could also include investigating the feasibility of predicting a draw or predicting the

outcome of a match as it is being played. Further analysis should be performed with other classification techniques such as neural networks to identify the maximum accuracy with which predictions can be made. The main objective of the analysis has partially been achieved with the discovery of a model that can successfully predict the outcome of a football match with an accuracy of 0.783 when evaluating between a win or a loss.