Building a model to determine the difficulty of passes made in a football match from match-event data

Sachin Mishra and Dinesh Adhithya

Abstract

Using data collected from 3 football leagues and 1 world cup, we used several machine learning algorithms to determine the probability of completing a pass. Our models produced upto a 0.888 AUC-ROC curve and an 81% accuracy in determining outcome of an attempted pass. We also used this metric in a sample match from the dataset to further analyse the roles of the players and to check the sanity of the model.

Introduction

In a football match, credit to a goal scored is generally given to the goalscorer alone. An “assist” is awarded to the player providing the last pass to the goalscorer which immediately leads to the goal being scored. This is inherently a flawed metric in determining offensive contributions of a player, as deep lying players often do not perform these actions but play pivotal roles in offensive buildup. Additionally, not all goals require the same amount of skill to execute. A popularly used metric, xG is used to determine the probability of scoring a goal given the location of a shot. This metric obviously favours strikers and a similar metric, xA favours the players who often make the final pass.

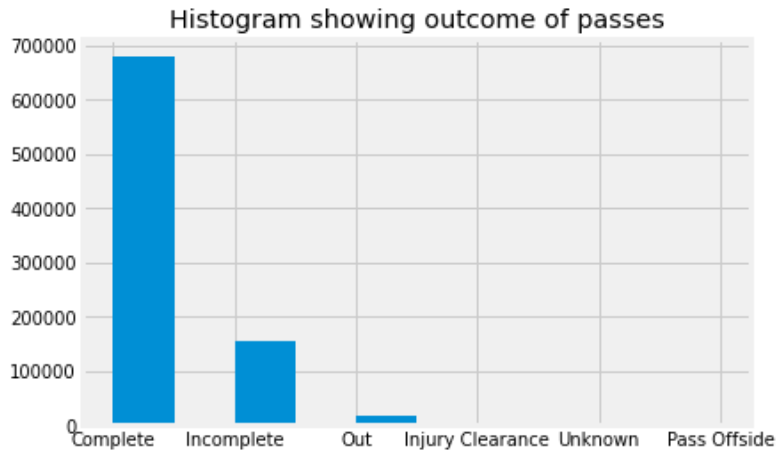
xGChain is a metric that sums the xG of all the possessions a player was involved in to give a total score that represents the involvement of a player in offensive buildup. This metric still favours players with high xG and xA but allows for some underappreciated players to shine through. xGBuilgup is similar to xGChain but it excludes shots and assist qualifying passes as contributions to purely observe the role of a player in offensive buildup. A glaring issue with these metrics which xG has but these do not, is the difficulty of the action that is attempted. A simple horizontal pass or a backpass is not as challenging as a pass through opposition lines or lob across the pitch. These metrics give an equal weightage to each action which is often not appropriate, especially in long possessions with a large number of passes which lead to a shot.

Another issue is that the number of involvements are not taken into account- a player can (hypothetically) make 7 relatively difficult passes in a possession and will get the same contribution to the xG as a deep lying midfielder who made an easy back pass to a defender. This means that these metrics are often not balanced in smaller sample sizes like a match or a few matches, but may be useful in large sample sizes, for example over the course of a season of football.

We intend to develop metrics that tackle these very flaws with the current metrics. Using event wise match data made available by statsbomb, we studied the characteristics of over 8,50,000 attempted passes to train models to determine the probability of an attempted pass being completed. Each pass is described by its length, duration, angle, start & end locations and outcome.

Data Wrangling

The event level data is available in .json format for over 850000 passes. We first unravel the data by plotting the distribution of passes since there will be far more completed passes than incomplete passes. The histogram is given below:



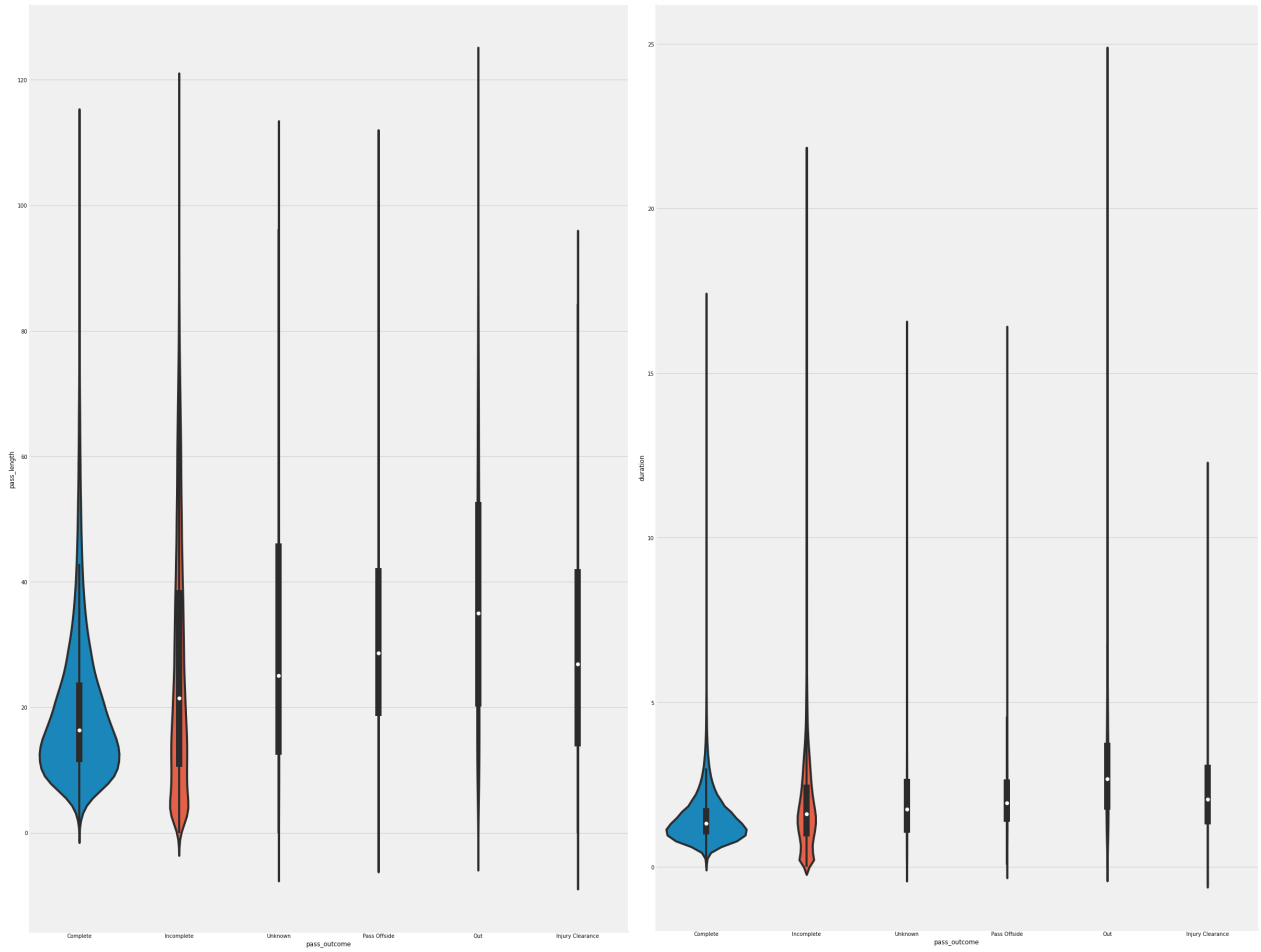
Clearly, there are far more complete passes than incomplete passes and almost insignificant amount of injury clearances and offsides. We grouped all non complete passes as incomplete since injury clearances are quite rare and insignificant in comparison to the others, while the rest are still failed passes. To avoid causing any wrong inferences, we reduce data to make it unbiased. This resulted in a working dataset of around 350,000 attempted passes.

Some additional data that would definitely help model the difficulty of passing better would be a positional information of all the players on the pitch when a pass is attempted, so as to determine the extent of coverage the passer and receiver are subjected to and account for the extent of pressing that the opponent team is executing. But, due to the large size of the dataset, we expect certain positional trends of coverage to translate onto the success rate in those regions, for example a pass in the defensive half is probably going to be significantly easier than a pass deep into the opponent's half since there are more opponents vying for the ball in their defensive half. This is confirmed by finding the percentage completion of passes aimed at the defensive half (88.33%) when compared to that in the final third (63.9%). This is further exaggerated as one moves closer to the opponent's goal, with the accuracy falling to 48.66% in the final sixth.

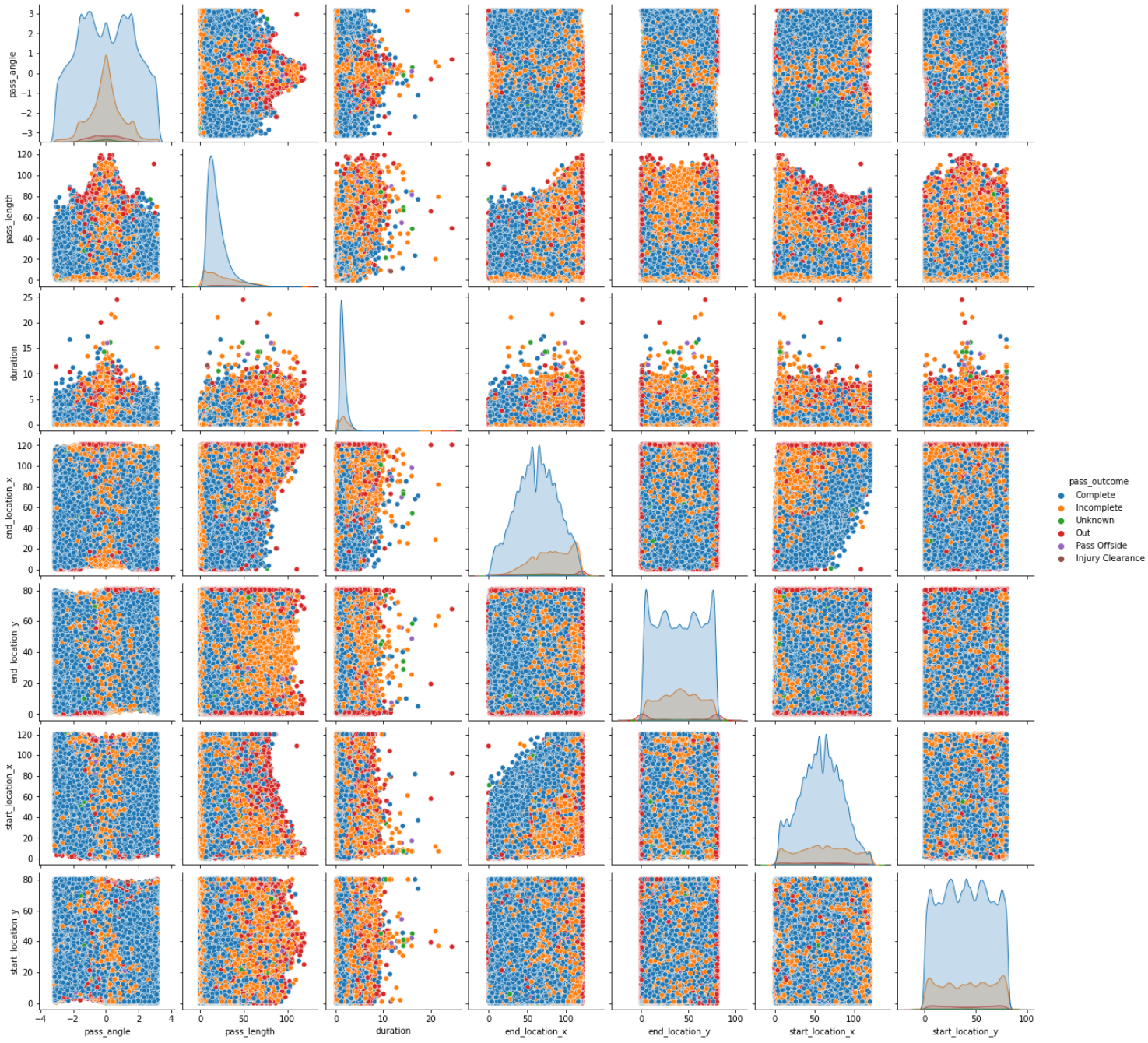
The given data lacked an outcome for completed passes which we fixed by replacing the pass outcome entry NaN with “Complete”. Since the data had positional values in cartesian values stored in a single variable, these had to be converted to separate x and y coordinate floating point values. We then made the result a binary result with zeros and ones for ease of use in calculating probability while training the model. We used the sklearn MinMaxScaler to preprocess the data and then shuffled and split the dataset into a train and test dataset using sklearn toolkit’s train_test_split.

Data Exploration

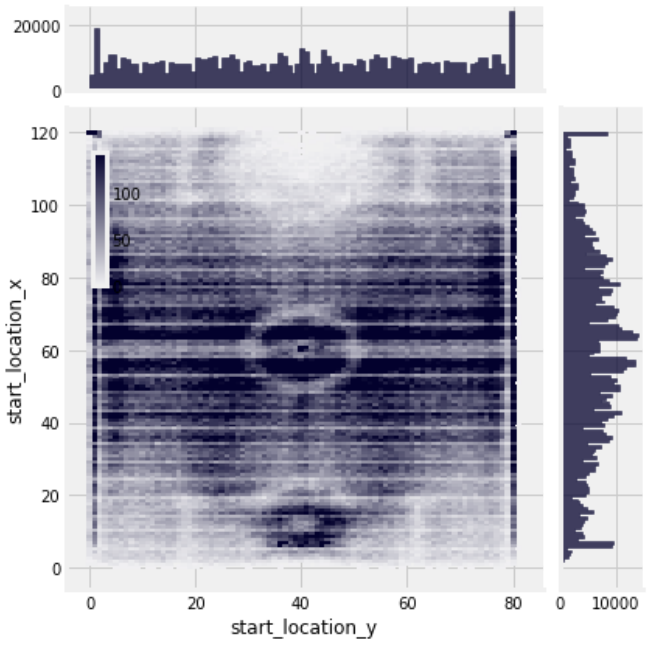
Frequencies of various pass outcomes plotted against pass length(left) and pass duration(right)



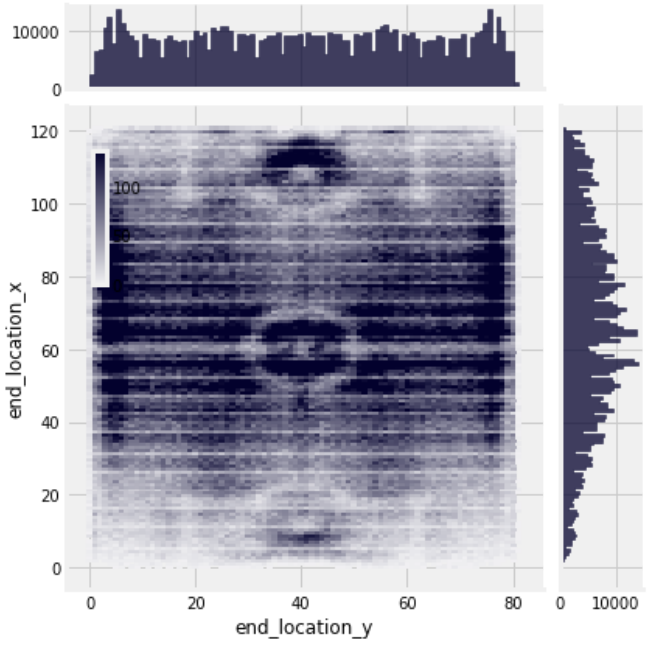
Pair plot of various parameters studied:



Heatmap of pass start locations



Heatmap of pass end locations

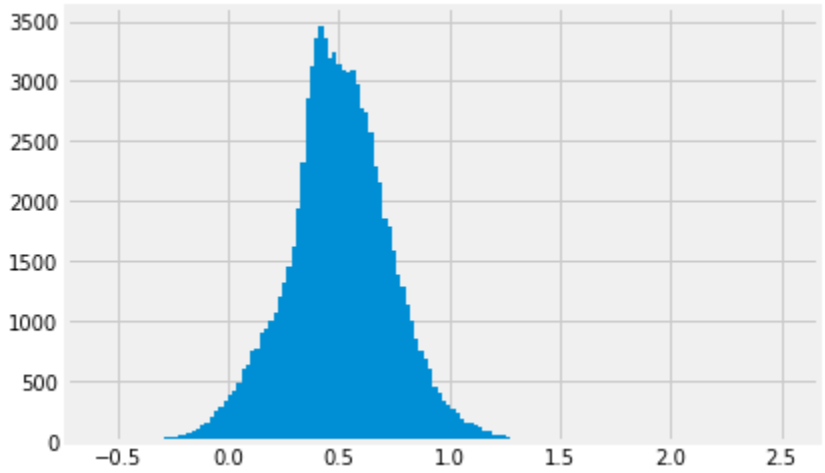


Model Training

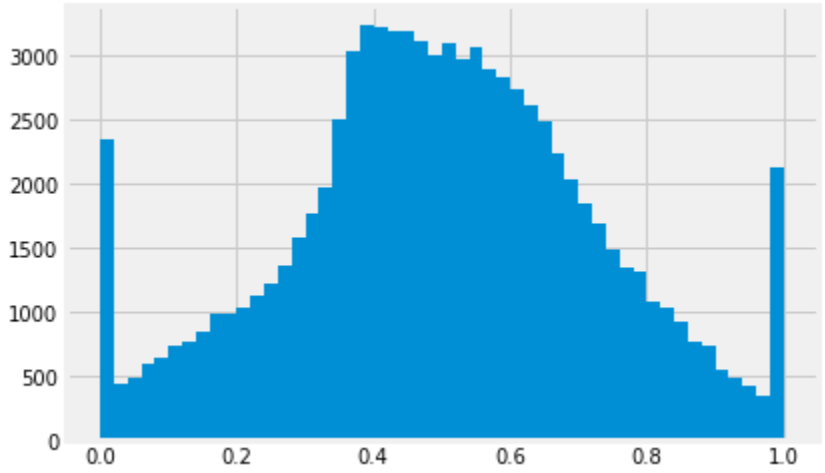
Linear Regression

We used scikitlearn's Linear Regression algorithm to try and predict the pass outcome which is an ordinary least squares linear regression model. Since the model is a regressor, it only tries to fit each variable to a line by minimizing the residual sum of squares and gives a

predicted float value which is interpreted as predicted probability, since all the training data has results in either 1 or 0 form, corresponding to made or missed passes respectively. Due to this, there are certain discouraging results like negative results and results that exceed 1, which are to be expected when such a simple regression algorithm is used to give binary output. But, since the algorithm gives the output based on the trained input data, it predicting high output values being classified as 1 and just considered as very confident guesses by the regressor we clip the output and reduce the graph to the range (0,1)

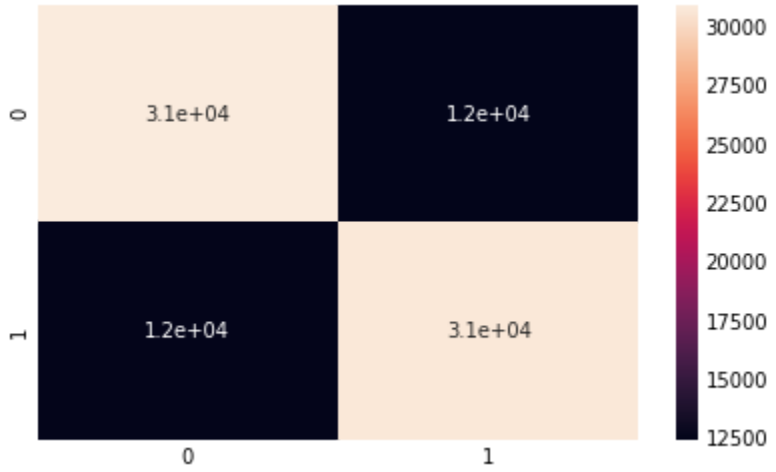


This is the plotted output of the regressor. As one can see it does not fit the (0,1) boundary. But as explained above, considering these outputs as 0 or 1 should not be a problem. This is plotted in the below.

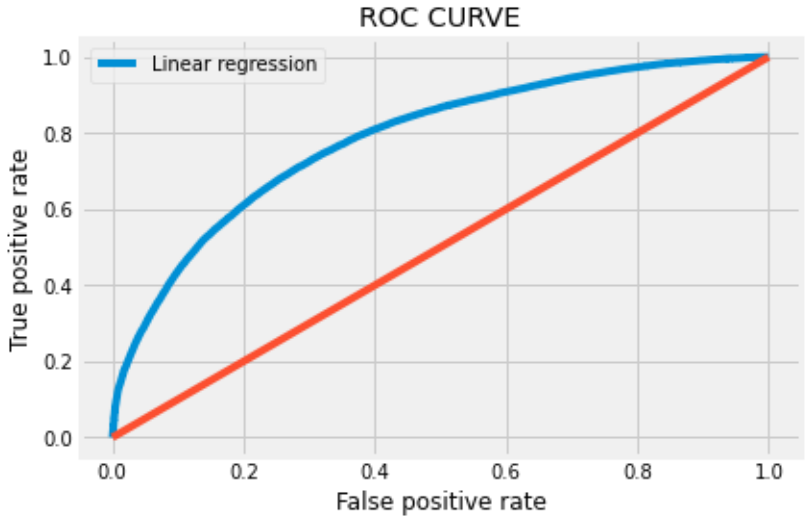


As one can see the regressor is not very confident for a significant chunk of the passes and gives an output very close to 0.5. Since we are considering it as a guess as 0 or 1, we separate the values above and below 0.5 and label them as 1 and 0 respectively. Nevertheless, the closeness of the predicted “probability” to the mean value indicates poor confidence in the predictions. We saw that the average difference between the prediction and actual value was almost 0.4, which falls to 0.39 when the probabilities are forced to 0 and 1. This again shows

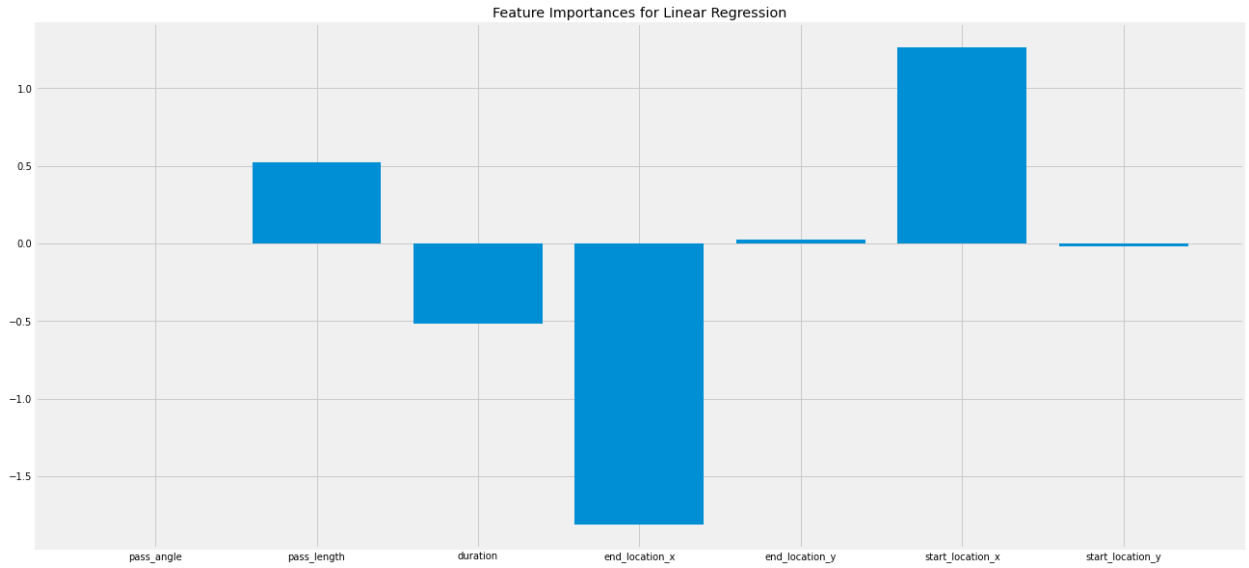
that the model is not very confident. We then plotted the confusion matrix of this prediction where we produce a heatmap of predicted 1s and 0s against actual 1s and 0s.



Following is the ROC curve for the regression model



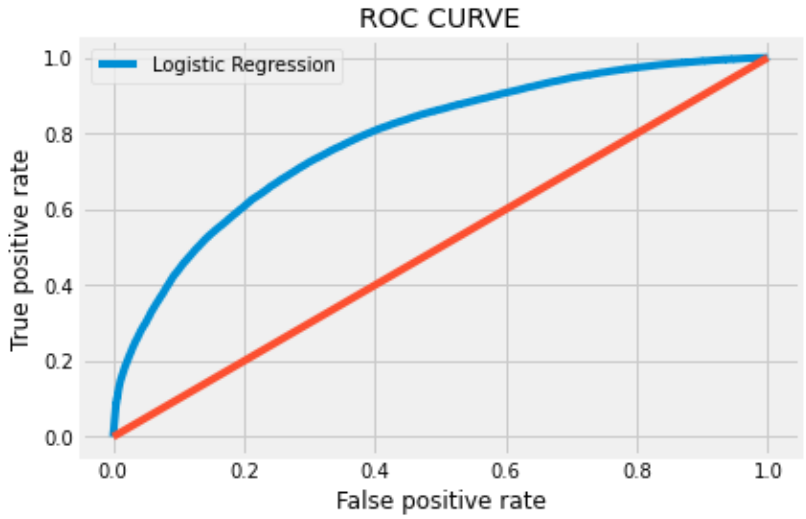
As one can see from both the confusion matrix and the ROC curve that the model works to a certain extent and is capable of predicting the outcome of an attempted pass. We obtained an accuracy score of 0.7132095527492461 and a AUC-ROC score of 0.7132086081170325, which indicates that it is a fair bordering on a poor way to classify the passes. We further analyse the feature importances that the linear regression algorithm picked up over the training to make a sanity check on the models methods.



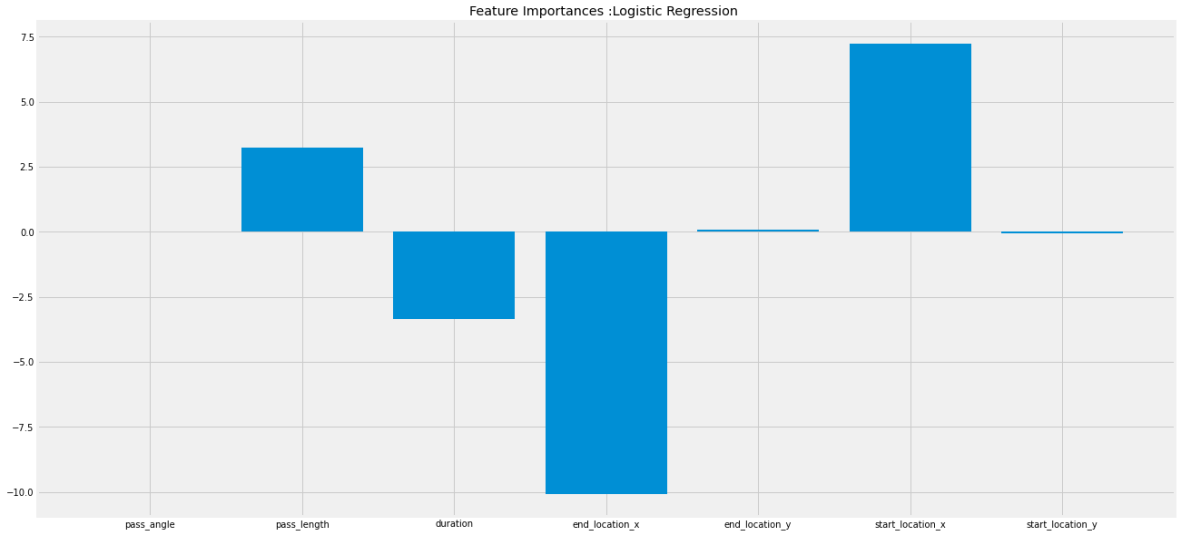
Here we observe that the model gives little weight to the y-axis coordinates-which is a good sign since the width of the pitch lies within 0 and 80, a linear scaling to the y coordinate should not come up during training. A large negative coefficient for end location is also very encouraging, since attempting to send a pass deep in the opponent's half is not a high percentage pass. The large positive coefficient for the start location may be due to the larger number of passes attempted in the midfield being successful when compared to the lower number of passes in the attacking third being attempted and failing. A positive weightage for pass length is difficult to interpret but the negative coefficient for duration shows that the model recognises that quick crisp passes are higher percentage passes when compared to slower ones.

Logistic Regression

We used sklearn's Logistic Regression algorithm here, which uses a lbfgs algorithm to optimise the loss. Since the logistic regression algorithm is a classifier, we do not need to further interpret the predictions and we can directly analyse the accuracy, ROC and confusion matrix. We see a mean difference between prediction and actual value of 0.288, which compared to the linear regression predictions performs slightly better.

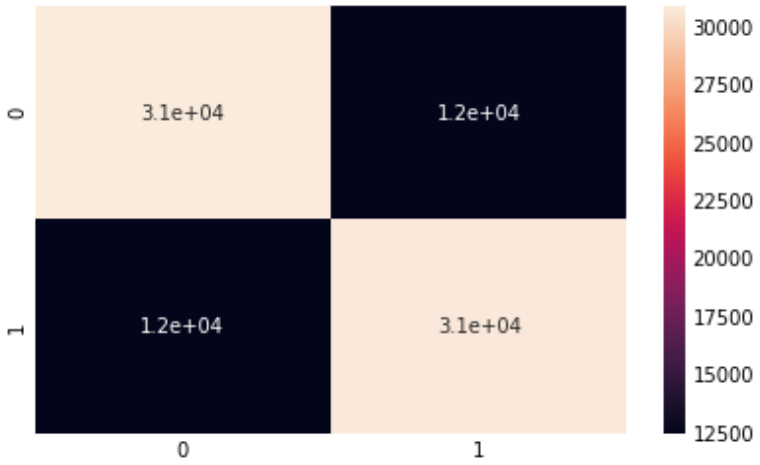


Following are the feature importances arrived at by the model:



Some encouraging observations are the negligible weight for the Y coordinates, large negative weight for destination X coordinate. Unfortunately the model has a positive weight for the pass length which is unintuitive looking at the violin plot of pass outcomes wrt distance.

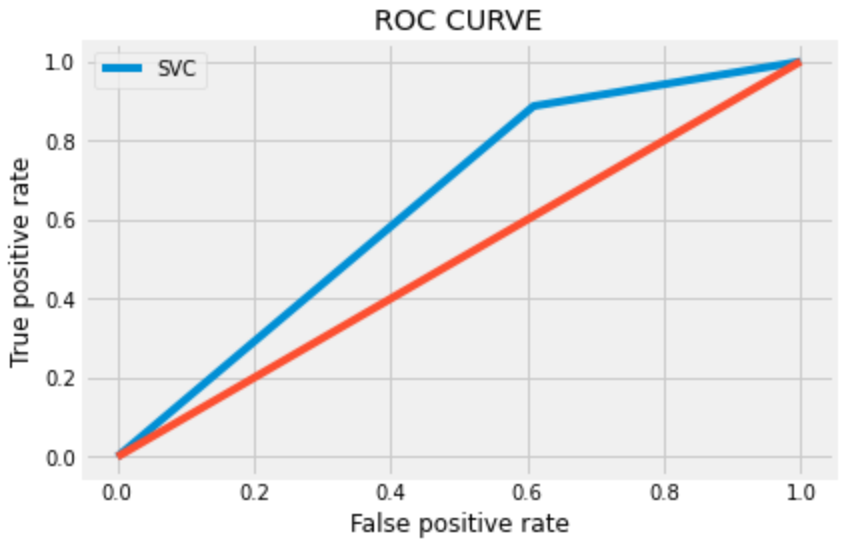
By observing the confusion matrix and ROC curve of the logistic regression, it is hard to see a significant difference between the linear regression model and the logistic regression model in terms of the quality of predictions



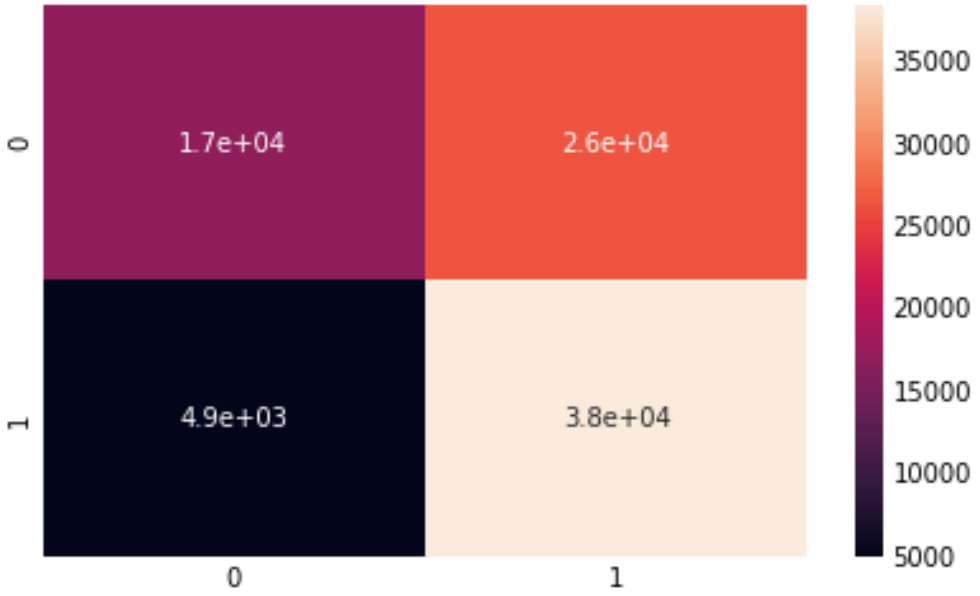
This is further validated by the accuracy score and AUC-ROC score which turn out to be 0.7132095527492461 and 0.7132086081170325 respectively. This, like the linear regression model, shows that the logistic regression model does not perform very well but is capable of making some sort of guess as to the outcome of the pass.

Support Vector Classifier

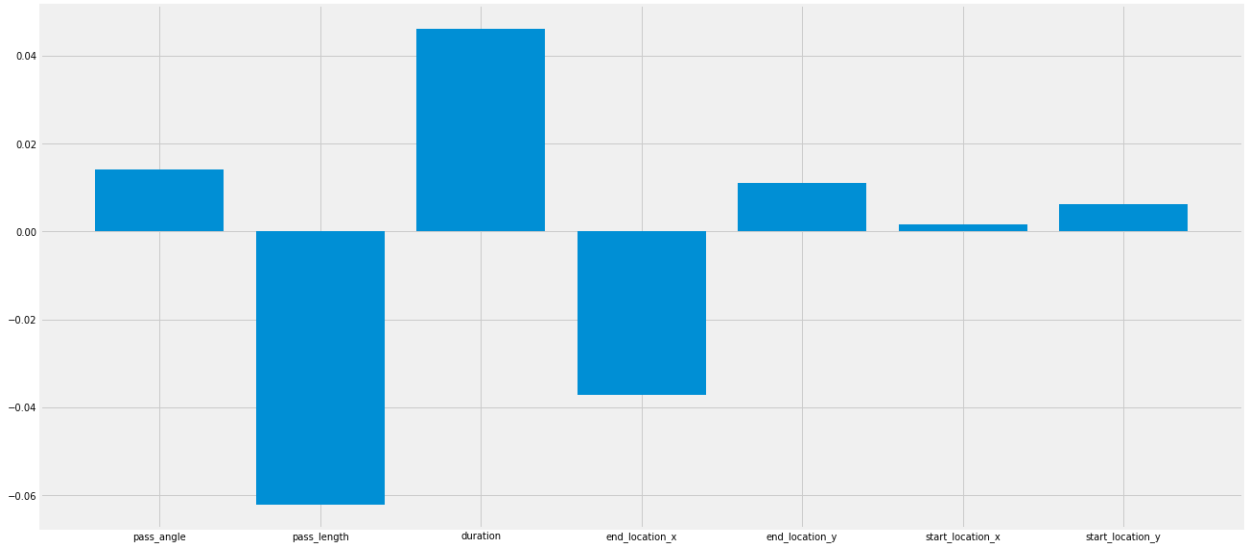
We used a scikitlearns support vector classifier based on the support vector machine library. Linear kernel was used at a max iteration limit of 350. The model still performed poorly, having the worst accuracy, AUC-ROC score and confusion matrix amongst all the models used.



Analysis of the confusion matrix shows that the model classifies passes as successful very easily and is reluctant to classify an attempted pass as a failed one, which is seen by the high false positive rates and simultaneously lowest false negative rates.

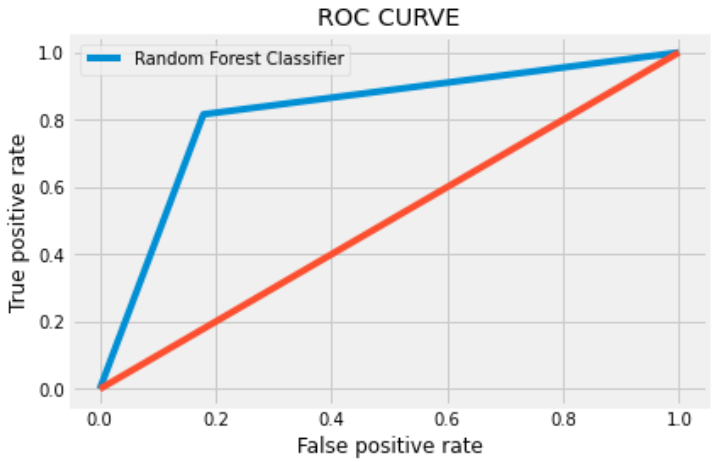


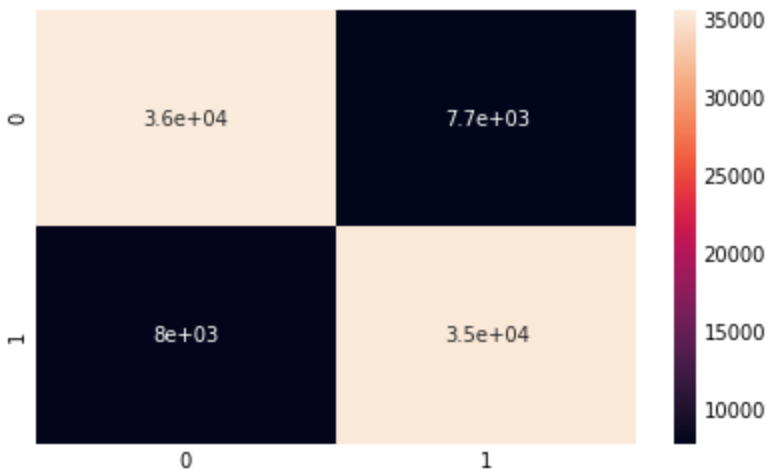
A look at the feature importances of the SVC shows very non intuitive conclusions - more weightage towards Y coordinates than start X coordinate, along with large positive weight for pass duration, show why the model is not very good at predicting pass outcomes, which is reflected by the low accuracy (0.6388) and AUC-ROC (0.639) scores.



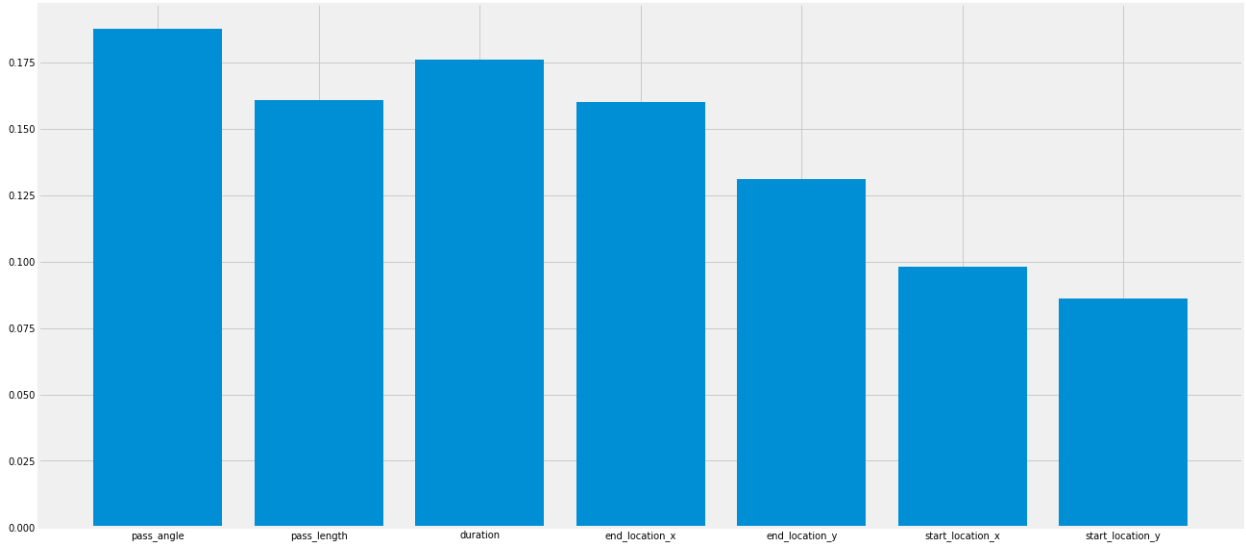
Random Forest Classifier

We used the scikitlearns Random Forest Classifier which performed at an impressive accuracy score of 0.8186 and AUC-ROC score of 0.8186. Below are the performance graphs of the model:





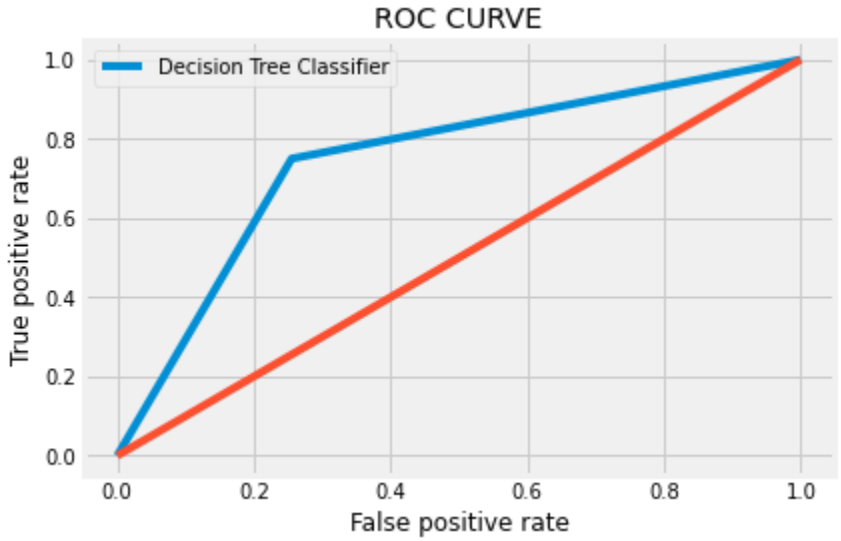
The confusion matrix shows both good positive and negative selections and the feature importance weightages are plotted below:

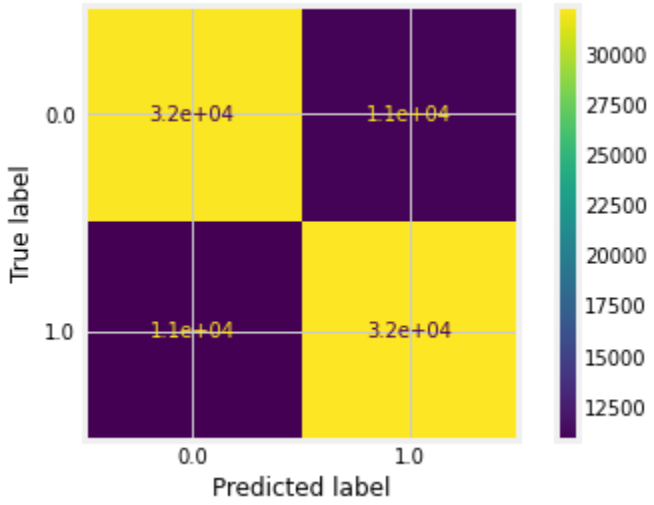


Some interesting features include the rather high weightage towards destination Y coordinate, which may be complex enough to understand the symmetry of the y coordinate and see the high percentage passes in the wings when compared to the centre, which can be observed in the pairplots of the parameters.

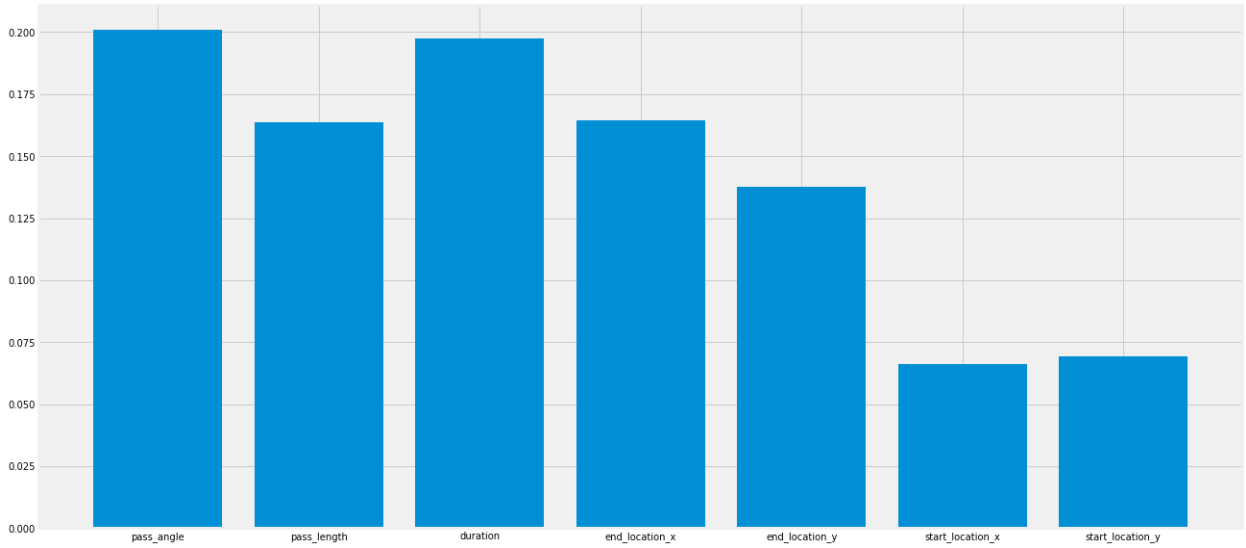
Decision Tree Classification

We also trained a decision tree classifier based on the scikitlearn library. The model performed at par with the linear regression and logistic regression models, with Accuracy and AUC-ROC scores of 0.7474. Below are the performance graphs for the model:



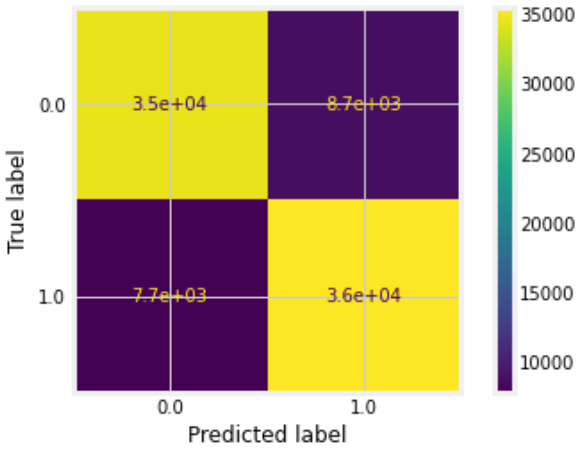


By studying the feature importances of the decision tree classification, we see that the model has similar weightages attributed to the parameters when compared to the Decision Tree Classifier, although the start location coordinates did receive more attention from the DTC model.

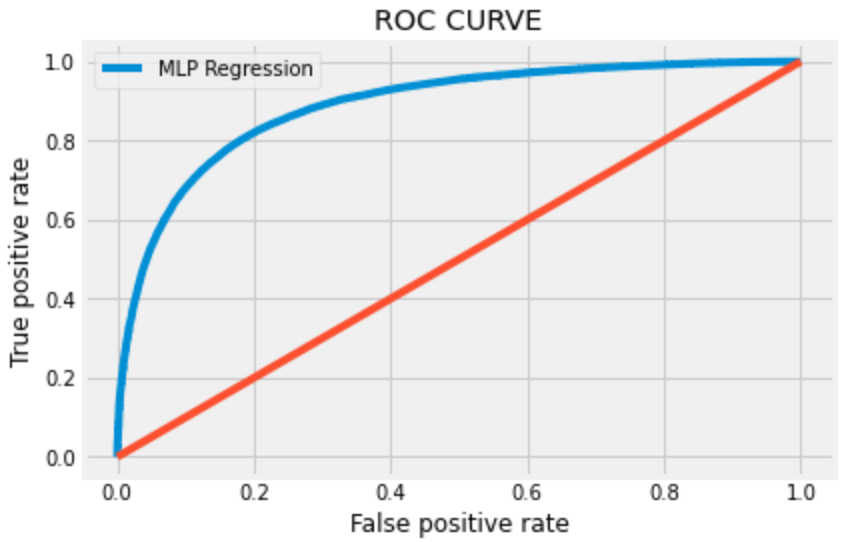


ANN

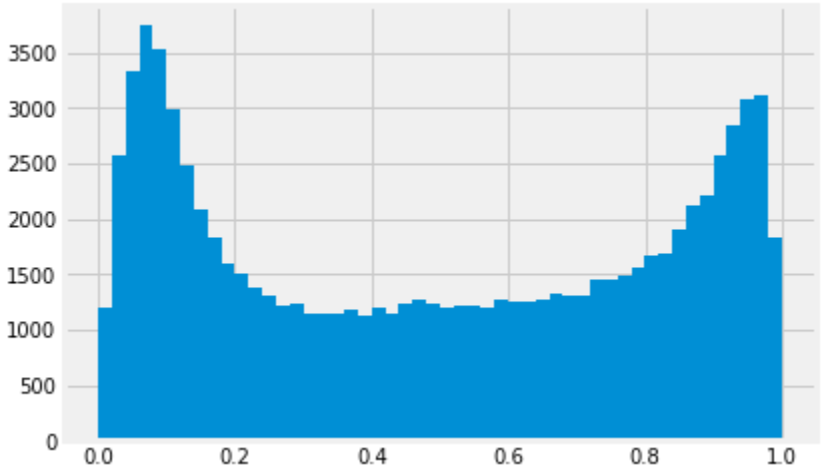
We used the scikit learn's Multilayer Perceptron classifier, which optimises a log-loss function using a stochastic gradient based optimization. We ran the model with an L2 penalty parameter of 0.01 and a maximum iteration limit of 500 for the optimisation function.



By studying the confusion matrix of this model we see that the model performs very well, about as good as the random forest classifier.



Even by studying the ROC curve, we see that the false positivity rate remains low as the true positivity rate rises, which indicates a good predictor. We see the prediction spread of the model, which shows a nice, confident distribution of passes. One can see that the model is very confident in predicting an attempted pass as incomplete and by looking at the confusion matrix, we see that it is largely successful in doing so.



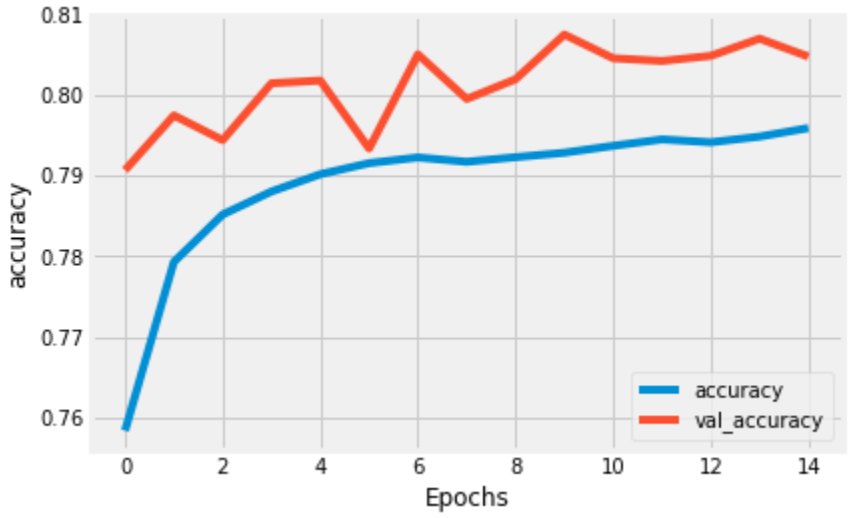
The model performs rather well with an AUC-ROC score of 0.888 and accuracy score of 0.8069.

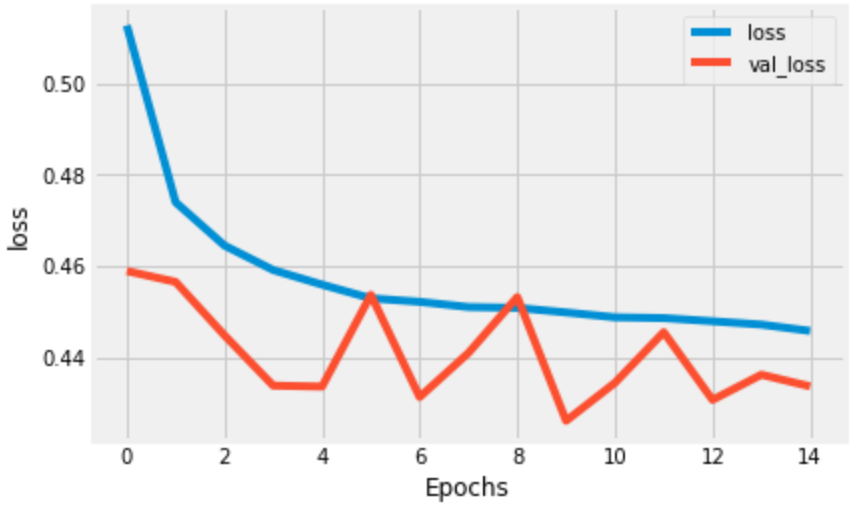
Deep Neural Network

This model was implemented using tensorflow API, with the sequential model being used. It performed at an accuracy score of 0.81 and AUC-ROC score of 0.888, making it the best performing model in this project. The model was trained using over 8000 nodes and 2,730,286 trainable parameters. To avoid overfitting, dropouts and batch normalisation were used. The model is summarised below:

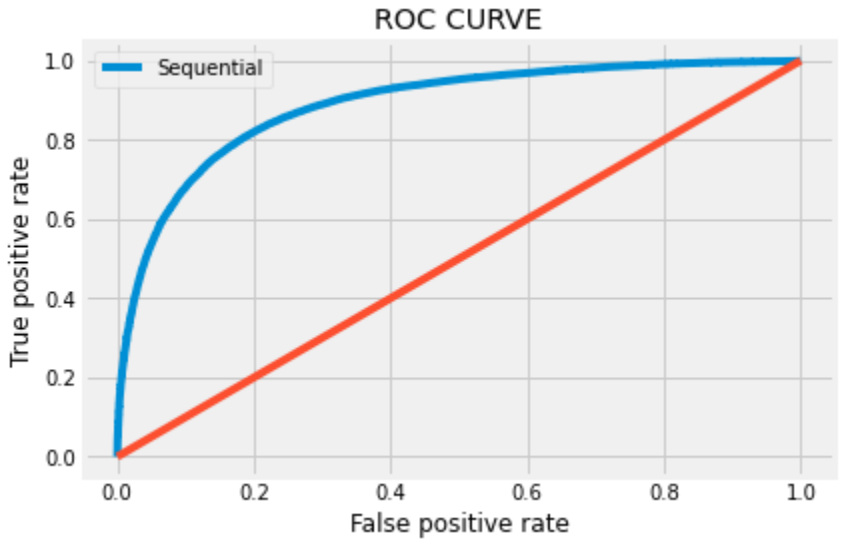
```
Model: "sequential_2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_7 (Dense)              (None, 1028)              8224
_________________________________________________________________
batch_normalization_5 (Batch (None, 1028)              4112
_________________________________________________________________
dense_8 (Dense)              (None, 1208)              1243032
_________________________________________________________________
dropout_3 (Dropout)          (None, 1208)              0
_________________________________________________________________
batch_normalization_6 (Batch (None, 1208)              4832
_________________________________________________________________
dense_9 (Dense)              (None, 1208)              1460472
_________________________________________________________________
dropout_4 (Dropout)          (None, 1208)              0
_________________________________________________________________
batch_normalization_7 (Batch (None, 1208)              4832
_________________________________________________________________
dense_10 (Dense)             (None, 2)                 2418
=================================================================
Total params: 2,727,922
Trainable params: 2,721,034
Non-trainable params: 6,888
_________________________________________________________________
```

The model performance while training is plotted below along with the performance with a validation dataset :

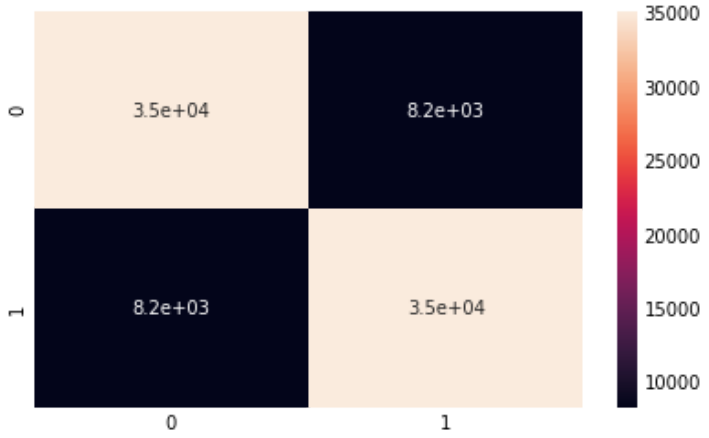




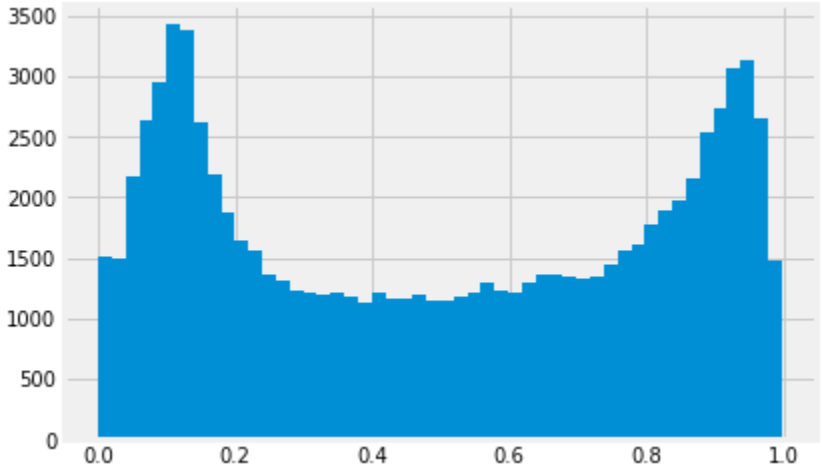
By studying the ROC curve we see that the model maintains a high true positive prediction rate and low false positive prediction rate, which is indicative of a good prediction model.



By looking at the confusion matrix of the model we observed that the model is equally good at classifying attempted passes as complete and incomplete



Following is the prediction spread of the model. It shows that the model is quite confident in predicting outcomes and the accuracy of prediction shows that the model is also trained fairly well.



The combination of good accuracy, AUC-ROC score and symmetry of prediction accuracy for success and failure made the sequential model the overall best model and was used on match event data.

Results and interpretations

The models were all trained over the same dataset and gave varying results in terms of AUC-ROC scores, Accuracy Scores, etc. More simplistic models like linear regression, logistic regression, Decision trees and especially support vector classifiers were less effective and couldn't match up to more tedious algorithms used. MPLC, Deep Learning Models and Random forests were very effective in predicting the outcome of the attempted pass and made intuitively consistent conclusions for the parameter weights (at least wherever observable), which was not the case for the simpler models used.

Below is a tabular comparison of accuracy and AUC-ROC scores

	Model Name	Accuracy Score	ROC AUC Score
0	LinearRegression	0.713210	0.713209
1	LogisticRegression	0.712331	0.712331
2	SVC	0.638803	0.639012
3	RandomForestClassifier	0.818558	0.818556
4	DecisionTreeClassifier	0.747409	0.747411
5	MLPRegressor	0.806980	0.888301
6	Sequential	0.810100	0.888066

Due to the combination of good accuracy, AUC-ROC score and symmetry of prediction accuracy for success and failure made the sequential model the overall best model and was used on match event data.

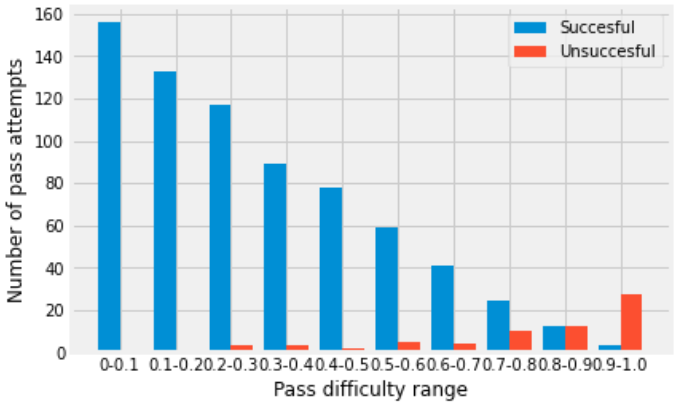
Model Application

The model was applied to a match between Granada and FC Barcelona during the 2011-12 La Liga season. The match ended 0-1 in the favour of FC Barcelona, a match where Barcelona dominated possession but could not convert any shots from open play, the lone goal of the match came from a Xavi Hernandez freekick. Barcelona had 17 attempts at goal, 14 of which were from open play and none of which lead to a goal. We analyse the pass data to see whether the outcome of the open play shots are a result of a poor shooting night or due to a better show by Granada to stop a dominant team.

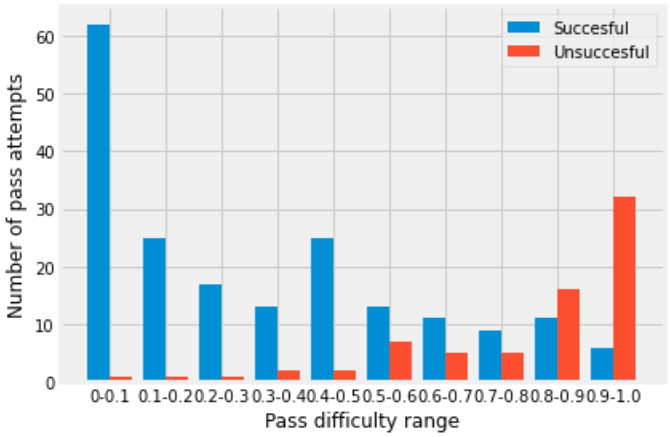
Pass Distribution

First we see the distribution of successful and unsuccessful passes based on pass difficulty by either team

Barcelona:



Granada:



Here we see that Barcelona has a significantly higher volume of passes when compared to that of Granada, which is expected from the most possession dominant teams. Barcelona also show a very structured distribution of pass difficulty, which indicates that the possessions are quite structured, since there are no abrupt increases in difficulty of passes, the pressure on the defence is built and the difficult and decisive passes are made. On the other hand, Granada in possession shows a very high volume of low risk passes and a sudden drop off in volume of passes in higher risk passes. For example having comparable number of passes in the 0.3-0.4 difficulty and 0.8-0.9 difficulty shows a lack of build up and more risky, less reliable style of offence.

Additionally, Granada have several failed passes, even those which are the easiest to execute - which indicates both poor distribution and great pressure on the offence by Barcelona. Barcelona on the other hand do not miss a single pass until the 0.2-0.3 difficulty range despite attempting almost 300 passes of the easier variety when compared to the less than 100 by Granada. Even in the more difficult passes, Barcelona have fewer misses than Granada in the 0.9-1.0 difficulty range, which indicates a more desperate playstyle adopted by Granada.

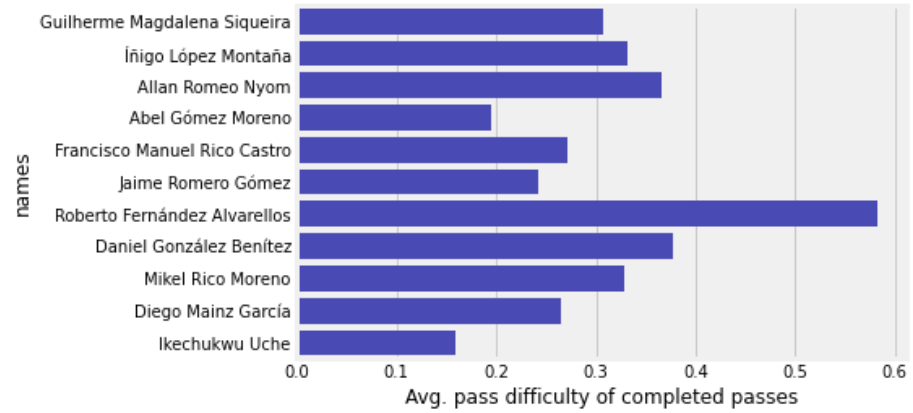
Granada had a pass completion percentage of 72% while Barcelona had a pass completion percentage of 91%.

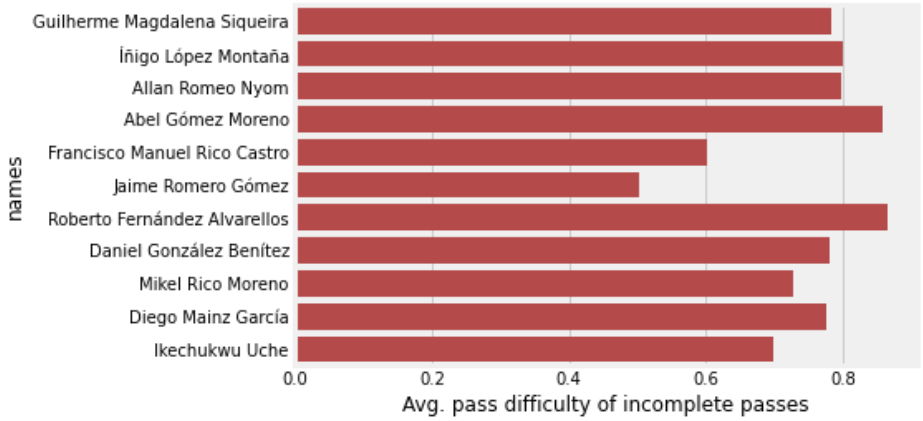
Despite having no goals from open play, Barcelona clearly had a good offensive showing, but lacking the clinical finishing touch. We analysed the performance of all the players in the match to see their contribution of the players towards the control and offensive buildup of the game.

Individual Performances and Roles

We see the average pass difficulty of passes by each player of both successful and unsuccessful passes to determine the role of the player in the game.

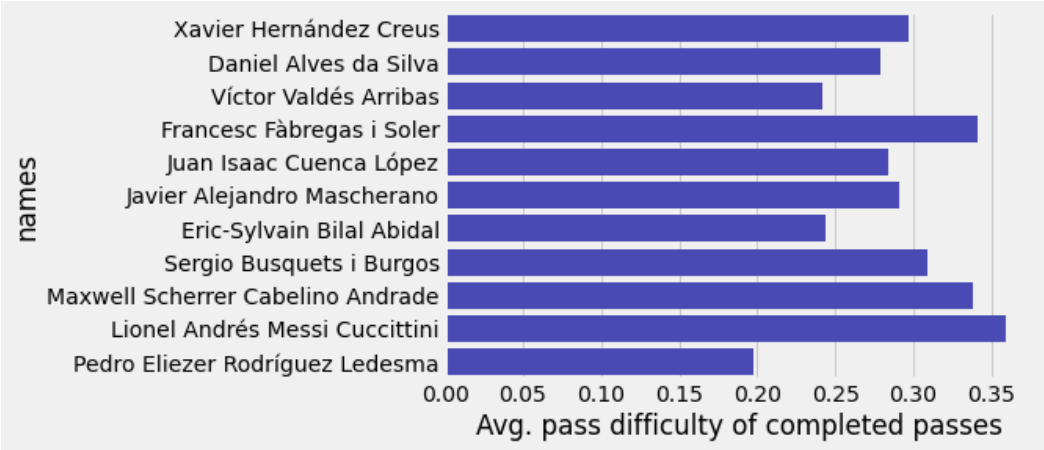
Granada:

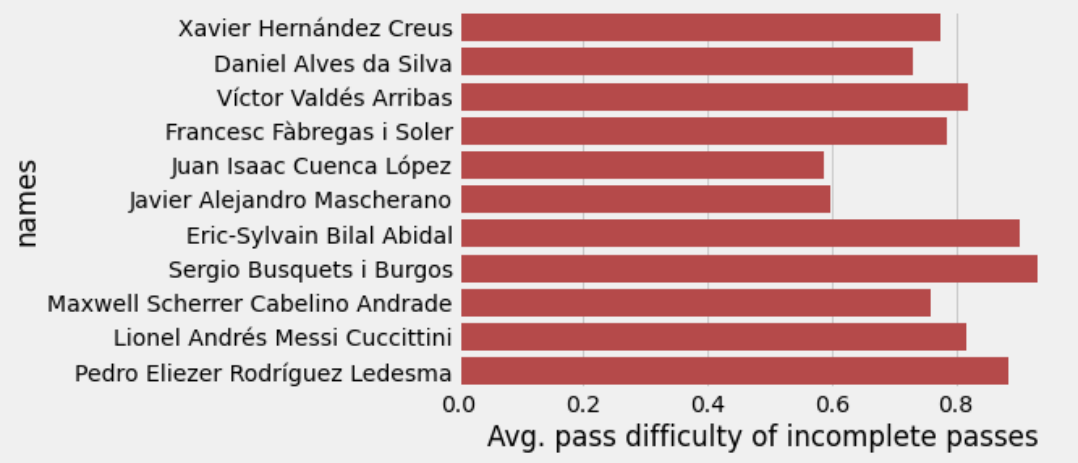




When we analyse the Granada players, we see that players like Allan Romeo Nyom(RB) and Daniel Benitez(LM) have the highest average difficulty of completed passes among the outfield players. This is also reflected in the statsheet as Nyom makes the only key pass for Granada. The high average pass difficulty for Roberto Fernandez, the goalkeeper shows that the team played more lobs from the goalkeeper rather than easier short passes to start from goal kicks. 38/49 passes by Fernandez are long passes of which 17 were successful. A low average difficulty of passes by central midfielders like Ikechukwu Uche, Abel Moreno, Fran Rico and Diego Mainz (all below 0.3) shows a rather passive midfield when compared to the average difficulty of passes by Guilherme Siquera, centre back which is around 0.3. Additionally, the high average failed pass difficulty shows that there weren't even enough attempts to make slightly more progressive plays without resorting to desperation passes. The low average failed pass difficulty for the other centre half, Jaime Romero, shows poor distribution of the ball from the player.

Barcelona:





While analysing the passes for Barcelona, we see that the highest average pass difficulty is done by players like Lionel Messi and Cesc Fabregas, both creative playmakers. Additionally, we see that all the midfielders have an average pass difficulty in the range of 0.3 and upwards, which is great especially considering the volume of passes in the lower difficulty. We see that the fullbacks (Alves and Maxwell) are involved in difficult passes while the wingers (Pedro and Cuenca) attempt rather easy passes. This shows which players are more involved in the buildup. The low pass difficulty of Valdes(GK) shows that Barcelona had more short passes to start goal kicks - 7 long balls attempted out of 25 total - shows that Barcelona like to build their offence from the back and don't rely too much on long balls to teammates.

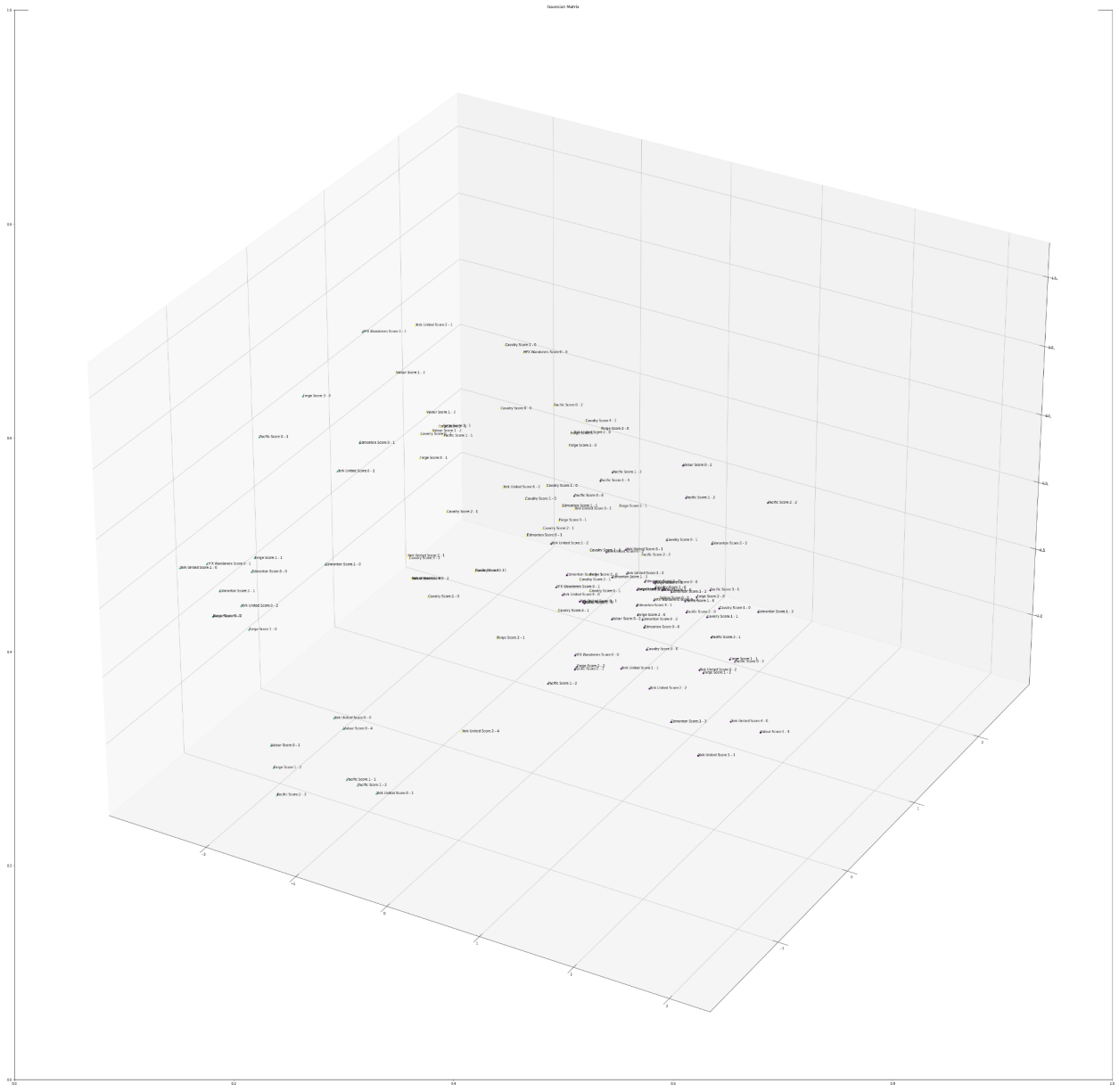
We see that Barcelona clearly dominate the possession aspect of the match, with 15 key passes to Granada's 1, 17 shots to Granadas 2, and the superiority of the possession quality as well, indicated by the analysis above. In the end, poor shooting from Barcelona(7/17 on target, zero converted from open play) and good (22 clearances and 15 interceptions), aggressive (14 fouls and 7 total yellow cards) from Granada prevented Barcelona from scoring any goals from open play and the lone goal from a freekick made the game look much closer on the surface level than it actually was.

Team Performance Clustering using Unsupervised Learning

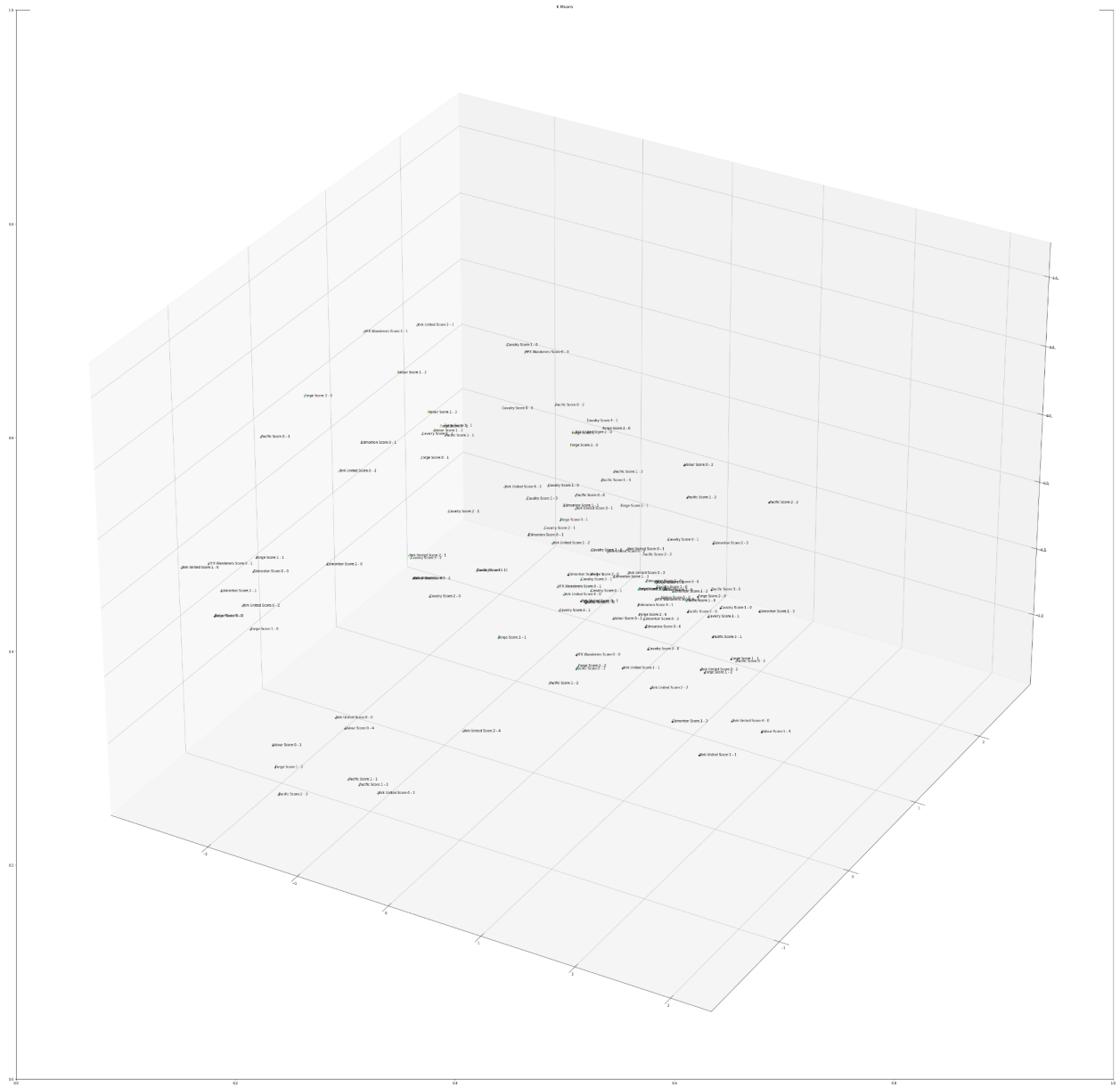
The data is taken from understat , and has detailed match event data from the Canadian Premier League 19/20 season. Methods used were PCA, K-Means Clustering, Spectral Clustering and Gaussian Mixture clustering.

The dataset covered 200 match performances and spanned 129 different features. The features which were in percentage values like Aerial%, Pass%, TcklMade%, etc were stripped of the % sign and string was converted to a float value. Data was scaled by converting each value to the ratio of the value and the maximum value in the column.
Match performance similarity clusters were generated using K-Means Clustering, Spectral Clustering and Gaussian Mixture clustering. These clusters can be used when counter strategizing against teams to study team performances against similar teams, or to classify matches as outliers in terms of results given the general trend of matches with similar performances.

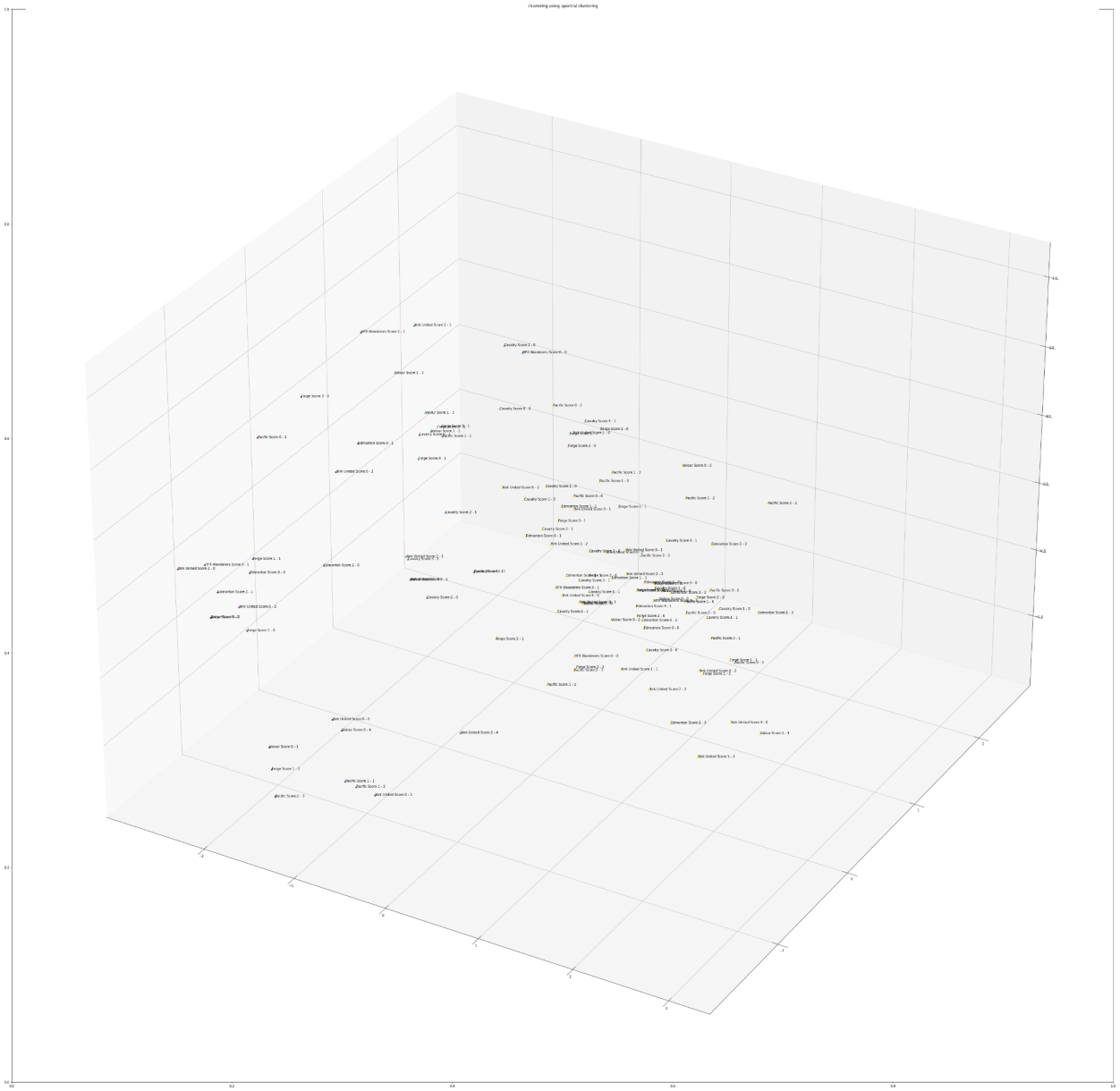
1. Gaussian Mixture



2. K-means clustering



3. Spectral Clustering



Results and interpretations

The K-means clustering model sorted best at 5 clusters, whose results are described below:

	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5
Wins	7	7	17	9	6
Losses	8	12	18	7	2
Draws	11	4	7	5	3
Goals Scored	32	21	50	29	19
Goals Conceded	31	28	48	25	12

As one can see the model clusters some of the matches into winning and losing clusters like #2 and #5 outright while some classifications are more close.

The Gaussian Mixture model sorted best at 3 clusters, whose results are described below:

	Cluster 1	Cluster 2	Cluster 3
Wins	16	6	24
Losses	24	12	11
Draws	16	5	9
Goals Scored	61	20	70
Goals Conceded	71	26	47

This model almost perfectly splits the data into clear winning, losing and close matches. Cluster 3 is a clear winning cluster while cluster 2 is a clear losing cluster. Cluster 1 is not very well classified, the high number of draws compared to wins indicates close games but the 1.5 times

number of losses indicates a losing cluster. The narrow goal difference considering the number of losses is another factor which made us label it as close games.

The Spectral Clustering model sorted best at 4 clusters, whose results are described below:

	Cluster 1	Cluster 2	Cluster 3	Cluster 4
Wins	5	9	6	26
Losses	13	4	3	27
Draws	5	4	4	17
Goals Scored	20	21	19	91
Goals Conceded	30	15	15	84

This model struggles to sort a large chunk of the games but sorts the rest of the games quite well into neat winning and losing clusters, by looking at both results and goal difference. Even by increasing the clusters, the large chunk of data wasn't sorted much better and matches from clusters 1,2,3 were further split into smaller clusters.