xG Model

Initial trial

8 models were compared. The logistic regression came out on top, and due to its inherent explainability it was the preferred method of choice, compared to other neural network-based approaches which fared roughly as well.

Due to the limited sample size neural network-based approaches had limited effectiveness at capturing the higher order behaviour. Angle (size of the goal in the field of vision of the striker) and distance to goal were the two first order factors with the biggest predictor of xG. The second order variables included in the data set were the number of players in the shot line and interference on the shooter, they accounted for about 2-3% increase in the performance of the model. The noise on the data set was about $1/\sqrt{N} \approx 1\%$ so higher order behaviour could not be observed at this current time. Another reason for favouring a logistic regression model. If we had a larger data set, I am sure neural nets would prove more effective.

The models scored about 80% on the ROC curve. (This represents 80% area under the curve)

I picked the two models with the best ROC-AUC score to continue testing and exploring their features.

	Model	Accuracy	Precision	Recall	F1 Score	ROC-AUC	TP	TN	FP	FN
0	Logistic Regression	0.895406	0.641026	0.111111	0.189394	0.794184	25	1807	14	200
1	Random Forest	0.881720	0.433071	0.244444	0.312500	0.723477	55	1749	72	170
2	Gradient Boosting	0.895894	0.607143	0.151111	0.241993	0.787582	34	1799	22	191
3	SVM	0.890029	0.000000	0.000000	0.000000	0.636163	0	1821	0	225
4	KNN	0.879765	0.391753	0.168889	0.236025	0.700724	38	1762	59	187
5	Naive Bayes	0.870479	0.272727	0.106667	0.153355	0.745012	24	1757	64	201
6	MLP	0.893451	0.555556	0.155556	0.243056	0.792180	35	1793	28	190
7	AdaBoost	0.896872	0.645833	0.137778	0.227106	0.790104	31	1804	17	194
8	XGBoost	0.884653	0.430380	0.151111	0.223684	0.753138	34	1776	45	191

Comparison and suspected non-linear behaviours

I subdivided the data set to look at what behaviours we suspect that the models are poorly identifying. The threshold for identifying behaviours that are suspected to be poorly understood by the model is $1/\sqrt{N}$ where N is the number of goals (this is consistent with Poisson models. This is a rough and ready estimate and obviously when you subdivide any data set you will always get divergence from mean behaviour even to one or two sigma if you divide it up enough times. However, this was the method which was chosen given the time constraints and constraints over data volume, it also is intelligible to a wide audience.

Number of intervening opponents = 0

```
In [135]: no_int_opp = X_test[X_test["Number_Intervening_Opponents"]==0]
```

```
In [137]: no_int_opp["goal"].value_counts()
Out[137]: goal
          1    32
          0    14
          Name: count, dtype: int64
```

```
In [138]: no_int_opp["xG_mlp"].sum()
Out[138]: 18.544593739925165
```

```
In [139]: no_int_opp["xG_log_reg"].sum()
Out[139]: 26.278505900415233
```

The threshold for this is $1/\sqrt{32}$ which roughly 5.6 goals. With confidence we can say the MLP classifier has underpredicted the number of goals when there is 0 intervening opponents. This is also true to a lesser extent for the logistic regression. We would need a bigger data set to confirm this anomaly. However, it will be corrected in the modelling stage.

Number of intervening opponents = 1

```
In [147]: one_int_opp = X_test[X_test["Number_Intervening_Opponents"]==1]
```

```
In [152]: one_int_opp["goal"].value_counts()
Out[152]: goal
          0    663
          1    203
          Name: count, dtype: int64
```

```
In [154]: one_int_opp["xG_mlp"].sum()
Out[154]: 205.4368027764844
```

```
In [155]: one_int_opp["xG_log_reg"].sum()
Out[155]: 218.62371187527978
```

The model understands this behaviour well, perhaps because most goals are in this category. The model understands the primacy of the goalkeeper, but struggles to linearly abstract that for more defenders.

Number of intervening opponents = 2

```
In [141]: two_int_opp= X_test[X_test["Number_Intervening_Opponents"]==2]
```

```
In [143]: two_int_opp["goal"].value_counts()
Out[143]: goal
          0    1409
          1     133
          Name: count, dtype: int64
```

```
In [145]: two_int_opp["xG_mlp"].sum()
Out[145]: 193.6277196846336
```

```
In [146]: two_int_opp["xG_log_reg"].sum()
Out[146]: 169.66660865013216
```

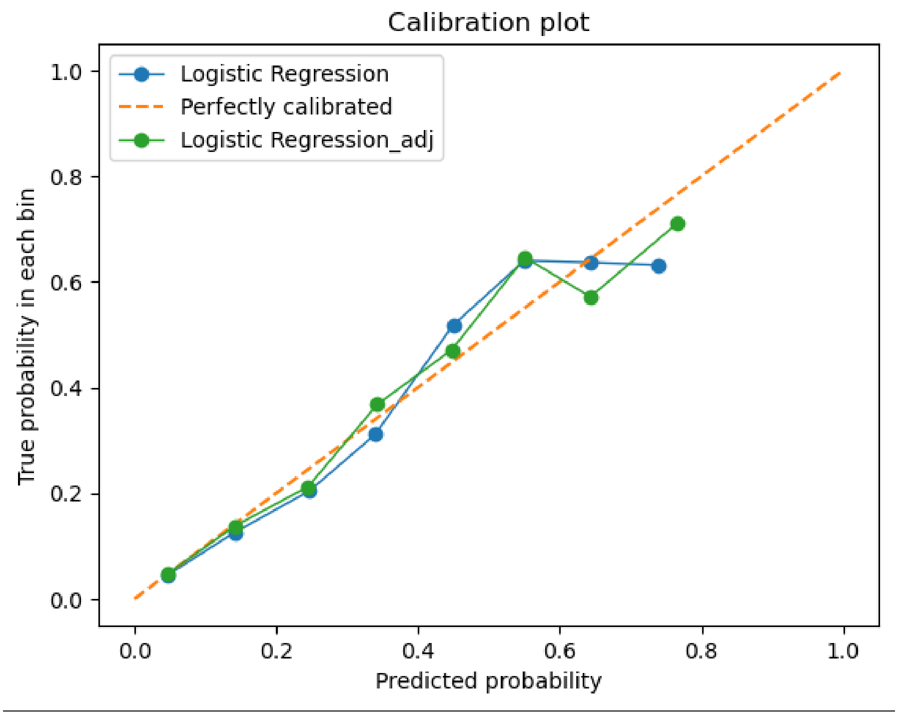
The threshold for this comparison is $1/\sqrt{133}$ which is roughly 11.7 goals both models clearly predict way too many goals. They do not understand the primacy of one blocking defender in stopping a goal. The data quality is such that we do not know if there is one goalkeeper and a defender or two defenders in between the goal and the attacking player.

<u>Number of opponents = 3 and 4</u>

```
In [157]: three_int_opp = X_test[X_test["Number_Intervening_Opponents"]==3]

In [158]: three_int_opp["goal"].value_counts()
Out[158]: goal
          0    955
          1     70
          Name: count, dtype: int64

In [160]: three_int_opp["xG_log_reg"].sum()
Out[160]: 56.79853170494015

In [161]: three_int_opp["xG_mlp"].sum()
Out[161]: 60.699558640762476

In [247]: four_int_opp = X_test[X_test["Number_Intervening_Opponents"]==4]

In [248]: four_int_opp["goal"].value_counts()
Out[248]: goal
          0    402
          1     15
          Name: count, dtype: int64

In [249]: four_int_opp["xG_mlp"].sum()
Out[249]: 8.59816017514208

In [250]: four_int_opp["xG_log_reg"].sum()
Out[250]: 14.462996197251954
```

The logistic regression starts to predict this behaviour well again, the MLP is still struggling with 4 defenders.

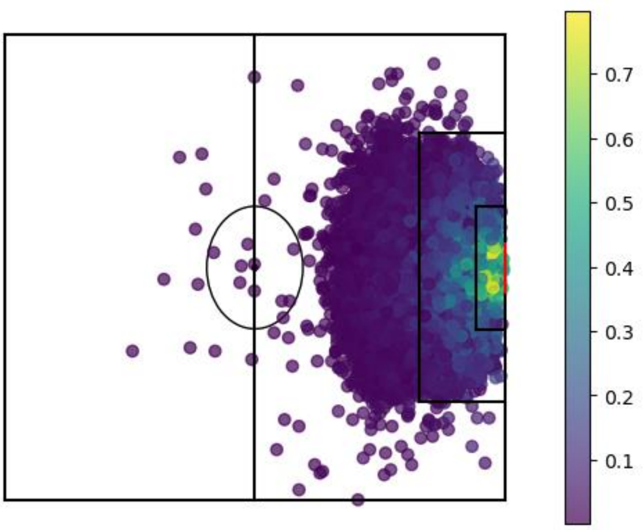
Fine tuning and calibration



Given the logistic regression showed the most accurate behaviour its weights were adjusted for the number of intervening opponents and a better calibration was observed. An open goal was given more weighting and having presumably the goalkeeper in goal with an extra defender was given less weighting.

Plotting

The open play plot is below.

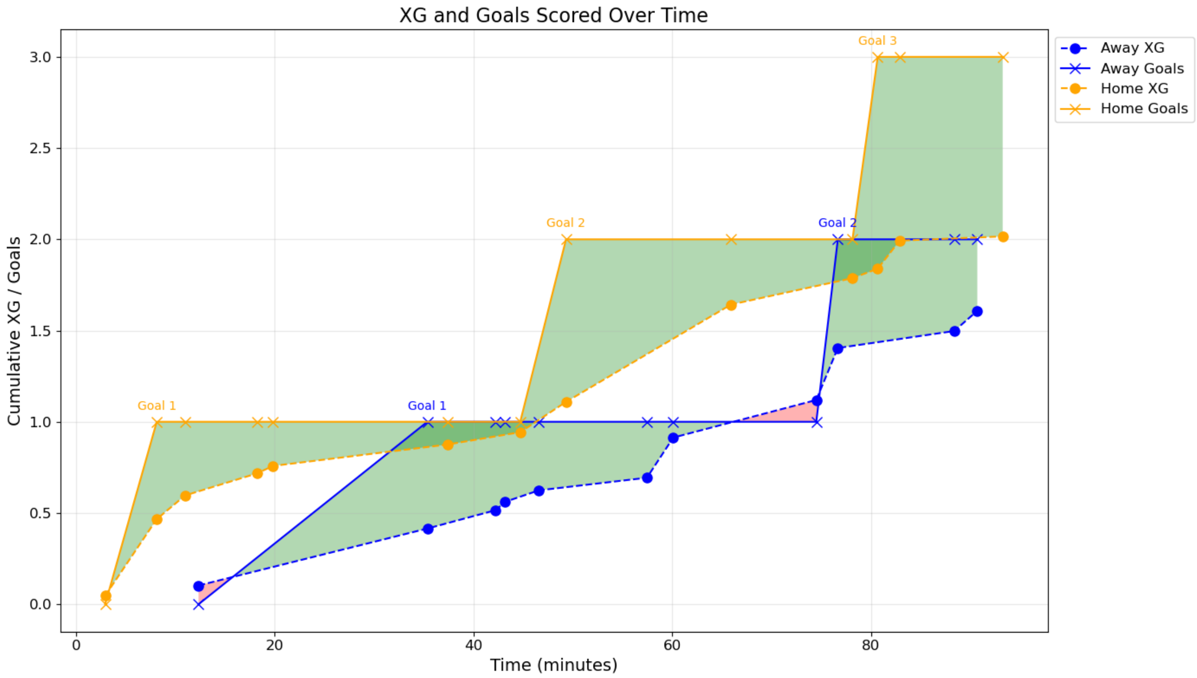


Match Report

xG match analysis

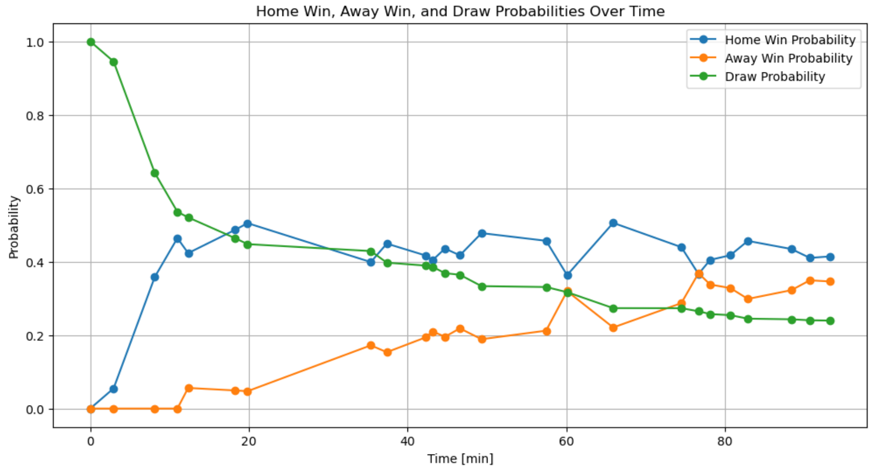
The home team led by 0.5xG from roughly the 10th minute of the game, they quickly replied both in terms of shots, xG and actual goals scored whenever they were in a drawing game state to quickly recontrol the game.

I have conducted an analysis below which seeks to explore the game more thoroughly by creating a binomial model to show the likelihood of winning game state given the shots. Then I have created a strategy function which seeks to determine what is the best strategy for each team at each point in the game.



Binomial model to predict win probability

I created a binomial model with each individual shot treated as a statistically independent event where the game state was ignored. The model showed the probability of a winning scoreline at each point in the game given the quality of each teams shots up and until that point. For example if a team had two shots, one with a xG of 0.2 and one with an xG of 0.3. the chance of scoring 0 goals is $(1 - 0.2) * (1 - 0.3) = 56\%$. The chance of scoring one is $(0.2 * 1 - 0.3) + (0.3 * 1 - 0.2) = 38\%$ and the chance of scoring two is $(0.2 * 0.3) = 6\%$. This was used to see the probability of game states and thus the probability of a team being in the lead or drawing.



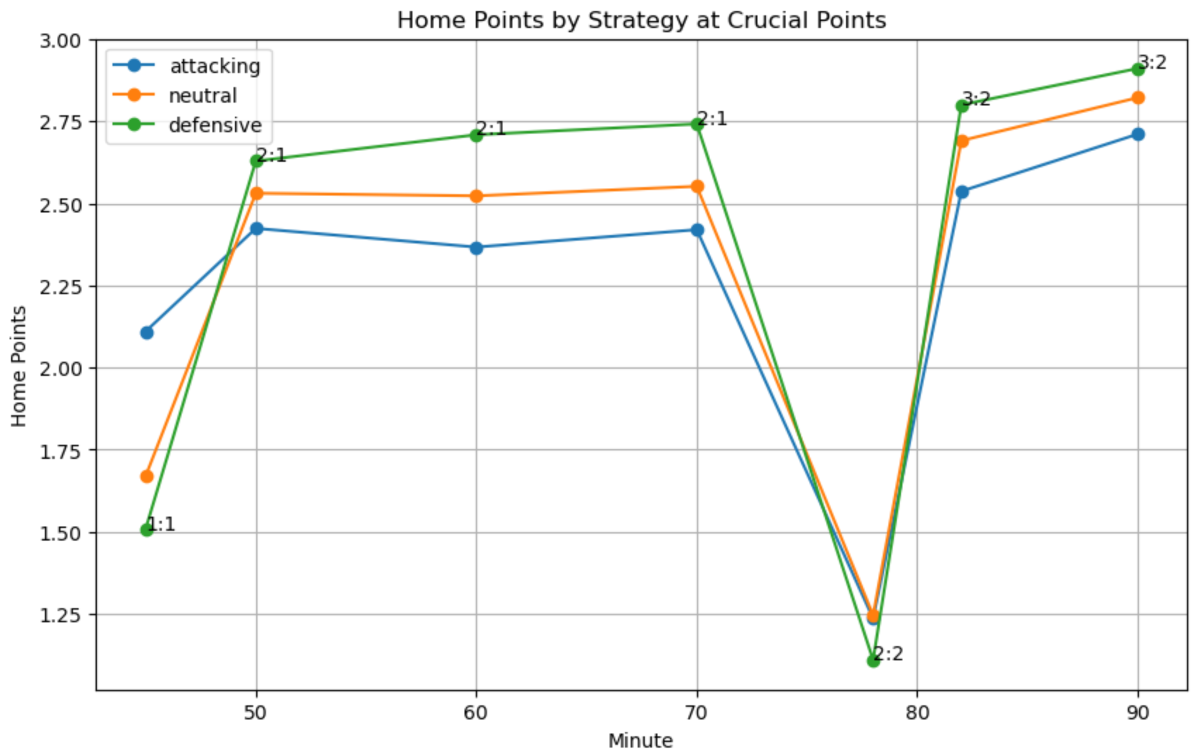
For most of the game the home team winning was the most probable outcome, having roughly a 45% outcome of winning for most of the game. The away team gradually grew into the game and created roughly equal chances to the home team by the end of the game.

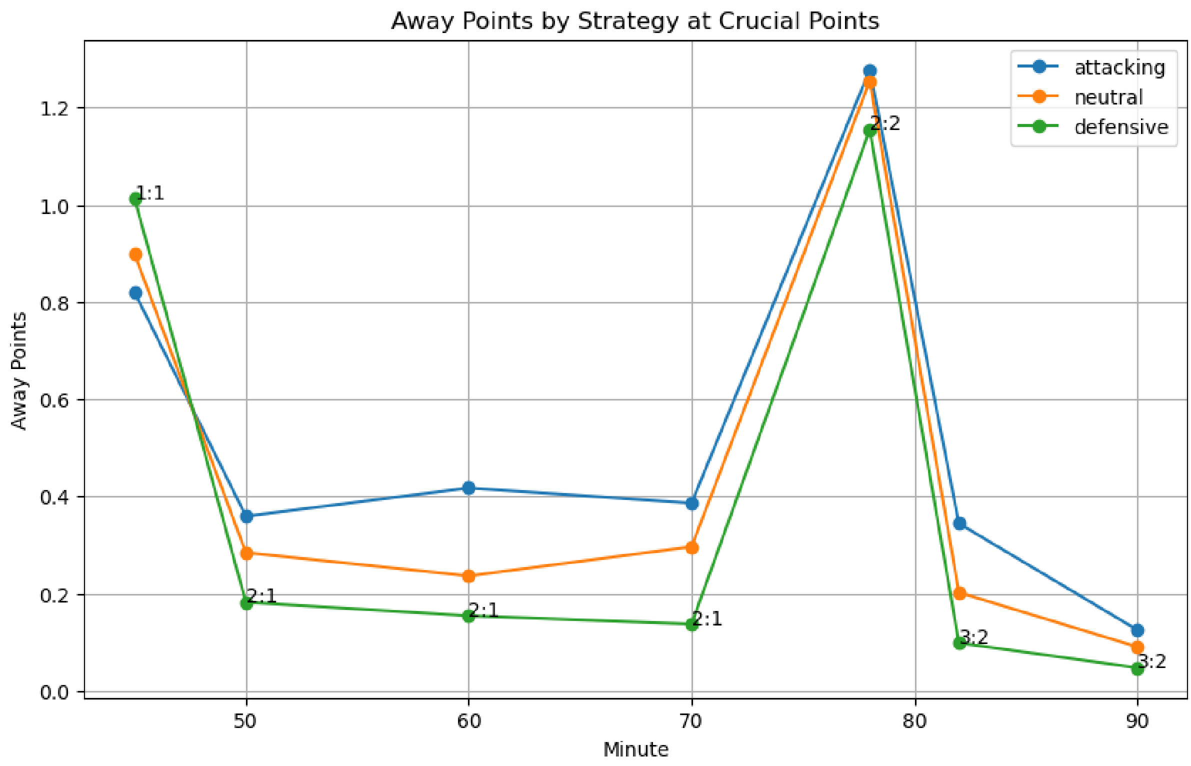
To say one team deserved to win the game is difficult, the xG of both teams at the end is similar. The home team always did what they needed to immediately reply when conceding a goal and managed the game state the best regardless of whether the away team eventually caught up with them in terms of xG.

If you are winning and go to a defensive strategy, you may give up more shots, but the chance of the game state changing is smaller, even though the other team may in that case have more xG than you. Does that still mean you deserve to win?

<u>Strategy</u>

Thinking about strategy I made this model which shows the strategy each team should take depending on which part of the game they were in.





The model took in the xG data up an until that point in the game as a marker of the strength of each team (obviously has its drawbacks). Each strategy could either double its xG output and double its input by going attacking, or half its xG output and input by going defensive.

In every point in the second half it was judged that it was best for the away team to attack to maximise points. The strategy for the home team varied. It was usually better to defend while it was in front.

Verdict

I would conclude that the home team deserved to win, given that they controlled the game state and always replied swiftly to a drawing game state with a high number of quality shots. Even if the xG of teams ended up roughly similar. Both teams overperformed their xG. Given purely the quality of shots you would expect the home team to win roughly 41% of the games, the away team 37% and a draw around 23%. However, this is slightly misleading as it does not take into account the way teams actively manage a game. So again I would reaffirm that the home team deserved to win.