From 90 minutes to Infinity: The cloaked Mathematical Vastness of a Football Match

Riley Marchant

April 11, 2026

1 Introduction

The beautiful game- I am sure you have heard of it- football. The game that transcends billions of fans into euphoria or despair each year; the game that unites us, one and all in times of victory and fracture alike. Yet as mathematicians, we are conditioned to ask a different question: how can we look at this through a mathematical lens?

This is precisely why a what seems like 'silly' or 'simplistic' game of football conceals a multitude of hidden structures lying beneath the roars of the crowd. From the moment the ball (which it-self covers a plethora of mathematical wonders) is struck- the power P=0.5mv/t, the angle of contact θ, the height of the trajectory h, the subtle influence of air resistance R, every variable playing into a family tree of mathematics, each branch entwining into a Universal language through which we so dearly attempt to decode the world.

However, the Mathematics of football extends beyond projectile motion or the physics behind a curling free kick. Beneath the seemingly finite 90 minutes regular fans see, what I see is something much more profound; a realm of infinite possibilities. From the infinitely many combinations the 22 players can be positioned on a seemingly finite 105x68m pitch, to the myriad subtle variations in a pass that could change a game by 2 or 3 goals, is what truly excites me.

2 Expected Goals (xG)

2.1 What is xG?

xG, If you dont stay up to date with football you probably dont see those two letters hanging around with eachother very often. However, if you do, im sure you understand the chaos it brings to our footballing fantasy. xG, rather known as 'expected goals' is a very divisive statistic within football. Some football maniacs (myself) love it; however it seems many players arent so fond of it, with Aston Villa superstar, Morgan Rogers, saying 'its a whole load of nonsense' and Roy keane the infamous Manchester legend adding that 'Goals will always be the superior stat'. But is it actually useful and what really is it?

Expected goals is a mathematical model used to estimate the probability that a shot will lead to a goal. Each shot is assigned a value between 0 and 1, where 0 represents a shot that is impossible to score from, and 1 represents an absolute banker, a sitter, you'd be ridiculed if you missed it. The model takes into account several factors such as the distance from goal, the angle of the shot, or even which part of the body is used to strike the ball.

2.2 The math behind xG

The question is: how can we actually generate this goal-predicting model using mathematics? Is there a way to transform qualitative peices of information about a shot, into a number between 0 and 1 that represents the probability of scoring? If you've come across 'logistic regression' before, you probably saw this coming. If not here's a quick introduction to what it actually is: Logistic regression is a ML method used to predict the probability of an outcome- here, whether or not a shot results in a goal. However, there is lots of maths involved!

By feeding the model lots of data, it learns and can predict how different factors- like distance- influence the likelihood of scoring.

In order to output a specific numerical value $0 < z < 1$ given a plethora of different variables we use the Sigmoid Function.

The Sigmoid Function: $\sigma(z) = \frac{1}{1+e^{-z}}$

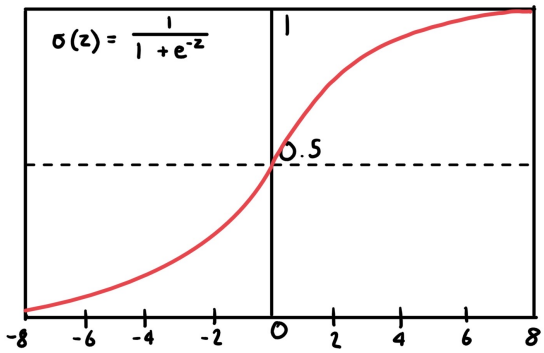

Figure 1: Sigmoid Function

As you can see via the diagram above, the Sigmoid Function has a range of $0 < \sigma(x) < 1$ Hence this function is perfect for modelling goal probability, as it maps any real-valued input onto a bounded interval. The Sigmoid Function enables us to input a value z, $z \in \mathbb{R}$ that is sculpted from an array of different variables to output a value $0 < P(goal) < 1$ that in our model represents the likelihood of scoring a goal. You may also notice a dashed line at 0.5. This is what's called a decision boundary, and any value above or below it will be categorized as goal or no goal $\sigma(z) > 0.5$, P(Goal)=1. $\sigma(z) < 0.5$, P(Goal)= 0

This is where the math starts! In order to calculate a value for z we need to first create a weighted linear combination of variables.

This is what our combination looks like:

$$Z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4 + \beta_5 x_5$$

$\beta_1, \ldots, \beta_5$ = weights for each factor

$x_1, \ldots, x_5$ = shot variables

As $z \to \infty$, P(goal)$\to$ 1. Hence when all x_n variables are at their optimum, the probability of a goal will =1

As $z \to -\infty$, P(goal)$\to$ 0. Hence when all $x_n$variables are at values that make it hardest to score from, the probability of scoring a goal is P(goal)=0.

However, once we pass a certain value for z, P(goal) doesn't change very much:

$$Z = 5 \Rightarrow P \approx 0.993$$
$$Z = 10 \Rightarrow P \approx 0.99995$$
$$Z = 20 \Rightarrow P \approx 0.999999998$$

Therefore, we can say that any value that satisfies $z > 5$ corresponds to shots with probabilities very close to 1. Similarily, for $z < -5$ shots have probabilities very close to 0.

What makes the model being weighted so important? Well put yourself in the shoes of a footballer for a second, you are gliding along the pitch as if you were dancing among a stage, the star player and you received the ball in a great position. What do you think is more important to whether or not you score, the shot type you use? Or how many defenders there are between you and the goal? If you said the amount of defenders then you would be absolutely correct, hence we would need the number of opposition defenders to stand out more than the shot type when calculating z.

Now to define our variables. For this model I am using 5 variables as adding any more would start to make it increasingly complicated.

x_1 = distance to goal line (m)
x_2 = angle between ball and goal posts (θ)
x_3 = number of defenders between you and goal posts (n)
x_4 = shot type (t)
x_5 = ball speed before shot (ms^-1)

In order to calculate the weight corresponding to each variable we will need a testing data set. However, this data is extremely rare to find on the internet, as big tech companies like to keep it to themselves for analysis. Therefore, with the aid of AI software I was able to generate an ultra-realistic simulated shot data set designed to reflect realistic scenarios.

Shot No.	x_1: Distance	x_2: Angle	x_3: Defenders	x_4: Shot Type	x_5: Ball Speed	xG
1	2.5	65	0	1	2.0	0.92
2	11.0	32	1	2	6.5	0.18
3	18.5	18	2	3	9.0	0.06
4	7.2	50	1	2	4.0	0.42
5	25.0	10	3	4	11.5	0.02
6	5.0	55	0	1	3.0	0.75
7	14.0	28	2	3	7.5	0.10
8	3.8	60	1	2	2.5	0.65
9	30.0	8	4	5	12.0	0.01
10	9.5	40	1	2	5.5	0.30

Table 1: Simulated shot data with variables x_1 - x_5.

You may have noticed that in the shot type column, we have values 1-5 instead of qualitative information; this is because we can assign numerical values to different shot types based on how positively or negatively they affect P(goal).

1 = Dominant sidefoot
2 = Dominant laces
3 = Weak foot (laces or side foot)
4 = Header
5 = Any other body part

Now that we have visualized our data set, we need to scale it so that each variable has the same range. This is because at the moment our data have all sorts of different ranges and units between variables. Without scaling, if we were to code a python optimization problem for this data, larger values such as angle would dominate due to its numerical magnitude, influencing the model disproportionately. This also helps to keep the value of z within a reasonable range (mostly between -5 and 5) preventing the logistic function from saturating at certain values.

Scaling: I have used a range of $0 < scaled_x < 10$

For each variable, I scaled using the minimum and maximum realistic value for each x_n, e.g for distance $x = 0m$ would give a scaled value of 10 as we are looking to make the optimum value for each variable = to a scaled value of 10.

x_n Variable	Min / Max for shot
x_1: Distance to goal line	0 – 88 m
x_2: Angle between ball and goalposts	0 – 180°
x_3: Number of defenders between ball and goal	0 – 11
x_4: Shot type	1 – 5
x_5: Ball speed of approach	0 – 25 m/s

Figure 2: Minimum/Maximum realistic values for x_n

Scaling Formula:

$$x_{\text{scaled}} = \left(\frac{x_{\max} - x_{\text{value}}}{x_{\max} - x_{\min}} \right) \times 10$$

(For variables in which larger numerical values weaken the probability of scoring i.e a larger distance we must also add a 1-... to the formula)

Now that we can scale all of our data points to have a value $0 < x_n < 10$ where 10 = optimum, A logistic regression model can be implemented in Python to estimate the coefficients, using all of the scaled input variables. The model can be trained on the dataset to minimise prediction error and produce weights (β_n) and expected goal (xG) values for each shot. Python outcome:

Shot No.	x_1: Distance	x_2: Angle	x_3: Defenders	x_4: Shot Type	x_5: Ball Speed	xG
1	9.72	3.61	10	10	9.2	0.92
2	8.75	1.78	9	7.5	7.4	0.18
3	7.90	1.0	8	5	6.4	0.06
4	9.18	2.78	9	7.5	8.4	0.42
5	7.16	0.56	7	2.5	5.4	0.02
6	9.43	3.06	10	10	8.8	0.75
7	8.41	1.56	8	5	7.0	0.10
8	9.57	3.33	9	7.5	9.0	0.65
9	6.59	0.44	6	0	5.2	0.01
10	8.92	2.22	9	7.5	7.8	0.30

Table 2: Scaled shot dataset

```
Intercept (β0): -16.3802009139562
Coefficients (β1–β5):
x1_distance: 0.3778
x2_angle: 0.7045
x3_defenders: 0.1851
x4_shot_type: 0.4623
x5_ball_speed: 0.7499
```

Figure 3: Python calculated β_n values

2.3 Finalised Xg Model

$$z = -16.3802 + 0.3778x_1 + 0.7045x_2 + 0.1851x_3 + 0.4623x_4 + 0.7499x_5$$

We can interpret β_0 = -16.38 to give a probability of 0.000008% of a shot being scored when all variables are at their 'hardest to score' value.

$$\text{As } \sigma(z) = \frac{1}{1 + e^{-(-16.38)}} = \frac{1}{1 + e^{16.38}} \times 100 = 0.000008\%$$

Alternatively when every x_n value is at its optimum e.g 0m from the goalline or 0 defenders in the way of the goal, the probability of scoring is at 99.98%

$$z = -16.3802 + 10(0.3778 + 0.7045 + 0.1851 + 0.4623 + 0.7499) = 8.4158$$

$$\text{As } \sigma(z) = \frac{1}{1 + e^{-(8.4158)}} = \frac{1}{1 + e^{-8.4158}} \times 100 = 99.98\%$$

Evaluating weights: As a larger value for β_n indicates a greater importance to P(goal), we can say from our data that the most important factor to scoring is the ball speed/angle and the least most important is the number of defenders in front of goal, this is most likely to be due to the fact that beyond some value for n, the extra difference each defender makes to your shot is minimal. (And the fact this isn't real shot data that we used).

While the model assigns precise probabilities to each shot, football itself remains inherently unpredictable. A value of 0.2 Xg may suggest low probability, yet in reality, a single unpredictable factor can completely alter the outcome. Just as an example, when up against Bayer Leverkusen in the Champions League R016 Arsenal star Eze recieved the ball on the half turn outside the box, he glanced at the opportunity and struck the ball straight into the top left corner, sending the Arsenal fans into pure ecstasy. That shot was predicted to have an XG of 0.02 illustrating perfectly how a moment of magic can shatter an entire mathematical framework. So despite the structure imposed by the model, trying to fully encapsulate football with a mathematical framework is limited. We can clearly see, while Xg models provide us with valuable insight, they will never be able to capture the boundless nature of the game it-self. That is why we love it.

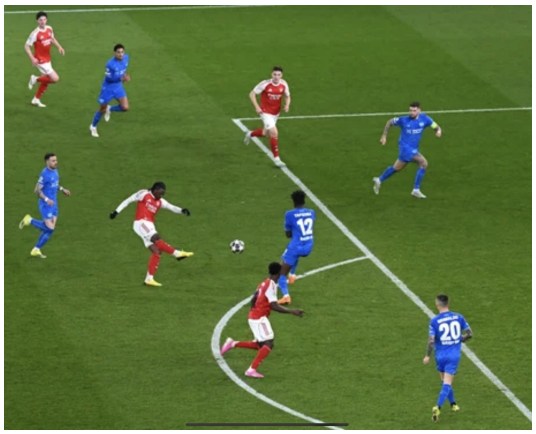

Figure 4: Eze's Beauty

3 Scoreline/Result Predictors

3.1 Intro

Shooting past Xg, I have this question for you: Have you ever clicked onto your favorite Sports app; Sky sports score for example; to check the % of a Win, Loss or Draw for your favorite team who will be playing in the evening? Or even slipped on a cheeky bet for predicting the correct scoreline at Full Time? Well if you said yes you are in for treat because i will be showing how we can create a 'most probable' scoreline predictor using Mathematics. And well, if you said no, dont be disheartened as there will be plenty of maths involved!

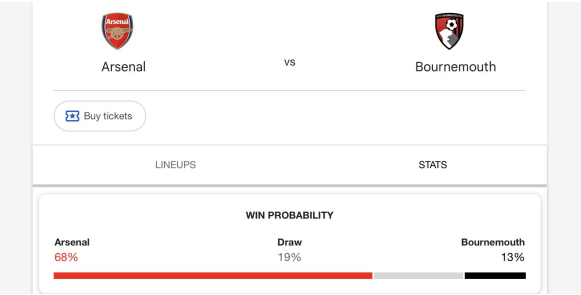

Figure 5: Probability of W,L,D for ARS vs BOU (BBC SPORT)

3.2 Poisson Distribution

What is Poisson Distribution? And how does it help us to predict the probability of Football scorelines?

Poisson Distribution predicts the amount of occurances of an event (goal being scored) within a fixed interval (90 minutes). Once we have the probability of a certain number of goals scored by each team we can combine the values to produce a 'highest probability' scoreline between two teams.

Poisson Distribution Formula:

$$P(X = k) = \frac{e^{-\lambda}\lambda^k}{k!}$$

$$\begin{cases} P(X = k) & \text{Probability of exactly } k \text{ events occurring i.e exactly 2 goals} \\ k & \text{Number of occurrences (e.g. goals scored)} \\ \lambda & \text{Expected number of goals scored in the match based on Seasonal data} \\ e & \text{Euler's number (approximately 2.718)} \\ k! & \text{Factorial of } k \end{cases}$$

In order to achieve a probability for each indivual amount of goals scored within a match (0-6; anything above 6 is unrealistic) I will first calculate the number of expected goals scored (λ) using this formula:

$\lambda_A = (\text{Home Attack Strength}_A) \times (\text{Away Defense Strength}_B) \times (\text{League Average Home Goals})$

$\lambda_B = (\text{Away Attack Strength}_B) \times (\text{Home Defense Strength}_A) \times (\text{League Average Away Goals})$

$$\begin{cases} \text{Home Attack Strength}_A = \frac{\text{Team A home goals per game}}{\text{League average home goals per team per game}} \\ \text{Away Attack Strength}_B = \frac{\text{Team B away goals per game}}{\text{League average away goals per team per game}} \\ \text{Home Defense Strength}_A = \frac{\text{Team A home goals conceded per game}}{\text{League average away goals per team per game}} \\ \text{Away Defense Strength}_B = \frac{\text{Team B away goals conceded per game}}{\text{League average home goals per team per game}} \end{cases}$$

For this demonstration I will be predicting the scoreline for the Arsenal Vs Bournemouth match on Saturday:

Types	**League**	**Arsenal**	**Bournemouth**
Average home goals scored per game	1.498	2.3	1.4375
Average home goals conceded per game	1.401	0.6	1.0625
Average away goals scored per game	1.374	1.625	1.5333
Average away goals conceded per game	1.665	0.8125	2.0667

Table 3: League, Arsenal, and Bournemouth scoring parameters

Index	**Arsenal**	**Bournemouth**
Home attacking strength	2.3/1.498 = 1.54	1.4375/1.498 = 0.96
Home defensive strength	0.6/1.401 = 0.43	1.0625/1.401 = 0.76
Away attacking strength	1.625/1.374 = 1.18	1.5333/1.374 = 1.12
Away defensive strength	0.8125/1.665 = 0.49	2.0667/1.665 = 1.24

Table 4: Arsenal and Bournemouth strength indices

Now that I have calculated each strength for both Home/Away side, we can find λ.

λ_A = 1.54 x 1.24 x 1.498 = 2.86.

λ_B = 1.12 x 0.43 x 1.374 = 0.66.

You may think that we are finished as we have the expected amount of goals scored by both teams, however our λ values are just constants in helping us to find the probabilities of 0,1,2... goals scored.

Now I will run each value (0-6) through our poisson distribution equation and record each probability:

k (goals)	**Arsenal** ($\lambda = 2.86$)	**Bournemouth** ($\lambda = 0.66$)
0	0.057	0.515
1	0.163	0.340
2	0.233	0.112
3	0.222	0.025
4	0.159	0.004
5	0.091	0.001
6	0.043	0.000

Table 5: Poisson probabilities for Arsenal and Bournemouth goals (0–6)

In order to interpret these probabilities and combine them to predict the probability of each scoreline I have inputted them into Excel and colored each result dependant of the score.

Number of goals	Bournemouth	0	1	2	3	4	5
Arsenal	Probability	0.515	0.34	0.112	0.025	0.004	0.001
0	0.057	0.029355	0.01938	0.006384	0.001425	0.000228	0.000057
1	0.163	0.083945	0.05542	0.018256	0.004075	0.000652	0.000163
2	0.233	0.119995	0.07922	0.026096	0.005825	0.000932	0.000233
3	0.222	0.11433	0.07548	0.024864	0.00555	0.000888	0.000222
4	0.159	0.081885	0.05406	0.017808	0.003975	0.000636	0.000159
5	0.091	0.046865	0.03094	0.010192	0.002275	0.000364	0.000091
6	0.043	0.022145	0.01462	0.004816	0.001075	0.000172	0.000043

Figure 6: Excel Datasheet

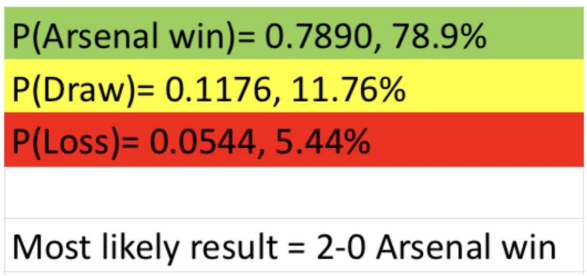

Figure 7: Probabilities for results

3.3 The day of the game

As Saturday unfolded, I sat among the stands as our mighty Arsenal were beaten 2-1 at home to Bournemouth, possibly capitulating Arsenal's Premier League title charge. An unfortunate result, to say the least, but it did prove one thing: football is and will always remain unpredictable. No matter how rigorously we seek to find a way to truly capture the game with statistics and analysis, the infinite possibilities of a single match will always prevail.

Figure 8: My view as Arsenal lose 2:1

4 A little conclusion to a mathematical and footballing fantasy.

Through uncovering a world of complex problems and models within the beautiful game, I have not only enjoyed combining two of my endless passions, but I have also expanded my knowledge of statistical systems themselves. I have learned quite valuably that when we introduce mathematics into things that we love, we are not just attempting to provide structure and order to them but instead to prove the wonderful realm of possibility. And that is exactly why i wrote this piece; The cloaked Mathematical Vastness of a Football Match.

5 Referances

Futmob In-App Statistics

FootballMatic Parallels- Application of math in Football YT vid

hudl.com/blog/expected-goals-xg-explained

Player Quotes- Sky Sports Interviews