



Football Analytics

making inferences from football data with python



- Apply linear regression to football data to make valuable inferences from it ; analyzing the linear relationship between goal difference and points.
- Calculate Expected Threat (xT); recently introduced metric that gives value to a player.



What is Linear Regression?

- Comparing the relationship between two variables

We have independent and dependent variables

- Independent

The variable we use to predict the dependent variable - (x)

- Dependent

The variable being predicted - (y)




- How a quantity in an earlier period can predict the same quantity during a latter period ?



- Goal Difference = No of goals scored - No of goals conceded
- A win results to 3 points
- A draw results to 1 point
- A loss results to 0 point
- 38 games are played by a team in a season ;the team with the highest number of points at the end of the season wins the title.



- In general, the team with a larger goal difference in the table will often win more games and finish higher in the table.
- A linear relationship exists between the goal difference and the points earned by a team at the end of the season.

- 
- In order to explore the relationship between goal difference and the points scored, we applied linear regression to the final standings of the premier league from 2010/11 season upto 2020/21 season.
 - A table was created to record the final ranks, the points, goals scored and the goals conceded. Goal difference was later calculated by subtracting the goals allowed from the goals scored.
 - Using various mathematical and machine learning libraries available in python, we were able to analyse the impact of the goal difference on the final point standings.

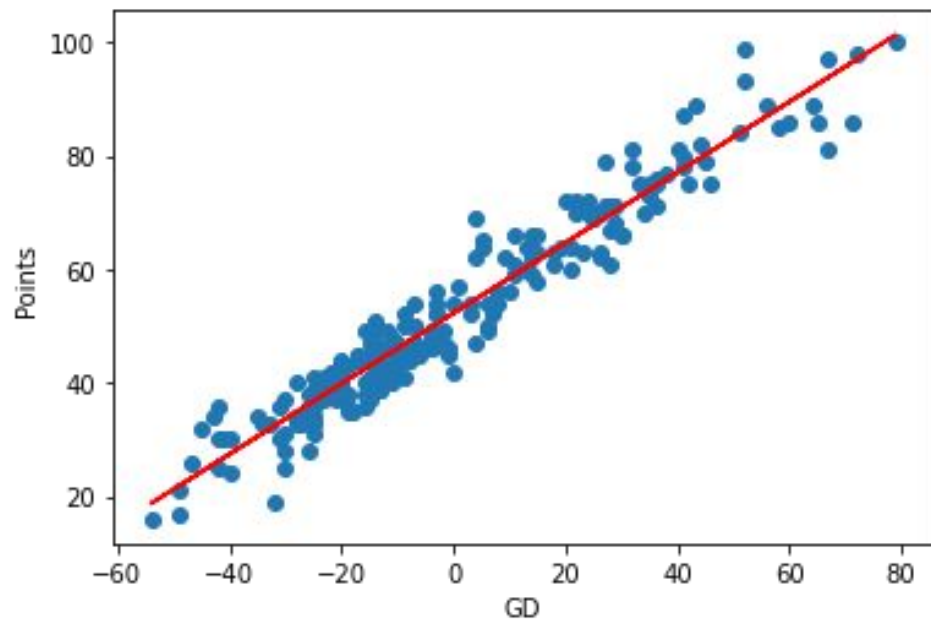


Libraries used :

- **pandas**: fundamental package for data analysis and manipulation.
- **numpy** : mathematical functions and support to operate on multidimensional arrays and matrices.
- **sickit-learn**: machine learning library that features regression algorithm and allows integration with **matplotlib** for plotting.
- **matplotlib**: plotting library for python.



- The independent variable was the goal difference .
- The dependent variable was the points.
- Each data point was imported to the plot as well as the best fit line.

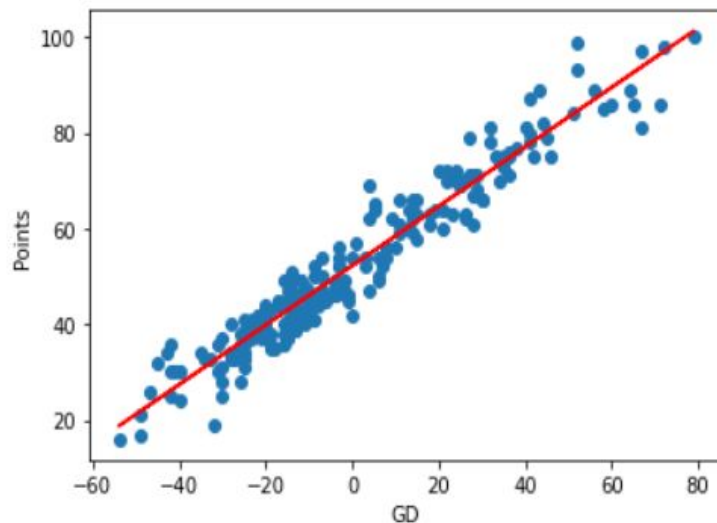




It was observed that a linear relationship existed between the two given variables. In the plot, the line of the best fit has the α equal to the gradient and the β equal to the y-intercept.

From calculating the gradient, we were able to infer that if a club's goal difference increases by one, at the end of the season their total points will increase by 0.62 points.

Similarly, from the y-intercept, we were able to infer that if a club has the the amount of goals that it scores equal to the amount of goals it concedes, then the club will finish the season with a total of 52.26 points.



```
► slope=linreg.coef_  
  intercept=linreg.intercept_  
  Rsqr=linreg.score(x,y)
```

```
► print("Slope=",slope,"intercept=",intercept,"R squared=",Rsqr)
```

Slope= [0.62036799] intercept= 52.268181818181816 R squared= 0.9344078657511434

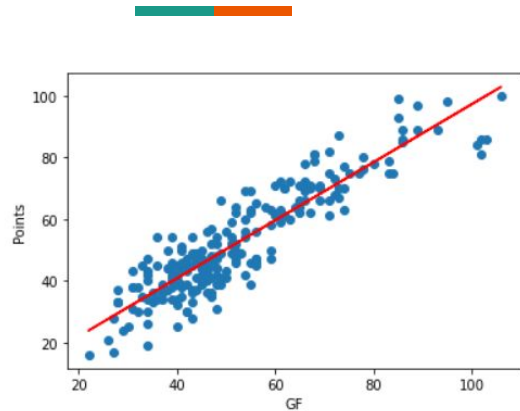


Should you buy an attacker or a defender ?

- One of the most fiercely debated topic in the premier league is whether you should buy an attacking player or spend the same amount of money buying a defender.
- We tried to quantitatively analyse the problem with the help of linear regression.



- The procedure we did was exactly the same as before, but this time, we replaced the X-axis variables with the goals scored and goals conceded, instead of the goal difference.
- After importing each data point to the plot, we observed a linear relationship as before. But, the correlation was weak as compared to the previous plot.

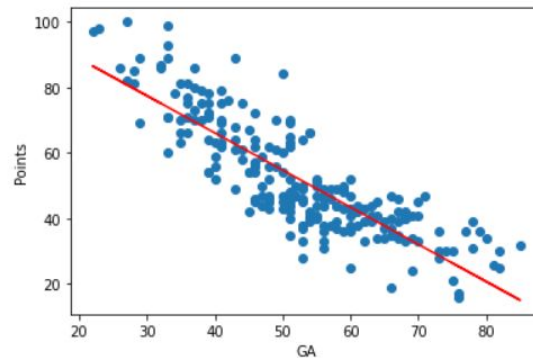


```
3]: slope=linreg.coef_
   intercept=linreg.intercept_
   Rsqr=linreg.score(x,y)

1]: print("Slope=",slope,"intercept=",intercept,"R squared=",Rsqr)

Slope= [0.9403055] intercept= 3.1756866346585113 R squared= 0.8251657335272903
```

A




```
3]: slope=linreg.coef_
   intercept=linreg.intercept_
   Rsqr=linreg.score(x,y)

1]: print("Slope=",slope,"intercept=",intercept,"R squared=",Rsqr)

Slope= [-1.13403848] intercept= 111.47529989554648 R squared= 0.7129300603593791
```

B

- 
- Slope of best fit line in A is positive as more goals scored should lead to higher points. Similarly, slope of best fit line in B is negative as less goals conceded should also lead to higher points.
 - We observed α to be 0.94 and -1.14 in A and B respectively. From this, we can infer that one more goals scored can increase the final point of a team by 0.94 and one goal less conceded can increase the final point of a team by 1.14.
 - So, in terms of the result we obtain from linear regression, a team is better off buying a defender than a scorer in the premier league, as a goal less conceded is worth more points than a goal scored.

The **Expected Threat** measures how the probability of scoring a goal changes before the action and after it, giving value to actions that lead your team towards more dangerous situations.

Dataset

The dataset we have used is the entire last season The 2021 UEFA Europa League Final.

In order to evaluate actions we look at how an action changes the probability of scoring. It is this change in probability of scoring which is the **expected threat (xT)**. If a player makes a pass which moves the ball from a place where it is unlikely for their team to score, to a place where they are more likely to score, then they have increased the xT in favour of their team. In general, the nearer you get the ball to the goal the more likely your team is to score (although if you look carefully passes back to the goalkeeper are also valuable).

0.001	0.002	0.002	0.003	0.003	0.004	0.005	0.006	0.007	0.009	0.011	0.013	0.016	0.017	0.017	0.016
0.002	0.002	0.003	0.003	0.004	0.005	0.006	0.007	0.008	0.01	0.012	0.015	0.018	0.021	0.02	0.021
0.002	0.003	0.003	0.004	0.004	0.005	0.006	0.007	0.009	0.01	0.013	0.016	0.021	0.025	0.027	0.024
0.002	0.003	0.003	0.004	0.005	0.006	0.007	0.008	0.009	0.011	0.014	0.018	0.024	0.029	0.039	0.031
0.003	0.003	0.004	0.004	0.005	0.006	0.007	0.008	0.01	0.011	0.014	0.019	0.027	0.055	0.091	0.071
0.004	0.004	0.004	0.004	0.005	0.006	0.007	0.008	0.01	0.012	0.014	0.019	0.033	0.077	0.142	0.332
0.004	0.004	0.004	0.004	0.005	0.006	0.007	0.008	0.009	0.012	0.014	0.02	0.034	0.085	0.134	0.32
0.004	0.003	0.004	0.004	0.005	0.006	0.007	0.008	0.01	0.012	0.014	0.02	0.028	0.062	0.095	0.085
0.002	0.003	0.004	0.004	0.005	0.006	0.007	0.008	0.009	0.011	0.014	0.018	0.025	0.035	0.042	0.033
0.002	0.003	0.003	0.004	0.005	0.006	0.007	0.008	0.009	0.011	0.013	0.017	0.021	0.026	0.026	0.022
0.002	0.002	0.003	0.003	0.004	0.005	0.006	0.007	0.009	0.01	0.013	0.016	0.019	0.021	0.02	0.02
0.001	0.002	0.002	0.003	0.004	0.004	0.005	0.007	0.008	0.009	0.012	0.014	0.016	0.018	0.017	0.017

This grid can be utilised in a variety of ways, with the primary output being how players improve their team's chances of scoring by carrying or passing the ball. In reality, any movement of the ball between the different zones, such as crossing or even looking at passes received that pull the team further upfield, can be utilised to compute the xT a player adds through their actions. As a result, xT places an $M*N$ grid over the pitch to split it into zones. The value $xT(z)$ assigned to each zone z shows how dangerous teams are in terms of scoring at that place. As a result, xT places an $M*N$ grid over the pitch to split it into zones. The value $xT(z)$ assigned to each zone z shows how dangerous teams are in terms of scoring at that place.