Fantasy Premier League DreamTeam Prediction

Documentation Report

AIS301-1A

Under Supervision of:

Dr. Ghada Khoriba

&

Eng. Tasneem Wael

By:

Yousif Adel Khalil	**202000238**
Mostafa Samer Dorrah	**202000125**
Ali Fouad Marzban	**202002450**
Mohamed Ragab Salem	**202002035**

Table of Contents

1-Problem Statement

FPL DreamTeam prediction project is mainly about statistics, FPL is a game based on the real Premier League event that casts you in the role of a Fantasy manager of Premier League players. Points are calculated according to each player performance and numbers during a specific week (game). Our goal in this project is to build a ML model to predict the DreamTeam of each week; the DreamTeam is a team consisting of 11 players who scored the most points in a certain game week (game). The methods we will approach is that we will get the data from a trusted source, then we will do some cleaning and feature engineering on it, then apply a machine learning model on the data and then tuning the hyperparameters of the model to make the best model for the data with the least percentage of error.

2-Related Work

There is a couple of Related work listed below:

1. https://rstudio-pubs-static.s3.amazonaws.com/577042_d4492e2e511848fb97ef839be077e9b8.html
   Here they did some visualization and analysis on FPL data and chose Random Forest as their model to predict the total points of each player in the premier league.

   Similar to us but not exactly the same.

2. https://github.com/saheedniyi02/fpl-ai
   In this repository, he made 2 models, one is depending on the other. The first one is a classification model to predict whether a player will start the game or not, and the 2nd model is a regression model to predict the total points of the players that the first model predicted to start the game.

As the first one, it is a similar and more precise one, but also not the same idea exactly.

3-Model Architecture

3.1-Logistic Regression (Yousif Adel Khalil)

Logistic Regression is a classification model, it is relevant to Linear Regression (Regression) model, but how can we convert the linear regression equation shown down below to logistic regression using the Sigmoid/Logistic function?

Linear Regression Equation

$$\hat{y} = \beta_0 x_0 + \cdots + \beta_n x_n$$

$$\hat{y} = \sum_{i=0}^{n} \beta_i x_i$$

Sigmoid/Logistic Function

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

To convert it, we need to plug in the linear regression equation into the sigmoid/logistic function to create a logistic regression

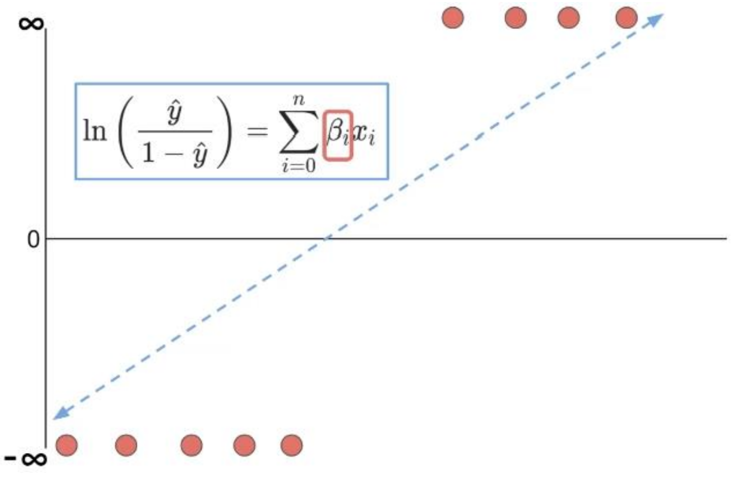
$$\hat{y} = \beta_0 x_0 + \cdots + \beta_n x_n$$

$$\hat{y} = \boxed{\sum_{i=0}^{n} \beta_i x_i} \qquad \sigma(x) = \frac{1}{1 + e^{-x}}$$

$$\hat{y} = \sigma(\beta_0 x_0 + \cdots + \beta_n x_n)$$
$$\hat{y} = \sigma\left(\sum_{i=0}^{n} \beta_i x_i\right)$$

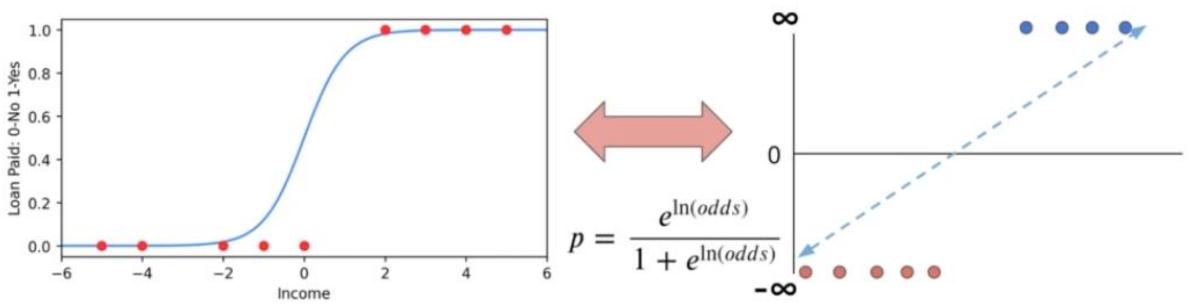
After Solving for ln(odds)
$$\ln\left(\frac{\hat{y}}{1-\hat{y}}\right) = \sum_{i=0}^{n} \beta_i x_i$$

Here, the relationship between ln(odds) and the β Coefficient is linear. Where odds are the ratio of the probability of event happening to the probability of the same event not happening.

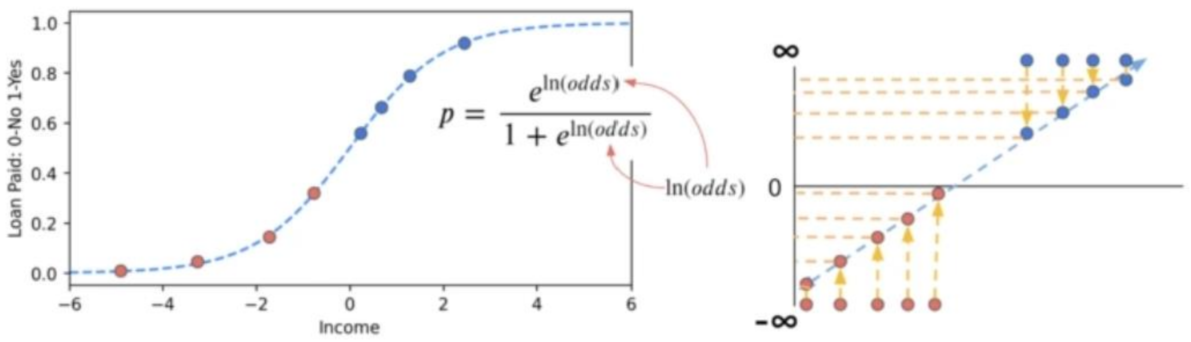

∞
0
-∞
$\ln\left(\frac{\hat{y}}{1-\hat{y}}\right) = \sum_{i=0}^{n} \boxed{\beta_i} x_i$

Generally, Logistic Regression is non-linear in terms of odds or probability, however Logistic Regression is linear in terms of ln(odds) or log odds.

So, we generally say it is a linear model because we use ln(odds) in the Maximum Likelihood Estimation instead of just the probability.



Maximum Likelihood's goal in the first step is to revert the ln(odds) back to a probability. (From the RHS to the LHS in the figure above)



After the first step, measure the likelihood of these probabilities.

Notice while the points in the RHS is on infinity and -infinity, so the points are being projected to the straight line and then measure the likelihood.

Now with the cost function which is called also ‘log loss’:

$$J(\mathbf{x}) = -\frac{1}{m}\sum_{j=1}^{m} y^j \log\left(\hat{y}^j\right) + (1-y^j)\log\left(1-\hat{y}^j\right)$$

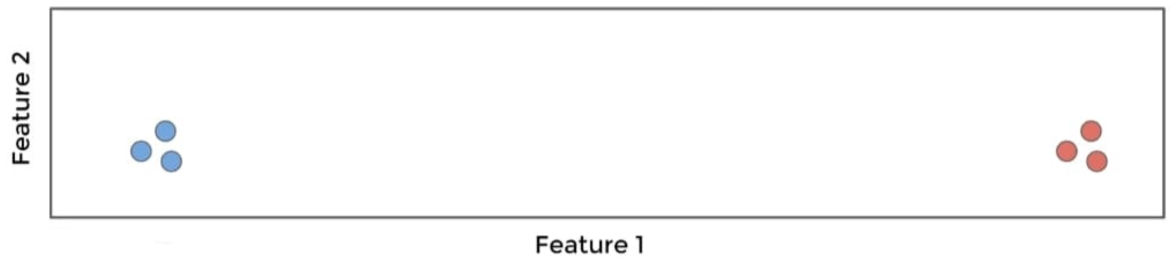
A regularization term can be added to the log loss function, it can be l1 (lasso) regularization and l2 (Ridge) regularization.

In the case of our project, Logistic Regression performs better with l1 rather than l2 because it helps with Feature selection as it tend to shrink the coefficients to a small number or zero, on the other hand l2 tends to shrink coefficients evenly without shrinking it till zero.

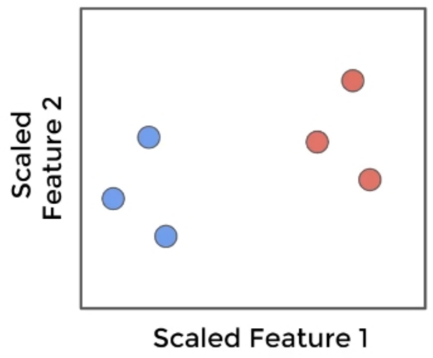
Let’s finish this section by talking about Logistic Regression Solvers. As I stated above, we tend to use l1 penalty term we must choose a solver that supports it at least. Saga and liblinear solers are the only solvers that support l1 penalty. However, liblinear is only efficient and good for small and simple datasets, while ours is neither small nor simple, we must choose Saga because it is also very efficient in large datasets and a fast convergence guaranteed when the data is on the same scale.

3.2-KNN(K-Nearest Neighbor) (Mostafa Samer Dorrah)

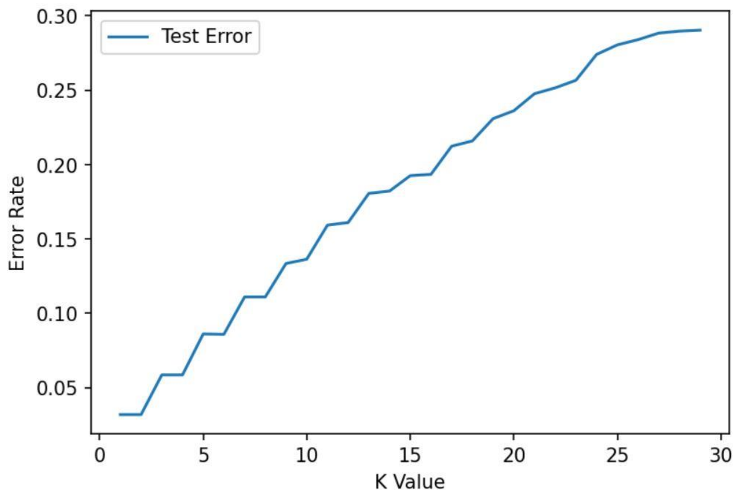
KNN or K-nearest-neighbors is a machine learning algorithm that works by assigning a label to a new data based on the distance between the old and the new data. It Determine the number of links by the value of K.



In KNN some points can be far from some features and for KNN the distance matter, so it is always important to Scale the data.



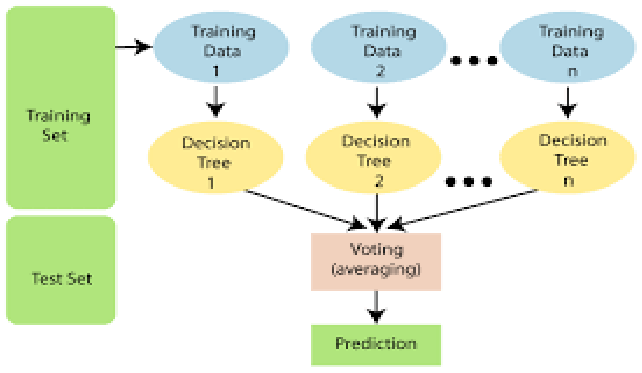
The K is the most important parameter in KNN so to find the best one I did Grid search which resulted in the best value of 1. So, to make sure the number is correct I did the Elbow method then graphed the K's.



Here we Can see the that the error rate is increasing when the k increases. Therefore, the best K is 1.

3.3-Radnom Forest (Ali Fouad Marzban)

Random forest is a Supervised Machine Learning Algorithm that is used widely in classification and Regression problems. It builds decision trees on different samples and takes their majority vote for classification and average in case of regression.



One of the most important features of the Random Forest Algorithm is that it can handle the data set containing continuous variables as in the case of regression and categorical variables as in the case of classification. It performs better results for classification problems.

3.4-Naïve Bayes Classifier (Mohamed Ragab Salem)

naive Bayes classifiers are a family of simple "probabilistic classifiers" based on applying Bayes' theorem with strong (naive) independence assumptions between the features (see Bayes classifier). They are among the simplest Bayesian network models

Bayes' Theorem

Bayes' Theorem finds the probability of an event occurring given the probability of another event that has already occurred. Bayes' theorem is stated mathematically as the following equation:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Basically, we are trying to find probability of event A, given the event B is true. Event B is also termed as evidence.

P(A) is the priori of A (the prior probability, i.e. Probability of event before evidence is seen). The evidence is an attribute value of an
unknown instance (here, it is event B).

P(A|B) is a posteriori probability of B, i.e. probability of event after evidence is seen.

Now, with regards to our dataset, we can apply Bayes' theorem in following way:

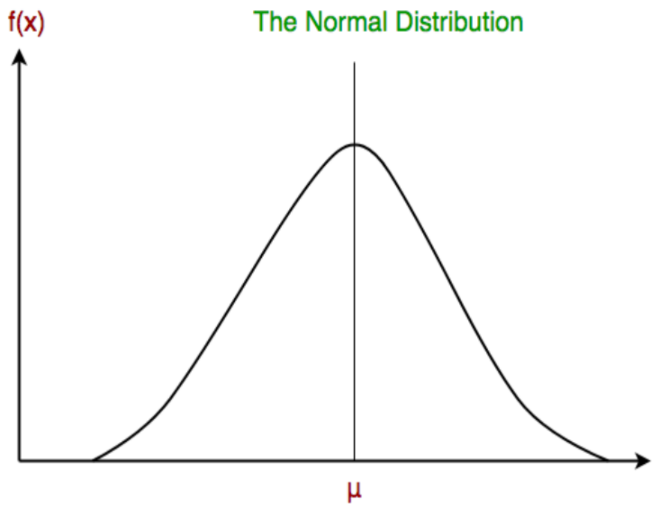
$$P(y|X) = \frac{P(X|y)P(y)}{P(X)}$$

where, y is class variable and X is a dependent feature vector (of size *n*) where:

$$X = (x_1, x_2, x_3, ..., x_n)$$

Gaussian Naive Bayes classifier

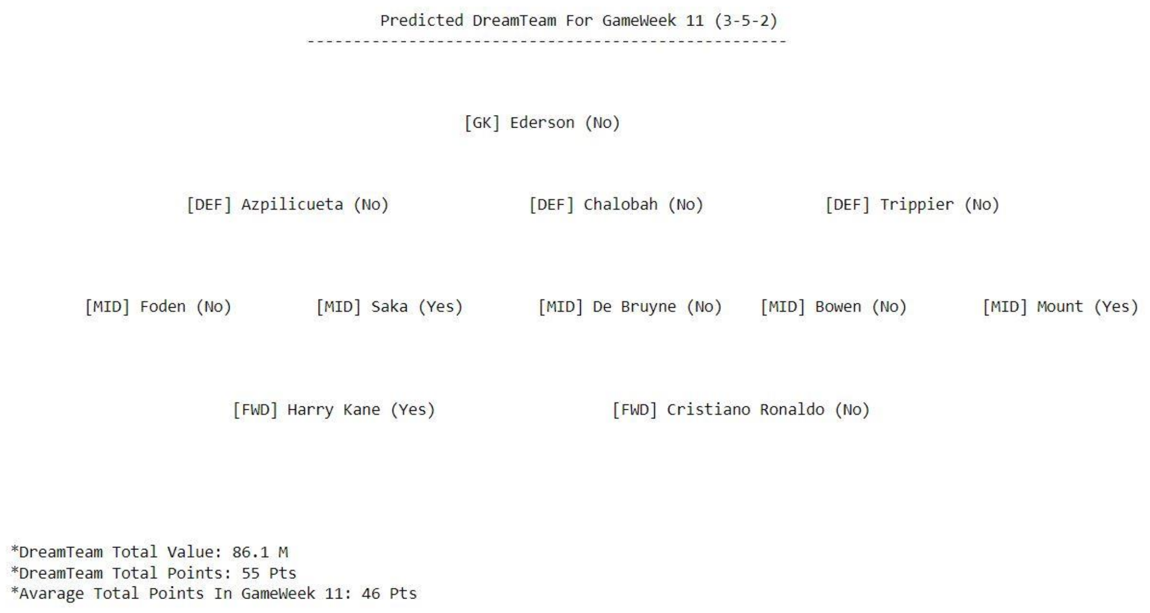
In Gaussian Naive Bayes, continuous values associated with each feature are assumed to be distributed according to a Gaussian distribution. A Gaussian distribution is also called Normal distribution. When plotted, it gives a bell-shaped curve which is symmetric about the mean of the feature values as:


f(x)
The Normal Distribution
μ

4-Evaluation & Results

4.1-Logistic Regression (Yousif Adel Khalil)

The best model evaluated on Test Data was Logistic Regression:



```
                          Predicted DreamTeam For GameWeek 11 (3-5-2)
                      ---------------------------------------------------


                                        [GK] Ederson (No)


                 [DEF] Azpilicueta (No)              [DEF] Chalobah (No)            [DEF] Trippier (No)


          [MID] Foden (No)        [MID] Saka (Yes)        [MID] De Bruyne (No)    [MID] Bowen (No)       [MID] Mount (Yes)


                     [FWD] Harry Kane (Yes)                    [FWD] Cristiano Ronaldo (No)




  *DreamTeam Total Value: 86.1 M
  *DreamTeam Total Points: 55 Pts
  *Avarage Total Points In GameWeek 11: 46 Pts
```

The Classifier (Logistic Regression) has been final evaluated and tested on the GameWeek 10 of Season 2022/2023, which it means the model's goal is to predict the DreamTeam of GameWeek 11 according to sum statistics and values of the past.

The model has predicted 3 correct players out of the 11 in the DreamTeam of GameWeek 11. Comparison to human error is that the Scout's Selection for GameWeek 11 has only 1 correct player. Also, there is one mutual prediction within the model and the Scout's Selection, which is the Spurs Attacker (FWD) 'Harry Kane' who was found to be in the DreamTeam of GameWeek 11.

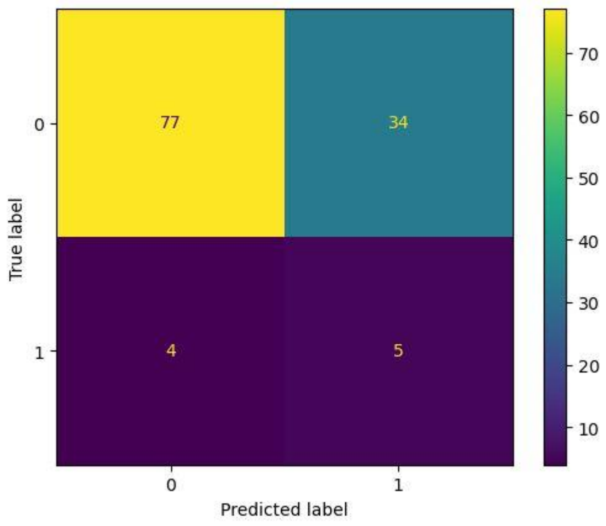
-Accuracy Score of the Model in the Final Test Data: 0.6833333333333333

This Accuracy Score means that it is 68% Accurate on predicting the whole Test set regardless which class is being predicted more than the other.

-	precision	recall
0	0.95	0.69
1	0.13	0.56

While Precision is the ratio of True Positives to Total Predicted Positives. And Recall is the ratio of True Positives to Total Actual Positives.

Both of These Classification Metrics will be understood after visualizing the Confusion matrix below.



True Negatives: 77

True Positives: 5

False Negatives: 4

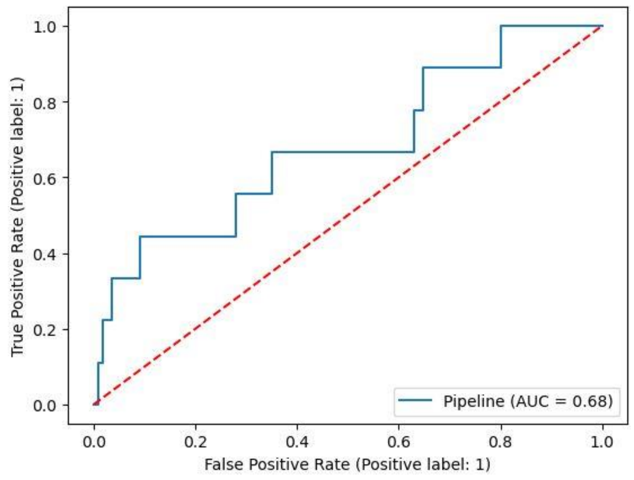
False Positives: 34

Precision = TP/ (Total Predicted Positives) = 5/39 = 0.13

Recall = TP/ (Total Actual Positives) = 5/9 = 0.56

Notice We only focus on evaluating the model's performance on predicting the class '1' which is our major concern in the project.

Now, let's see the roc curve



The red-dashed line is representing a random prediction. So, if the model's roc curve is over the red-dashed line that means that it is performing well. And it is a very good indicator and metric to test classification models.

4.2-KNN (Mostafa Samer Dorrah)

In the KNN model the results came back disappointing With Classification report with 0.00 precision and recall in the 1's.

	precision	recall
0	0.92	0.95

1	0.00	0.00

So, let’s understand what went wrong. First KNN works with using the o ld data distance from the new data which in our situation (Data) is not th e best approach. KNN works on memorization and our data needs model s like logistic regression with optimization techniques.

So, the sum up the results we can conclude that KNN is not the best mod el for our type of data.

4.3-Radnom Forest (Ali Fouad Marzban)

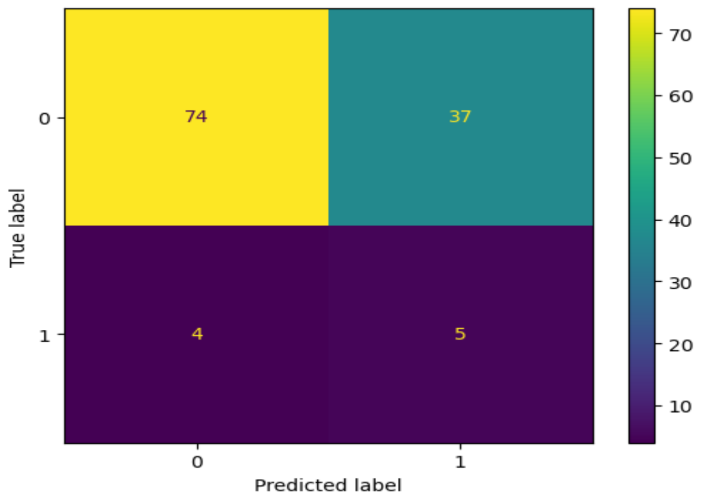
-Accuracy Score of the Model in the Final Test Data:
0.6633333333333333

This Accuracy Score means that it is 66% Accurate on predicting the whole Test set regardless which class is being predicted more than the other.

-	precision	recall
0	0.95	0.67
1	0.12	0.56

While Precision is the ratio of True Positives to Total Predicted Positives. And Recall is the ratio of True Positives to Total Actual Positives.

Both of These Classification Metrics will be understood after visualizing the Confusion matrix below.



True Negatives: 74

True Positives: 5

False Negatives: 4

False Positives: 37

Precision = TP/ (Total Predicted Positives) = 5/42 = 0.12

Recall = TP/ (Total Actual Positives) = 5/9 = 0.56

Notice We only focus on evaluating the model's performance on predicting the class '1' which is our major concern in the project.

4.4-Naïve Bayes (Mohamed Ragab Salem)

-Accuracy Score of the Model in the Final Test Data: 0.7571

This Accuracy Score means that it is 75.71%

Accurate on predicting the whole Test set regardless which class is being predicted more than the other.

and we calculate the difference between training data and test data we don't find overfitting in our data

Training set score: 0.7616

Test set score: 0.7571

And I use the naive Bayes algorithm because of my personal opinion all of future depend on each other I know naive bayes algorithm using of independent features, but we are multiplying all probability of each features that's dependent

5-References

1-https://github.com/vaastav/Fantasy-Premier-League/tree/master/data

2-https://fantasy.premierleague.com/

3-
https://docs.google.com/spreadsheets/d/112sIeBPL84j9VLhguaw4ViAcNYSP0iLWp0OKkx-01mE/edit#gid=1446219884

4-https://www.fourfourtwo.com/features/fpl-what-is-fantasy-football-and-how-does-it-work

5-
https://www.udemy.com/share/103I0w3@hr15sskCnPmbYgiPutXz7PnPx5ytF9MpPw3tEeKnUofdo_d7y5GSvZ3ikNHQ0_hIeg==/