Scientific Journal Impact Factor (SJIF): 3.518
International Journal of Modern Trends in Engineering and Research
ISSN (ONLINE) : 2349 - 9745
ISSN (PRINT) : 2393 - 8161

Forecasting English Premier League Match Results

Kusum Lata[1], Prabha Gupta[2]
[1]*Computer Science and Engineering, Delhi Technical University*
Delhi India
[2]*School of IT, Centre For Development Of Advanced Computing*
Noida India

Abstract—In this paper, we predicted the results of soccer matches in the English Premier League (EPL) using artificial intelligence and machine learning algorithms. A feature set that includes the game day performance of the player was generated from historical data.

After gathering the data set, an average of the performance of all the players was calculated which is used for categorization of the data set. Then a random graph of a player was selected from the given category. A cumulative team performance was calculated by the performance of all the players of the team. Then the performances of the two teams were compared to predict the final result. The prediction was in one of three classes for each game: win, draw, or loss**.**

Keywords— Baseline, vertical distance, schema

I. INTRODUCTION

The Premier League is an English professional league for men's association football clubs. At the top of the English football league system, it is the country's primary football competition. Contested by 20 clubs, it operates on a system of promotion and relegation with the Football League. Besides English clubs, the Welsh clubs that compete in the English football league system can also qualify to play.

The Premier League is a corporation in which the 20 member clubs act as shareholders. Seasons run from August to May, with teams playing 38 matches each (playing each team in the League twice, home and away). So, 380 match per season in total. Usually games are played in the afternoons of Saturdays and Sundays, otherwise during weekday evenings. It is currently sponsored by Barclays Bank and thus officially known as the Barclays Premier League and is colloquially known as the Premiership. It is commonly known to as English Premier League mostly outside the UK [5].

The main object of our work was to predict the outcome of the match played by any of the 2 out of the 20 teams in the league. A machine learning application was developed to predict this outcome. The basis of the prediction used was player performance factor throughout the game.

Machine learning explores the analysis and design of algorithms that can learn from and make predictions on the basis of dataset. Machine learning is a branch of computer science that evolved from the study of pattern recognition and computational learning theory in artificial intelligence. Such algorithms operate by building a model from example inputs in order to make data-driven predictions or decisions, rather than following strictly static program instructions.

Here the basis of the algorithm was to study the performance of each player for each match. After studying the performances of all the players, the cumulative performance of the entire team was computed. Then these team performances were compared to determine the probability of each team winning, loosing or the match being a draw.

II. LITERATURE SURVEY

Some organization bet on the result of match for money making. These are gambling organizations who want to forecast the match result for the benefit of odds makers. One example of the many that we examined comes from a CS229 Final Project from Autumn 2013 by Timmaraju et al.[3] . Their work focused on building a highly accurate system to be trained with one season and tested with one season of data. Because they were able to use features such as corner kicks and shots in previous games, as well as because their parameters were affected by such small datasets, their accuracies rose to 60% with an RBF-SVM.

The idea for this paper also taken from the work of Rue et al., who used a Bayesian linear model to predict soccer results. Notably, they used a time-dependent model that took into account the relative strength of attack and defence of each team [2].

Another attempt at predicting soccer works was done by Joseph et al. This group used Bayesian Nets to predict the results of Tottenham Hotspur over the period of 1995-1997. It relies upon trends from a specific time period and is not extendable to later seasons [10].

Other paper has used only a single attribute i.e. overall team performance, to determine the outcome of the match. However, to determine the outcome of the match here we have considered total SEVEN attributes (performance at every 15 minute interval) instead of just one hence providing a more detailed way of comparing the teams.

III. DATA SET GENERATION

Input data for this particular problem, taking a 'player' as the smallest entity, contained the performance of a player at every 15 minutes interval in a 90 minute football match. Because there is confusion with the player name the data point, performance of a player at every 15 minutes interval in a 90 minute football match, had the team name and the player jersey number as the pair (team-jersey number) is unique for a player [4][5]. A team usually assigns a jersey number to a player depending on the choice or the availability of the number. Taking the pair as a unique ID is a better option than name of the player as the input data may have conflicts with the player name spell. The schema for this particular table in the input data is defined as follow

(Team name, Date, Jersey number, Performance of player at 0 min for that match, Performance of player at 15 min for that match, Performance of player at 30 min for that match, Performance of player at 45 min for that match, Performance of player at 60 min for that match, Performance of player at 75 min for that match, Performance of player at 90 min for that match)

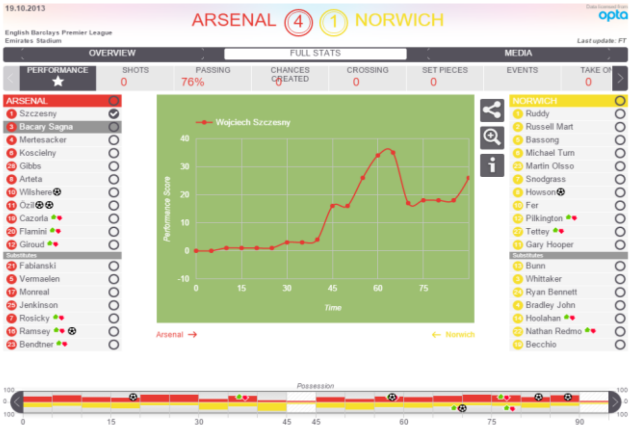


Fig. 1 A sample line graph for the performance of the player at 15 min interval.

For computation purposes the input data contains another set of attributes that links the unique team-Jersey number pair to the player name. There may appear a pair that may be repeated but that has only one possibility that the pairs are for two different seasons. Thus the input data for different seasons is supplied in two mutually exclusive sets such that they do not hinder with each other and may lead to errors. This data set or table combined with the above set of input will give the input

data a transformation that will be explained in the next section. The schema for this particular table is defined as follow
(Team name, Jersey number, Player name)

Both the above defined sets or tables, are defined separately for each of the seasons that are used in evaluation. If one has to expand the data set these two tables should be added per season.
Another set in the input data that contains the details of a particular match. This input set contains all the matches that are used as a training set for the machine learning. This set is the basis for categorization for the training and testing set. The schema for this set is as follows
(Date of match, Team 1, Team 2, Full time score, Half time score)
The above defined table or set contains all the matches as it contains the date field , name for team 1 and name for team 2. This triplet will be a unique value thus all the matches can be stored in a table without any fear of repetition as any date may contain multiple matches but the triplet is unique
The input data also contains another set tables as given problem had a major issue of the players that are substituted in the game. A substitution is a replacement for a player in the playing eleven and a player on the bench just takes his place in the playing eleven. The substituted players in most cases do not play the entire match length but is a replacement for a player.
Now here there is a contradiction for the evaluating the performance for the players in substitution and those in the playing eleven. Here the approach assumes that the replacement for the players just replaces their overall performance as a team and the substituted player performance is only considered in evaluation if it is not zero. The reason for taking the non-zero performance is that whilst the player was on the bench his performance was zero thus this zero in the performance field at any time t it is not taken into consideration as it may degrade or upgrade his performance.
The first table is also defined separately in the input data set for players on the bench.
Thus this makes three tables per season to be added in the input data set for increasing the number of data points per season

IV. PRE-PROCESSING AND TRANSFORMATION

The first transformation was to substitute the Team-Jersey number pair as defined in the previous section with the name of the player using the second set or table in the data set.
After the transformation the schema for the new table be :

(Player Name, Team name, Date, Performance of player at 0 min for that match, Performance of player at 15 min for that match, Performance of player at 30 min for that match, Performance of player at 45 min for that match, Performance of player at 60 min for that match, Performance of player at 75 min for that match, Performance of player at 90 min for that match)
The same is done for the table in the input data set for the players on the bench. The schema for the table for the players on the bench will be

(Substituted Player Name, Team name, Date, Performance of player at 0 min for that match, Performance of player at 15 min for that match, Performance of player at 30 min for that match, Performance of player at 45 min for that match, Performance of player at 60 min for that match, Performance of player at 75 min for that match, Performance of player at 90 min for that match)
And the procedure is repeated for all the seasons in the data set.

The second transformation used for this problem is to generate the overall difference in team performances for a match. Let us suppose there are two teams, say A and B in a match. The first task is to evaluate the overall performance of the teams. This task is accomplished by taking all the players for the particular team including the players on the bench into consideration and adding the performance of each player at each time t, taken into consideration at an interval of 15 minutes, separately. This will lead us to an overall performance of the team as a whole. The schema for the overall team performance is as follows:

(Team name, Date, Performance of team at 0 min for that match, Performance of team at 15 min for that match, Performance of team at 30 min for that match, Performance of team at 45 min for that match, Performance of team at 60 min for that match, Performance of team at 75 min for that match, Performance of team at 90 min for that match)

This procedure gives the overall performance of the teams in a match.
As mentioned above the teams are assumed to be A and B, and the performances are compared by the algorithm that will be discussed in the next section.
The transformed data set will now look like the following schema:

(Team 1 name, Team 2 name, Date, Performance at 0 min for that match, Performance at 15 min for that match, Performance at 30 min for that match, Performance at 45 min for that match, Performance at 60 min for that match, Performance of player at 75 min for that match, Performance at 90 min for that match, Full Time score, Half Time score)

The third transformation is to combine all the tables formed by the first transformation for the players in the playing eleven and similarly for the players on the bench. Thus this transformation would lead to two tables, one with all the players performances those were in the playing eleven combined together irrespective of their seasons and the second with all the players performances those were on the bench combined together irrespective of their seasons. The schema will be same as the schema in the first transformation.

The fourth and the final transformation include the tables or the sets that are generated by the third transformation. This transformation is only for optimization purposes. This transformation is kind off a hashing for the third transformation .The tuples or the records in the tables generated by the above transformation are indexed, now this transformation sorts the above tables in a format such that there is only one entry for all the players and the index numbers in the above table they show up.

Thus the table generated by this transformation by only one search query of the payer name the program can access all the records in the table produced by the third transformation.
The schema for this table :

(Player Name, Index 1, Index 2, Index 3,…)

The number of columns in this table is variable
Separate tables are generated by using the same transformation in the same way for the players in playing eleven and the players on the bench.
The schema for the table generated for the players on the bench :

(Substituted Player Name, Index 1, Index 2, Index 3,...)

This will conclude the transformations.
The modifications or transformations does not make the false as the transformations are simple basic substitutions that does not hinder with the data for the performances of the player thus they are valid.

V. APPROACH

The task here is to predict the result for a match using machine learning. The approach here used is to categorize the matches into training set and testing set. The matches in training set are used for the past result as in the matches in the test set are with the aim to predict their result using the computation of the matches in the training set. This will be done by calculating the performance of the overall teams and then comparing the graph and comparing this result with matches in the training set.

The first task is to calculate the performance of the overall teams for this we need to calculate the performance of each player playing the match for the team. First we will categorize all the graphs of a player in a team. The categorization is on the bases of a baseline.

Baseline: A baseline is an average performance line calculated from all the players in the playing eleven and on the bench in the training set for all the points at time t with interval of 15 minutes for a 90 minutes match.

a) Baseline Calculation

While calculating the average performance at any time t for any interval of 15 minutes the performance for player in playing eleven it is directly included in the sum and the count is incremented but if the player was on the bench and the performance at any time t is 0 it is not considered in the calculation as it would increase the count would not affect the sum thus it is ignored as the player was not a part of the match until that point of the match

```
for all player graphs :
sum[t]=0
count[t]=0
for time t at every 15 minute interval:
if not(perfor[t]==0 and player on the bench) :
sum[t]=sum[t]+perfor[t]
count[t]++
```

After establishing the baseline for the training set the player's graphs can be categorized whether his performance was better than average or below average. The average is the baseline.

b) Player graph categorization

Now there is a graph that represents the baseline. All the performances for the player are also represented as graphs. Now for the particular player the graphs are categorized according to the baseline

The indexes of the graphs above and below are categorized and clubbed in two separate arrays

c) Comparing two graphs

Here another task is allocated to compare two graphs in general with discrete set of points. The approach used here is just comparing the vertical distances. A counter is set to zero before encountering the first discrete point. Then taking the convention that first graph is above the second graph as positive and if the opposite is true the value becomes negative. The vertical distances are cumulated according to the convention in the counter. Thus at the end after the last discrete point the value of counter will decide whether the assumption the first graph is above the second is true or not.

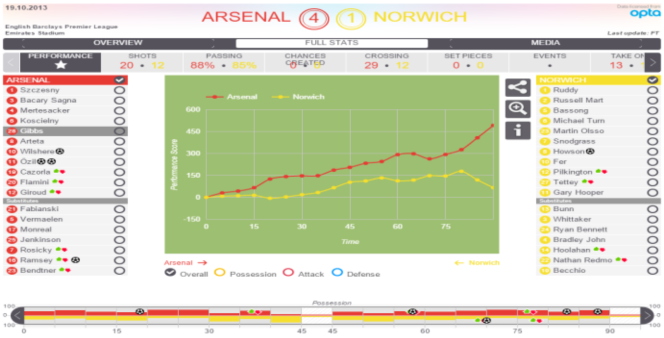


Fig.2 A sample line graph to compare performance of two random players at each interval of 15 min.

```
counter=0
for time t interval of every 15mins:
```

```
count=count+graph1[t]-graph2[t]
If counter value is positive graph1 is above graph2
Else if counter value is negative graph 2 is above graph1
Else both the graphs are more or less the same
```

Now all the graphs for a player are categorized. Now the need is to select a graph for the player. The categorization of the graphs will lead us to probabilities or randomness. As the graphs are categorized whether they are above the baseline or below there will be probabilities related to both the events.

Then there is a random selection on the bases of the probabilities of a side above or below of the graph. The selected event in the random process will lead to a side. Then an existing player graph is selected randomly from the selected side.This is the process to select a graph of a particular player.

d) Prediction of the testing match

In the similar way the players for the testing match a graph for each player involved in the match is selected and the overall performance for the teams are calculated.

Calculation of overall performance of the team

```
for time t at every interval of 15 minutes :
sum[t]=0
for all players playing a match for a team:
sum[t]=sum[t]+perfor[t]
```

Suppose the testing match is between Team A and Team B the overall performance of the teams is calculated by adding all the player performances for the team including the players on the bench as shown above in the algorithm.

Then there are two graphs corresponding to each of the teams. Taking the same convention in mind that the first graph is above thus the Team A graph is assumed above the Team B graph. The vertical difference between the graphs will lead us to another graphs having the vertical differences between the discrete points for both the graphs.

e) Graph formed by vertical differences

```
for time t at every interval of 15 minutes:
diff[t]=teamA[t]-teamB[t]
```

This graph of vertical differences is evaluated for all the matches in the training set. This graph is then compared with all the graphs of the training set and then categorized according to the final score for the graphs. The graph would contain the diff array for each of the matches in training set

Now the there are two graphs to compare created by the discrete set of points from the diff[t] as shown above in the algorithm.

As shown above in the snippet that the comparison would produce a counter. The counter values generated in the comparison of the graphs are appended into three arrays win, draw and lose depending upon their results.

The result of any arbitrary match say team1 vs team2 in the training set is compared with teamA vs teamB in the testing set the graphs formed by vertical differences are compared as:

```
If  team1 won:
counter is appended in win array
else if team1 lost (team2 won):
counter is appended in lose array
```

```
else
counter is appended in draw array
```

(f) Calculating the prediction percentages

The average value for each array will give the average vertical displacements between the graph. The counter is a measure how close or how far the two graphs are. While taking the average the absolute value is taken into consideration

Avg counter win (w)

```
sum=0
count=0
for all counters in win array:
sum=sum+counter[count]
count++
w=sum/count
```

Similarly average counter values for lose (l) and raw (d) can be calculated.

Now the values w,l and d are inverted because the higher the value of any of these variables the lower the possibilities match.

$w=1/w, l=1/l, d=1/d$

Now the percentages can be evaluated

Win percentage for teamA

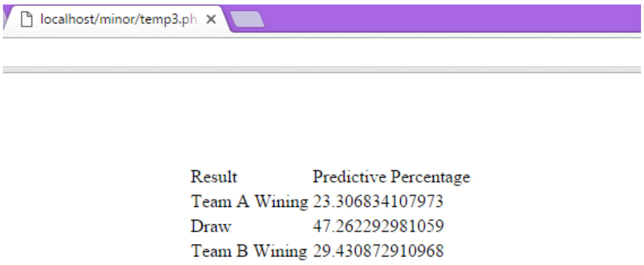
$per(w)=w/(w+l+d)$

Draw percentage

$per(d)=d/(w+l+d)$

Win percentage for teamB

$per(l)=l/(w+l+d)$



Result	Predictive Percentage
Team A Wining	23.306834107973
Draw	47.262292981059
Team B Wining	29.430872910968

Fig.3 Prediction by the system with percentage factor.

VI. CONCLUSIONS

This approach was developed to predict the outcome of a soccer match. The dataset included the performance of the players during games at every 15 minute interval in the form a graph. These graphs were compared to determine how a player would play during a given game. After the performances of all the players were determined, they were used to calculate the performance of the team during that match. The performances of the two teams were compared to determine who would win.

After testing this project using K- fold cross validation method, a maximum accuracy of 41.2% was obtained when the value of K was 10. The accuracy of the project could be improved using a larger data set.

VII FUTURE SCOPE

1. To improve the accuracy for this problem a different approach towards the selection for a player's graph could be used.
2. The graph comparison is been computed using a counter and calculating the vertical distances a different approach towards graph comparison can be used.
3. While evaluating the performances the user can select any of the eleven players given in the squad these options are generally classified into forward, mid, defense etc. hence these categorizations can be taken into considerations.
4. A different method to induce probabilistic selections can be used.
5. The classification of the counters to calculate the prediction percentage is simple average different mathematical and logical relations can be used.
6. As this graph comparison is similar to the fitted distribution statistical analysis, those techniques can be used to check whether they increase the accuracy.

REFERENCES

[1] Ben Ulmer and Matthew Fernandez, "*Predicting Soccer Match Results in the English Premier League"*, cs229.stanford.edu, 2014.

[2] H. Rue and O. Salvesen, *"Prediction and retrospective analysis of soccer matches in a league"* Journal of the Royal Statistical Society: Series D (The Statistician) 49.3 (2000): 399-418

[3] A. S. Timmaraju, A. Palnitkar, & V. Khanna, Game ON! Predicting English Premier League Match Outcomes, 2013.

[4] "Barclays Premier League Squad Lists 2013/14", <http://www.premierleague.com/en-gb/news/news/2013-14/sep/premier-league-squad-lists-201314.html>

[5] "Premier Leaque 25-man Playing Squad Season 2012-13",

[6] http://www.myfootballfacts.com/Premier_League_Squads_2012-13.html

[7] "Premier Leaque by Seasons ",
http://www.myfootballfacts.com/Premier_League_by_Seasons.html

[8] "Team Directory", http://www.squawka.com/team-directory

[9] https://github.com/footballcsv/eng-england

[10] A. Joseph, A. E. Fenton, & M. Neil, Predicting football results using
Bayesian nets and other machine learning techniques. Knowledge-Based Systems 19.7 (2006): 544-553