Predicting the Winner of Spain LIGA BBVA(Soccer) Match

Mohanakrishna Vanamala Hariprasad
The University of Texas at Dallas
mxv160430@utdallas.edu

Vinaya Ganiga
The University of Texas at Dallas
vsg160130@utdallas.edu

1 Introduction

Football or Soccer, is a team sport played between two teams of eleven players with a spherical ball. It is played by 250 million players in over 200 countries and dependencies making it the world's most popular sport. The game is played on a rectangular field with a goal at each end. The object of the game is to score by getting the ball into the opposing goal. The game consists of 11 players including the goal keeper [3].

In this project, we attempt to predict the most probable winner given two teams using the different attributes of this game such as: the players and their performance, overall teams' performance, and whether the team is playing in its home ground or not. We have scraped the data from a subset of the dataset : http://football-data.mx-api.enetscores.com/ and https://www.kaggle.com/hugomathien/soccer , of which we are concentrated in predicting the most probable winner given match statistics in "Spain LIGA BBVA".

2 Problem Definition and Algorithm

2.1 Task Definition

One of the things that makes predicting outcomes tricky is the significant incidence of draws (neither team wins, if they both score the same number of goals) as compared to other sports. Most popular games viz., tennis, cricket, baseball and American football either have no draws or very few draw occurrences. Consider this - out of the 380 games in the 2012-13 season, 166 games were won by the home team, 106 games by the away team and there were 108 draws! To put these numbers in perspective, we calculate the season's entropy. We regard home wins (1), away wins (2) and drawn games (3) as three separate classes and compute the fraction of each of these outcomes.

p1 = 166/380 p2 = 106/380 p3 = 108/380

Since, we have a 3-way classification, we use the base-3 logarithm.

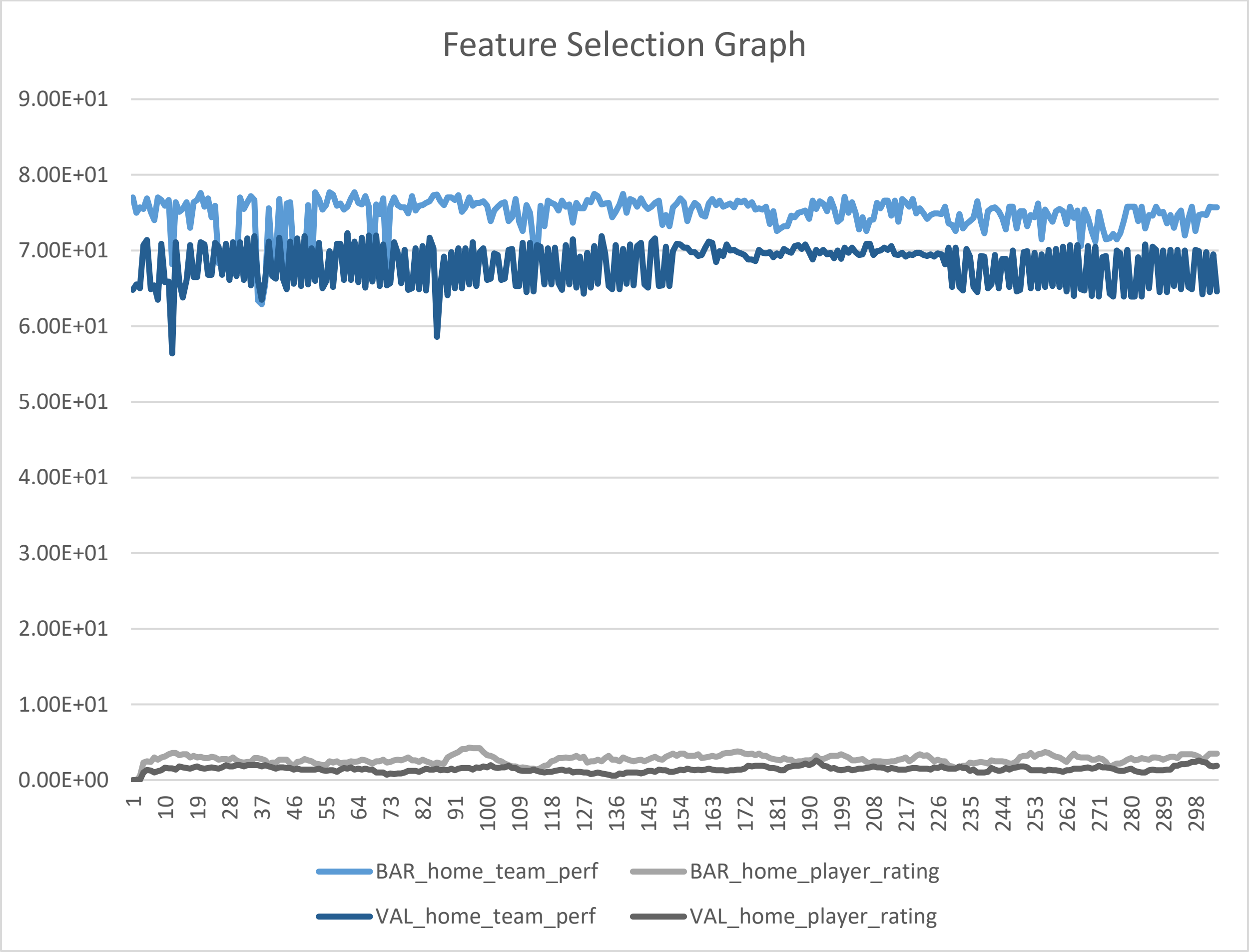
$\text{Entropy} = -(p1 * \log3(p1) + p2 * \log3(p2) + p3 * \log3(p3)) = 0.9789$

An entropy of 1 would correspond to a perfectly random setting (p1 = 1/3, p2 = 1/3, p3 = 1/3). So, considering this, a random prediction would give an expected accuracy of 33.33%. Another naive, but a marginally better approach would be to predict a home win always, which would deliver 1 44-45% accuracy (about the average fraction of home wins in EPL). These are however, not very good numbers. [1]

2.2 Algorithm

We believe the performance of the team over past matches considerably helps in predicting the team's performance in future matches. The rating of each player provided by the dataset is used to get a team's overall player rating which provides a deeper insight of how good a goal keeper, striker, mid-fielder of one team is compared to another. The team playing in home ground has an edge over the one visiting to play. Hence we feel that the team past performance, average of all players rating and home team or not features are justified to use for our match predictions.

The dataset provides with information such as the date on which a match is played, the 11 players who played for a team, the player rating for each player, the league information, the venue, the number of goals scored by each team.



As shown in the above graph, the features team performance and player rating clearly distinguishes the probability of the winning team. Here, BAR_home_team_perf refers to performance of Barcelona team performance and VAL_home_team_perf refers to Valencia CF team's performance

To understand the dataset, it is essential that we first understand the below notations used in the dataset:

date = Match Date (dd/mm/yy)
home_team = Home Team
away_team = Away Team
home_team_goal = Number of Home Team Goals
away_team_goal = Number of Away Team Goals
home_player_(i) = i-th player of the home team
away_player_(i) = i-th player of the away team
past_perf_home = Past performance of home team
past_perf_away = Past performance of away team
home_player_avg = Average of home players' rating
away_player_avg = Average of Away players' rating
result = Winner of the match

We used the below algorithm for the feature selection and classified data based on different classifiers.

```
Initialize Home Team
Initialize Away Team
Initialize k=10

def past_perf ():
For i in the range (1, k):
    Compute the Average of number of Home Team Goals for past k matches
    Compute the average of number of Away Team Goals for past k matches

def player_rating_avg():
    For i in the range (1,11):
        home_player_avg += Rating of home_player_i
        return (home_player_avg/11)
    For i in the range (1,11):
        away_player_avg += Rating of away_player_i
        return (away_player_avg/11)

def result ():
    if (home_team_goal > away_team_goal):
        result = 2 // Home Team Won
    elif(home_team_goal < away_team_goal):
        result = 1 // Away Team Won
    elif(home_team_goal = away_team_goal):
        result = 0 // Match ended in draw

Update the dataset with past_perf_home,past_perf_away, home_player_avg,
away_player_avg,result
Split the Attributes and Class variables
Run the Classifer on Training set
Compute the Accuracy on Test set
```

We have applied the below classifiers to the computed dataset:

1.) **Support Vector Machines:** The reason for choosing the above SVM is because its defined by a convex optimization problem (no local minima). We have used RBF kernel to solve the optimization problem.

The **RBF kernel** on two samples x and x', represented as feature vectors in some input space, is defined as:

$$K(x,x') = \exp((- \|x-x'\|^2) / (2 * \sigma^2))$$

Where $\|x-x'\|$ is the Euclidean distance of the two feature vectors.

2.) **Logistic Regression:** The reason for choosing logistic regression is that don't have to be normally distributed, or have equal variance in each group. The maximum-likelihood ratio is used to determine the statistical significance of the variables.

3.) **K-Nearest Neighbor:** The reason for choosing KNN is because the dataset used is very large and since KNN is robust to noisy data, we will be able to achieve a better accuracy with KNN. We calculated the accuracy on the test set for different values of k.

As a part of the conclusion, we will compare the accuracy of the different classifiers.

3 Experimental Evaluation

3.1 Methodology

The past performance of the home team and the away team is not available in the dataset. We have computed the past performance of the team by calculating the average of the number of goals scored by the team in past k-matches. (We have performed the calculations for the value, k=10)

The past_perf_home and past_perf_away is computed as below:

$$\text{past\_perf\_home} = \frac{\text{Number of goals scored by the home team in past k matches}}{k}$$

$$\text{past\_perf\_away} = \frac{\text{Number of goals scored by the away team in past k matches}}{k}$$

The home_player_avg and away_player_avg is computed as below:

$$\text{home\_player\_avg} = \frac{\sum \text{player\_rating of all players in home\_team}}{11}$$

$$\text{away\_player\_avg} = \frac{\sum \text{player\_rating of all players in home\_team}}{11}$$

The result column in the dataset is computed by comparing the number of goals scored by the home team and away team based on the below conditions:

Condition I: if (home_team_goal > away_team_goal)
result = 2; // Home Team Wins

Condition II: if (home_team_goal < away_team_goal)
result = 0; // Away Team Wins

Condition III: if (home_team_goal = away_team_goal)
result = 1; // The match ended in a draw

The above computation provides us with the complete dataset which we will split the data as training and testing data to test the accuracy of the algorithm chosen to predict the outcome of the match.

Example:

Table 1: Example Dataset

home_team	away_team	past_perf_home	past_perf_away	home_player_avg	away_player_avg	result
MAL	NUM	0.2	1.666666667	66.40918917	55.09536234	Win
HUE	BIL	0.333333333	1	58.69522083	67.89222291	Draw
MAL	VAL	0.5	1.166666667	64.99589809	63.45186542	Lose

Based on the above metrics we apply different classifier and test their accuracy with respect to the data in the test set.

3.2 Results

Results of Support Vectors Machine:
Percent: 62.5
Results of Naive Bayes:
Percent: 57.49999999999999
Results of Logistic Regression:
Percent: 62.5
Results of 70 Nearest Neighbor:
Percent: 60.83333333
Results of Random Forest:
Percent: 56.666666666666664

The results for KNN based on different values of k are given below:

Table 2: KNN Results

	Accuracy (Percentage)
K=10	53.33333333
K=30	57.5
K=35	58.33333333
K=50	56.66666667
K=70	60.83333333

The results of SVM based on different kernels:

Table 3: SVM Results for different Kernels

	Accuracy (Percentage)
RBF kernel	62.5
Linear kernel	60.83333333
Polynomial kernel	60.83333333

4. Conclusion & Discussion

We train and test all our models from the **Spain LIGA BBVA** league data available in the dataset. As reported in the results above, SVM with Radial Basis Function Kernel and Logistic Regression gives our best accuracy of 62.5% in comparison to the accuracy of expert pundits like Mark Lawrenson of BBC which is 52.6% [2].

Although, it is intuitive to use KNN with 30 nearest neighbors and expect the best accuracy, the accuracy seems to be greater when k=70. Also, SVM and Logistic Regression outperform KNN by a margin of 2%. We can conclude that the data seems to be marginally classified in terms of player rating and team rating rather than our intuition that it should be classifiable by teams. Hence, we think that it explains the results above.

5. References

[1] Sierra, Adrian, et al. "Football futures." *URL http://cs229. stanford. edu/proj2011/SierraFoscoFierro-FootballFutures. Pdf*

[2] Lawrenson, Mark (2013) , https://www.pinnacle.com/en/betting-articles/soccer/mark-lawrenson-vs-pinnacle-sports

[3] https://en.wikipedia.org/wiki/Association_football