Soccer Matches Prediction

Milad Abbaszadeh[1], **Malek Trabelsi**[1]

[1]Technische Universität Berlin

Milad.abbaszadeh94@gmail.com, malek.trabelsi@student.ecp.fr

Abstract. *The aim of this project is to predict the outcome of recent soccer matches (in this paper, we predict the 2018 matches), based on data from previous years, that is used to train. In order to achieve our goal, we tried several approaches. In this paper, we present four of the approaches that we tried. These four approaches differ by the structure of the dataset (the first dataset focuses on the number of features, while the second is more about the number of observations), and/or the modelling of the task (as a classification or as a regression).*

Keywords*: World Cup, Soccer Matches, Data Integration, Modelling*

1. Introduction

Predicting the outcome of soccer matches has been subject to a lot of research recently. To mention but a few, the paper [1] that made use of a Bayesian probabilistic approach to predict the winner country of the FIFA World Cup iteratively, by predicting for each round the probability of each country to win, then qualifying the countries that have the highest probabilities for the next round.

Our goal in this paper is quite different, and our approach too, since we won't be focusing on the winner of the World Cup, but on predicting the outcome of each match, taken as a standalone. Results can be applied iteratively later on if need be, to predict the winner of a given tournament.

2. Problem Statement

Predicting the outcome of Soccer Matches has turned out to be a challenging task, as various factors are involved in the game and all of them cannot be scoped out in a single dataset, some of them may even be hidden, and influence indirectly the outcome of a match.

In this report, we propose a method to predict the winner country for each single match in our datasets, that took place in 2018. We modelled the task as a 3-class classification problem (winner, loser and draw) and as a regression, and in both cases, trained our model on matches from previous years.

3. Challenges

3.1. Modelling

In the beginning, we started modelling the problem as a classical Machine Learning problem: each match was modelled as one row in the dataset, with several features related to the match and/or to the playing teams. This classical formulation was, however, not accurate, and we will explain the reasons for that and give some solutions in this section. We will also talk about some of the challenges that we faced when collecting data.

3.1.1. Duplicating the Data

In table 1, we can see two rows modelling two matches, ordered alphabetically by the names of the countries. The problem here is that Germany appears in the first match as team1, but appears again in the second match as team2. Since these are two different columns, the Machine Learning model would be incapable of learning that Germany that played as team1 in match1 is the same as Germany that played as team2 in match2, which would mistaken our results. ⇒ One solution to solve the problem is to duplicate our data:

Team1	Team2	Team1 Goals	Team2 Goals
Germany	Italy	2	0
France	Germany	1	3

Table 1. Simple Formulation of the Dataset

duplicating means that we reproduce the same dataset, but we invert the order of the two countries (team1 - team2 becomes team2 - team1), as well as all the features that are country-related (for example, the number of goals scored in the past year by a given team, the surface of a country, ... etc). In this way, the order of appearance of a country won't matter anymore. Table 2 presents this solution on the data of table 1.

3.1.2. Removing the Names

After having duplicated the data, we decided to drop the columns containing the names of the teams, as well as the column containing the year. This can be understood as follows: we want to learn the strength of the playing teams. The names and the year don't matter anymore, but it's the performance according to some criteria that matters. For instance, the model could be able to learn that the performance of Germany in 2014 is so similar to that of Uruguay in 1930.

3.2. Data collection

We collected our data from several sources: we crawled data from the official FIFA website [2] and collected data on friendly matches [3] using Data Miner, an extension that can be added to a browser and enables us to crawl data. We also looked for features such as the population, mainly on the Our World In Data website [4], but we also looked for data in other websites, especially when we had missing values for some countries / years.

3.3. Data integration

Since we had to integrate data from several sources, in the schema mapping and matching phase, we faced a representation problem in the team's name field. On one hand, there

Team1	Team2	Team1 Goals	Team2 Goals
Germany	Italy	2	0
Italy	Germany	0	2
France	Germany	1	3
Germany	France	3	1

Table 2. Duplicating the Observations

were quite close team names, referring to different countries, such as Dominica vs the Dominican Republic, or Democratic Republic of Congo vs Republic of Congo. On the other hand, there were different names corresponding to the same country, either because of some spelling issues (for example, Cte d'Ivoire vs Cote d'Ivoire), or because of the existence of different names to one country: Republic of Korea and South Korea, or because of the use of abbreviations: United States of America vs USA. We used the names from the population dataset (downloaded from [4]) as a reference, and changed all other names that are different in other datasets, into the correct spelling.

4. Datasets

In our experiments, we will be using 2 datasets: the first one is a **wide** data set and second one is **long**. The main emphasis in wide data set is on the number of the features, but the number of data points isn't of big importance. This data set only consists of the matches from 1993 until 2018.
On the contrary, the long data set focuses more on the number of observations. Thus, we have collected matches from many international tournaments (Friendly, World Cup, African Cup of Nations, World Cup qualifications ...).In the following subsections we will describe each of the datasets in more detail.

4.1. Long Dataset

This dataset contains matches from 1872 to 2018. It has 38900 observations (before duplication), from different international soccer tournaments, and 30 features. Our features here are the number of matches played by each country in the year preceding the concerned match, the number and the ratio of goals scored by the countries, the number and the ratio of wins, losses and draws in all matches played in the previous year, and the number and the ratio of wins, losses and draws of the matches played against the opponent in the concerned observation. These features were computed using the scores of the matches, and a sliding window of width 1 year.

4.2. Wide Dataset

Since finding some of the features for very old matches was quite difficult (for example, finding the population of a country in the eighties), and sometimes impossible (finding the FIFA ranks and FIFA scores for years before 1993, because they were only introduced by the FIFA in 1993), we decided to construct a second dataset, where we add features, but limit the number of observations. So this second dataset has 9071 observations (before duplication), and 40 features. The features here are the ones we had in the long dataset, computed in the same way, and we added new features: FIFA Rank, FIFA Points, Population, Surface and Density of the countries. We also added some other features such as the GDP (a measure of the economic growth of a country), the continent of the country ... but finally decided to drop them because there were many missing values, and adding them decreased our accuracy.

5. Experiments & Results

5.1. First Serie: Classification

In the first place, we formulated the problem as a **classification** problem. This means that our target vector is a discrete one, and each observation can either have 0, 1 or 2

Classifier	Accuracy Score
Dummy Classifier	36.82%
Random Forest	48.17%
Bernoulli NB	46.92%
Extra Trees	41.58%
KNN	39%
MLP	44.66%
Nearest Centroid	48.61%
Ridge Classifier	48.76%
SVC	49.04%

Table 3. Accuracy Scores for the Classification Task (Wide Dataset)

Classifier	Accuracy Score
Dummy Classifier	36.27%
Random Forest	36.6%
Bernoulli NB	41.5%
Extra Trees	38.56%
KNN	34.64%
MLP	45.42%
Nearest Centroid	39.22%
Ridge Classifier	41.18%
SVC	42.48%

Table 4. Accuracy Scores for the Classification Task (Long Dataset)

as outputs. 0 stands for a draw match, 1 means that team 1 won the game and 2 means that team 2 won the game. The order of appearance of the two playing countries doesn't matter, since we duplicate the tuples prior to performing the experiment.

So we perform this experiment on the two datasets presented earlier in this paper. We scaled our data, then used multiple classifiers to test the performance of our datasets. The reference is a dummy classifier that outputs all the time the most frequent class in the labels' vector. And the other classifiers are Random Forests, Bernoulli Naive Bayes, Extra Trees, k-Nearest Neighbours, Multi-Layer Perceptron, Nearest Centroid, Ridge Classifier and Support Vector Classifier. Tables 3 and 4 show the accuracies that we obtained for respectively the wide and the long datasets.

5.2. Second Serie: Regression

In this experiment, we treat the problem as a **regression** task. This means that our target vector is continuous, and no longer discrete like it was in the previous experiment. The target is relative to the first team in the observation tuple, and since we duplicate our data, like we did before, all of the teams are treated in this Machine Learning approach.

The target value is calculated using the formula:

$$target = \frac{\text{number of goals scored by team 1}}{\text{number of goals scored by team 1} + \text{number of goals scored by team 2}} \tag{1}$$

Unlike the classification choice to formulate the problem, this formulation takes into consideration the strength of a team in a match. For example, let's say we have two matches:

Regressor	Accuracy Score	Accuracy (threshold = 0.03)	2-class Accuracy
Dummy Regressor	26.35%	26.35%	57.85%
MLP Regressor	45.82%	46.41%	62.62%
Gradient Boosting	48.76%	**49.34**%	65.20%
Random Forests	45.24%	42.17%	63.22%
AdaBoost	48.02%	43.19%	54.67%
Bagging Regressor	44.22%	45.39%	62.82%
Transformed Target	**48.90**%	48.02%	**66.60**%

Table 5. Accuracy Scores for the Regression Task (Wide Dataset)

Regressor	Accuracy Score	Accuracy Scores (threshold = 0.05)
Dummy Regressor	27.45%	27.45%
MLP Regressor	**43.79**%	**44.44**%
Gradient Boosting	41.83%	43.14%
Random Forests	37.91%	34.64%
AdaBoost	39.87%	39.87%
Bagging Regressor	39.22%	37.25%
Transformed Target Regressor	41.18%	41.38%

Table 6. Accuracy Scores for the Regression Task (Long Dataset)

France - Germany that ended with the score: 2 - 1
and France - Rwanda that ended with the following score: 6 - 1
In the first match, France will have a label score of $\frac{2}{2+1} = 0.67$ and Germany will have $\frac{1}{2+1} = 0.33$.
However, in the second match, France will have a score of $\frac{6}{6+1} = 0.86$, while Rwanda will have a score of $\frac{1}{6+1} = 0.14$.
$\rightarrow$ We can see that the score is the highest for France in the second match, which reflects the fact that it won with a large difference then. And this may increase the model's ability to learn, thus boosting our accuracy, because we will somehow be trying to predict a teams strength, then we will try to compare the outputs for both teams in one match, transform our regression score into a discrete value (0, 1 or 2 as we explained in the previous section), then compute the overall accuracy.
To predict this country strength, we used many regressors, like we did for the classification task. Our reference is a dummy regressor that always outputs the mean of the training set. Other used regressors are Multi-Layer Perceptron Regressor, Gradient Boosting, Random Forests Regressor, AdaBoost, Bagging Regressor and Transformed Target Regressor.
We scale our data, then do the predictions as explained, for the wide and the long datasets. The results that we get are presented in tables 5 and 6, for the wide and long dataset, respectively. When inspecting the results, we realized that the frequency of predicting a draw was so low, compared to what we should have predicted. And this is due to the fact that while converting the continuous predicted target into a discrete one (having as possible values 0, 1 or 2), we only considered a draw match when the two predicted country strengths of the playing teams were exactly equal to each other. And this is not frequent in a regression task, since the range of possible values that our prediction can take is infinite. To remedy this problem, we introduced a **threshold**, and considered that when the

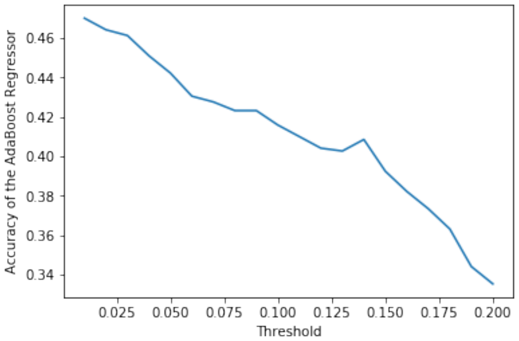


Figure 1. Influence of the Threshold on the Accuracy for AdaBoost Regressor

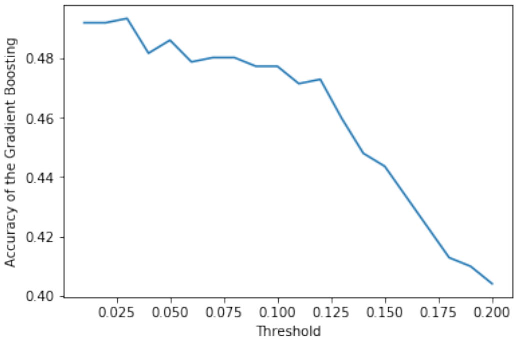


Figure 2. Influence of the Threshold on the Accuracy for Gradient Boosting Regressor

absolute difference between the two predicted regression values is less than this threshold, then the outcome of the match is considered as a draw.

This solution helped increase the accuracy measure for some of the regressors (see results in the third column of the tables 5 and 6, for respectively the wide and the long datasets), but for some of them, the accuracy dropped a little bit, and we could explain this by the fact that the threshold depends on the used algorithm itself, but we used the same threshold for all of them. For future work, tuning the threshold for each of the regressors individually could help boost the performance by a few percents.

For instance, we can see on figures 1 and 2 that for the Adaboost regressor, the best threshold is rather small (less than 0.01), but for the Gradient Boosting regressor, it's around 0.025.

5.3. Inspecting our Output

5.3.1. Draw Matches are Difficult to Predict

Since our scores were still not that high, we thought that maybe the prediction of draw was decreasing our accuracy, because it's the most difficult thing to predict, since usually the two countries have rather the same strength, which makes the prediction of the winning country harder, even for a human being. To make sure that our hypothesis is correct, we deleted all the observations corresponding to draw from our train set, and we transformed the problem into a two-class classification. We display in figures 3 and 4 the confusion matrices, after having applied Support Vector Classifier on the 2-class classification problem. We can see that as we might think, removing the draw matches helps improve the accuracy, from around 45% to up to 66%. This was also done using the regression formulation discussed in section 5.2. The results obtained are presented in the 2-class Accuracy column of table 5. The conclusion is the same.

5.3.2. Teams' Strengths

In order to get deeper insights into the problems that our modelling is facing, we also tried to verify the hypothesis that when two teams that are relatively close to each other in terms of performance, the accuracy of predicting is not that good. However, when we only

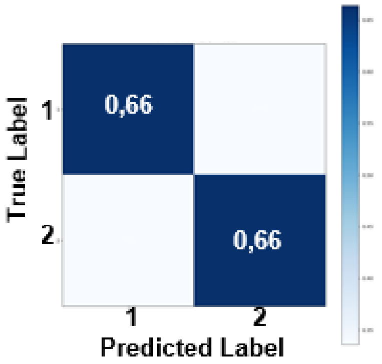


Figure 3. Confusion Matrix for a 2-class Classification, on the Wide Dataset

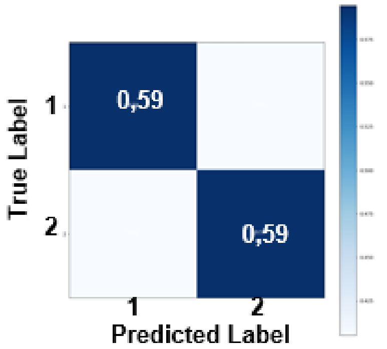


Figure 4. Confusion Matrix for a 2-class Classification, on the Long Dataset

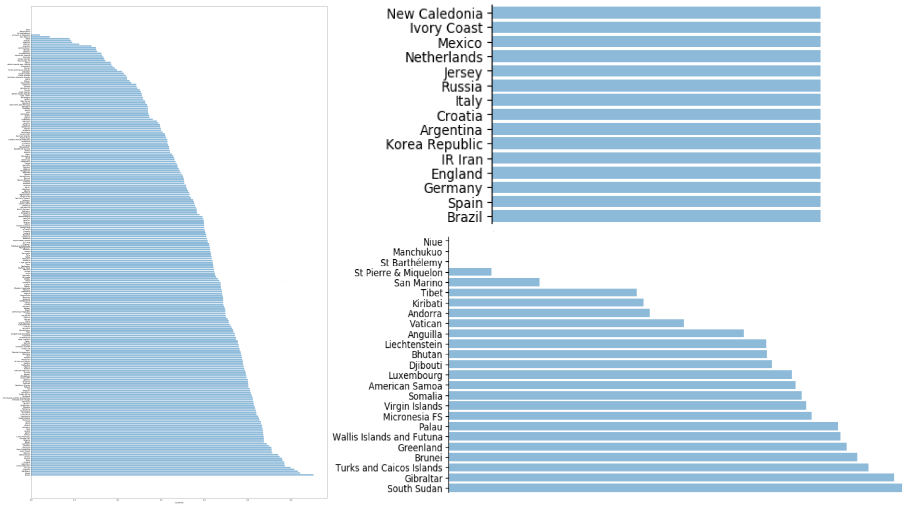
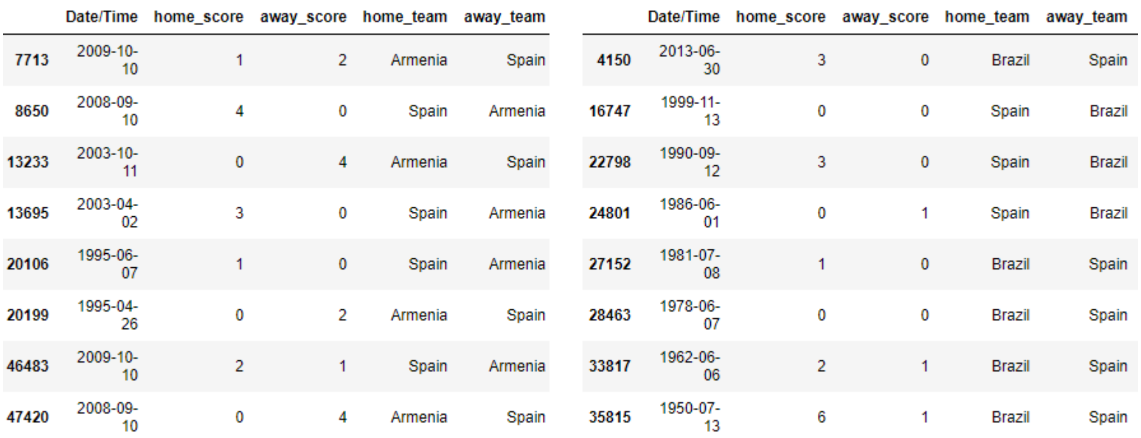


Figure 5. On the left: Histogram of all countries ; On the top right: Zoom into the strongest countries ; On the bottom right: Zoom into the weakest countries

consider the matches where a strong team is playing against a weak one, the prediction is almost always correct.

So to verify the hypothesis, we started by plotting a histogram, where the x axis is the strength of the country, computed on all matches as explained in section 5.2, and then we average over the matches played by each of all the countries present in the dataset, to get the final score, and the y axis corresponds to the countries. We can see the plot on figure 5 So to verify our hypothesis, we set a threshold. Teams having a score greater than this threshold are considered to be strong, others are weak. This approach wasn't really good, because in case countries that have strengths close to the threshold (one that has a strength slightly greater than the threshold and the other slightly smaller) play against each other, we end up having the same problem discussed earlier in this section. So we needed to do some separation: we only considered countries that are so strong against those that are so weak, discarding the intermediate ones. The intersection is not really important, because strong countries are less likely to play against weak ones. But we could verify for example, that when Spain (strong team) plays against Armenia (weak team), Spain was almost always the winner, however, when Spain plays against Brazil (2 strong teams),



	Date/Time	home_score	away_score	home_team	away_team
7713	2009-10-10	1	2	Armenia	Spain
8650	2008-09-10	4	0	Spain	Armenia
13233	2003-10-11	0	4	Armenia	Spain
13695	2003-04-02	3	0	Spain	Armenia
20106	1995-06-07	1	0	Spain	Armenia
20199	1995-04-26	0	2	Armenia	Spain
46483	2009-10-10	2	1	Spain	Armenia
47420	2008-09-10	0	4	Armenia	Spain

	Date/Time	home_score	away_score	home_team	away_team
4150	2013-06-30	3	0	Brazil	Spain
16747	1999-11-13	0	0	Spain	Brazil
22798	1990-09-12	3	0	Spain	Brazil
24801	1986-06-01	0	1	Spain	Brazil
27152	1981-07-08	1	0	Brazil	Spain
28463	1978-06-07	0	0	Brazil	Spain
33817	1962-06-06	2	1	Brazil	Spain
35815	1950-07-13	6	1	Brazil	Spain

Figure 6. Right: Armenia VS Spain ; Left: Brazil VS Spain

predicting the outcome of the match is harder (See figure 6)

5.3.3. Features Importance

We also tried to see which features were the most important for our prediction, and we found out that the most important ones, regardless of the algorithm that we were using, were: the number and the ratio of goals scored in the past for the two teams, and the number of matches played in the past year, and also the rank, points, population and density from the wide dataset.

6. Conclusion and Future Work

As we could see, throughout this project, we tried to explore many methods, varying the datasets, the modelling approaches, as well as the algorithms. The results showed that the best accuracies were obtained for a wide dataset, using a regression formulation. So as future work, we could try making the dataset bigger by thinking of more features and adding them to it. Also, in order to solve the draw matches prediction problem, we could use some online learning, in order to ask experts for their expectations about the outcome of a match that is difficult to predict for our algorithm. Also, as we mentioned earlier, training the threshold in the regression formulation, for the different algorithm could help increase the accuracy. One last thing to mention, the parameters' tuning for the different algorithm wasn't really helpful to increase the accuracy, but maybe after having added features, we could retry it, to boost the accuracy.

References

[1] Suzuki, A. K., Salasar, L. E. B., Leite, J. G., Louzada-Neto, F. (2010). A Bayesian approach for predicting match outcomes: the 2006 (Association) Football World Cup. Journal of the Operational Research Society, 61(10), 1530-1539.

[2] https://www.fifa.com/.

[3] https://www.flashscore.com/football/world/friendly-international/archive/.

[4] https://ourworldindata.org/.