Project Final Report

La Liga games predictions

Nour Jamoussi, Marco Klepatzky

20-01-2022

Abstract

Different supervised methods to predict the result of matches of La Liga Santander. The aim is to find methods giving the best accuracy. We used the Data of the 20-21 season. It was collected by scraping websites and analysis were performed using python libraries such as pandas, and sklearn.

1 Introduction

The project aim is to have an accuracy as good as possible on predicting football matches results. Our choice of the Spanish league is based on the recent regulations that made the teams more balanced because of the limited budget they can spend. The input to our algorithms is a collection of features like the Team, the opponent Team, average scores of the teams (goals, shots, blocks, FIFA players' scores, etc.) and the "home/away" feature. We then used different models; KNN, Random Forest, SVM and Logistic Regression, to output the predicted result (0:Loss - 1:Win - 2:Draw). Then, we did feature selection on the mentioned models (RFE - Lasso regularization) to minimize the number of features and achieve better accuracy.

2 Related work

After some prior investigation, we found several studies and reports on guessing the odds in various sports. Given the amount of money in circulation and the websites offering betting services, many clients try to maximize their profits by investing on the correct team, with the risk of losing their bets. An extensive analysis (Kempa, 2020) on the brazilian football league [1], depicts an interesting approach on how to treat, manipulate and process this sport data, the author explains the non-linearity of the data, and makes efforts to find the most relevant features that can determine the outcome of the match. A similar approach (Yezus, 2014) [2] on the data scrapping for soccer predictions was used in this paper. The study differs in considering a very limited subset of features as the author pre-selects them based on three relevant questions.
Moreover, a paper (Warner, 2010) [3] adapts the analysis done to a certain type of spread used by bookmakers so as to build a bet-oriented classifier. In this study, the author sticks to using input data for only two teams of the NFL league and considers exogenous features, such as the team's home city temperature.
In the analysis, the author (Hamadani, 2006) presents a win estimator for the NFL league [4]. He makes an effort on classifying the feature by its nature, demonstrating that some models achieve better results depending on the distribution and type of data they are used with.
Lastly, the paper (Tax, Joustra, 2015) [5] describes a public data based match prediction system for the Dutch Eredivisie. The authors used principal component analysis to reduce the dimensions of the model in conjunction with Naive Bayes and Multilayer peceptron. They managed to perform a reduction from 51 features to 3 components.

3 Dataset and features

The most important part of the project is the Dataset used and how we manipulate it. We looked for Datasets on Kaggle, but no one was good enough for our project so we made the choice to prepare it by our own. First, we scrapped this page [6] to get the links of all the reports of matches in season 20-21 and we did the same process for the matches of the season 21-22. Lastly, for each encounter, we extracted the match report. An example of a report is shown in this page [7].
The data shows scores for each player of the team in the match. In addition to the scores extracted from the match, we added 4 FIFA scores [8] per player: age, potential, overall rating and his market value. But since we would take the whole team in consideration, we grouped it by match, team, date, the stadium and the crowd. Then we summed the players' scores of the match and did an average on the FIFA scores. We also added a column to mention if the team is playing home or away (1 if Home and 0 if Away), and a column for the result (the label). We assigned 0 for losers, 1 for winners and 2 if it is a draw. Since the models we used can't take into consideration string features, some encoding was also done. At this level, we had all the scores for the match played in our input data. Our goal, however, is to predict the result of the match before

the match. Thus, we assumed that we could take into consideration the average of the scores of the whole season (before the match) and the average of the three last matches. To emphasize the contrast between playing teams, we computed a substraction of scores as shown in Table 1.

Table 1: Example of the data used

Table 2: Previous Data

Match	Team	AvgScore
Alaves vs Sevilla	Alaves	0.5
Alaves vs Sevilla	Sevilla	0.8

Table 3: Processed Data

Match	Team	AvgScore
Alaves vs Sevilla	1	0.5-0.8=-0.3
Alaves vs Sevilla	2	0.8-0.5=0.3

The final dataset structure and the features used are shown in Figure 1 and Figure 2. The data is composed by 41 columns, including 37 features and by 734 non null samples. We split it into a training set(80%), a validation set (10%), and a testing set (10%). We made the choice of (80,10,10) because the training data is not huge.

Match	Date	Team	Opponent_Team
Stadium	Crowd	Home/Away	Results
last_3_avrg_Gls	avrg_Gls	last_3_avrg_Ast	avrg_Ast
last_3_avrg_PK	avrg_PK	last_3_avrg_PKatt	avrg_PKatt
last_3_avrg_Sh	avrg_Sh	last_3_avrg_SoT	avrg_SoT
last_3_avrg_CrdY	avrg_CrdY	last_3_avrg_CrdR	avrg_CrdR
last_3_avrg_Touche	avrg_Touches	last_3_avrg_Press	avrg_Press
last_3_avrg_Tkl	avrg_Tkl	last_3_avrg_Int	avrg_Int
last_3_avrg_Blocks	avrg_Blocks	last_3_avrg_Potentia	avrg_Potential
last_3_avrg_Overall	avrg_Overall Rating	last_3_avrg_Age_y	avrg_Age_y
last_3_avrg_Value	avrg_Value		

Figure 1: Dataset structure

Team	Opponent_Team	Home/Away	
last_3_avrg_Gls	avrg_Gls	last_3_avrg_Ast	avrg_Ast
last_3_avrg_PK	avrg_PK	last_3_avrg_PKatt	avrg_PKatt
last_3_avrg_Sh	avrg_Sh	last_3_avrg_SoT	avrg_SoT
last_3_avrg_CrdY	avrg_CrdY	last_3_avrg_CrdR	avrg_CrdR
last_3_avrg_Touche	avrg_Touches	last_3_avrg_Press	avrg_Press
last_3_avrg_Tkl	avrg_Tkl	last_3_avrg_Int	avrg_Int
last_3_avrg_Blocks	avrg_Blocks	last_3_avrg_Potentia	avrg_Potential
last_3_avrg_Overall	avrg_Overall Rating	last_3_avrg_Age_y	avrg_Age_y
last_3_avrg_Value	avrg_Value		

Figure 2: Features

As shown in Figure 2, we haven't used the crowd feature since the matches of the 20-21 season were held without an audience due to COVID sanitary restrictions. The data has been improved several times after computing some tests to reach its final version. And for accuracy purposes, we skipped using the data of the current season (21-22), as it mismatches with the previous year samples in several points, and this ultimately affected the prediction accuracy.

4 Methods

Our method of working was collecting, cleaning and processing data, then testing different models to see if we needed to consider more features or to process it in a different way. We tried to curate the data until we got similar accuracy values compared with previous works on the subject. The classification models used to test are KNN, Random Forest, SVM, Logistic Regression and Logistic Regression with Lasso regularization. Then we did Feature engineering with recursive feature elimination (RFE) and kept track of the normalization and standardization on the inputs in order to test the models afterwards. A short description of each method used is shown below.

Logistic regression: is a statistical method for computing the probability of an event occurrence.

Support Vector Machines (SVM): SVM constructs a hyperplane in multidimensional space to separate different classes.

K-Nearest Neighbors (KNN): This algorithm is a non-parametric model making the classification depending on the classes of the k-nearest points.

Random Forest (RF): Creates several decision trees and groups them into one "Forest" of trees. The model results perform a majority voting among several estimators and can end up being better than the overall of the previous estimators used.

Recursive Feature Elimination (RFE): works by searching for a subset of features by starting with all features in the training dataset and successfully removing features until the desired number remains. This is achieved by fitting the given machine learning algorithm used in the core of the model, ranking features by importance, discarding the least important features, and re-fitting the model.

Lasso regularization (L1) : is one common type of regularization, which involves penalizing the sum of absolute values (1-norms) of regression coefficients. Thanks to its ability to shrink some of the coefficients to zero, it represents an embedded method to select features.

Standardization (Z-score) and Normalization (MinMax): Both form part of the feature scaling process, they help ML algorithms to train and converge faster, avoiding bias. The first handles well the outliers in data, whereas the second is useful when we don't know the data distribution.

5 Experiments, results and discussion

To evaluate our models, we have used the accuracy score [9] and the confusion matrix [10] of sklearn library.

5.1 General experiments before feature engineering

We ran random forests, KNN, logistic regression and SVM on our data with an input of 37 features. The results are shown in Table 4 and Figures 3, 4, 5 and 6.

Table 4: First Experiment

Method	Accuracy
Random forests	0.4593
KNN	0.4496
Logistic regression	0.4689
SVM	0.4625

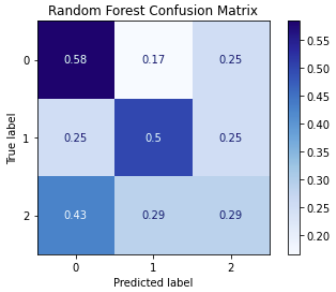

Figure 3: Confusion matrix of RF

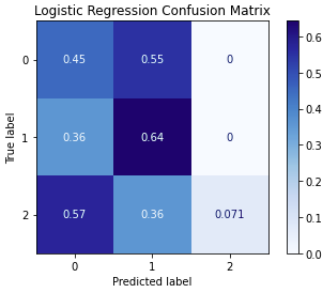

Figure 4: Confusion matrix of LR

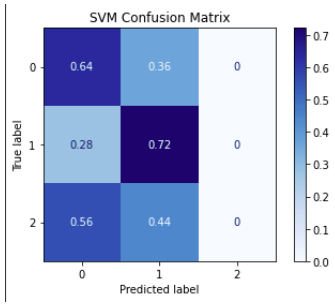

Figure 5: Confusion matrix of SVM

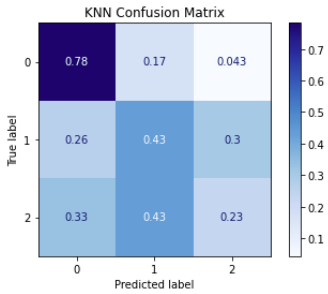

Figure 6: Confusion matrix of KNN

5.2 General experiments after feature engineering

We ran RFE on both RF and LR models, then we picked the best 8-6 features selected for each classifier.
Then we tested the models with the selected features and we tested the logistic regression with Lasso as well. The results are shown in Table 5.

Table 5: Second Experiment

Method	Accuracy
Random Forest after RFE	0.4736
Logistic regression with Lasso regularization	0.5263
Logistic regression after RFE	0.4682
Logistic regression with Lasso regularization after RFE	0.5
SVM after standard scaling	0.5

Figure 7: RFE on logistic regression

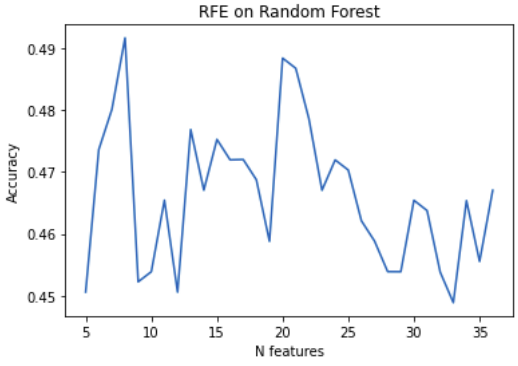

Figure 8: RFE on random forests

5.3 Some experiments on Real Madrid matches

5.3.1 General accuracy

We trained and tested models only on Real Madrid data. The accuracy of predicting Real Madrid matches is presented in Table 6.

Table 6: Third Experiment

Method	Accuracy
Random forests after RFE	0.5459
Logistic regression with Lasso regularization	0.75
Logistic regression with Lasso regularization after RFE	0.75

5.3.2 Accuracy of predicting 2 specific matches

We selected a bottom team like Eibar and the Liga winner : Atletico Madrid. We wanted to know the accuracy of the prediction of their matches against Real Madrid. For each team above we trained the model with its data and the Real's data. Then we tested on a Match between them. The result is shown in Table 7.

Table 7: Fourth Experiment

Table 8: Real Madrid vs Atletico Madrid

Method	Accuracy
Random forests	1
Logistic regression	0.5

Table 9: Real Madrid vs Eibar

Method	Accuracy
Random forests	1
Logistic regression	1

5.4 Discussion

Feature Engineering has improved the accuracy of SVM and Random Forest. It minimized the number of features from 37 to less than 10 which has decreased a bit the time spent on the execution. Logistic regression shows a similar accuracy before features selection and after. Logistic regression with Lasso regularization has shown the best result with 52.63% of accuracy on general data. When we select the samples related to Real Madrid and proceed to train and test the models, we notice a very good accuracy, especially for the Logistic Regression with Lasso (75%). The prediction's accuracy even reaches 1.0 when models learn the statistics of a leading team and a modest one and do the test on a match between them. But whenever we test the prediction between the first two leading teams, the accuracy of the logistic regression drops to 50% which is completely understandable.
Firstly, overfitting was prevented by constantly checking training/testing accuracies. Furthermore, we visualized an increasingly optimization on the accuracy values for the models analyzed by implementing feature scaling and feature selection. However, each model behaves differently in predicting the outcomes even if they feature similar accuracies. A first glimpse in the confusion matrix can provide more granular details on the predictions strengths and weaknesses. For example, the probability of ending a match with a draw is practically non-existent for models such as LR and SVM, in contrary to RF and KNN, where they have more margin of occurrence. And even though KNN showed the worst overall accuracy, it was the best model to predict a loss.

6 Conclusion and future work

To summarize, we gave as input data with Spanish League matches scores and Fifa scores to 4 models. The best general accuracy was 52.63% given by the Logistic Regression with Lasso Regularization. Thus, we can state that we achieved the initial approach of the problem, by developing an algorithm that could determine with a decent level of accuracy the outcome of a football match between two teams. The use of the methods described achieved better values than the 33.3% blind random pick or the 46% trend that the "Home Team" always wins. We think that the feature elimination done by the Lasso regularization was a key part on determining the best model to approach this kind of data.
The link for our repository can be found here [11].
Some future improvements for this work would include retrieving a larger and more curated database, implementing custom parameter tuning for the models described, and pursuing progressive increments on the overall accuracy. We may also focus on the accuracy for each team or for specific matches as we did for Real Madrid. Another idea is to code an application that benchmarks multiple instances of models on the same input data. The user would get all the relevant information on the models benefits and weaknesses. By doing so, he would visualize the side by side results with an assistance on the decision making.

7 Contributions

Nour Jamoussi : Data Collection-Cleaning-Processing and Running Experiments.
Marco Klepatzky : Data Cleaning and Feature Engineering.

References

[1] Kempa, M. (2020). Machine Learning Algorithms for Football Predictions.
https://medium.com/@matheuskempa?p=51b7d4ea0bc8

[2] Yezus, A. (2014). Predicting outcome of soccer matches using machine learning.
https://www.math.spbu.ru/SD_AIS/documents/2014-12-341/2014-12-tw-15.pdf

[3] Warner, J. (2010). Predicting Margin of Victory in NFL Games: Machine Learning vs. the Las Vegas Line.
http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.402.8415&rep=rep1&type=pdf

[4] Hamadani, B. (2006). Predicting the outcome of NFL games using machine learning.
http://cs229.stanford.edu/proj2006/BabakHamadani-PredictingNFLGames.pdf

[5] Tax, N., Joustra, Y. (2015). Predicting The Dutch Football Competition Using Public Data: A Machine Learning Approach.
https://www.academia.edu/16272629/Predicting_The_Dutch_Football_Competition_Using_Public_Data_A_Machine_Learning_Approach?pop_sutd=false

[6] The Web Page used to get the links of each match report (season 20-21) :
https://fbref.com/en/comps/12/10731/schedule/2020-2021-La-Liga-Scores-and-Fixtures

[7] Example of a match report :
https://fbref.com/en/matches/9613fb68/Alaves-Real-Madrid-August-14-2021-La-Liga

[8] A Github repository containing a dataset of FIFA 2021 Player ratings scraped from web.
https://github.com/othmbela/fifa-21-web-scraping

[9] Accuracy score of sklearn library : https://scikit-learn.org/stable/modules/generated/sklearn.metrics.accuracy_score.html

[10] Confusion Matrix of sklearn library : https://scikit-learn.org/stable/modules/generated/sklearn.metrics.confusion_matrix.html

[11] Github Repository : https://github.com/NourJamoussi/LaLigaPrediction