Predicting football scores using machine learning techniques

Josip Hucaljuk, Alen Rakipović
University of Zagreb, Faculty of Electrical Engineering and Computing
Unska 3, 10000 Zagreb, Croatia
josip.hucaljuk@fer.hr, alen.rakipovic@fer.hr

Abstract - Predicting the results of football matches poses an interesting challenge due to the fact that the sport is so popular and widespread. However, predicting the outcomes is also a difficult problem because of the number of factors which must be taken into account that cannot be quantitatively valued or modeled. As part of this work, a software solution has been developed in order to try and solve this problem. During the development of the system, a number of tests have been carried out in order to determine the optimal combination of features and classifiers. The results of the presented system show a satisfactory capability of prediction which is superior to the one of the reference method (most likely a priori outcome).

I. INTRODUCTION

Sporting events have always been very interesting to a wide range of population. One of the most popular sports is football and the Champions League is the toughest and the most prestigious club football competition in the world.

There are three possible outcomes of a match: home win, draw and away win. Due to its popularity and the small number of possible outcomes of games, predicting results is a very interesting and seemingly simple challenge. However, it is very difficult to predict the final outcome because the way the team plays on a particular day depends on many factors, such as the current form, the last team meetings, rivalries, offensive and defensive skills, individual abilities of key players and even the psychological impact of fans in the stands. Football is a game where the average sum of scored goals is pretty little (two to three per game), which means that a moment of brilliance or stupidity of an individual can decide the final outcome. For this reason, it is a big challenge to choose the features and the way of classification which would facilitate the prediction.

In this paper, we will be developing a software system that can predict the outcome of Champions League matches with around 60% accuracy. The Champions League consists of two phases. In the first stage of the competition teams are divided into groups. Each group has four teams, who each play two games, one at home and one away. Two best teams from each group pass to the second round where they play in a knock-out tournament system.

The outcome of a match depends on a number of features and that requires a large number of experiments to determine the best subset which has the greatest impact on the final outcome of the match. In order to achieve the best possible properties of prediction we have tested a large number of classifiers. The final part of the project was the choice of classifiers which showed the best results in predicting the outcome of matches.

The initial set of more than 30 features was reduced to 20. The accuracy of the most successful classifier has reached a limit of 65%.

The next section gives an overview of related articles. The third section describes the procedure for determining the system through the selection process of the most important features and the most successful learning algorithms. The next section presents the achieved results. The article ends with the conclusion and references.

II. RELATED ARTICLES

In the paper [1] authors are describing approaches of predicting the results of football matches using Bayesian networks. They predict the outcomes of the games of the football club Tottenham Hotspur in the 95/96 and 96/97 seasons of the Premier League. Beside Bayesian network classification, they use other classifiers like: naive Bayes, MC4 (a member of the decision trees group), k-NN and Hugin's achievement of Bayesian networks. The set of selected features contains 30 attributes of which 28 are logical variables that indicate whether a particular player plays a certain match. The remaining two features indicate the quality of the opposing team and meeting place (home or away game).

In the other paper [2] the problem of predicting the outcome of football matches is approached primarily using artificial neural networks. In this paper, the system predicts the outcomes of games for all teams participating in Premier league in the 06/07 season. The obtained results are compared with the results achieved by the naive Bayesian classifier, k-NN algorithm, and J48 (member of the decision trees group). The selected set of features consists of the number of played matches, points achieved at a certain point, the outcome of the

Authorized licensed use limited to: University College Dublin. Downloaded on June 02,2024 at 09:57:38 UTC from IEEE Xplore. Restrictions apply.

meeting played in the home and away ground, and the current form.

These papers approach the problem of predicting outcomes with a similar set of classifiers, but with significantly different characteristics. The set of classifiers which is used in this paper covers a large part of the above classifiers, and the feature set contains all the features used in the paper [2] with the addition of several important attributes that can affect the final outcome like the number of injured players.

III. SYSTEM DESIGN

When designing a system for classification, the feature selection and choice of the learning algorithm can greatly affect the performance of classification.

A. *Feature Selection*

A large number of factors such as the form in which the teams is, the home court advantage, the overall quality of the team, the individual quality of players on the team, etc affect at the final outcome of a football match. The problem of selecting the features can be approached in two ways [1]. The first way implies that we have some knowledge about the problem (in this case the football matches) and that based on this knowledge we select those features that we believe would most affect the final result. The other way implies that we have very little or no knowledge about the problem and therefore choose all features that could affect the result, and then gradually try to determine those that have the greatest impact. Due to a very large set of features which could affect the outcome of the match and the many problems in collecting them, in this paper we used the first approach.

For the purpose of this study the following features have been selected:

- The current form of teams shown on the basis of results achieved in the last six games,
- The outcome of the previous meeting of the teams that play the game,
- The current position in the rankings,
- Number of injured players from the first team,
- The average number of scored and received goals per game.

In order to find the optimal mixture of the above features numerous tests have been carried out. Based on these tests we concluded that the optimal results are achieved by separation of features. To explain it a little more we could take the form of teams for an example. The initial idea was to show the number of obtained points in the last six games. The testing however has shown that the separation of these features into three features (number of wins, losses and draws) actually shows better results. In the same manner we separated other features. In the final set of features there are 20 features included. In addition, another set of features is constructed that includes all of the above listed features in addition to the subjective assessment of the quality of each team by experts in this field. The aim of this paper is to see to what extent adding a subjective expectation of outcomes affects the final accuracy of predictions.

B. *Selecting learning algorithms*

In order to achieve better prediction results we used many learning algorithms to determine which give the best results. Taking into account the experiences and results from the papers [1] and [2], the following algorithms have been chosen:

- Naive Bayes,
- Bayesian networks,
- LogitBoost,
- The k-nearest neighbors algorithm,
- Random forest,
- Artificial neural networks.

The selected algorithms are the most popular algorithms from the following categories: probabilistic, targets and lazy classifiers and decision trees. Artificial neural networks are added as they are universally applicable.

Naive Bayes algorithm belongs to the class of probabilistic classifiers, which assume that all features are independent [3]. The classification is done by applying the Bayes theorem.

Bayesian network is a representative of the probabilistic graphical models [4] group. The network is presented using a directed acyclic graph where nodes represent a random variable. The edge between two nodes X_i and X_j, where X_i is a parent node, represents a conditional probability $P(X_j, X_i)$.

The LogitBoost algorithm is a boosting algorithm [5]. Boosting algorithms consist of a set of weak classifiers, which perform badly if they are used as standalone classifiers (accuracy slightly better than 50%), but in combination give good results. Decision Stump algorithm (decision tree) is a commonly used weak classifier.

The representative algorithm from the group of lazy classifiers [3] is the k - nearest neighbor algorithm. Classification of new examples is done by finding k nearest neighbors in the space of features (typically using Euclidean distance) from the examples in the learning set. Based on these examples voting or another method determines classification of new examples.

In the group of decision trees algorithms the random forest is the most common algorithm. Random forest consists of a large number of decision trees which are built in some global style [6]. An example is classified in the class that is among the most common outputs of all the trees.

Artificial neural network is a mathematical model inspired by biological neural networks [7]. The network consists of mutually connected artificial neurons; each

Authorized licensed use limited to: University College Dublin. Downloaded on June 02,2024 at 09:57:38 UTC from IEEE Xplore. Restrictions apply.

connection between two neurons has a weighting factor. Network learning is based on the adjustment of weight factors based on relationships examples for learning.

C. *Implementation*

The software system has been implemented in the Java programming language with use of Weka API. Microsoft Excel was used for designing the data set. Since Weka API requires input data in a separate format (. ARFF), authors coded the converter in C# which forms data from Excel files in the requested format.

IV. RESULTS

A. *Data set*

In the group stage of Champions League 96 matches are played. Each match represents one example in the data set so we are working with a relatively small set. Each example consists of 20 features explained in chapter III. Basic statistical data are shown in table 1. Data from the past seasons would be more than welcome despite the fact that every year different teams participate. Unfortunately the necessary data was too hard to collect, so we had to settle with the current seasons results. Data has been collected manually from various websites with football statistics since no site contained all the necessary informations in one place.

Table 1 – Statistical data

Home win	Draw	Away win
52	27	17

B. *Performance evaluation*

For the purpose of testing data set was divided in 3 ways:

- Training and validation sets contain matches from the first 3 rounds, testing set the remaining 3 rounds
- Training and validation sets contains matches from the first 4 rounds, testing set the remaining 2 rounds
- Training and validation sets contains matches from the first 5 rounds, testing set the remaining round

In each round 16 matches are played. The optimal parameters for each classifier were determined by 10 fold cross-validation on the training and validation sets. Testing set contained only the matches which were played after every match in the training set. The reason for such a decision is really simple: in real life we don't know the outcome of future events.

The performance of each classifier is shown by its accuracy and F1 measure. For the sake of completeness, results from the reference method have also been added. In this case reference method classifies new examples in the most common class.

C. *Results analysis*

Test results are shown in Table 2. From these results we can observe how much of an impact feature and classifier selection has on the prediction capabilities. During testing one interesting property of data sets was observed which is also described in the following chapters.

1) *Feature selection*

If we observe the results from the point of feature selection we can see that our basic set (one without subjective estimates of team qualities) actually shows better results than expertly constructed set. From that we can draw two conclusions:

- Teams which participate in the Champions league are evenly matched so classification in the favor of the "favorite" of a match won't always show good results
- Expert who performed the labeling of the second set might not be that skilled after all

2) *Classifier selection*

Naïve Bayesian classifier shows the worst prediction accuracy among all the tested classifiers. Such bad results, which are comparable to reference method, are much unexpected if we take into account that feature set contains minimally interdependent features. Such set should in theory favor the naïve Bayesian classifier. The last claim can be confirmed if look at F1 measure results where each and every classifier (including the naïve Bayesian) shows significantly better results than the reference method.

Bayesian network shows a bit better results than naïve Bayesian classifier which is consistent with the results obtained in [1]. An especially interesting feature of Bayesian networks is its consistency across feature sets. Minimal difference in the accuracy between the two sets of features leads us to believe the selected sets might not be best suited for the structure of Bayesian networks. In order to confirm this assumption it is necessary to carry out additional testing.

Unlike Bayesian networks, the LogitBoost classifier shows significant difference in performance when changing feature sets. Better results were achieved on expertly built set which is in contrast with the behavior of other classifiers. To draw any valid conclusion for such behavior additional tests are required on larger data sets. Overall results achieved by LogitBoost algorithm belong to the top of the tested classifiers. Only artificial neural network shows better results.

Authorized licensed use limited to: University College Dublin. Downloaded on June 02,2024 at 09:57:38 UTC from IEEE Xplore. Restrictions apply.

Table 2 - Results

Feature set	Training/test set size		Naive Bayes		Bayessian net		LogitBoost		k-NN		Random forest		ANN		Most common	
	Trainng	Test	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc
Expert	3	3	0.5	47.9%	0.546	56.3%	0.5	50.0%	0.535	54.2%	0.511	52.1%	0.541	54.2%	0.22	50.0%
	4	2	0.542	53.1%	0.546	53.1%	0.594	59.4%	0.526	53.1%	0.551	53.1%	0.692	68.8%	0.24	56.3%
	5	1	0.554	56.3%	0.552	56.3%	0.675	68.8%	0.548	56.3%	0.468	50.0%	0.578	56.3%	0.22	50.0%
Basic	3	3	0.543	52.1%	0.567	56.3%	0.46	47.9%	0.569	56.3%	0.534	56.3%	0.587	58.3%	0.22	50.0%
	4	2	0.542	53.1%	0.534	56.3%	0.641	62.5%	0.638	62.5%	0.644	65.6%	0.692	68.8%	0.24	56.3%
	5	1	0.554	56.3%	0.517	56.3%	0.547	56.3%	0.548	56.3%	0.468	50.0%	0.578	56.3%	0.22	50.0%

Results achieved with k-NN classifier were done by taking into account 5 nearest neighbors. Overall results are relatively poor, just marginally better that those achieved by Bayesian networks. One possible cause for such results is the property of k-NN algorithm to take into account all the features or in other words its inability to filter out nonessential features. Before the tests were carried out all the features were normalized to unit interval.

Random forest classifier shows excellent performance on basic set of features and relatively average performance on expertly built set. Assignment of too big priority for subjective team quality marks might be the reason for lower performance on expertly built set. Results achieved on the last round are really poor and comparable to those of the reference method. The most probable reason for such behavior might be tree overfitting. Further tests are necessary to confirm such claim.

And in the end the best results were achieved by Artificial Neural Network (ANN) with prediction accuracy up to 68%. Network consisted of 5 hidden layers. Backpropagation algorithm was used to train the network.

3) *Training/test set size selection*

During our test process we observed one very interesting situation. By increasing the number of examples in the training set we expect to see an improvement in classification performance. In our tests we achieved better performance with the increase of training set from 3 to 4 rounds which was expected. On the other hand when we increased the training set to 5 rounds, classification performance dropped significantly. That is very surprising. One possible reason for such behavior might be our inability to model every possible significant feature. Namely, if we observe the results from the last round, we can see that there were many "surprises". By "surprises" we mean matches where objectively better teams, which secured their place in the next phase of competition, played under their usual level. There are many reasons for such behavior like protecting the main players from injuries or trying to avoid some teams in the next phases. Unfortunately things like that cannot be modeled and such examples can be seen as outliers.

D. *Final analysis*

Generally speaking, achieved results can be considered satisfying especially when using artificial neural networks. In comparison with the reference method (classification in the most common class), the results are significantly better. Even though direct and objective comparison with the results from similar articles is not possible, we can establish that our approach in best case scenario produces similar or slightly better results than those from other articles mentioned in chapter 2 (59% [1], 65% [2]).

V. CONCLUSION

Predicting the results of football matches poses an interesting challenge due to the fact that the sport is so popular and widespread. However, predicting the outcomes is also a difficult problem because of the number of factors which must be taken into account that cannot be quantitatively valued or modeled.

As part of this work, a software solution has been developed in order to try and solve this problem. During the development of the system, a number of tests have been carried out in order to determine the optimal combination of features and classificators.

The results of the presented system show a satisfactory capability of prediction which is superior to the one of the reference method (most likely a priori outcome). The goal set at the start of this project (achieving accuracy around 60%) was greatly surpassed so from that point of view we can consider this project successful.

Of course, there is room for further improvement, primarily in the area of feature selection. If we were to model the form for each and every player in the match we could probably achieve better results. This way we could monitor each players form during the season and determine its influence on the final score.

Larger data set for learning would also help to predict future outcomes.

ACKNOWLEDGEMENTS

We would like to thank all the anonymous reviewers for their helpful comments.

Authorized licensed use limited to: University College Dublin. Downloaded on June 02,2024 at 09:57:38 UTC from IEEE Xplore. Restrictions apply.

REFERENCES

[1] A. Joseph and N.E. Fenton and M. Neil, “Predicting football results using Bayesian nets and other machine learning techniques ” Jelsevier B.V., 2006.

[2] R.K. Balla, ‘’Soccer Match Result Prediction using Neural Networks’’, unpublished

[3] Tom M. Mitchell, “Machine Learning”, McGraw-Hill Science/Engineering/Math, 1. Edition, 1997.

[4] S. J. Russell and P. Norvig, ‘’ Artificial Intelligence A Modern Approach’’, 1. Edition, 1995.

[5] J. Friedman and T. Hastie, R. Tibshirani, “Additive logistic regression: a statistical view of boosting”, Annals of Statistics, 2000.

[6] L. Breiman, “MACHINE LEARNING - Random Forests”, Kluver Academic Publishers, volume 45, 5-32, 2001.

[7] B. Dalbelo Bašić, M. Čupić, J. Šnajder, Umjetne neuronske mreže, notes for Artificial intelligence course, FER Zagreb, 2008.

Authorized licensed use limited to: University College Dublin. Downloaded on June 02,2024 at 09:57:38 UTC from IEEE Xplore. Restrictions apply.