Using Weather Condition and Advanced Machine Learning Methods to Predict Soccer Outcome

Denny Asarias Palinggi	Francisco Ramos	Joaquín Torres-Sospedra	Sergio Trilles	Joaquín Huerta
INIT	INIT	INIT	INIT	INIT
Universitat Jaume I	Universitat Jaume I	Universitat Jaume I	Universitat Jaume I	Universitat Jaume I
Castellón, Spain	Castellón, Spain	Castellón, Spain	Castellón, Spain	Castellón, Spain
al373634@uji.es	jromero@uji.es	jtorres@uji.es	strilles@uji.es	huerta@uji.es

Abstract

Massive amounts of research have been doing on predicting soccer matches using machine learning algorithms. Unfortunately, there are no prior researches used weather condition as features. In this work, three different classification algorithms were investigated for predicting the outcomes of soccer matches by using temperature difference and several other historical match statistics as features. More concretely, the dataset consists of statistic information of soccer matches in La Liga and Segunda division from season 2013-2014 to 2016-2017 and meteorological data in every host city. The results show that the Support Vector Machine model has better accuracy score compare to K-Nearest Neighbours and Random Forest with 45.32% for temperature difference below 5° and 49.51% for temperature difference above 5°. Our test results have shown that weather information can be important factors to improve the prediction accuracy of soccer matches outcome.

1 Introduction

Soccer is currently the most popular team sport. The 2018 edition of FIFA World Cup was broadcast live to every territory around the world with an estimate 3.572 billion viewers watch the event (FIFA, 2018). With such a large amount of attention, soccer forecast has huge potential to become a profitable business. According to *Sportradar* director Darren Small, the international sports match-betting industry is worth an estimated $700 billion to $1 trillion annually a year which 70% of that trade has been estimated to come from soccer betting (Keogh & Rose, 2013).

The easy access to the Internet can be considered as the main reason for the growing revenue of betting industry since people can use a Smartphone application to bet on-line. Due to the financial assets involved in the betting process, the decision of which team is likely to win becomes of important relevance; thus bookmakers, fans, and potential bidders are all interested in approximating the odds of a game in advance (Bunker & Thabtah, 2017). Concurrent with the increase of soccer-betting, more people become enthusiasm to research on soccer forecast.

Soccer gambler usually prefers betting on predicting the Full-Time Result (FTR) even though there are also other outcomes that users can bet such as total goals, goal-scorer, halftime result and so on. There are only three possible outcomes of FTR which are home team win, draw, and away team win, therefore, predicting FTR can be categorized as a multiclass classification problem. One of the intelligent approaches that have been proven in classification domain is Machine learning (ML) (Bunker & Thabtah, 2017). In the past, ML methods were used to forecast the result of soccer matches. Those previous researches forgot to incorporate weather condition as one of the variables to predict the soccer outcome. For the work we are introducing, temperature difference between two teams and rain precipitation are used as weather information with main focus on temperature difference since the dataset is split based on temperature difference. The weather absolute difference is computed using the main location of the home and away teams.

This paper aims to show preliminary results in predicting the outcome of soccer matches using ML techniques and the focus will be on determining the FTR. Compare to other sports, soccer is very unpredictable since there are many factors need to be considered such as players quality, location injuries, and so on. To fulfil the goal, the specific objectives are:

- To design and implement various ML classification algorithms and optimize the hyperparameters to improve the accuracy of each algorithm.
- To compare the performance of some ML classification algorithms to find the best model.
- To conclude how much the effect of temperature difference can influence the match outcome.

The motivation of this work is to perform an initial exploration of how weather conditions in home-team location and away-team location can be used to predict the FTR.

2 Materials and Methods

This section introduces several classification algorithms will also be explained and the past related works within the topic of soccer prediction ML modelling.

The ML term refers to the automatic process of finding meaningful patterns in data. In the past couple of decades, ML become a common paradigm to solve any task that requires information extraction from big data sets. The learning process on ML is a process of gaining experience and convert it into knowledge. In the case of ML, before able to generate knowledge or expertise first it needs to receive experience in the form of training dataset (Shalev-Shwartz & Ben-David, 2014). There are various kind of algorithms that are used in ML. Commonly these algorithms are grouped into two approaches; unsupervised and supervised learning. The former corresponds to find the pattern from an unlabelled dataset, which means that the dataset does not have a pre-established corresponding output value. In supervised learning, on the other hand, predictions are based on some already known examples whose class is well-known (labelled dataset).

Since this research uses labelled dataset, only supervised learning will be evaluated further. Supervised learning problems are categorized into "regression" and "classification" problems. The main difference between regression and classification is that the data type of the label/output. If the label/output value is an ordinal numerical value (e.g., home prices) then it belongs to regression problem while if the label/output value is a cluster or a group (e.g., gender) then it belongs to classification. As this research output is to predict FTR (home team win, draw, away team win) with three possible states, classification algorithms are considered.

2.1 Supervised Classification Algorithms

2.1.1 Random Forest (RF)

Random Forest (RF) algorithm is a development of the Classification and Regression Tree (CART) method by applying bootstrap aggregating (bagging) and random feature selection methods. Even though Decision Tree (DT) algorithm is easy to interpret and not having many hyperparameters to over-tune but it is prone to overfitting. Overfitting is the ability of correctly classify the patterns used in training, but failing in non-seen examples. RF algorithm reduces the danger of overfitting is by constructing an ensemble of trees (Shalev-Shwartz & Ben-David, 2014).

Unlike DT, the RF method combines many trees to make classifications and prediction classes. In RF tree formation is done by doing training sample data. The selection of variables used for split is taken randomly. The classification is executed after all the trees are formed. This classification of RF is taken based on votes from each tree and the most votes are winners.

2.1.2 K-Nearest Neighbours (KNN)

K-nearest neighbour (KNN) is a supervised algorithm learning where results from new instances classified according to the majority of the closest K-neighbour category. For instance, we want to predict whether “a” is “cat” or “dog”, if K=4 and 3 of the closest is “cat” while only one is “dog”. From this result, the conclusion is “a”=”cat” because the majority of 4 closest neighbours of “a” is “cat”.

There are many ways to calculate the distance, for this research we choose three most famous distance formula, which are: Euclidian, Minkowski, or Manhattan.

The advantages of using KNN are it is a simple algorithm to explain and understand. The main disadvantage of the KNN algorithm is that it is a lazy learn which mean the way the algorithm perform classification is by use the training data itself rather than learn from it (Karthikeyan et al., 2016).

2.1.3 Support Vector Machines (SVM)

The current standard of Support Vector Machines (SVM) were introduced by Cortes and Vapnik back in 1995. Basically, SVM is an algorithm to separate data by using what is called a hyperplane into different groups with same classifier. For instances, in two dimensions, a hyperplane is a flat one-dimensional subspace (line). In three dimensions, a hyperplane is a flat two-dimensional subspace (plane). In $\rho > 3$ dimensions, it can be hard to visualize a hyperplane, but the notion of a $\rho - 1$ dimensional flat subspace still applies (James et al., 2013).

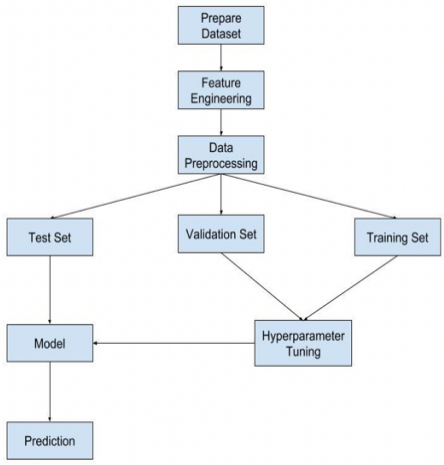
Since there are many ways to make hyperplane, the best possible hyperplane can be determined by measure the distance between the support vectors and the hyperplane. The best hyperplane is the one with the largest distance between the hyperplane and the support vectors, which can be called Maximum Margin Hyperplane (MMH). The support vectors are the points in the dataset from both classes that are closest to the MMH. The support vectors allow the algorithm to be memory efficient even with large amounts of data, as only the vectors need to be saved for future reference (Frölich, 2017).

Beside able to performing linear classification, SVMs can also perform a non-linear classification where it will mapping the input data into high-dimensional feature spaces, this method is known as the kernel function. By using kernel, the best hyperplane between classes can be found by measuring the maximum hyperplane margin between non-linear input spaces and characteristic spaces (Cortes & Vapnik, 1995). The commonly used kernel functions are: Linear kernel, Polynomial kernel and Radial Basis Function (RBF) kernel.

2.2 Methodology

This subsection discusses every steps of the research implementation which includes hardware and software, data gathering and pre-processing, create and select features that will be used for the models and develop the models. Figure 1 is the flowchart to visualize the methodology of this paper.

Figure 1: Flowchart of the proposed methodology



2.2.1 Hardware and Software

Python is chosen as programming language for this work because it has many options of inbuilt libraries that very useful for scientific computing. In this project, we used various libraries such as pandas for data manipulation and analysis, and seaborn for data visualization. PhpMyAdmin also used to manipulate the dataset on MySQL database especially when created all necessary features. As for machine learning library, scikit-learn was used because it features various machine learning algorithms. The experiments have been run in a computer with an Intel(R) Core(TM) i7-7500 2.90 GHz processor, an Asus UX530UX motherboard and 8 GB of DDR4 memory.

2.2.2 Data Sources

The historical matches dataset contains matches of *La Liga and Segunda division* (also known as *La Liga 2*) from season 2013/2014 until 2016/2017. La Liga is men's top professional soccer competition in Spanish soccer league system, while Segunda division is 2nd behind La Liga. Every season since 2010-2011, top two teams and the play-off winner between teams rank 3rd - 6th promoted to La Liga for the next season replacing three lowest rank teams, this means every season the composition of teams played in La Liga and Segunda division always different from previous season.

Totally, there are 3830 matches from season 2013-2014 until 2016-2017, however for this paper not all matches included in the final dataset since only matches with complete weather information will be eligible. In the end, only 3335 matches are eligible for final dataset.

The weather dataset is from the *Agencia Estatal de Meteorología* (AEMT). In order to get the weather conditions for every match, first thing to do is find the closest weather station for each stadium. After that, join both datasets based on weather station ID and the date of matchdays.

3 Experimental Evaluation

3.1 Data Preprocessing

Data pre-processing is needed to make sure the data are in a good quality to be used for machine learning algorithm, data with a lot of noise and irrelevant input can lead to misleading results when predicting unseen data. This step requires a lot of time since it involves not only cleaning and normalizing the data but also transforming and extracting feature.

Some ML algorithms can really be affected by the different scale of the features. For example, KNN classifier tries to measure the distance between data points when trying to predict the label, this means features on large scale will dominate the prediction. To solve this issue, features need to be re-scaled as an initial step. All features for this paper are normalized, which means they are from the interval [0; 1].

3.2 Data splitting

There are many ways to split the dataset, but due to the fact that there is a time-element in the professional soccer dataset then it is better to split data between training and testing historically. Therefore, the seasons 2013/14 to 2015/16 were selected for training and validation, whereas the season 2016/17 was selected for testing.

After performing the training process, the final model should be able to predict the label/output of testing dataset correctly; but most of the time the final model overfits and is only able to correctly predict the training data (Reitermanova, 2010) and accuracy decreases for testing data. .

One of the solutions to avoid overfitting is to use k-fold cross-validation where the data are split into *k* parts of the same size. The *k*-th part of the dataset is used for validation while the rest of the dataset used for training the model, In most of cases, k = 10 is chosen which mean this process is repeated 10 times for each part of the data. This process able to reduce the risk of overfitting because for each iteration the final model is using a different combination of training and validation data

3.3 Hyperparameter Optimization

Beside data splitting, another factor that need to be considered to find the best algorithm is the choice of its hyperparameters values. Every algorithm has different hyperparameters, for example K in KNN or kernel in SVM.

Usually, the value of hyperparameters is choosing randomly and then pick the hyperparameters value with the best accuracy result. But it can be a very exhausting process especially there is more than one hyperparameter for each algorithm, therefore it is better to use an algorithm to find the best hyperparameter combination automatically such as grid-search.

This process definitely takes a lot of time. But even though the grid-search process takes a lot of time, it is pretty straightforward and safer compare to other methods which avoid doing an exhaustive parameter search (Hsu et al. (2003)). Table 1 shows the dictionary of parameters and their corresponding values for KNN, SVM, and RF algorithms, the methods considered for this work.

Table 1: The dictionary of parameters

ALGORITHM	PARAM	VALUE
KNN	**K**	3,..,50
	Weight	uniform, distance
	Metric	City block, Minkowsky, Euclidean
SVM	**Kernel**	Linear, RBF
	Gamma	0.1, 1, 10, 100, 500, 1000
	C	0.1, 1, 10, 100, 500, 1000
RF	**Estimators**	10, 50, 100, 150, 200
	Min leaf	0.1, 1, 10, 100, 200, 500
	Max features	auto, sqrt, log2

4 Experimental Evaluation

To understand the impact of the weather into soccer outcome, we decided to split the dataset into two different datasets. For each match, we calculate the temperature difference of the station next to the home team stadium and the station next to the away team stadium. We considered that for an absolute difference temperature higher than 5º, the away team performance might be affected by the weather conditions. Therefore, one dataset only contains the data from matches where the absolute temperature difference was above 5º, whereas the other dataset contains the data from matches were the absolute temperature difference was 5º or lower.

Furthermore, we split the data into two different case studies:

- Case study 1, where only weather features are used (temperature difference and rain precipitation) to predict the FTR.
- Case study 2, where weather and historical statistics (the total points of home team in the last 4 home matches; the total points of away team in the last 4 away matches; the difference between number of goal scored and conceded of the home team in the last 4 home matches, and the difference between number of goal scored and conceded of the away team in the last 4 away matches) are used to predict FTR.

Moreover, Grid Search method combined with 5-Fold Cross-Validation, determines the best hyperparameters value. The best hyperparameters value combination are picked based on the accuracy score and are selected for each dataset (below 5º and above 5º) and features (Case study 1 and Case study 2).

4.1 Model Accuracy

Table 2 show the accuracy score for KNN, SVM, and RF algorithms. The way to calculate the accuracy is by sum total number of samples correctly predicted divided by total number of samples in dataset.

Table 2: % of matches correctly predicted for each model.

	KNN		SVM		RF	
FEAT.	**Below 5º**	**Above 5º**	**Below 5º**	**Above 5º**	**Below 5º**	**Above 5º**
CASE 1	42.68	47.11	44.79	47.59	43.73	46.63
CASE 2	43.38	41.82	45.32	49.51	41.62	43.26

In the experiment where only weather features were used (Case 1), all models showing better accuracy score for dataset with temperature difference above 5° compare to below 5°. SVM model shows the best accuracy with 47.59% but KNN is better in terms of accuracy improvement from below 5° to above 5°. KNN model shows the best improvement of accuracy (4.43%), followed by RF (2.9%), and SVM (2.8%). In the experiment where all features are used to predict surprisingly SVM is the only model to show improvement of accuracy prediction for both below and above 5°, these results are unexpected since it was assumed that by adding historical statistics as features it will improve the prediction accuracy for every model. KNN model is even show decrease of accuracy prediction from below 5° to above 5° (-1.56%), SVM accuracy for dataset above 5° is 49.51% which is an improvement of 4.19% compare to dataset below 5°.

4.2 Missclassification Rate

Choose the best model based solely on accuracy score can be misleading because in many situations where the dataset has large class imbalance, a model can predict the value of the majority class for every prediction and achieve a high classification accuracy. Since most of the times the home team wins the match, misclassification rate also needs to be calculated in order to find an ideal model. Table 3 shows the misclassification rate for FTR classes.

Table 3: The dictionary of parameters

		CASE STUDY 1		CASE STUDY 2	
MODEL	**LABEL**	**Below 5º**	**Above 5º**	**Below 5º**	**Above 5º**
KNN	**home win**	15.44	11.76	23.93	30.39
	draw	94.51	88.88	85.97	83.33
	away win	90.27	96.15	81.94	86.53
SVM	**home win**	3.86	9.8	3.86	0.98
	draw	100	94.44	85.97	96.29
	away win	96.52	92.30	81.94	100
RF	**home win**	13.51	13.72	32.81	31.37
	draw	95.12	88.88	80.48	83.33
	away win	88.88	94.23	79.16	78.84

The results of classification (see Table 3) shows that SVM classifier gives the best performance in terms of classification accuracy but it also gives high misclassification rate on both draw and away team win. Further, in one case SVM classifiers even show 100% misclassification rate on away team win class: which mean it failed to predict every sample in that class. Based on solely on misclassification rate, we can say that RF model is more balance since only two times it has class with more than 90% misclassification rate.

5 Conclusions

Weather conditions show a good potential to improve predictions of the outcome of soccer games. Using the SVM algorithm the final test classification accuracy of the outcome was 44.79% for predicting matches with temperature difference below 5° and 47.59% for temperature difference above 5°. When other historical statistics features also used the accuracy rate improves significantly with 45.32% for temperature difference below 5° and 49.51% for temperature difference above 5°. However, based on misclassification rate calculation the SVM model accuracy rely too much on majority class which is home team win. Future work can be performed on this subject; for example, other weather data could be used such as the average speed of wind during the matchday, the weather data during the exact time span of the match also could improve the accuracy of the model, and more variation on dataset samples such as match between two team from different country or continent could also improve the accuracy since the temperature difference can be more significant.

As future work, we consider to extend this work in order to consider more weather and location-based features and other well-known advanced and deep learning models.

6 Acknowlegments

Denny Asarias Palinggi would like to thank Francisco Ramos and Joaquín Torres-Sospedra for all correction and improvement suggestion for this work. He would also like to thank friends and colleagues motivating him to do this work.

7 References

Bunker, R. P., & Thabtah, F. (2017). A machine learning framework for sport result prediction. Applied Computing and Informatics.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. Machine learning, 20(3), 273-297.

Frölich, I. (2017). Machine Learning for Classifying Cellular Traffic. Master's thesis, Chalmers University of Technology.

Hsu, C. W., Chang, C. C., & Lin, C. J. (2003). A practical guide to support vector classification.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An introduction to statistical learning (Vol. 112). Springer.

Karthikeyan, T., Ragavan, B., & Poornima, N. A comparative study of algorithms used for leukemia detection (2016). International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume, 5.

Keogh, F. & Rose, G. (2013). Football betting - the global gambling industry worth billions. https://www.bbc.com/sport/football/24354124. [accessed 28/12/18].

Mitchell, T. M. (1997). Machine learning (mcgraw-hill international editions computer scienceseries).

Reitermanova, Z. (2010). Data splitting. In WDS (Vol. 10, pp. 31-36).

Shalev-Shwartz, S., & Ben-David, S. (2014). Understanding machine learning: From theory to algorithms. Cambridge university press.