Predicting the World's Biggest Competition, The World Cup

Hannah Harhai

Data Analytics, University of Pittsburgh

***Abstract*— This project seeks to determine if one can accurately predict the outcome or winner of the FIFA Men's World Cup. With the power of machine learning, I believe this task to be possible. Within this paper, I will discuss the importance of the World Cup along with background information and the rules of the competition. I will also thoroughly discuss and explain the data/datasets used, the techniques and tools used to create, enhance, and evaluate the model, and lastly the results of the model and the impact/value it could provide.**

I. Introduction

The FIFA World Cup, often referred to simply as the World Cup, is an international soccer competition played by men's national teams. The tournament is held every 4 years, starting in 1930, and as of today, there have been 22 total tournaments. The tournament involves a qualification phase that takes place in the preceding 3 years, to determine which teams qualify, and from there, 32 teams will compete for the title. The World Cup is the most prestigious soccer tournament in the world, as well as the most widely viewed and followed single sporting event in the world. That being said, is there a way we can predict the winner of future World Cups? This question is fascinating to me, not only as a huge soccer fan but also because every four years when the World Cup rolls around almost everybody has an opinion on who they believe will win it all. Being able to actually predict the result of this competition with high accuracy would make watching what truly unfolds all the more interesting. This project would be of value and fun for avid fans all around the world. The World Cup is arguably the most important sporting event in the world, it is a global competition, and for hundreds of millions of people worldwide, it is the ultimate cultural expression. I believe this project can be done with the help of machine learning in which I will examine historical data and look at particular metrics.

II. Methodology

A. Feature Selection

The predictors I will use to try and predict my response variable, the overall winner of a World Cup competition, will be total number of player awards each team has received, number of wins in World Cup competitions, number of goals scored in World Cup competitions, and previous tournament standings for each competition (top 4 teams are ranked). I chose these metrics as my predictors for several reasons. The total number of player awards each team has received will be an indicator for which countries generally produce the best players. The total number of wins and goals scored by each team across all World Cup competitions will be a great indicator of how well we'd expect them to do in an upcoming competition. Lastly, previous tournament standings for each competition will give us data on which teams historically will most likely make it the farthest in the competition.

B. Examining and Parsing the Data

To examine and parse my data I used Jupyter Notebook with the help of various libraries such as pandas, numpy, matplot, etc. I collected my data/datasets from the Fjelstul World Cup Database found on Github. "The Fjelstul World Cup Database

is a comprehensive database about the FIFA World Cup created by Joshua C. Fjelstul, Ph.D. that covers all 22 World Cup tournaments (1930-2022). The database includes 27 datasets (approximately 1.1 million data points) that cover all aspects of the World Cup" [1]. From this database I pulled four different datasets including award_winners.csv, goals.csv, team_appearances.csv, and tournament_standings.csv to examine total player awards, total wins, total goals, and tournament standings respectively. First I loaded in each dataset and created dataframes of them. Each dataset contained many columns/features so my next task was to retrieve only the data necessary for my model. Using pandas I was able to create four new dataframes all with the exact data I needed. I now had dataframes with the total number of awards received by each country/team per tournament, the total number of goals scored by each team per every tournament, the total number of wins each team earned per tournament, and lastly each teams final rank in every tournament (only top 4 specified) plus each overall winner. After parsing through all the data to collect only what I needed I merged all four datasets into one. Since many columns contained null values I replaced them all with zeros. For total goals, 0 will indicate 0 goals scored. For total player awards, 0 will indicate no player awards were received by that team. For final rank, 0 will indicate that the team didn't rank in the top four of said tournament. Lastly, as I mentioned above, for winner, 0 will indicate that said team did not win the tournament. To see a snapshot of the final merged dataframe see below.

tournament_id	team_name	total_wins	total_goals	total_player_awards	final_rank	winner
WC-1930	Argentina	4	18.0	1.0	2.0	0.0
	Belgium	0	0.0	0.0	0.0	0.0
	Bolivia	0	0.0	0.0	0.0	0.0
	Brazil	1	5.0	0.0	0.0	0.0
	Chile	2	5.0	0.0	0.0	0.0
	France	1	4.0	0.0	0.0	0.0
	Mexico	0	4.0	0.0	0.0	0.0
	Paraguay	1	1.0	0.0	0.0	0.0
	Peru	0	1.0	0.0	0.0	0.0
	Romania	1	3.0	0.0	0.0	0.0
	United States	2	7.0	1.0	3.0	0.0
	Uruguay	4	15.0	1.0	1.0	1.0
	Yugoslavia	2	7.0	0.0	4.0	0.0
WC-1934	Argentina	0	2.0	0.0	0.0	0.0
	Austria	2	7.0	0.0	4.0	0.0
	Belgium	0	2.0	0.0	0.0	0.0
	Brazil	0	1.0	0.0	0.0	0.0
	Czechoslovakia	3	9.0	1.0	2.0	0.0
	Egypt	0	2.0	0.0	0.0	0.0
	France	0	2.0	0.0	0.0	0.0

Fig. 1 Final Dataframe of Collected Data

My next task was to gather descriptive statistics for each feature, check for missing values, and check for any outliers. I performed calculations to find the mean, median, standard deviation, and variance for all features for each country across all tournaments. These calculations were to give me insight into what kind of numbers I would be working with and to help me select the best approach for my machine learning model. As expected there were no missing values for me to deal with as I already converted the null values to zeros as in my case they made sense for the dataset. My next course of action was to check for outliers and determine if any needed to be dropped. Due to the nature of my dataset, each of my features contained multiple outliers. Total wins contained outliers but this is due to the fact that the one team who wins each tournament gets more wins than all the other teams who competed. Total goals had multiple outliers, however, after examining the dataset where the outliers lie and doing some fact checking, these values were indeed

correct. Total player awards outliers were due to the fact that the majority of teams, especially those who do not make it to the final 4, generally do not receive any player awards for their respective teams. While the teams who reach the end receive the majority of these player awards. Final rank's outliers, again, are due to the fact that only 4 teams per tournament are ranked and the rest are not. Lastly, winner's outlier is because in this case winning the competition is the outlier as we know only one team wins per tournament while all the others lose. Therefore, I dropped none of the outliers as they were correctly meant to be in the dataset and were datapoints I needed to make the most accurate model.

C. *Selecting Approaches*

For this project I will be attempting two different machine learning models to see which one gives the best results/performance. Based on the fact that I have labeled data this can be categorized as a supervised machine learning problem. Subsequently, since the output of my model is a class/label this is considered a classification problem. That being said for this project I will be making use of two classification algorithms, the Random Forest algorithm and the linear kernel SVM algorithm. Random Forest is one of the most popular machine learning algorithms today and is widely used in classification and regression problems. Random forest consists of a large number of individual decision trees that operate as an ensemble. Each tree then produces a class prediction, the class with the most votes will become the model's prediction. I chose Random Forest for this reason as I was drawn to its predictive capabilities and robust algorithm. SVM stands for Support Vector Machine, and it works by "plotting each data item as a point in n-dimensional space (where n is the number of features you have), with the value of each feature being the value of a particular coordinate. Then, classification is performed by finding the optimal hyper-plane that differentiates the two classes. I selected SVM because of its classifying capabilities and also due to the fact that its algorithm isn't sensitive to outliers which my data has multiple of.

D. *Preprocessing the Data*

Preprocessing the data often called feature engineering is a critical step in trying to create a machine learning model. The techniques I made use of were scaling by standardization, and performing discretization on features. "Feature scaling is a data preprocessing technique that involves transforming the values of features or variables in a dataset to a similar scale. This is done to ensure that all features contribute equally to the model and to prevent features with larger values from dominating the model" [2]. When examining the scales of my features I found that they are all on fairly small scales, however, one has to take into account the type of algorithm(s) they are using. Since I am creating one model using an SVM algorithm I decided to use the standardization scaling technique which centers the data around the mean and scales to a standard deviation of 1. I chose the standardization technique as it is less sensitive to outliers and therefore fits the data I'm working with the best. Scaling is crucial for this type of algorithm as SVM takes into consideration the distance between observations. The Random Forest algorithm is not a distance based model, it instead is based on partitioning data to make predictions, therefore, using scaled data/unscaled data would result in the same outcome. Lastly, I'd like to

discuss the process of discretization. "Discretization, or binning, is the process of transforming continuous variables into discrete variables by creating a set of contiguous intervals, also called bins, that span the range of the variable values" [3]. I chose to use discretization on all features for the Random Forest model as alternative decision-tree models are not best suitable for continuous features. I used an equal width discretization method, which "separates all possible values into '**N**' number of bins, each having the same width" [4]. This allowed me to separate my continuous variables into a specified number of bins to create new discrete variables for my Random Forest model. Since I am using a linear kernel for my SVM model I will not be using the discretized version of the features, and will be sticking with the original continuous data for that model.

E. Training and Testing the Data

After completing all the preprocessing and feature engineering it was now time to finally train and test the data. I first defined the X and y variables, X being the features total_wins, total_goals, total_player_awards, and final_rank, y being the winner category. Next, I split X and y into training and test sets that were now ready to be inputted into my chosen machine learning algorithms.

III. Results

A. SVM Model

I chose four evaluation techniques to determine the performance of the SVM model. I calculated the accuracy score, AUC score, performed a classification report, and graphed a confusion matrix of the results. The classification accuracy score represents the percentage of correct predictions. The AUC score is the percentage of the ROC plot that is underneath the curve. ROC is a probability curve and AUC represents degree or measure of separability. Therefore, the ROC curve is a performance measurement for classification problems that can be used at various threshold settings. AUC tells how much the model is capable of distinguishing between classes. The higher the AUC score, the better the model is at predicting 0s as 0s and 1s as 1s. The classification report provides metrics to assess the quality of the model. The confusion matrix is a table that describes the performance of a classification model which we will see visualizations of below. The results of the four evaluation techniques are shown/described below.

```
Accuracy Score: 0.9795918367346939
AUC Score: 0.8946236559139785

Classification Report:
              precision    recall  f1-score   support

           0       0.99      0.99      0.99        93
           1       0.80      0.80      0.80         5

    accuracy                           0.98        98
   macro avg       0.89      0.89      0.89        98
weighted avg       0.98      0.98      0.98        98
```

Fig. 2 Accuracy Score, AUC Score, Classification Report Results for SVM Model

Note - Confusion matrix attached in appendix at the end of report for better visualization

As we can see from the results we got an accuracy score of about 98% which means that out of all the model's predictions 98% were correct. However, one can not rely on just that metric to determine if the model was truly accurate hence the next few subsequent tests. The AUC score gave us a result of 0.89 which is also a great result. This number means our SVM model did a thoroughly good job at classifying the observations correctly. The classification report has multiple metrics for us to

examine. The precision column corresponds to the percentage of correct positive predictions relative to total positive predictions. The recall column is the percentage of correct positive predictions relative to total actual positives. The f1-score is a weighted harmonic mean of precision and recall. The closer to 1, the better the model [5]. After examining the numbers within the classification report we can clearly see our SVM model performed incredibly well and garnered high scores across all columns. Lastly, when examining the confusion matrix (attached in appendix) we can clearly see that our model almost every time correctly categorized the true negative values (0.99) and very rarely incorrectly classified false negatives (0.2). On the other side, our model correctly identified true positives with a score of 0.8, while only incorrectly identifying false positives with a score of 0.011. All evaluation metrics indicate that the linear kernel SVM model performed with high results.

B. Random Forest Model

I conducted the same four evaluation techniques as above, and will now proceed to show the results of the Random Forest model below.

```
Accuracy Score: 0.9693877551020408
AUC Score: 0.7946236559139785

Classification Report:
              precision    recall  f1-score   support

           0       0.98      0.99      0.98        93
           1       0.75      0.60      0.67         5

    accuracy                           0.97        98
   macro avg       0.86      0.79      0.83        98
weighted avg       0.97      0.97      0.97        98
```

Fig. 3 Accuracy Score, AUC Score, Classification Report Results for Random Forest Model

Note - As above the confusion matrix is attached in the appendix.

As we can see the Random Forest model also performed very well. The accuracy score is about 97% so just one less than our SVM model. The AUC score evaluated to about 0.79 which is 0.10 less than the SVM model. Nonetheless, an AUC score of 0.79 still indicates that the model was pretty accurate at identifying the correct classifications. Next, examining the classification report gives us more insight into how well the model performed. The precision, recall, and f1-scores were all lower in our Random Forest model than the previous SVM model. However, the numbers are just slightly below the numbers our SVM model produced and still indicate a good performance from the classifying model. The confusion matrix for our Random Forest model is also very similar to the one our SVM model produced. The Random Forest model correctly predicted the true negatives almost all the time (0.99) while only incorrectly predicting false negatives with a score of 0.4. Again, on the other side of the matrix, the model correctly predicted true positives with a score of 0.6 and only incorrectly classifying false positives similarly to our SVM model with a score of 0.011. Therefore, all evaluation techniques indicate that the Random Forest model also performed with high accuracy as a classifier.

C. Comparing Models

Now that we've thoroughly examined results from both models we can determine which one statistically performed better. The SVM model has better statistics/numbers across all evaluation techniques and therefore, was the better classifier. Although, it must be noted that both models performed exceedingly well and worked great as classifiers for the data I had. It's also important to note what these results mean in the context of my initial problem. These high results indicate that both models would be able to accurately classify the losers of a World Cup tournament as well as classify the winner. In other words, both machine learning models would be able to predict the outcome of a World Cup tournament with high accuracy.

IV. Discussion

My original question that started this project was if one could accurately predict the World Cup competition using the power of machine learning. After performing this experiment I can conclude that both models I created worked incredibly well and I believe have the power to do exactly that. However, I must admit that this was a rather naive first attempt which may have resulted in the high scores I received for both respective models. If I were to modify and go further with this project in the future I would get more features to test and also research and examine more datasets to use to be able to create an even more robust model. Nonetheless, I do believe that this was a great starting point for the intended model I wanted to create and also provided me with a great learning experience. All in all, with the power of machine learning there are infinite possibilities of what one could achieve, and I believe that today I was able to answer my initial question, that it is indeed possible to predict the world's biggest competition, the World Cup.

V. References

[1] Fjelstul, Joshua C. "The Fjelstul World Cup Database v.1.0." July 8, 2022.

[2] Bhandari, Aniruddha. "Feature Scaling | Standardization vs Normalization." Analytics Vidhya, 3 Apr. 2020, www.analyticsvidhya.com/blog/2020/04/feature-scaling-machine-learning-normalization-standardization/.

[3] BISHT, SHUBHAM. "Discretization Continuous Variables." Kaggle.com, www.kaggle.com/code/mrbisht/discretization-continuous-variables. Accessed 27 Apr. 2023.

[4] Gupta, Rohan. "An Introduction to Discretization in Data Science." Medium, 6 Dec. 2019, towardsdatascience.com/an-introduction-to-discretization-in-data-science-55ef8c9775a2.

[5] Zach. "How to Interpret the Classification Report in Sklearn (with Example)." Statology, 9 May 2022, www.statology.org/sklearn-classification-report/.

[6] "Does the Random Forest Algorithm Need Normalization?" KDnuggets, www.kdnuggets.com/2022/07/random-forest-algorithm-need-normalization.html.

VI. Appendix

Confusion Matrix:

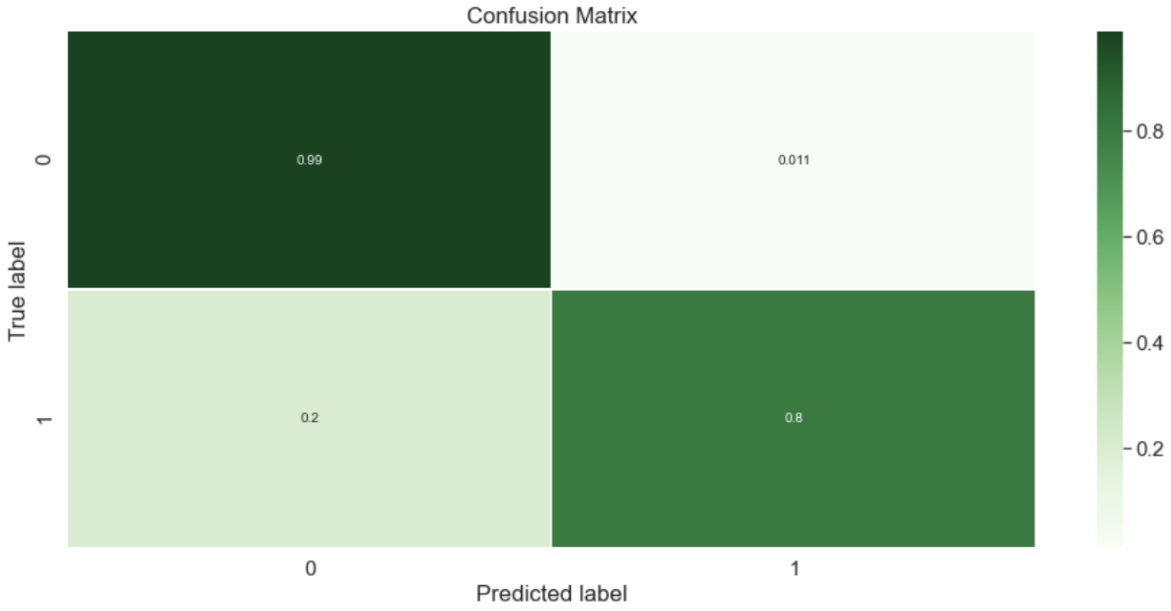

Fig. 4 Confusion Matrix for SVM Model

Confusion Matrix:

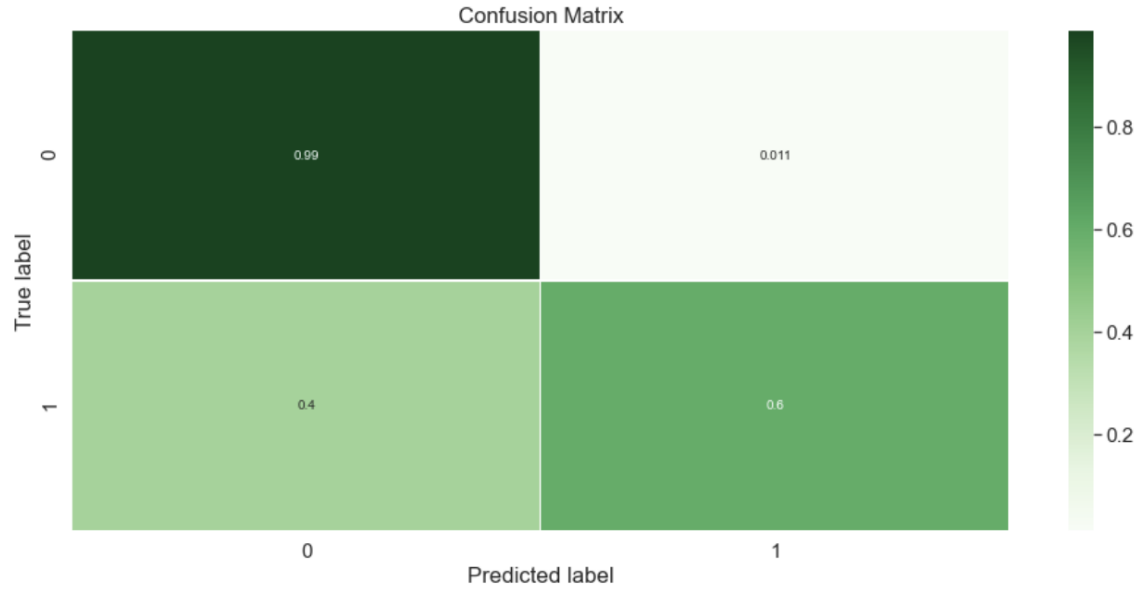

Fig. 5 Confusion Matrix for Random Forest Model