Predict the First 10 in 2022 FIFA World Cup

Yueyang Liu, Violet Chen, and Alejandra Taboada

Dr. Zhengling Qi

DNSC 4280

The George Washington University

Table of Contents

Problem Statement

The 2022 FIFA World Cup is an international football tournament featuring the men's national teams of FIFA member associations. The 22nd World Cup held in Qatar from November 20 to December 18, 2022; It is the first World Cup to be held in the Arab and Muslim worlds, and the second to be held entirely in Asia after South Korea and Japan in 2002. The World Cup is a game that people all over the world pay attention to and the audience participates in the game in different forms, such as watching the game, betting on the game, and so on. It is the carnival for both football aficionados and everyman, and it is lucrative.

Our goal for the project is to figure out which team is going to win the World Cup based on historical tournament data. We try to predict each match's outcome and simulate the knockout bracket for the World Cup. Although we know that winning or losing on the field is largely influenced by the player's form on the day and other external conditions, we still believe that data analysis and machine learning can provide some reference in predicting the World Cup. With our prediction, one may better figure out which team will win the matches and become the winner of the World Cup.

Data Selection

We used complete data from all international football matches played since the 90s. On top of that, the strength of each team is provided by incorporating actual FIFA rankings as well as player strengths based on the EA Sports FIFA video game. The original data set was scraped from https://www.fifa.com, and we acquired the compiled data set from Kaggle.com. This data set is extensive, containing 23,920 rows for each occurrence of each football match and 25 columns of various factors to describe the match details. These factors includes:

1 bool variable: shoot_out

14 numeric variables: home_team_goalkeeper_score, away_team_goalkeeper_score, home_team_mean_defense_score, home_team_mean_offense_score, home_team_mean_midfield_score, away_team_mean_defense_score

Away_team_mean_offense_score, away_team_mean_midfield_score
home_team_fifa_rank, away_team_fifa_rank, home_team_total_fifa_points,
away_team_total_fifa_points, home_team_score, away_team_score

10 categorical variables:
date , home_team, away_team, home_team_continent, away_team_continent, , tournament, city, country, neutral_location, home_team_result

Data Pre-Processing

Data Filtering

For data preprocessing, we first select the teams who will participate in the 22nd World Cup, since those are the teams we may use to simulate the match knockout bracket and the only teams we care about:

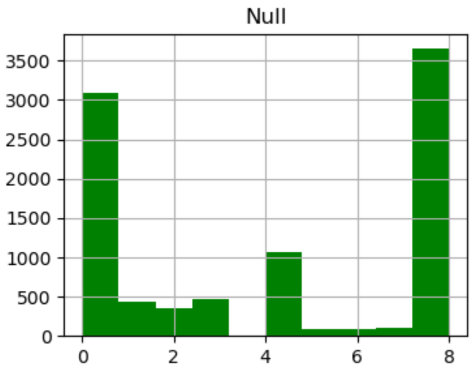
> ['Qatar', 'Germany', 'Denmark', 'Brazil', 'France', 'Belgium', 'Croatia', 'Spain',
> 'Serbia', 'England', 'Switzerland', 'Netherlands', 'Argentina', 'IR Iran',
> 'Korea Republic', 'Japan', 'Saudi Arabia', 'Ecuador', 'Uruguay', 'Canada', 'Ghana',
> 'Senegal', 'Portugal', 'Poland', 'Tunisia', 'Morocco', 'Cameroon', 'USA', 'Mexico',
> 'Wales', 'Australia', 'Costa Rica']

Then, we dropped a few columns that include information we deem as not important based on domain knowledge, which include demographic variables such as continent and unrelated variables such as tournament. We keep the home team and away team because we believe that the names of teams, as all football fanatics will assume, carry important and non-explicit information. For example, when you see Portugal against Qatar, you will predict that Portugal will win. We also make the assumption that the Team will have the highest variable importance in our ensemble classification models. After the preliminary filtering, we now have a data frame with 9321 rows and 13 columns. We alter the column names to make them more interpretable and here are the columns we have left for fitting the model:

Name	Modeling Role	Measurement Level	Description
Team1	predictor	categorical	country name of home team
Team2	predictor	categorical	country name of away team
Team1_FIFA_RANK	predictor	int64	home team FIFA rank
Team2_FIFA_RANK	predictor	int64	away team FIFA rank
Team1_Goalkeeper	predictor	float64	home team goalkeeper score
Team2_Goalkeeper	predictor	float64	away team goalkeeper score
Team1_Defense	predictor	float64	home team defense score
Team2_Defense	predictor	float64	away team defense score
Team1_Offense	predictor	float64	home team offense score
Team2_Offense	predictor	float64	away team offense score
Team1_Midfield	predictor	float64	home team midfield score
Team2_Midfield	predictor	float64	away team midfield score
Team1_Result	target	categorical	home match result with {'Win':1, 'Draw':2, 'Lose':0}

Handling Missing Values

We checked for missing values, and found that around half of our numeric predictors contain missing values (see graph 1).

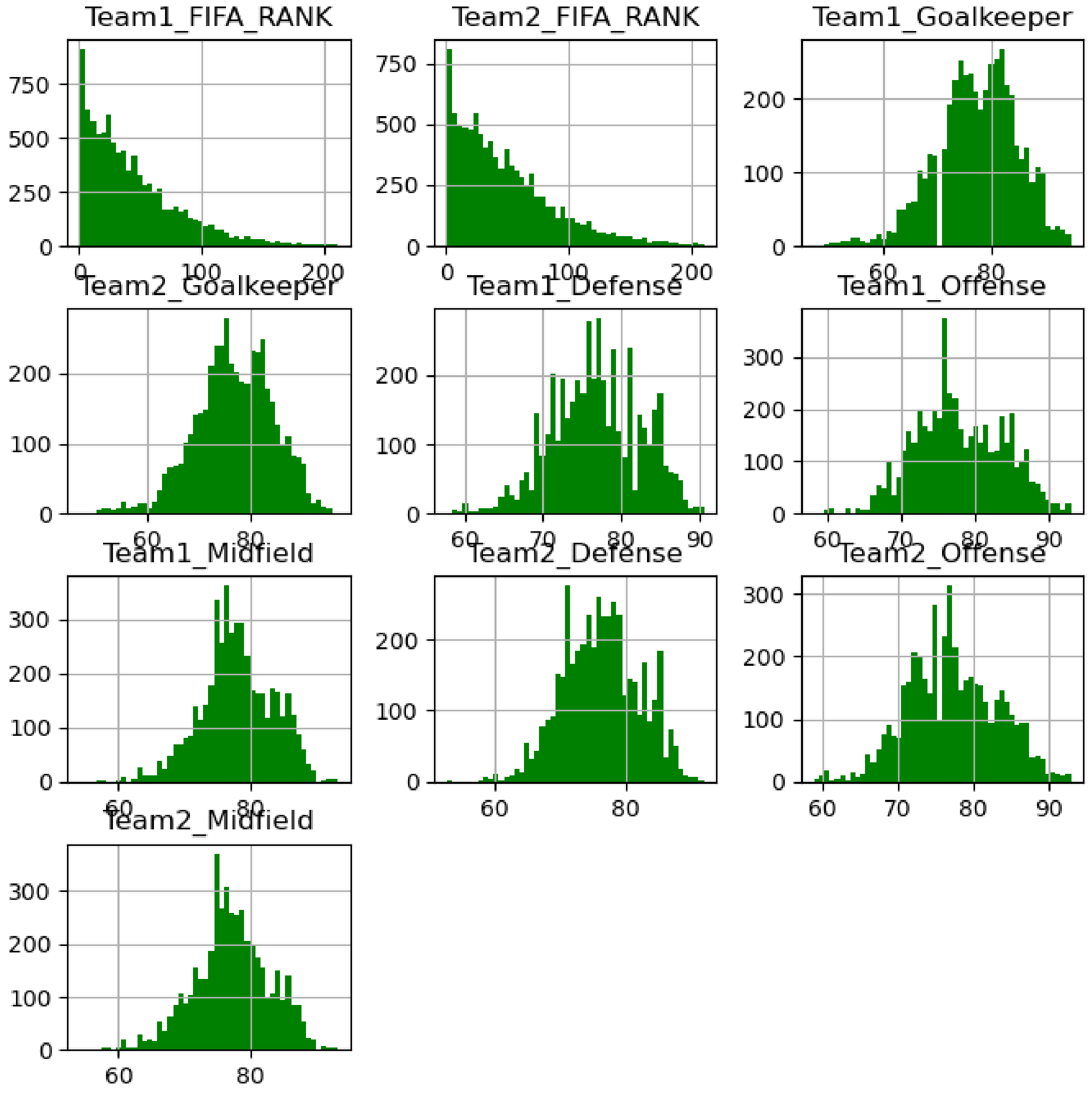


(Graph 1 - missing values distribution before filling in missing values)

We wanted to keep as much data as possible, so we only drop rows with more than 4 NAs. Then, we filled in the rest using column mean. By doing this, we now have 5396 rows with no missing values.

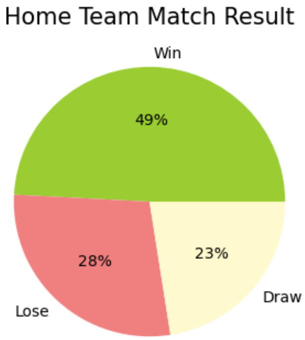
Descriptive Analysis

We plot the histogram with all numeric columns (see Graph 2). By the plot, we decide that the numeric value has a distribution that looks relatively like a bell shape within a reasonable range, except for Team1_FIFA_RANK and Team2_FIFA_RANK. Thus, we assumed a normal distribution and decided not to standardize our numerical data.



(Graph 2 - histogram for all numeric columns)

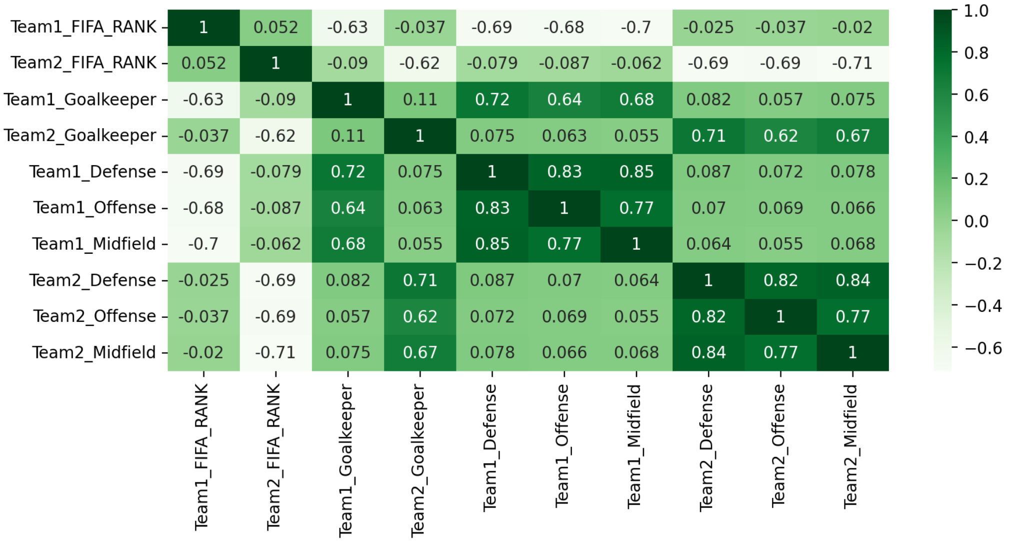
We also want to see if the target variable has a balanced distribution. Thus, we plot the pie chart for the target variable Team1_Result (see Graph 3). We can see that it is reasonably balanced data with 49% winning, 28% is lost, and 23% being drawn. Thus, we do not process any balancing treatment in regard.



(Graph 3 - pie chart for target variable)

Multicollinearity

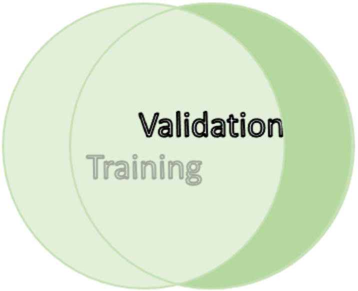
Multicollinearity usually causes inaccurate results and significance of variables. This dataset has a multicollinearity issue because a good team may be excellent in all aspects – with both a good goalkeeper and good defense (see Graph 4). We cannot eliminate multicollinearity here nor can we mitigate it due to the limited predictors we have in hand. More discussion will be available in the limitation section of this report.



(Graph 4 - heatmap for variable correlations)

Data Partition

We partitioned the data into 70% training and 30% validation sets. Our current training set has 3777 rows and 12 predictor variables, and 1 predictor variable. Our current validation set has 1619 rows and 12 predictor variables, and 1 predictor variable. However, due to the high cardinality of our dataset – that's too many unique values in categorical predictors "Team1" and "Team2" (211 unique country names) – we have to identify those labels that exist in the validation set but not in the training set (see Graph 5) and drop the corresponding rows (shown in darker green in Graph 5). After dropping those 17 rows, we have 3777 rows and 12 + 1 variables in the training set and 1602 rows and 12 + 1 variables in the validation set.



(Graph 5 - visualization of rows to be dropped in darker green)

Data Encoding

Because we used Python to conduct the analysis (more information will be available in the mining technique section), many available packages and pre-coded models in this platform do not take categorical data as it is. We need to generate labels for "Team1" and "Team2." To avoid data leakage, we used "Team1" and "Team2" in the training data set to create label encoders and fit encoders to both training and validation sets. We specified the types of these columns as type "category". Then, we convert our target variable by mapping it to {'Win':1, 'Draw':2, 'Lose':0}.

All 6 phases were completed, and we finished our model preprocessing and moved to models.

Mining Technique Selection

Basic Information

Before getting into models, we like to present and recapitulate some basic information related to the data and models for documentation purposes and replicability considerations.

- Person developing models:
    - Yiqi Chen, ychen20@gwu.edu
    - Alejandra Taboada, ataboada@gwu.edu
    - Yueyang Liu, yliu37@gwu.edu
- Models date: December 14, 2022
- Models version: 1.0
- License: MIT
- Model implementation code: DNSC4280_ML_Project_WorldCup_Group3.ipynb
- Intended Use
    - Intended uses: Models are used to predict the winner of the 2022 World Cup, as the major deliverable of DNSC 4280 group project.
    - Intended users: Students and the professor in GWU DNSC 4280 class.
    - Out-of-scope use cases: Any use beyond an educational example is out-of-scope.
- Random seed: 42
- Data partition: 70% training, 30% validation
- The number of rows in training and validation data:
    - Training rows: 3,777
    - Validation rows: 1,602
- Columns used as predictors in the models: Team1', 'Team2', 'Team1_FIFA_RANK', 'Team2_FIFA_RANK', 'Team1_Goalkeeper', 'Team2_Goalkeeper', 'Team1_Defense', 'Team2_Defense', 'Team1_Offense', 'Team2_Offense', 'Team1_Midfield', 'Team2_Midfield'
- Column used as the target in the models: 'Team1_Result' – {'Win':1, 'Draw':2, 'Lose':0}
- Version of the modeling software:
    - Python version: 3.9.13
    - Sklearn: 1.0.2
    - Keras: 2.11.0
    - Xgboost: 1.7.1

Model of Choices

Given that our target variable is a categorical/nominal variable with 3 levels of value – {'Win':1, 'Draw':2, 'Lose':0} – we decided to use classifiers to perform classification on 'Team1_Result'.

One common classification technique for machine learning as well as statistics is logistic regression. Logistic regression by default is usually a binary classifier, but we can use multinomial logistic regression – an extension of logistic regression for multi-class classification – to predict the target variable with 3 classes.

The second immediate model we can think of is the naive Bayes classifier. It manages data that is continuous and discrete, like our data. When we filtered rows with countries who participated in the 2022 World Cup, we used "or" instead of "and" to preserve as many data points as possible. Therefore, our "Team1" or "Team2" columns also contained countries who do not participate in the 2022 World Cup. It is capable of making predictions based on available information and we try to maximize the use of each row even if a pair of countries does not present. Also, it is not sensitive to unimportant characteristics.

K-nearest neighbor (KNN) is also a great classifier, using the nearest k nearby data points to predict. KNN is a lazy learner and it does not make assumptions about data. Our data set has a reasonable size, a reasonable number of predictors, and an approximately normal distribution. Also, we took care of all missing values which KNN is sensitive to. Thus, we deem that KNN will be a good modeling technique to be used in this data set.

The decision tree is another popular classifier and is highly interpretable for humans. However, we do not think our decision tree will perform too well, because it is highly unstable and usually subject to the random seed used (in our case, all random seed = 42). We fitted a decision tree as a comparison to the random forest.

Based on numerous base trees, a random forest is robust and is capable of classification tasks. It avoids overfitting by using multiple trees and ensemble techniques. In comparison to the decision tree method, the random forest algorithm offers a higher level of accuracy in outcome prediction.

A neural network is a type of artificial intelligence system that is modeled after the human brain. It is composed of interconnected nodes, or neurons, that are designed to process data and learn from it. Neural networks are used in a variety of applications, such as image recognition, natural language processing, and autonomous vehicles. You can use a neural network in this project to analyze data related to the teams competing in the World Cup and their performance in past tournaments. The neural network can then use this data to make predictions about which team is most likely to win the World Cup.

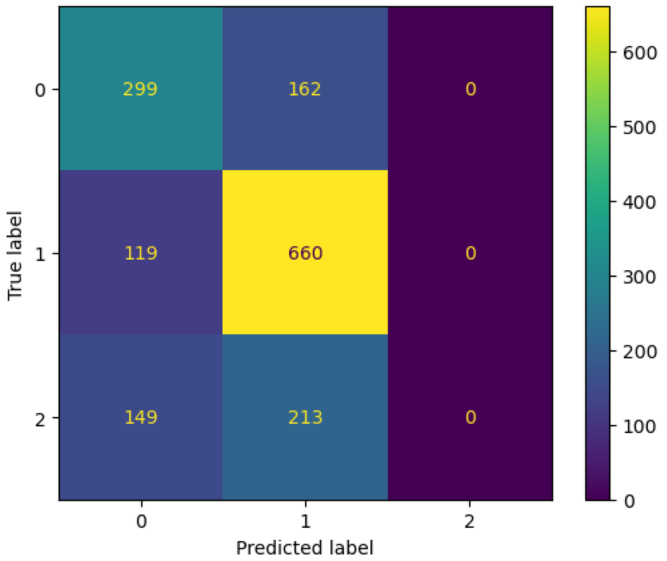
XGBoost is an advanced implementation of gradient boosting that uses decision trees as its base model. It is an efficient and powerful algorithm that can be used for both regression and classification problems. XGBoost works by building an ensemble of decision trees, each of which is trained on a subset of the data. The predictions from each tree are then combined to make a final prediction. XGBoost is known for its speed, accuracy, and scalability, making it a popular choice for machine learning.

In conclusion, we are going to fit the following models: logistic regression, naive Bayes classifier, KNN, decision tree, random forest, neural network, and XGBoost. Please see more detail in the next section for our results.

Data Mining Results

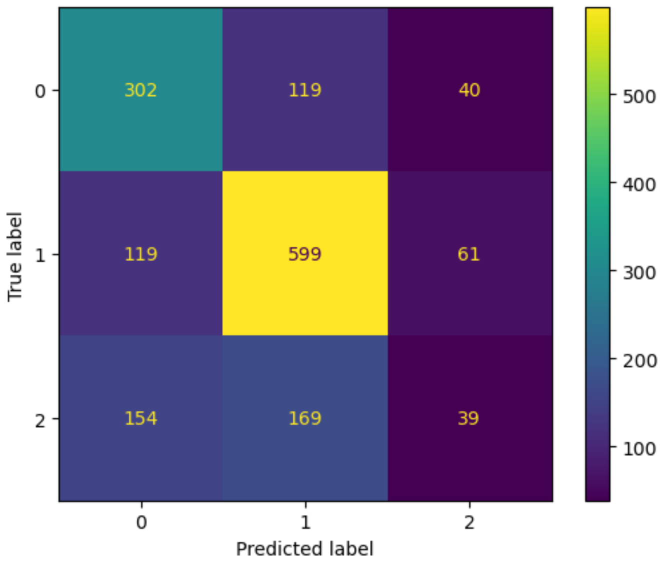
Logistic Regression

A logistic regression was fitted using multi_class='multinomial'. Misclassification rate = 0.4014.



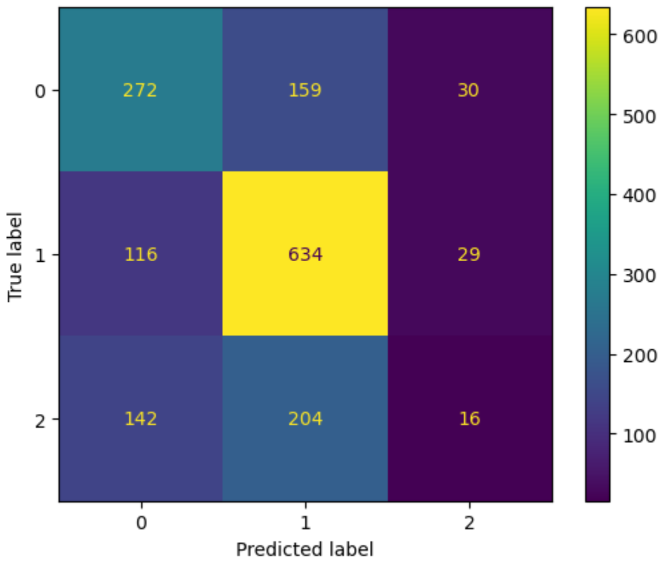
Naive Bayes

A naive Bayes was fitted using GaussianNB because we assume the predictors are normally distributed. Misclassification rate = 0.4132.



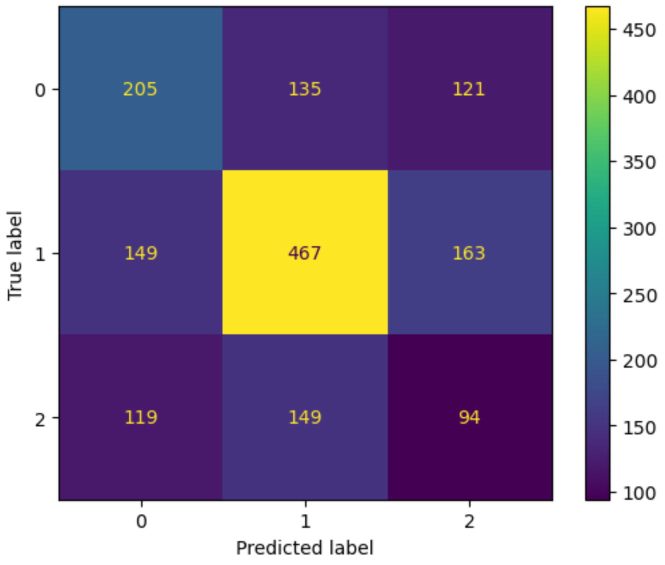
K-Nearest Neighbor

A KNN was fitted using a ‘uniform’ weigh – assigning all nearest neighbors with equal weights – with n_neighbors = 26, the best hyperparameter under k = 30 when using grid search. Misclassification rate = 0.4245.



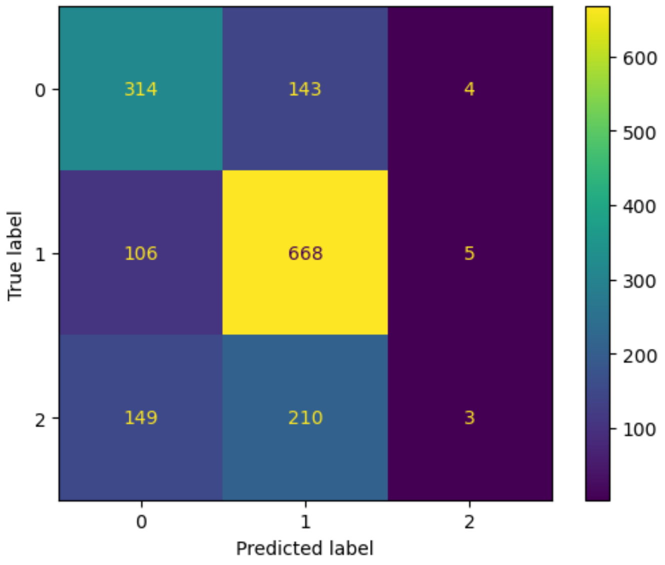
Decision Tree

A decision tree is fitted using random_state = 42 to make sure the result is consistent. Misclassification rate = 0.5218.

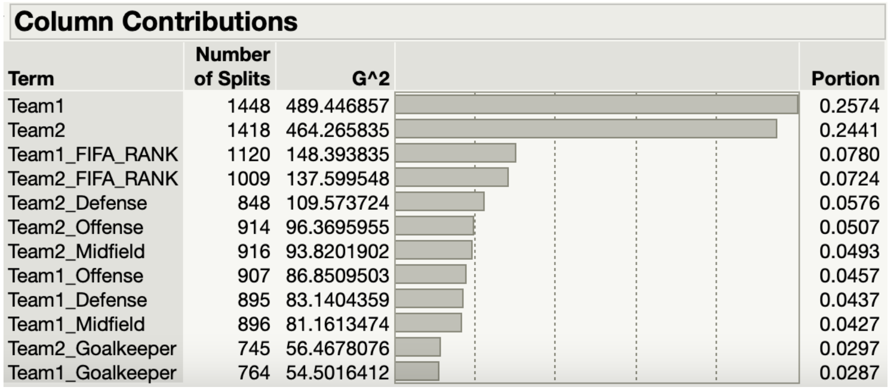


Random Forest

A random forest is fitted using RandomForestClassifier(criterion = 'entropy', n_estimators = 120, max_depth = 100, max_features = 'sqrt', min_samples_leaf = 15, min_samples_split = 5, random_state=42) using grid search. Misclassification rate = 0.3851.



The variable importance of our random forest model using the same hyperparameters and random seed as stated above are plotted as expected, with Team 1 and Team 2 taking up around 50% of the total variable importance/ column contribution.



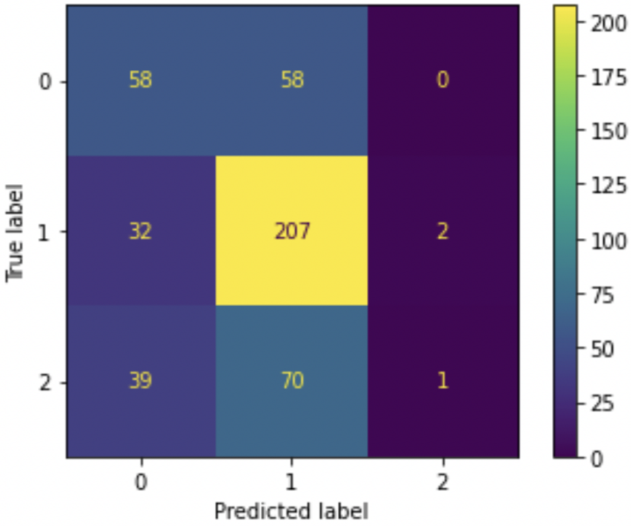
Column Contributions

Term	Number of Splits	G^2		Portion
Team1	1448	489.446857		0.2574
Team2	1418	464.265835		0.2441
Team1_FIFA_RANK	1120	148.393835		0.0780
Team2_FIFA_RANK	1009	137.599548		0.0724
Team2_Defense	848	109.573724		0.0576
Team2_Offense	914	96.3695955		0.0507
Team2_Midfield	916	93.8201902		0.0493
Team1_Offense	907	86.8509503		0.0457
Team1_Defense	895	83.1404359		0.0437
Team1_Midfield	896	81.1613474		0.0427
Team2_Goalkeeper	745	56.4678076		0.0297
Team1_Goalkeeper	764	54.5016412		0.0287

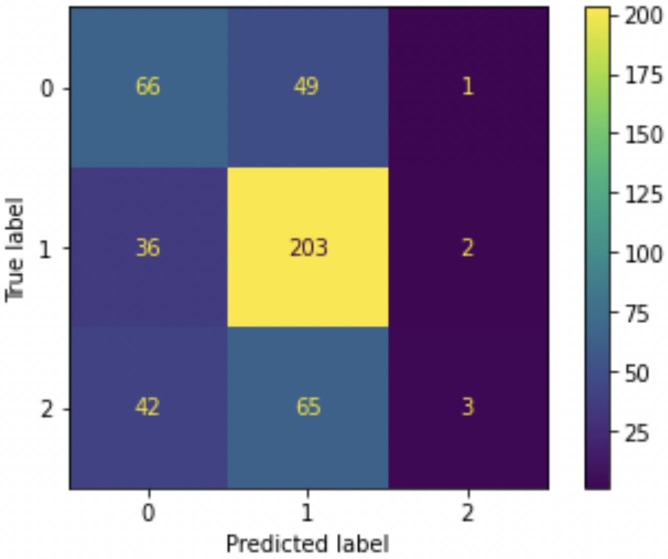
(Graph 6 - column contribution of the random forest)

XGBoost

The AdaBoost Classifier is used to create a strong classifier from a set of weak classifiers. This is done by combining multiple weak classifiers into a single strong classifier. By using the AdaBoost Classifier, we can create a strong classifier that can then be used as the basis for the XGBoost algorithm. The XGBoost algorithm is an ensemble learning method that uses the AdaBoost Classifier as its base classifier. By using the AdaBoost Classifier as the base classifier, the XGBoost algorithm can more accurately and efficiently identify patterns in data and make predictions. AdaBoost Classifier: from sklearn.ensemble import AdaBoostClassifier metrics_display(AdaBoostClassifier()).



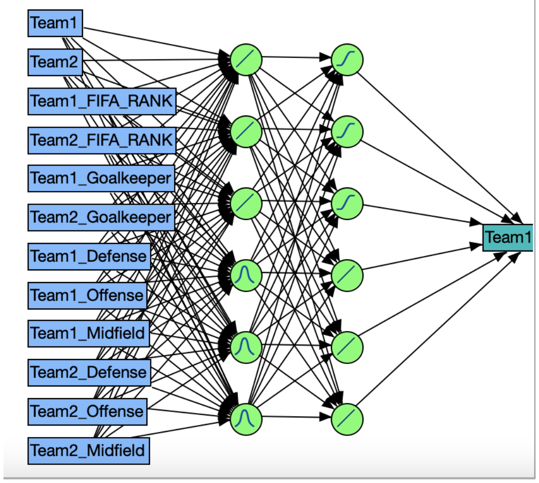
- XGB Boost - Misclassification rate on validation set = 41.76%

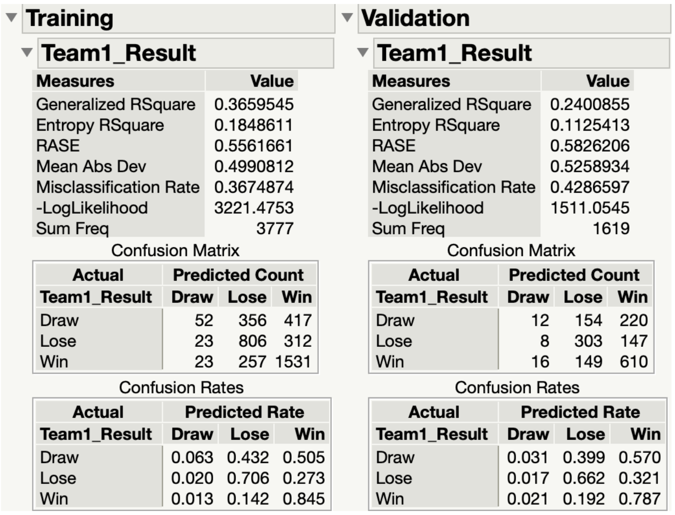


The XGB Boost model has achieved a misclassification rate of 41.76% on the validation set, which indicates that the model is performing fairly well. This result is likely a result of hyperparameter tuning to optimize the model's performance. It is important to remember that the model's performance can be further improved by trying different techniques.

Neural Network

A neural network with 2 hidden layers and 6 nodes each is used to fit the model using random seed 42. Misclassification rate = 0.4287.





Training

Team1_Result

Measures	Value
Generalized RSquare	0.3659545
Entropy RSquare	0.1848611
RASE	0.5561661
Mean Abs Dev	0.4990812
Misclassification Rate	0.3674874
-LogLikelihood	3221.4753
Sum Freq	3777

Confusion Matrix

Actual	Predicted Count		
Team1_Result	Draw	Lose	Win
Draw	52	356	417
Lose	23	806	312
Win	23	257	1531

Confusion Rates

Actual	Predicted Rate		
Team1_Result	Draw	Lose	Win
Draw	0.063	0.432	0.505
Lose	0.020	0.706	0.273
Win	0.013	0.142	0.845

Validation

Team1_Result

Measures	Value
Generalized RSquare	0.2400855
Entropy RSquare	0.1125413
RASE	0.5826206
Mean Abs Dev	0.5258934
Misclassification Rate	0.4286597
-LogLikelihood	1511.0545
Sum Freq	1619

Confusion Matrix

Actual	Predicted Count		
Team1_Result	Draw	Lose	Win
Draw	12	154	220
Lose	8	303	147
Win	16	149	610

Confusion Rates

Actual	Predicted Rate		
Team1_Result	Draw	Lose	Win
Draw	0.031	0.399	0.570
Lose	0.017	0.662	0.321
Win	0.021	0.192	0.787

Model Comparison and Best Model

Model Name	Misclassification rate
Logistic Regression	0.4014
Naive Bayes	0.4132
KNN	0.4906
Better KNN	0.4245
Decision Tree	0.5218
Random Forest	0.3964
XGB Boost	0.4176
Neural Network	0.4287

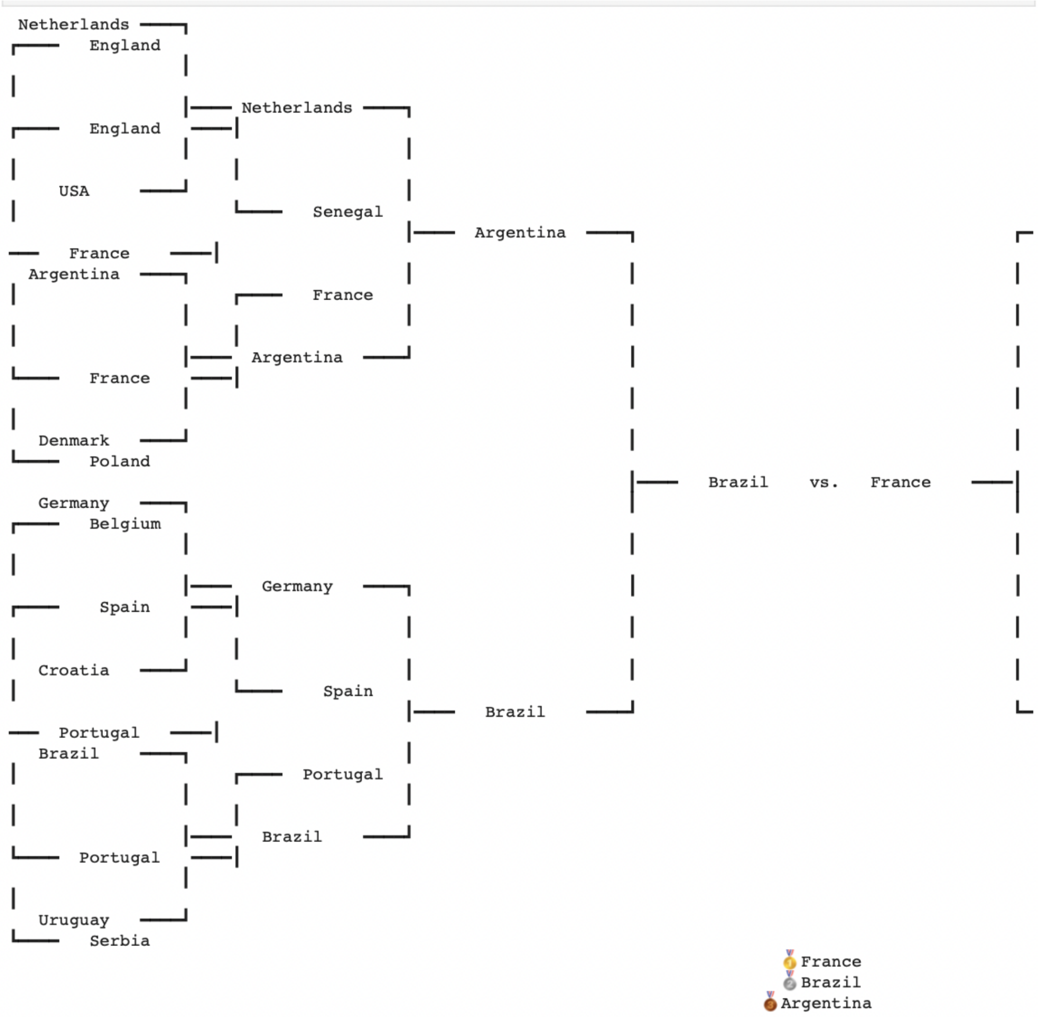
Based on the misclassification of the different models above, we could see that the random forest model has the lowest misclassification rate,0.3964 and the KNN has the highest misclassification rate In machine learning, 0.4906. The misclassification rate is a metric that tells us the percentage of observations that were incorrectly predicted by some classification model. It shows that random forest models predict models more correctly than other models.

When comparing the misclassification rates of the different models, it is clear that the random forest model is the most accurate among the three. This is because of its ability to capture important and complex relationships between variables, which allows for better generalization and predictive performance. Furthermore, the random forest model is able to handle a large number of features, which makes it suitable for the classification task. This is due to the fact that the model is able to identify the most important features and discard irrelevant ones.

Additionally, the KNN model has the highest misclassification rate. This is likely due to the fact that KNN is lazy, does not follow a training process, and in consequence, it does not try to optimize any effectiveness measure such as misclassification rate or accuracy.

Thus, based on the misclassification rate analysis, the random forest model is the best performer among our models, with the lowest misclassification rate.

Model Deployment - Tournament Knockout Bracket



(Graph 7 - visualization of our simulation result using our best model)

While predicting the 2022 World Cup is difficult at this point, the predictions are quite impressive. Football is a sport where many factors, both on and off the field, can influence the outcome of a match or tournament, making it difficult to predict the exact winner. Countries like Morocco and Korea have less established sides, which makes them harder to predict.No one expected Morocco to make it so far in the World Cup due to their lack of experience in international tournaments. Additionally, Morocco was not expected to perform well due to their lack of star players and the fact that they were competing against some of the best teams in the world. All of these factors contributed to the low expectations for Morocco.

Using models like Random Forest, Decision Tree, and Naive Bayes can help with the prediction process by incorporating a range of variables and taking into account the uncertainty of each variable. However, it is important to remember that models can only go so far in predicting outcomes. For example, there may be new players or changes in tactical approach that are not accounted for in the data used to train the model. As we have seen in this year's World Cup, France made it to the Final but Brazil didn't despite being favored in the predictions. This shows how unpredictable certain matches can be and highlights the need to always be prepared for the unexpected.

Limitations

Technical Limitations

Time and hardware constraints: We only have limited time to complete our project, which limits our ability to find more robust models. Also, even though the data set is not as large, using grid search to define the best hyperparameters combination is challenging for our personal laptops. When more than 1,000 candidate models with different hyperparameter sets are fitted, it may take more than 5 minutes to complete and is often restricted to limited options. This is a major limitation, especially when we don't have access to powerful hardware like a high-end GPU.

Subjectivity to hyper-parameters: We used random seed = 42 to partition our data into 70% training and 30% validation sets, and we delete the rows in the validation set but not in the training set. Using a different seed will result in a different partition and the rows that need to be deleted will also change. Beyond the use of the random seed, the performance of a machine learning model is often heavily dependent on the values of the hyperparameters used to train it. This means that the choice of hyperparameters can have a big impact on the model's performance. As discussed above, we can only try a limited combination of hyperparameters, which makes it a limitation on our model training and selection. In addition, for models like the decision tree, due to their instability nature, their performance largely depends on the random seed used and is thus subject to the selection hyper-parameters (seed = 42 in our case). We did not proceed with any underspecification test. Thus, changes in the random seed may result in unexpected outcomes.

Data Selection Limitations

Data timeliness and representativeness: Our training data can be dated back to 1993-08-08. The timeliness of our data can also be a limitation, especially for a sport like a football where a constant retirement of players or joining of new blood exists. If the data we used to train your model is out of date – it does not represent the current tournament – the usefulness of your model will be constrained. Other than timeliness, whether data sufficiently represent the real problem we are trying to solve also requires extra considerations. There are teams joining the World Cup and teams leaving; some team has changed their team members or coaches. These may lead us to reconsider whether historical team performance is the best indicator of real tournament performance.

Null values: Our dataset contains a large amount of null or missing values, which take up almost 50% of all rows. We have to drop some nulls and compute the other using column means. These can cause inaccurate predictions when training a machine learning model.

Multicollinearity: Multicollinearity occurs when two or more predictor variables in a model are highly correlated with each other. As discussed in the data preprocessing section, we cannot remove nor remediate multicollinearity in our data set. Because if a team plays well, the team can do well in all aspects, including goalkeeper, midfield, defense, and offense. This can be a problem because it can make it difficult to interpret the individual contributions of each predictor variable to the model's predictions, and may also lead to inaccurate prediction outcomes.

High cardinality issue in categorical variables: We are using categorical variables “Team1” and “Teams” in the models, which include 221 unique values in total out of 3,777 rows of training and 1,602 rows of validation data. We believe that our assumption of team name containing underlying information to be captured holds, but some of these names will have insufficient data points to output an accurate result.

Potential Future Steps

Knowing the limitations of our project, we may propose some future possible steps we can take to improve our model performance. Here are 2 major things we can do:

Find better data: The majority of our limitations are a result of poor data quality – a large percentage of incomplete data, out-of-date data, and multicollinearity. Thus, one practical approach is data augmentation. Also, given the possibility that the historical team performance may not be the best predictor of real tournament performance, we can also resort to players’ data, incorporating each player’s performance, or computing additional predictors based on the information related to the player’s performance in a team in total.

Tune more models: The other major limitation is hardware issues. We may resort to a less computationally intensive tuning package other than grid search. By doing this, we may be able to find better hyperparameter combinations, especially for models with ensemble techniques. We also see the potential of neural networks, whose performance is highly dependent on the network architecture. Determining which architecture and activation function to be used is hard, but with more time and better hardware, we may be able to try out more combinations and locate a robust model.

Conclusion

The 2022 FIFA World Cup is a highly anticipated event for football fans around the world, and our predictions using data analysis and machine learning techniques may provide valuable insights for those interested in the tournament. In the end, of the six models, the random forest model is the best method to classify the winning chance of teams in the world cup based on historical tournament data. Although the misclassification rates were similar throughout the models, the random forest performed better than other models. Moreover, the contribution of the variables performs similarly to the real-life situation and is better than other models. Ultimately, we conclude the random forest model is better to predict the tournament in the world cup. We still have some limitations. To improve the performance of the model in the future, the project will focus on finding better data with fewer missing values and multicollinearity, as well as augmenting the data with information on individual players' performance. Additionally, the project will also try out different model architectures and hyperparameter combinations using less computationally intensive tuning techniques.

Predicting the 2022 World Cup has been an interesting and challenging task. While we may not get to see the rivalry between Messi and Ronaldo in the Final, we can still enjoy the tournament and all it has to offer. Football is a sport that has something for everyone, and there is always something new and exciting happening. We never know what will happen on the pitch, and that is part of what makes football the most popular sport in the world. No matter the outcome, football is the best sport to watch and play, and the 2022 World Cup is sure to be an unforgettable tournament.