PREDICTING THE WINNER FOR FIFA WORLDCUP

Using various Data Mining Techniques

STAT 557 DATA MINING

Himanshu Sukheja
hxs376@psu.edu

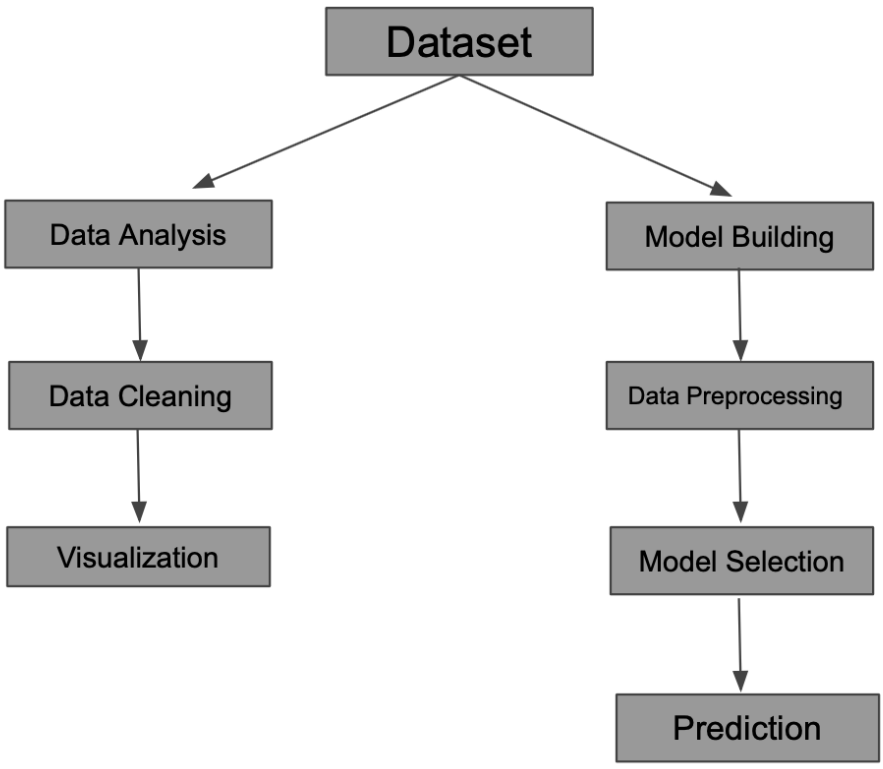
Nikhil Sunil Nandoskar
nxn59@psu.edu

1. Problem Statement

The FIFA World Cup is a global football competition contested by the various football-playing nations of the world. It is contested every four years and is the most prestigious and important trophy in the sport of football. he championship has been awarded every four years since the inaugural tournament in 1930, except in 1942 and 1946 when it was not held because of the Second World War. The current champion according to the dataset is Germany, which won its fourth title at the 2014 tournament in Brazil.

The purpose of this project is a twofold motive. Firstly, we used data pre-processing and data cleaning in order to make the dataset useable for applying various data analysis techniques to make sense out of data. In Second part of the project we try and predict the top 3 teams for World Cup 2018 using classification models to predict the exact results of the semi-finals, third place playoff and final.

2. Life Cycle



3. Dataset

The dataset files were obtained from Kaggle. They can be obtained from here. We will use results of historical matches since the beginning of the championship (1930) for all participating teams. Although, there was a primary limitation which we faced in this dataset which made it difficult for us to do a close to real prediction, the FIFA rankings were given beginning of 1990's that makes a huge portion of dataset lacking. So, we retained to historical match records.

We first did some exploratory analysis on the three datasets, do some feature engineering to select most relevant feature for prediction, do some data manipulation and data cleaning, choose a Machine Learning model and finally deploy it on the dataset.

The dataset had three different files

1. WorldCupMatches.csv
2. WorldCupPlayers.csv
3. WorldCups.csv

The World Cup Matches dataset had 4572 rows and 20 columns. Out of the 4752 rows, only the first 852 rows had data and remaining values were NAN for all columns. This was causing unnecessary noise in the dataset before we could analyse it. The following table shows us the corresponding values for the 20 columns in the World Cup matches data set.

Year	Datetime	Stage	Stadium
City	Home Team Name	Home Team Goal	Away Team Name
Away Team Goal	Win Conditions	Attendance	Half-time Home Goal
Half-time Away Goal	Referee	Assistant 1	Assistant 2
RoundID	Match ID	Home Team Initial	Away Team Initial

The World Cup players data set had 37784 rows and 9 columns. However, there were 9069 missing values in the events column and 4143 missing values in the position column. To gain more insight we learnt that event represented the time stamp when a player scored a goal and we could not enter a random value for that in order to complete the data set. This would cause unnecessary noise to be introduced in data set. The position column represented the position of a player, in our dataset unfortunately only two values are present for this column (Goalkeeper and captain). We came up with a unique solution for missing values problem in position column. If for a row in Position has an empty value then check the corresponding row's data in Event, if a goal is scored then it will be scored by a Captain. Missing values were filled in by taking into consideration of event column.

The following table shows all the possible columns in World cup players dataset.

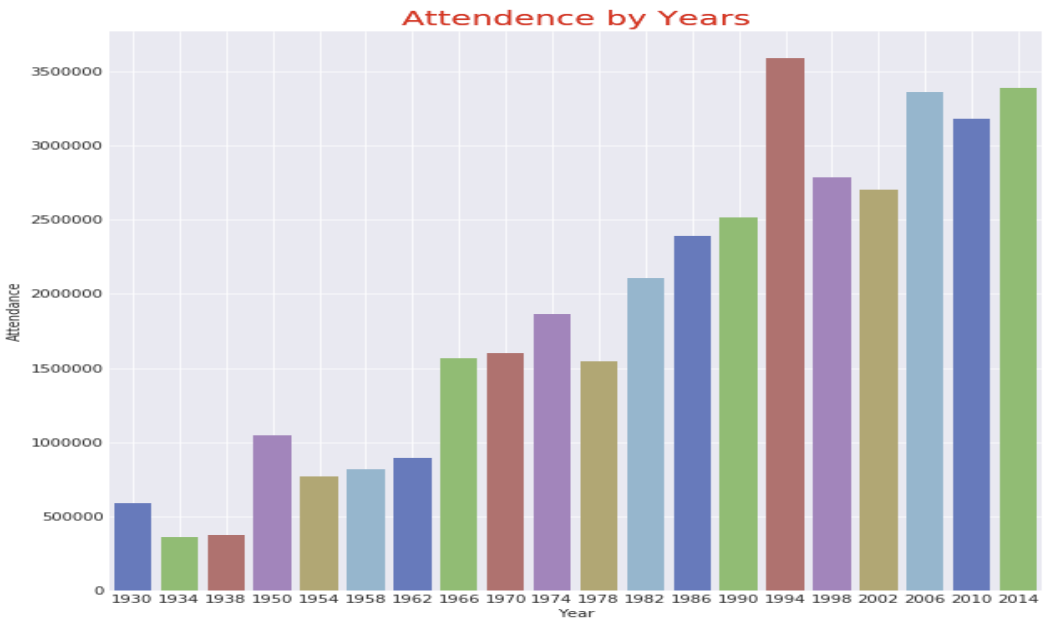
RoundID	MatchID	Coach Name
Line-up	Shirt Number	Player Name
Position	Event	Team Initials

The World Cup data set was the data set which was most beneficial for our prediction and model selection process. This dataset had 20 rows and 10 columns which had important information like the year the matches were played, and which team obtained their respective position and number of people attended the world cup that respective year. The following table describes the names of the 10 columns in this dataset.

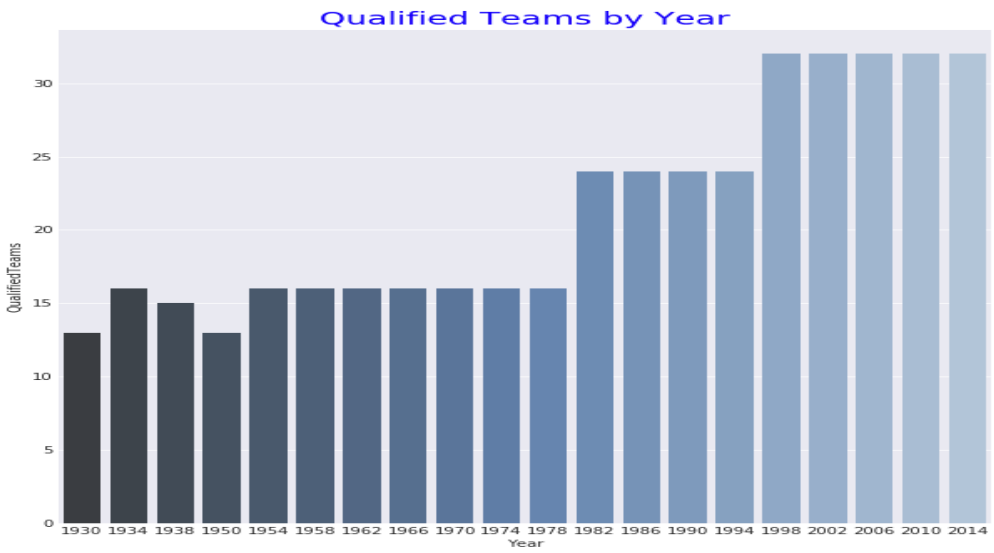
Year	Country
Winner	Runners-Up
Third	Fourth
Goals Scored	Qualified Teams
Matches Played	Attendance

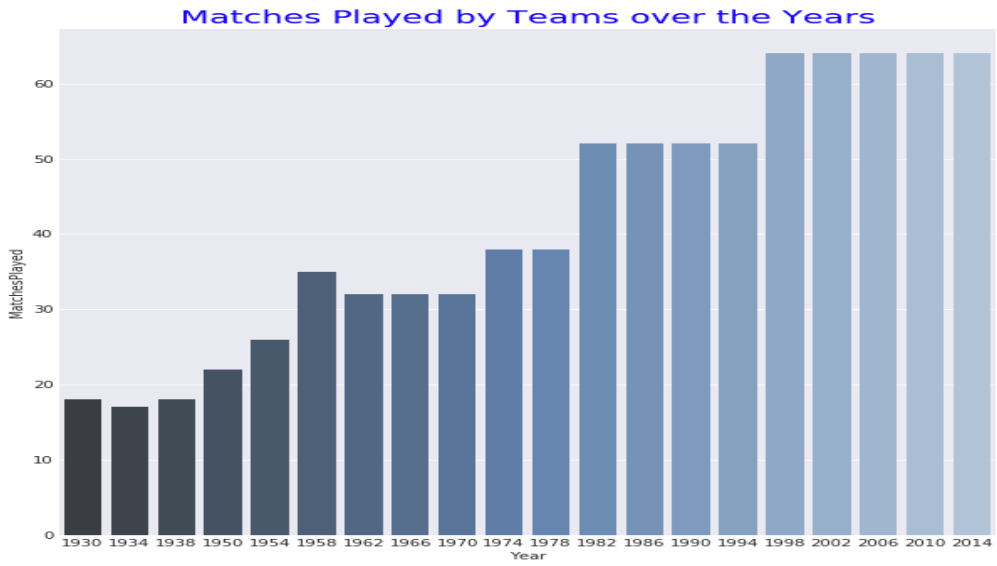
4. Data Visualizations

We did data visualizations on the data cumulating from the three data sets in order to assess the legibility and correctness of the data. Secondly, we wanted to see the general trends over the years which could help us in model building exercises and selecting the ideal model for making our prediction. Some of the visualization which made us reinforce the legibility and the correctness of the dataset are attached below along with their corresponding observations.

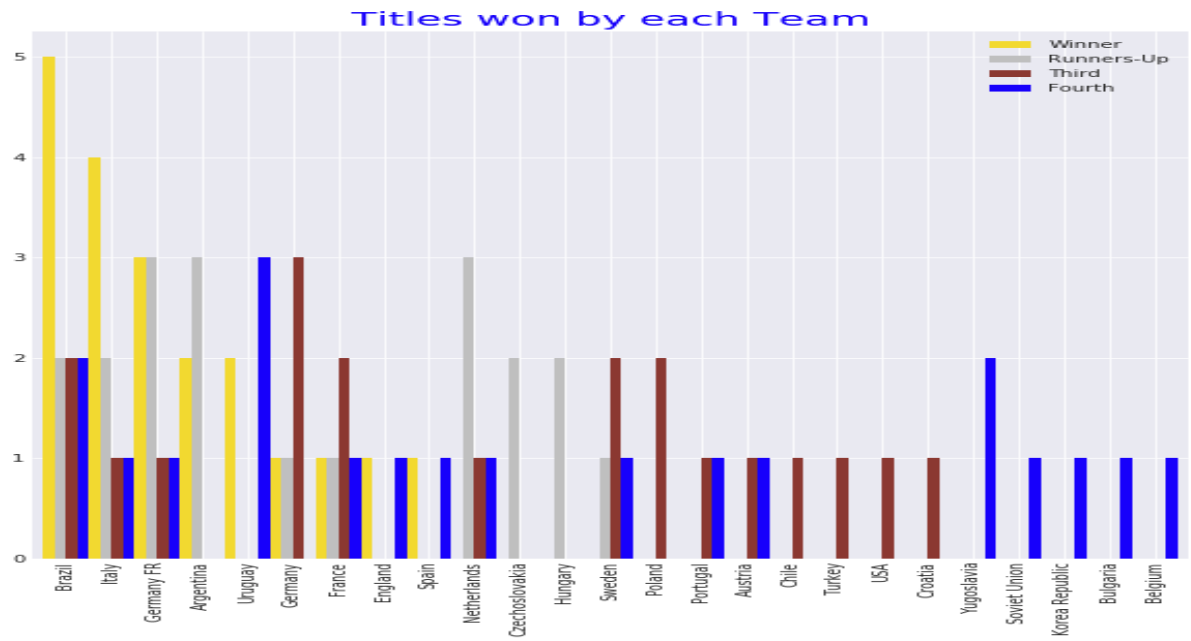


In the above visualization we saw the genral trend of increasing interest among people for the sport of FIFA in general. As, it can be seen easily from the trend there has been exponential increase in the attendance for the sport of football leading to more people attending the event. A total of 3.43 million people watched the 64 games of the 2014 FIFA World Cup in Brazil live in the stadium. This meant that the average attendance per game was 53,758, the highest average since the 1994 World Cup in the United States where average attendance is 68,991 per game. The championship has been awarded every four years since the inaugural tournament in 1930, except in 1942 and 1946 when it was not held because of the Second World War.





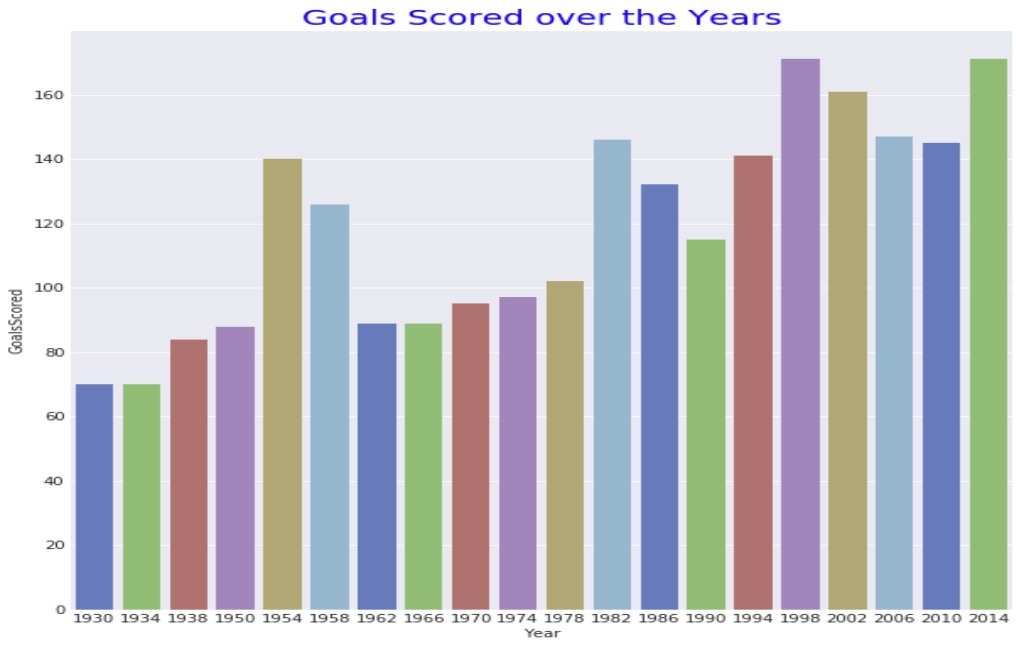
The above visualization depicts the graphs for the number of qualified teams for their respective years and number of matches played for the respective years. In the tournaments between 1934 and 1978, 16 teams competed in each tournament, except in 1938, when Austria was absorbed into Germany after qualifying, leaving the tournament with 15 teams, and in 1950, when India, Scotland, and Turkey withdrew, leaving the tournament with 13 teams. The tournament was expanded to 24 teams in 1982, and then to 32 in 1998, also allowing more teams from Africa, Asia and North America to take part.



The above visualization depicts the total number of titles won by each country over the period of 1930 to 2014. We observe and confirm that Brazil is the only country to have competed in every World Cup finals series. Moreover, it is the most successful country with five victories (1958, 1962, 1970, 1994 and 2002) and additionally they have also lost two World Cup finals in (1950 and 1998). Germany and Italy have won four world cups each. Germany was by far the most consistent team after 1950 as they not only won 4 world cups but also lost 4 world cup finals. Netherlands is the only team to have appeared in the final more than 2 times and failed to win the world cup trophy. They have lost three World Cup finals (1974, 1978 and 2010).

One of the major observations we made here was that countries like Germany had a different name in the history of it (Germany FR) and Russia was earlier called Soviet Union. So, one of the major benefits of visualizing this was that while model building for prediction we considered both these countries as the same.

Another interesting stat is that so far, only European or South American teams have won the World Cup. United States and Turkey are the only teams (outside Europe and South America) to have finished 3rd in world cups.



So far, during the Mundial of 2014 and the one in 1998, were scored the most goals, with 171 goals in 64 matches. The World Cup of 2002 is in the second place with 161 goals in 64 matches. Less goals were scored during the first two World Cups that have been organized. More specifically, in 1930 only 70 goals were scored in 18 matches and four years later, they were also scored 70 goals, but in 17 matches.

5. Model Building

Our goal for the second phase of this project was threefold. Firstly, we wanted to use Machine Learning to predict who is going to win the FIFA World Cup 2018. Secondly, we wanted to predict the outcome of individual matches for the entire competition. Lastly, we ran simulation of next matches i.e. quarter finals, semi-finals and finals. These goals present a unique real-world Machine Learning prediction problem and involve solving various Machine Learning tasks: data integration, featuring modelling and outcome prediction.

For a broader depth of knowledge, we decided to implement various machine learning techniques learned in class in order to compare the benefits and drawbacks of various techniques. To begin with, we decided to implement the following model building techniques on the FIFA World Cup dataset.

1. Logistic Regression (LR)
2. Support Vector Machine (SVM)
3. K-Nearest Neighbours (KNN)
4. Decision Tree
5. Random Forest (RF)
6. Bagging Classifier
7. Gradient Boosting (GB)
8. XGBoost
9. Naïve Bayes

I. Logistic Regression

Logistic Regression is the appropriate regression analysis to conduct when the dependent variable is dichotomous (binary). Like all regression analyses, the logistic regression is a predictive analysis. Logistic regression is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables. It returns a probability value which can be mapped to two or more classes.

When selecting the model for logistic regression analysis, another important consideration is the model fit. Adding independent variables to a logistic regression model will always increase the amount of variance explained in the log odds (typically expressed as R^2).

The loss function for the logistic regression can be given by the following expression

$$l(z) = -\log\left(\prod_{i}^{m} \mathbb{P}(y_i|z_i)\right)$$

The optimization algorithm we have used is Stochastic Average Gradient Descent. It is a breakthrough method in stochastic optimization. The stochastic gradient in SAGA is given by

$$\underbrace{g_{i_k}^{(k)}}_{X} - \underbrace{(g_{i_k}^{(k-1)} - \frac{1}{n}\sum_{i=1}^{n} g_i^{(k-1)})}_{Y}.$$

II. Support Vector Machine (SVM)

A Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyperplane. In other words, given labelled training data (supervised learning), the algorithm outputs an optimal hyperplane which categorizes new examples. In two-dimensional space this hyperplane is a line dividing a plane in two parts where in each class lay in either side. In our case we used the 'rbf' kernel with a regularization of 0.1.

Support vectors are data points that are closer to the hyperplane and influence the position and orientation of the hyperplane. Using these support vectors, we maximize the margin of the classifier. Deleting the support vectors will change the position of the hyperplane. These are the points that help us build our SVM. Loss function for the support vector machine is given by the following expression.

$$min_w \lambda \parallel w \parallel^2 + \sum_{i=1}^{n} (1 - y_i \langle x_i, w \rangle)_+$$

III. K-Nearest Neighbours (KNN)

KNN algorithm is one of the simplest classification algorithms and it is one of the most used learning algorithms. KNN is a non-parametric, lazy learning algorithm. Its purpose is to use a database in which the data points are separated into several classes to predict the classification of a new sample point. The distance in general is the Euclidean distance.

KNN Algorithm is based on feature similarity: How closely out-of-sample features resemble our training set determines how we classify a given data point. KNN can be used for classification — the output is a class membership (predicts a class — a discrete value). An object is classified by a majority vote of its neighbours, with the object being assigned to the class most common among its k nearest neighbours. It can also be used for regression — output is the value for the object (predicts continuous values). This value is the average (or median) of the values of its k nearest neighbours.

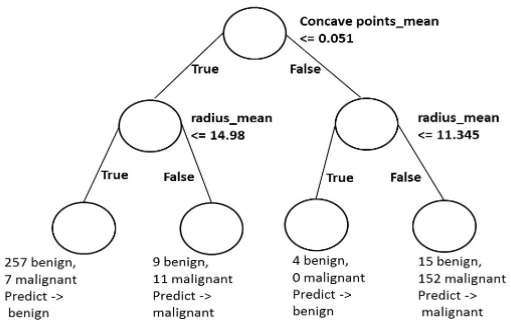
For multi-class *k*-NN classification, the upper bound error rate is given by

$$R^* \leq R_{kNN} \leq R^* \left(2 - \frac{MR^*}{M-1}\right)$$

IV. Decision Tree

The idea of a decision tree is to divide the data set into smaller data sets based on the descriptive features until you reach a small enough set that contains data points that fall under one label. Each feature of the data set becomes a root[parent] node, and the leaf[child] nodes represent the outcomes. The decision on which feature to split on is made based on resultant entropy reduction or information gain from the split.

However, Decision trees are likely to overfit noisy data. The probability of overfitting on noise increases as a tree gets deeper. Mechanisms such as pruning (not currently supported), setting the minimum number of samples required at a leaf node or setting the maximum depth of the tree are necessary to avoid this problem.



V. Random Forest

Random Forests grows many classification trees. To classify a new object from an input vector, put the input vector down each of the trees in the forest. Each tree gives a classification, and we say the tree "votes" for that class. The forest chooses the classification having the most votes (over all the trees in the forest). The forest error rate depends on two things: The correlation between any two trees in the forest. Increasing the correlation increases the forest error rate. The strength of each individual tree in the forest. A tree with a low error rate is a strong classifier. Increasing the strength of the individual trees decreases the forest error rate.

After training, predictions for unseen samples x' can be made by averaging the predictions from all the individual regression trees on x':

$$\hat{f} = \frac{1}{B}\sum_{b=1}^{B} f_b(x')$$

This bootstrapping procedure leads to better model performance because it decreases the variance of the model, without increasing the bias. This means that while the predictions of a single tree are highly sensitive to noise in its training set, the average of many trees is not, if the trees are not correlated. Simply training many trees on a single training set would give strongly correlated trees (or even the same tree many times, if the training algorithm is deterministic); bootstrap sampling is a way of de-correlating the trees by showing them different training sets.

VI. Bagging Classifier

A Bagging classifier is an ensemble meta-estimator that fits base classifiers each on random subsets of the original dataset and then aggregate their individual predictions (either by voting or by averaging) to form a final prediction. Such a meta-estimator can typically be used to reduce the variance of a black-box estimator (e.g., a decision tree), by introducing randomization into its construction procedure and then making an ensemble out of it.

Random forest is a better version of Bagging as it does not use this greedy algorithm and introduces very less covariance between the trees formed. Each classifier's training set is generated by randomly drawing, with replacement, N examples - where N is the size of the original training set; many of the original examples may be repeated in the resulting training set while others may be left out. Each individual classifier in the ensemble is generated with a different random sampling of the training set.

VII. Gradient Boosting

Gradient boosting is a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees.

In each stage, introduce a weak learner to compensate the shortcomings of existing weak learners. In Gradient Boosting, "shortcomings" are identified by gradients. Both high-weight data points and gradients tell us how to improve our model. So the boosting algorithm can be seen as a form of gradient descent that optimizes the squared error loss function, because in each step, it adds a sub-model that tries to mimic the negative gradient of this loss. (The learning rate η would be 1/2 in this case.) We have previously seen gradient descent in the context of linear models and neural networks: the difference here is that we now update the model by adding new sub-models, while previously we updated the model by changing the weights.

VIII. Naïve Bayes

Naive Bayes methods are a set of supervised learning algorithms based on applying Bayes' theorem with the "naive" assumption of conditional independence between every pair of features given the value of the class variable.

The Naive Bayes Classifier technique is based on the Bayesian theorem and is particularly suited when the dimensionality of the inputs is high. Despite its simplicity, Naive Bayes can often outperform more sophisticated classification methods.

Mathematically the Naïve Bayes classifier can be represented by

$$\hat{y} = \underset{k \in \{1,\dots,K\}}{\operatorname{argmax}} \; p(C_k) \prod_{i=1}^{n} p(x_i \mid C_k).$$

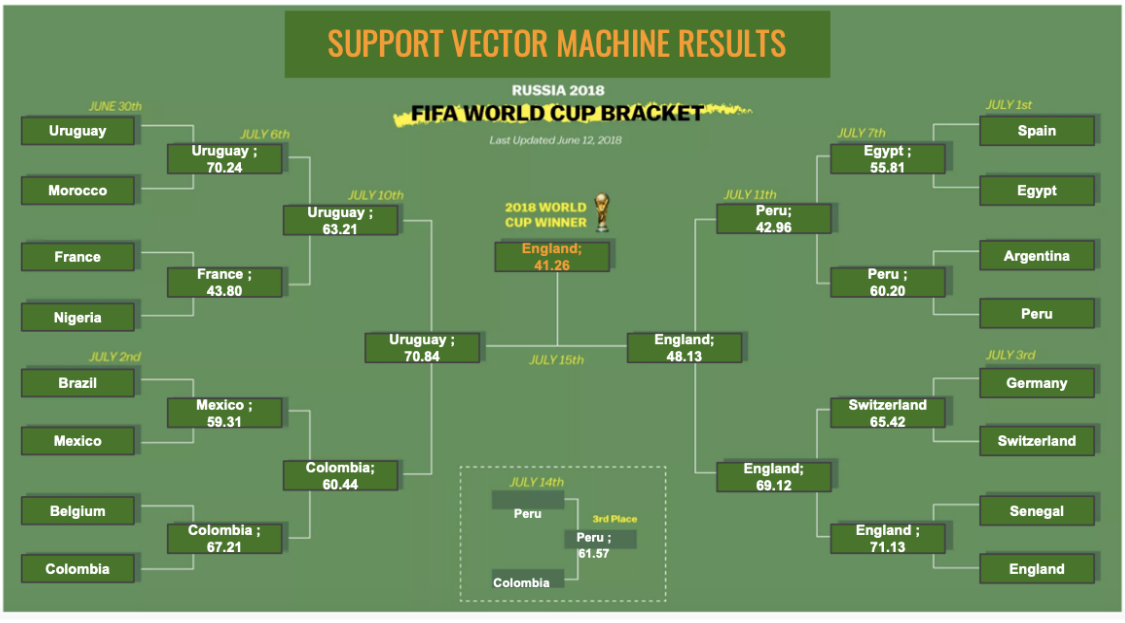
4. Results

We performed the above-mentioned models which we learned in class and calculated their test train accuracy based on the FIFA World Cup data sets. The results for these can be seen in the following table.

Model	Training Accuracy (%)	Testing Accuracy (%)
LR	62.41	54.68
SVM	**55.53**	**52.34**
KNN	66.94	52.73
Decision Tree	91.77	44.50
RF	91.77	44.92
Bagging Classifier	91.77	48.80
GB	75.16	50.78
XGBoost	67.11	56.25
Naive Bayes	62.08	57.03

We chose the SVM technique because it had the least overfitted results as compared to other modelling techniques. This can be seen by comparing the test and train accuracy to the closest difference.

The fixtures based on SVM technique can be seen in the following image



Cross-validation is primarily used in applied machine learning to estimate the skill of a machine learning model on unseen data. That is, to use a limited sample in order to estimate

how the model is expected to perform in general when used to make predictions on data not used during the training of the model.

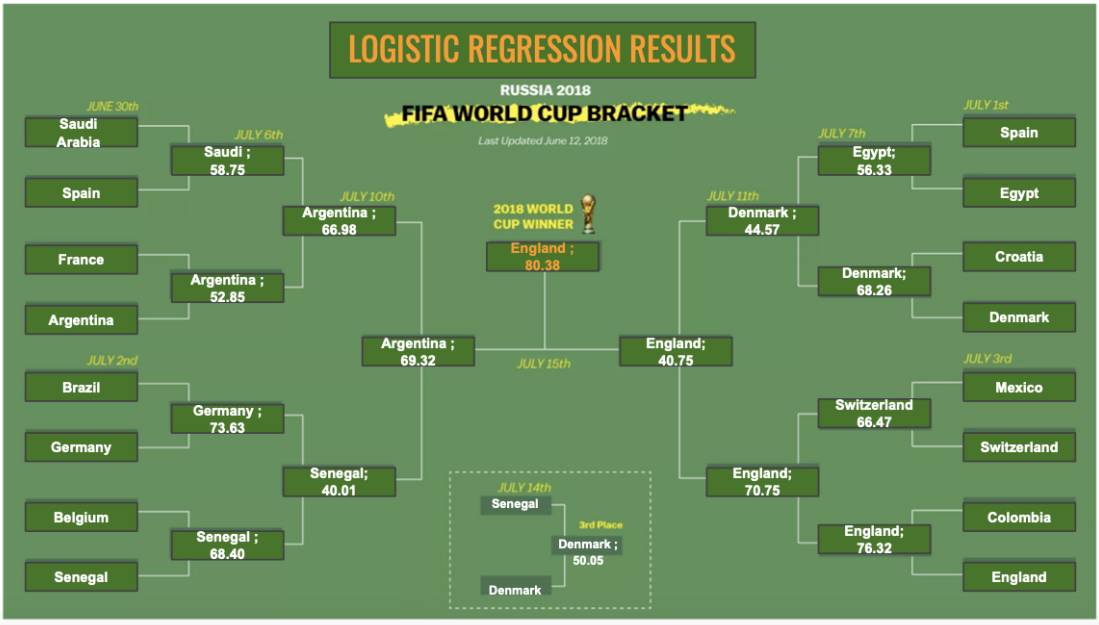
Importantly, each observation in the data sample is assigned to an individual group and stays in that group for the duration of the procedure. This means that each sample is given the opportunity to be used in the hold out set 1 time and used to train the model k-1 times. Due to this reason we decided to perform K-Fold Cross Validation on these models and compare them in order to get more accurate results.

The following are the results after K-Fold cross Validation.

Model	Training Accuracy (%)	Testing Accuracy (%)
LR	**60.26**	**60.00**
SVM	53.81	54.11
KNN	68.62	45.88
Decision Tree	90.61	40.58
RF	90.61	46.47
Bagging Classifier	90.61	45.88
GB	74.19	57.05
XGBoost	65.10	61.10
Naive Bayes	61.58	57.64

As it can be seen clearly that after performing K-fold Cross Validation, the most accurate and least overfitted model is that of Logistic regression. So, in order to compare the results of the two we performed the simulation for each stage again and found that winning probability of England has increased.

The simulation results after K-Fold cross Validation can be seen in the following fixture diagram.



5. Conclusion

In this project we tried to predict the winner of FIFA World Cup 2018. We tried two different techniques for splitting the data into training dataset and testing dataset. Firstly, using the train test function of sklearn we divided out train test split with ratio of 60% for training and 40% testing. However, the disadvantage of this technique is our model doesn't go through the entire dataset. Also, for better accuracy we require a good amount of dataset for testing too. The results obtained using this showed that most of the Machine learning techniques were 'overfitting'. In machine learning we always have a bias-variance trade off. To overcome this problem, we tried K-fold cross validation. In K-fold cross validation we divide our entire dataset in k chunks, and we train our model on k-1 chunks and validate on k^{th} chunk. We repeat this for k iterations. Due to this our model goes through the entire dataset and we get better results. This can be confirmed from the result section. Most of the models that were overfitting earlier gave better results.

6. References

1. https://www.kaggle.com/abecklas/fifa-world-cup
2. https://blog.goodaudience.com/predicting-fifa-world-cup-2018-using-machine-learning-dc07ad8dd576
3. http://personal.psu.edu/jol2/course/stat557/material.html
4. http://storm.cis.fordham.edu/%7Egweiss/papers/data-mining-chapter-2010.pdf
5. https://statathlon.com/an-in-depth-analysis-for-world-cups/