UFC Fight Result Prediction

Laxman Kumar
School of Information Studies
Syracuse University

Saheb Singh
School of Information Studies
Syracuse University

Abhiraj Singh
School of Information Studies
Syracuse University

Abstract
UFC events are conducted worldwide, and it also promotes the fighters. Building a system which can predict the results of the fight based on the fighter statistics and the past fight results can surely change the way the game is played. To accomplish this goal, we used the past 17 year data to analyze the data and tried to predict the winner using different machine learning models. We had also used deep learning which is one of the most advanced methodology used to making prediction in more complex scenario.

I. INTRODUCTION

Sports betting is a $155 billion industry. Fighting ranks among the top in the industry, and the Ultimate Fighting Championship (UFC) is currently taking steps to push it even further. Mixed Martial Arts (MMA) fighter statistics involve everything from skill centric values such as wins, and significant strikes landed to physiological measurements such as height and reach. There are over one hundred different features up to analyze before any given fight, and machine learning can be used to best understand which are most relevant, and to indent trends and predict the outcomes (win/draw/loss) of each fight. Mixed martial arts are examples of full-contact combat sport that allows grappling and striking while both standing on the ground. UFC (Ultimate fighting championship) is an American martial art that is mixed with different kinds of martial arts organizations based in Nevada and Las Vegas. UFC is the largest MMA promotion in the world and it also features top rank fighters of the martial arts. UFC produces events worldwide that showcase fights in the 12-weight division and follows the rules of Unified rules of Martial Arts. This is a highly unpredictable sport since sometimes new fighters can win from year-old champions.

The goal of this study is to explore our ability to predict the outcome of UFC fights based on each match's pre-fight statistics using machine learning models. An accurate prediction model could both inform the best-placed bets (and potential risk associated) for each fight, but also could provide insight to coaches when accepting fights, to begin with, simply by looking at the opponent's statistics relative to their fighter. It could also be used to help to identify which features are most significant in this prediction.

II. DATA DESCRIPTION

The dataset which we have used in this project is taken from Kaggle and the size is 10.74 MB. There are over one hundred different fighter statistics on UFC Stats for each of the fights in UFC record from 1993 to 2019, which include information such as fighters' height, weight, reach, and stance, as well as statistics such as win streaks, strike percentage, guard passes and strikes landed by location. The dataset used in this study was scraped from UFCStats and statistics

pertaining to the circumstances of the fight (i.e., location, number of allocated rounds, etc.) were removed as they were deemed irrelevant and did not align with the fighter-centric intention of the study.

From figure 1, UFC has conducted most events in the year 2014 with a total of 503 events. The total number of events got decreased in the year 2015 with only 473 events. When UFC first started conducting events, the total number of events conducted were around 50 events per year. But it started to boom after the year 2006 when UFC first crossed 100 events per year and since then it started to grow and reached a value of more than 500 events per year in 2014.

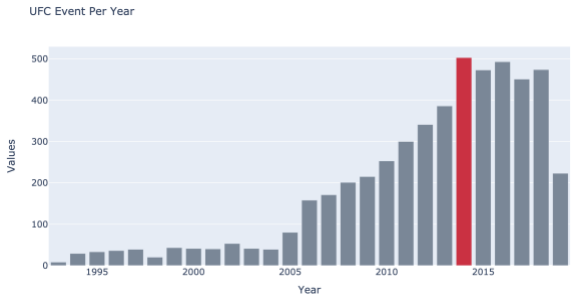

Figure 1 Total events conducted by UFC

events by country. UFC has conducted most events in the USA with 3392 total events. UFC is based in Los Angeles and Nevada and promotes American Martial Arts and therefore has most events in the USA. It does conduct events worldwide, but the number of events conducted in other countries is around 10% of the total events conducted in the USA.

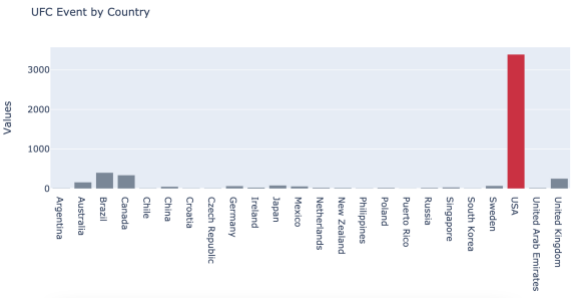

Figure 2 Total events by country

UFC has a very large number of fighters who are associated with the company. If we divide the fighters based on gender, there is a huge difference. There are only 6.1% of female fighters in comparison with the male fighters with 93.6%.

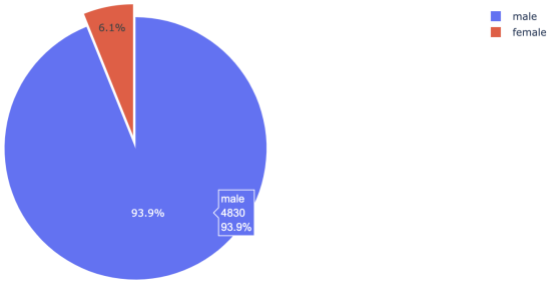

Figure 3 Total fighters by Gender

All the UFC fights conducted have two corners red and blue. And there is the relevance of fighters from which corner will win. If we see the distribution in figure 4, fighters fighting from the blue corner have won 67.5% of events while only 30.9% of total events have been won by the red corner. There is only 1.61% of the events were drawn.

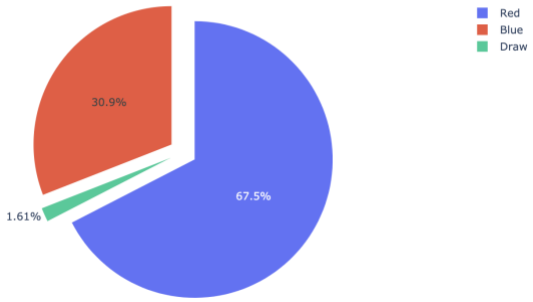

Figure 4 Win distribution by Team

III. APPROACH/ALGORITHMS

This section contains the methodology which we implemented for the project in order to attain quality results. For any machine learning tasks to work effectively, we must modify the data appropriately.

Preparation of the data was carried out before implementing machine learning techniques in the following manner. The preprocessing steps which were taken are:

3.1) Removal of null values
3.2) Data encoding
3.3) Data standardization

Removal of null values

Firstly, the dataset was consisting of null values. In order to achieve better results and improvised efficiency, we must remove the null values from the dataset. The columns of height and reach were converted from inches to centimeters. We have replaced the string terms present with space so that it does not affect the prediction. For instance, in the weight column, we have removed. lbs.

Data Encoding

Since some of the features were categorical, we must convert them into numerical. To achieve this, we need to perform data encoding. Data encoding helps to convert categorical data into numerical data. This is helpful for the machine learning models to learn the hidden pattern that might be present in the categorical variables. Therefore, this is a very crucial step.

Data Standardization

Different features present in the dataset have different scales. The machine learning model will get affected due to the large/small magnitude values present in the features. So, standardization of the data comes into the picture, using the standardization we can bring different features to the same scale. This will avoid the machine learning models to get influenced by the large/small magnitude.

The next section will provide a brief introduction to the various models which are being implemented for the required task. The results of each machine learning model will be discussed in the results section.

SVM Classifier

Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible.

For the SVM-C to work effectively we need to provide hyperparameter tuning. Therefore, we have tuned 'C' parameter. The C parameter is helpful to provide the penalty for the misclassified data point.

Logistic Regression

Logistic regression is a statistical model that in its basic form uses a logistic function to model a binary dependent variable. It is a predictive algorithm using independent variables to predict the dependent variable, just like Linear Regression, but with a difference that the dependent variable should be a categorical variable.

For this project, we have fine-tuned the logistic regression using the following hyperparameter. C is a hyperparameter that is helpful in controlling the penalty we need to impose for misclassification. Max_iter is another hyperparameter that provides the total number of iterations. Tuning these hyperparameters we were able to get the best logistic regression model which is discussed in the result section.

Naïve Bayes

Naive Bayes algorithm is a classification technique based on Bayes' Theorem with an assumption of independence among predictors. In simple terms, a Naïve Bayes

classifier assumes that the presence of a feature in a class is unrelated to the presence of any other feature. To get the best performing Naïve Bayes model, we performed hyperparameter tuning using the grid search. 'Alpha' is the hyperparameter which we have tuned appropriately in order to achieve the best model. It is helpful to achieve smoothing.

Random Forest Classifier

The random forest is a model made up of many decision trees. Rather than just simply averaging the prediction of trees (which we could call a "forest"), this model uses two key concepts that give it the name *random*: Random sampling of training data points when building trees. Random subsets of features considered when splitting nodes.

For the random forest to work appropriately, we performed hyperparameter tuning. Here, we have fine-tuned n_estimators. N_estimators is basically the number of trees in the forest. Max_depth is helpful for providing the number of levels in the decision tree. We have also fine-tuned max_features which allows the random forest to get the max number of features considered for splitting a node. Min_sample_split is helpful to get the minimum number of the data points which are placed in the node before the node is split. Min_sample_leaf provides the data points allowed in the leaf node.

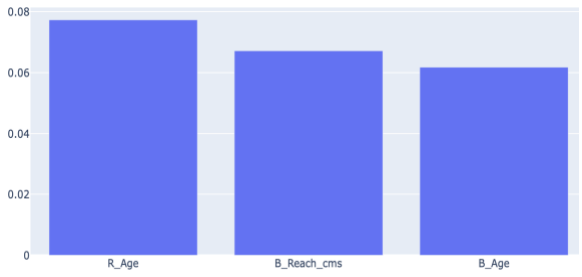

Figure 5 Feature Importance plot

Gradient Boosting Classifier

Gradient Boosting Classifier builds the model in a stage-wise fashion as other boosting methods do, and it generalizes them by allowing optimization of an arbitrary differentiable loss function.

For the Gradient boosting classifier to work at its best, we have fine-tuned the following hyperparameter. Min_samplesplit is useful to get the minimum number of the observations which the algorithm considers for performing the split. Min_samples_leaf provides the minimum samples which the leaf or the terminal node requires. It is helpful to control the overfitting of the model. Max_depth is also important for controlling the overfitting of the model. It is helpful to decide the depth of the tree. Max leaf_node is helpful to decide the maximum number of terminal leaves which the tree will contain.

Extra Tree Classifier

Extra Trees is an ensemble machine learning algorithm that combines the predictions from many decision trees. It is related to the widely used random forest algorithm. It can often achieve as good or better performance than the random forest algorithm, although it uses a simpler algorithm to construct the decision trees used as members of the ensemble. We have fine-tuned this model using the following features. N_estimator, max_features, max_depth, min_sample_leaf and min_sample_split. These hyperparameters are explained in the section random forest.

K-Means

K-Means is a clustering algorithm that divides observations into k clusters. Since we can dictate the number of clusters, it can be easily used in a classification where we divide data into clusters that can be equal to or more than the number of classes.

To achieve the best model in k-means, we have considered having 2 clusters since this is a binary classification problem.

Deep Neural Network

Artificial Neural Networks (ANN) are multi-layer fully connected neural nets. They consist of an input layer, multiple hidden layers, and an output layer. Every node in one layer is connected to every other node in the next layer. We can make the network deeper by increasing the number of hidden layers.

During the implementation of the network, we usually perform scaling since it mostly boosts the overall performance of the algorithm. Then we have trained the model using appropriate epochs, neurons, and activation function. We have utilized dropout in order to avoid overfitting. The loss function is binary cross-entropy because it is a binary classification problem.

IV. RESULTS

In this section we will discuss the results we have obtained using the above machine learning tasks which were implemented on the preprocessed dataset. The evaluation metrics which we have considered for the evaluation are:

4.1) Accuracy

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

4.2) Recall

$$\textbf{Recall} = \frac{\text{True Positive}}{\text{Predicted Results}} \quad \text{or} \quad \frac{\text{True Positive}}{\text{True Positive + False Negative}}$$

Model Name	Accuracy	Recall
SVM-C	68.88	86.10
Logistic Regression	69.53	72
Naïve Bayes	69	70
Random Forest	70.11	96
Gradient Boosting	70.66	95
Extra Tree Classifier	69.88	96
K-mean	53.16	65
Deep Neural Network	71.04	94

Figure 6 Results

Based on the above results we can clearly see that in terms of accuracy Deep Neural Network were able to perform best relatively, when compared to other learning models. Based on the above results we can see that the Random Forest and Extra tree classifier are performing best in terms of recall, when compared to other learning models. K-mean has performed the worst in terms of accuracy and recall as it provides an accuracy of only 53.16 and recall of 65. Naïve bayes classifier which performs fastest amongst the above algorithm provides a decent evaluation score of 69 and 70 as accuracy and recall respectively. Logistic regression able to perform this task with an accuracy and recall of 69.53 and 72 respectively. SVM-C able to provide the score of 68.88 and 86.10 for accuracy and recall respectively.

V. CONCLUSION

One of the biggest challenges we faced during our model building and evaluation was able to merge three datasets and processed the merged dataset for model building. During pre-processing we faced problem with feature engineering due to having more than 100 features. After

selecting most relevant features we used dummy variables to encode our target variable, we removed draws as they were only about 2 percent of the total dataset. For modeling we applied ensemble models such as Random Forest, Gradient Boosting, and Extra Tree classifier, for classification we applied Support Vector Machine, and Naive Bayes Classification. We also ran sequential three hidden layers neural network. After running models on pre-processed data we were able to get the highest accuracy of 71 percent from a 3 layer neural network, which is an improvement over the base accuracy taken of the UFC prediction model which is able to correctly predict the winner 57% of the time. After analyzing different models and applying Apriori Algorithm to get the most important features for the winner we can say that Age, Winning Streak, and reach of the fighter are the most important characteristics and must be taken into consideration while predicting the outcome of the match. A person can use these characteristics to place a bet.

VI. AKNOWLEDGMENT

We would like to thank our professor Dr. Ying Lin for being excellent instructor who taught us all the different models and gave us knowledge about analyzing the result of those models without which we would not have been able to complete this project. He also guided us through the entire project by providing complete instructions on the ways to approach this project.

VII. REFERENCES

1. https://www.kaggle.com/rajeevw/ufcdata
2. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html
3. https://scikit-learn.org/stable/search.html?q=random+forest
4. https://dash.plotly.com/
5. https://seaborn.pydata.org/tutorial.html

DASH LINK
https://ufcfight-da.herokuapp.com/