UFC fight winner prediction

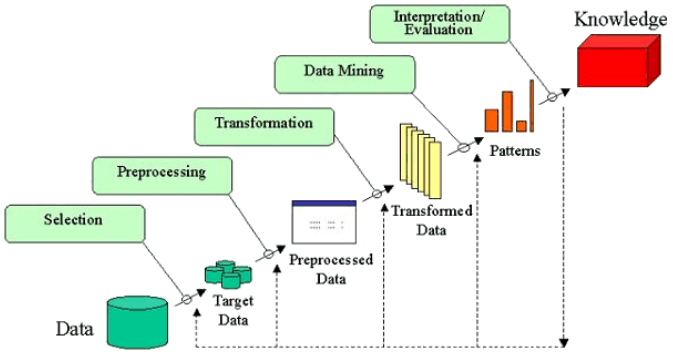
Motivation

For this project we decided to dive into the UFC universe. More specifically, we focused on predicting who the winner of an upcoming UFC match will be. Lately, the sport has received a major inflow of popularity and interest from the public, which naturally results to an increase of the amount of money being invested both for events and promotions. Moreover, more and more betting companies have begun to operate in this field, due to the number of people that are willing to bet on who the winner will be and various other types of bets. Hence, apart from the general interest of predicting the winner of a fight based on statistics and examining if UFC organizes fights in such a way that a fighter would be benefited, this project could potentially be of use to betting companies.

In order to carry out these predictions, we collected data from Kaggle, which are originally sourced from the official UFC api. The data initially included every fight that took place from 1993 to 2019. The UFC has undergone many changes during that time. In the beginning, the sport had a small number of rules, until 2001 when the franchise was sold and was converted to a highly organized, sanctioned and controlled combat sport. In 2013, the digital side of UFC launched, when pay per view events and other subscription services debuted, vastly increasing the popularity of the sport. Throughout the years new rules have been applied, mainly for safety reasons. Moreover, the sport has one of the most thorough anti-doping policies since the athletes are randomly tested for the use of substances, so that the high quality of the matches is ensured.

Data used and process followed

The data we used start from 2013 onwards, since the digital debut of the sport resulted to an increase of statistical data collected. Also, the data from 2013-2019 are more interpretable and solid, since there are very few to none missing values overall and a broad range of self-explainable statistical features for each match. During this period, a total of 3592 fights took place. This is a large enough sample to make some observations and try to make a prediction of the winner of a fight. The KDD process will be implemented in order to make predictions, illustrated in the following image.



The data selection refers to the gathering of data, relevant to our business question. In our case, the data selected are the data from 2013-2019 for the reasons that were previously explained. We then have to preprocess our data. In that stage, we dealt with the missing values, outliers, implemented standardization and engineered features that are useful for prediction-making. In order to apply machine learning models, we had to transform our data to format acceptable as input for the implemented algorithms. In order to do this, we used label encoding to transform the categorical variables. We then proceeded on testing different algorithms and evaluating them. To prevent overfitting and simultaneously discover the best set of parameters, we used GridSearchCV which runs the models using different parameters each time, chosen from an a priori set of parameters, and performs a 5-fold cross validation in order to evaluate overfitting. Finally, we run the algorithms with the recommended hyperparameter settings, evaluate and compare the results, using F1 score, confusion matrices and ROC-AUC. In the end, we conclude by examining if the result was expected, and whether we could use this model in real cases.

Challenges

During the process we encountered several challenges. Firstly, we had to preprocess the data and make sure everything is in order. We initially tried to get a grasp of the data by using EDA and then proceeded to check whether we have missing values. The dataset was well-documented and therefore, we did not have to deal with missing values. In case we had to, we would try to fill the missing values by checking if another fight of the same fighter is available in the dataset and would replicate the values that would not change. However, for missing values regarding the blows landed or attempted during a fight, it would not be possible to fill them and hence we would have to remove the relevant row.

We then conducted feature engineering in order to reduce the data dimensionality and create features that can be important for predicting the fight's winner. These features mainly regard the conversion of attempted and landed strikes or takedowns to a single feature which demonstrates the percentage of successful strikes or takedowns a fighter managed to achieve in a fight. This percentage is given by the division of landed by attempted strikes. Since most numbers were float and had many decimals, we had to round the whole dataset to 2 decimal digits.

In order to remove outliers and standardize the data, as well as proceed on making predictions, we have to split the dataset to test and train sets. The ratio we chose was 75-25%, 75% for training and 25% for testing. The main reason for splitting the dataset before removing outliers and standardizing is that we do not want any information from the test set leaking into the training set. Otherwise we could have the impression that our model performs well but in reality it might overfit.

In order to deal with outliers, we implemented LOF on the training set. LOF is an outlier removal technique that marks an observation as an outlier by checking the distance from its k nearest neighbors. We used this approach because our data has a very high dimensionality and usual outlier removal techniques like IQR would lead to poor results. We then standardized the data by implementing StandardScaler, a method that converts numerical variables so that they follow a normal distribution of a mean equal to 0 and standard deviation of 1.

While tree classifiers like Random Forest, Decision Trees and Gradient Boost can handle both categorical and numerical data as outputs, SVM cannot and therefore we used LabelEncoding in order to convert the categorical output (Red/Blue) to 1/0.

Since the data is transformed and everything is in order, we proceeded by applying our models. We used Random Forest, Decision Trees, Gradient Boost and SVM, all known to be able to handle data with high dimensionality and complex relationships, in order to predict the winner of a fight. We used all the features that were included in the dataset after the preprocessing stage.

To test for overfitting and optimize the hyperparameters of each model, GridSearchCV was implemented. This method runs models with a different set of hyperparameters each time, while simultaneously using k-fold cross validation to test for overfitting. k was set to 5, which is enough to capture any overfitting trends. Wherever possible, the best parameters were set and the model was run. For SVM it required a huge computational time and power and therefore, the parameters were set as those suggested by the scikit learn library.

To evaluate the models, we used Confusion Matrices, F1 score, and ROC-AUC, all known to be effective for model evaluation. F1 score and ROC-AUC are the best among the three, with ROC-AUC being the most effective, since it also captures the True Negatives, which F1 score tends to ignore. The ROC curve was plotted for each model.

Finally, a bar plot displaying the feature importance for each model was plotted so that we could get a better grasp of the features that affect the prediction of the winner in a UFC match.

Findings

All models presented a similar performance, with an average F1-score of around 75% and AUC score of 52.5%.

Model	F1-score	AUC score
Random Forrest	79.61	52.18
Gradient Boost	79.97	54.64
Decision Tree	70.81	54.32
SVM	64.70	50.91

SVM seems to have the poorest performance, which is expected since SVM requires a lot of data in order to train properly. Moreover, SVM has a hard time separating between two classes if they are similar. In our case, the tree models perform better, with Gradient Boost topping the charts. Both the AUC and F1-score for the tree models are similar. Tree models can perform better than SVM with less data available. However, trees have a deterministic nature, which can lead to the presence of challenges when two classes are very similar. In our case, both classes are similar, since both Red and Blue corner can include the same fighter, in different fights, and present very similar features. Therefore, the distinction is very difficult, even for a human. The models achieve an AUC score of 55% which is not that good, but better than random guessing. A more thorough analysis is presented in the notebook.

All the models seem to agree on the most important features. They consider *avg_guard_passed* and *total_rounds_fought* as well as submission attempts and win/lose streaks to be the most important factors to predict who the winner will be. A human might select similar features in order to predict the outcome of a fight, therefore, our models are efficient in terms of feature importance.

The feature importance can be further examined based on the graphs that are presented in the notebook.

Results and Future improvements

From these results we can conclude that there is no profound evidence that the fights are favoring a certain fighter or fighting corner. Moreover, we can conclude that it is very difficult to predict the winner of a fight, not only because of the unpredictable nature of sports but also because of the similarity of the two classes and the constant swap of corners throughout a fighter's career. Therefore, the model even though it performs better than random guessing, cannot be used in the industry yet. Maybe, with the collection of more data, we will be able to get better results. However, some important insights regarding the features that affect the outcome of a fight were observed in this project.

Finally, this work, like any other machine learning project, can further be improved. The collection of more data would be crucial for the development of models that can capture complex relationships and require a big volume of data in order to be trained. Such models are Neural Networks, SVM, etc. The limitation of the available data from 2013-2019 only is not providing enough training power to our models. Moreover, one important factor that affects the outcome is that we considered all the fights for all the weight divisions in this project. If we had enough data to split the dataset into subsets containing separate fights for each weight division, we could surely achieve better results.

Another improvement that can be made with the data available at the moment is the introduction of hybrid models, models that consist of different algorithms and decide on the prediction by majority vote. Apart from that, we could better optimize the hyperparameters of our algorithms by trying out different sets and more parameter combinations.

To conclude, even though the results are not really satisfactory, with the collection of more data and the implementation of more complex models we could better predict the winner of a UFC fight in the future.