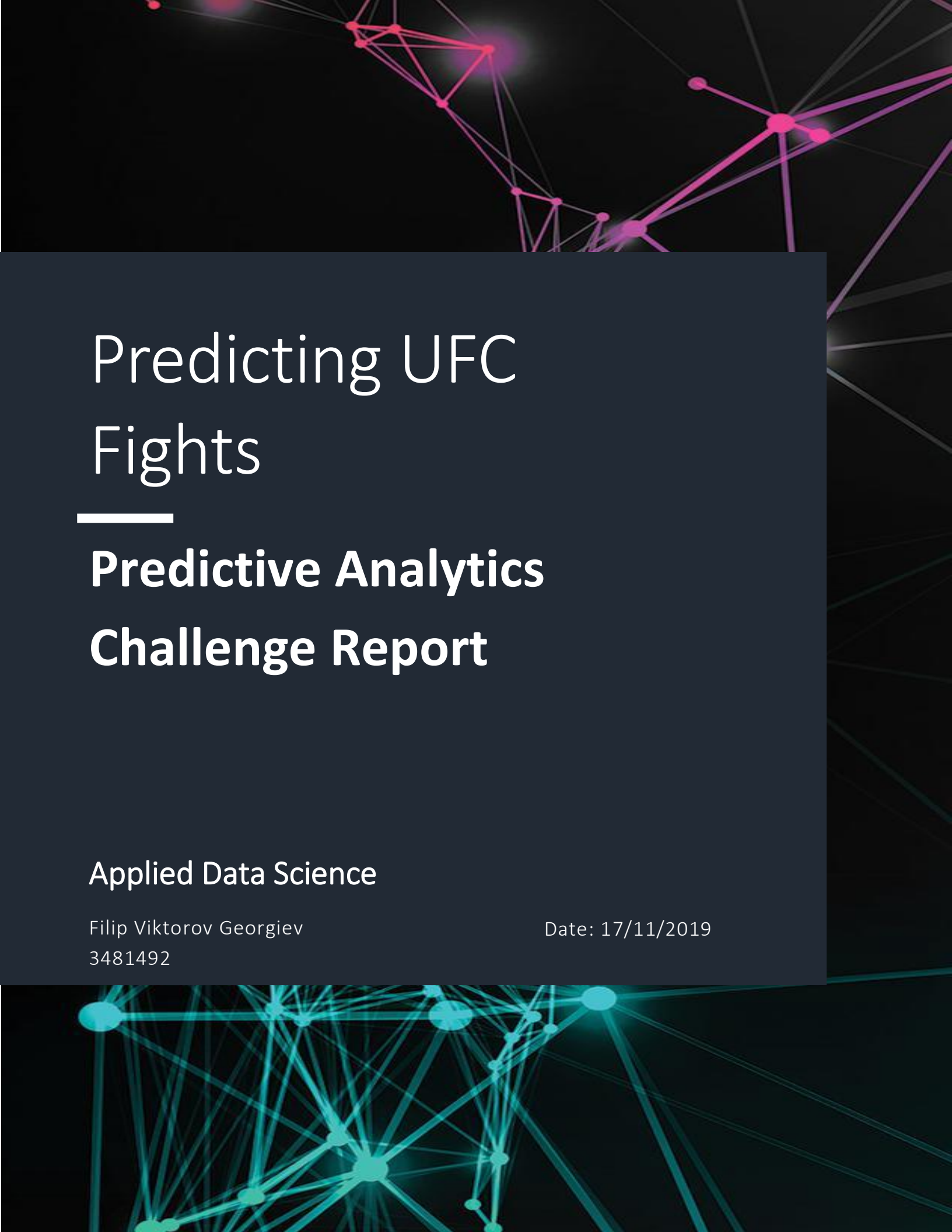

Predicting UFC Fights

Predictive Analytics Challenge Report

Applied Data Science

Filip Viktorov Georgiev
3481492

Date: 17/11/2019

Contents

I. Introduction

The purpose of this report is to describe the outcome of the predictive analytics done on the "UFC-Fight historical data from 1993 to 2019" dataset found on Kaggle[1]. This dataset claims to have extensive information about every fight in the age span mentioned in its title, which means that, perhaps, with the help of machine learning and data analysis it would be possible to predict who the winner of any match would be. The aim of this project is to find out if this is indeed achievable, as if the answer is yes and the predictions made are precise, this may mean that there is an extremely good investing possibility for any party that would be part of this project.

The reasoning behind this project is the pure interest I had in finding out whether it was possible to predict a sport, that is often categorized as the most "Unpredictable". The appeal of doing the impossible was definitely a good motivator for me.

The report will go over the data, analysis and methods used. The final results will also be shown together with a conclusion and recommendations. This document will have an Appendix, which will contain the business proposal and the reproducible research.

The main conclusion reached is that a prediction is possible, but its accuracy being 68% means that it is unreliable. Furthermore, the amount of data, research method and low variance further strengthen this statement.

II. Data

The dataset contains 5144 rows of data. Each of them contains a single match between two fighters. This means that you can find extensive data about both fighters. Their names, the date, location and referee of the match are the first columns. The winner of the match is also specified in a column. This column contains either Red, Blue or Draw, which means that either the fighter in the red corner(R_fighter) or the fighter in the blue corner(B_fighter) won. Of course, a draw is also possible. The dataset specifies whether the match was a title bout, in which weight class it was fought and in how many rounds. All of this can be seen in the picture below. (Figure 1 Base data)

[1] https://www.kaggle.com/rajeevw/ufcdata

```
RangeIndex: 5144 entries, 0 to 5143
Data columns (total 143 columns):
R_fighter                       5144 non-null object
B_fighter                       5144 non-null object
Referee                         5121 non-null object
date                            5144 non-null object
location                        5144 non-null object
Winner                          5144 non-null object
title_bout                      5144 non-null bool
weight_class                    5144 non-null object
no_of_rounds                    5144 non-null int64
```

Figure 1 Base data

Afterwards, you can find 67 columns containing specific fighter characteristics per fighter.(Figure 2 Extensive data) These characteristics include amount of wins and losses, longest and current win and lose streaks, amount of time spent in the ring (in rounds and in seconds), types of win (Knock out/Knockdown/Decision etc.), stance, age, height, reach and weight of fighter. What is more, the average of different fight techniques (Kicks, Punches, Ground, Reversal) received, attempted and landed is also documented.

```
R_current_lose_streak          5144 non-null float64
R_current_win_streak           5144 non-null float64
R_avg_BODY_att                 4494 non-null float64
R_avg_BODY_landed              4494 non-null float64
R_avg_CLINCH_att               4494 non-null float64
R_avg_CLINCH_landed            4494 non-null float64
R_avg_DISTANCE_att             4494 non-null float64
R_avg_DISTANCE_landed          4494 non-null float64
R_avg_GROUND_att               4494 non-null float64
R_avg_GROUND_landed            4494 non-null float64
R_avg_HEAD_att                 4494 non-null float64
R_avg_HEAD_landed              4494 non-null float64
R_avg_KD                       4494 non-null float64
R_avg_LEG_att                  4494 non-null float64
R_avg_LEG_landed               4494 non-null float64
R_avg_PASS                     4494 non-null float64
R_avg_REV                      4494 non-null float64
R_avg_SIG_STR_att              4494 non-null float64
R_avg_SIG_STR_landed           4494 non-null float64
R_avg_SIG_STR_pct              4494 non-null float64
R_avg_SUB_ATT                  4494 non-null float64
R_avg_TD_att                   4494 non-null float64
R_avg_TD_landed                4494 non-null float64
R_avg_TD_pct                   4494 non-null float64
R_avg_TOTAL_STR_att            4494 non-null float64
R_avg_TOTAL_STR_landed         4494 non-null float64
R_longest_win_streak           5144 non-null float64
R_losses                       5144 non-null float64
R_avg_opp_BODY_att             4494 non-null float64
R_avg_opp_BODY_landed          4494 non-null float64
R_avg_opp_CLINCH_att           4494 non-null float64
R_avg_opp_CLINCH_landed        4494 non-null float64
R_avg_opp_DISTANCE_att         4494 non-null float64
R_avg_opp_DISTANCE_landed      4494 non-null float64
R_avg_opp_GROUND_att           4494 non-null float64
R_avg_opp_GROUND_landed        4494 non-null float64
R_avg_opp_HEAD_att             4494 non-null float64
R_avg_opp_HEAD_landed          4494 non-null float64
R_avg_opp_KD                   4494 non-null float64
R_avg_opp_LEG_att              4494 non-null float64
R_avg_opp_LEG_landed           4494 non-null float64
R_avg_opp_PASS                 4494 non-null float64
R_avg_opp_REV                  4494 non-null float64
R_avg_opp_SIG_STR_att          4494 non-null float64
R_avg_opp_SIG_STR_landed       4494 non-null float64
R_avg_opp_SIG_STR_pct          4494 non-null float64
R_avg_opp_SUB_ATT              4494 non-null float64
R_avg_opp_TD_att               4494 non-null float64
R_avg_opp_TD_landed            4494 non-null float64
R_avg_opp_TD_pct               4494 non-null float64
R_avg_opp_TOTAL_STR_att        4494 non-null float64
R_avg_opp_TOTAL_STR_landed     4494 non-null float64
R_total_rounds_fought          5144 non-null float64
R_total_time_fought(seconds)   4494 non-null float64
R_total_title_bouts            5144 non-null float64
R_win_by_Decision_Majority     5144 non-null float64
R_win_by_Decision_Split        5144 non-null float64
R_win_by_Decision_Unanimous    5144 non-null float64
R_win_by_KO/TKO                5144 non-null float64
R_win_by_Submission            5144 non-null float64
R_win_by_TKO_Doctor_Stoppage   5144 non-null float64
R_wins                         5144 non-null float64
R_Stance                       5010 non-null object
R_Height_cms                   5140 non-null float64
R_Reach_cms                    4828 non-null float64
R_Weight_lbs                   5141 non-null float64
B_age                          4972 non-null float64
R_age                          5080 non-null float64
```

Figure 2 Extensive data

I only supply the data for the red fighter here, as the data for the blue fighter has the same structure.

III. Methods

Thankfully the data was already semi-processed so only a few steps in data cleaning had to be taken. Firstly, I found out that the columns for both fighters that indicated how many draws they had received in their careers had only zeroes. I decided to remove both columns as there seemed to be no logical explanation behind this problem. Afterwards, I filled in all the missing values, which you can see in Figure 2 Extensive data with their respective medians. The rationale behind this was that most columns were averages anyway, so if I were to fill them with the average of the averages this would be as close to the real data as possible. The column containing stances, which was not numerical also had missing data, so I made the assumption that these fighters fought “Orthodox” as this is the most common fighting stance. It was my intention to fill in all missing data instead of removing the rows containing it, due to the fact that I did not have a large amount of data in the first place and some of these columns would not be used anyway.

When I finished cleaning the data, I decided to inspect the categorical columns, or more precisely what were the values they possessed. I looked into the Winner column, which I discussed earlier, the Boolean column for the title bout, which showed me that only 335 fights were for a title and also the weight class column. This column showed me that there were 14 different classes, which could possibly later be used to make the predictions more precise by using grouping. I also looked into the stances, which were Orthodox, Southpaw, Switch, Open Stance and Sideways. Finally, I explored the column containing the number of rounds. This column was particularly interesting as a very large number of fights ended in the third round.

The final step taken before making any visualizations was to map the stances used and the chosen winner with numerical data, so that I can use them in my predictions later. I labeled the stances from 1 to 5 respectively. As mentioned above, the winner feature had the categories Red, Blue and Draw. I mapped them with -1, 0 and 1. This would be the feature that I would like to predict.

IV. Analysis

The analysis conducted had to be very well-thought out as the breadth of the features was very big. This meant that I had to find a way to see the best possible correlations that could be used to make my predictions good. I first made use of a heatmap, which was very hard to read for the same reason - breadth. My intuition told me to make the dataset smaller, so that I can focus on the more important features. I thought that this was a good idea, as I had misinterpreted the definition of the data and thought that half of the columns were fight-specific. I also briefly explored the Doctor Stoppage column, which made it clear that too few fights were won this way and was not worth exploring as a predictor. Perhaps a logical conclusion (Figure 3 Doctor stoppage)

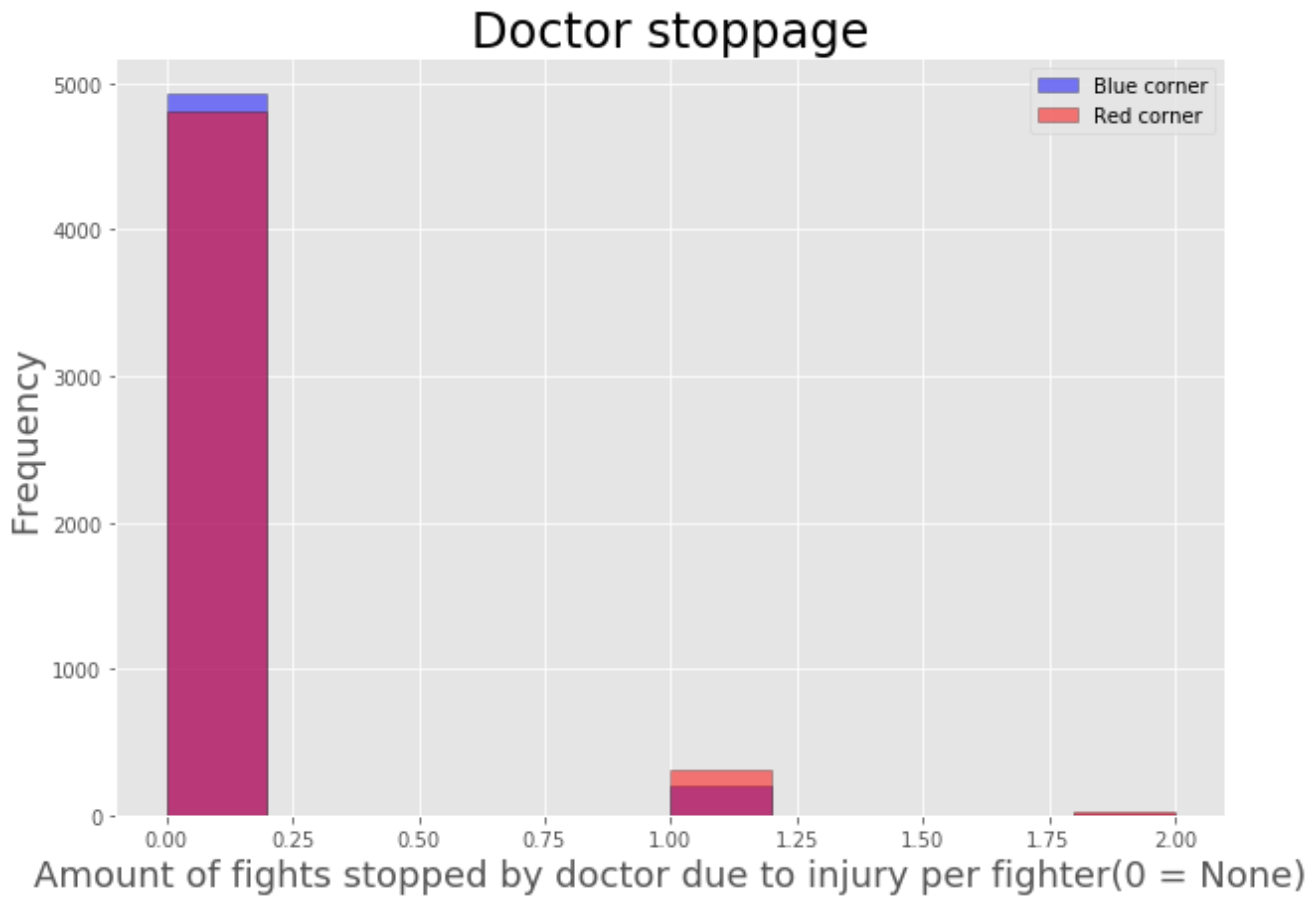

Figure 3 Doctor stoppage

I later found out that the fight-specific characteristics were actually average stats of the fighter for all of his fights. For this reason, you may find a divergence between the proposal and this report. In the end I scrapped everything and started working with the whole data again. This is the resulting heatmap (Figure 4 Heatmap). As you can see it is hard to read. If you copy the image and use paint or any other such tool on it, you can zoom in and examine it if you wish. What I discovered is that no feature correlates directly to the decision of who the winner(WinnerInt) would be.

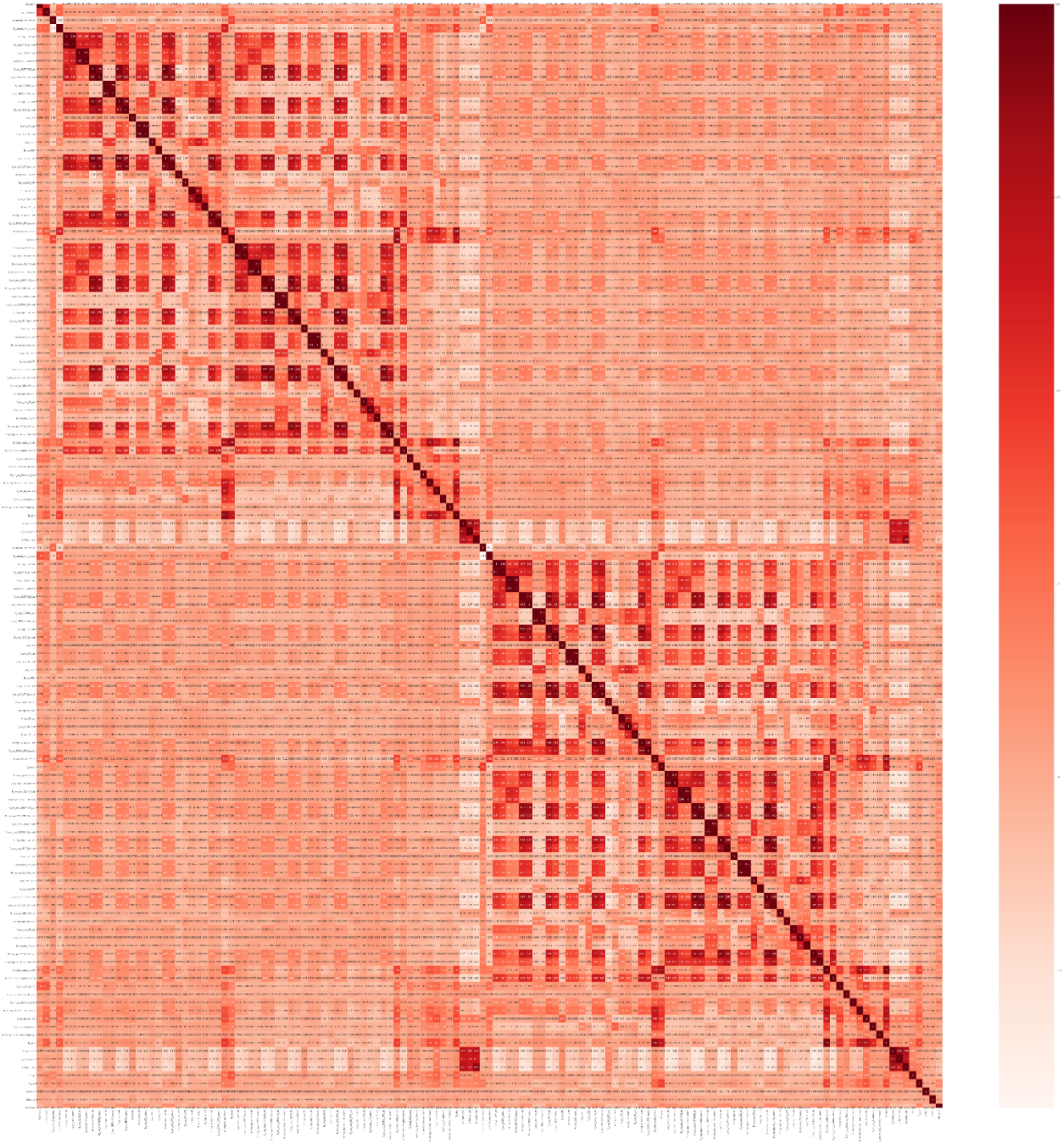

Figure 4 Heatmap

I reinforced this conclusion by using visualizations I had made before restarting the project. I had then explored whether the stance had any effect on the win rate of the fighter. It turned out that the win rate using a specific stance grew linearly with the usage of that specific stance, which meant that it had no correlation.(Figure 5 Wins per stance and amount of fighters per stance)

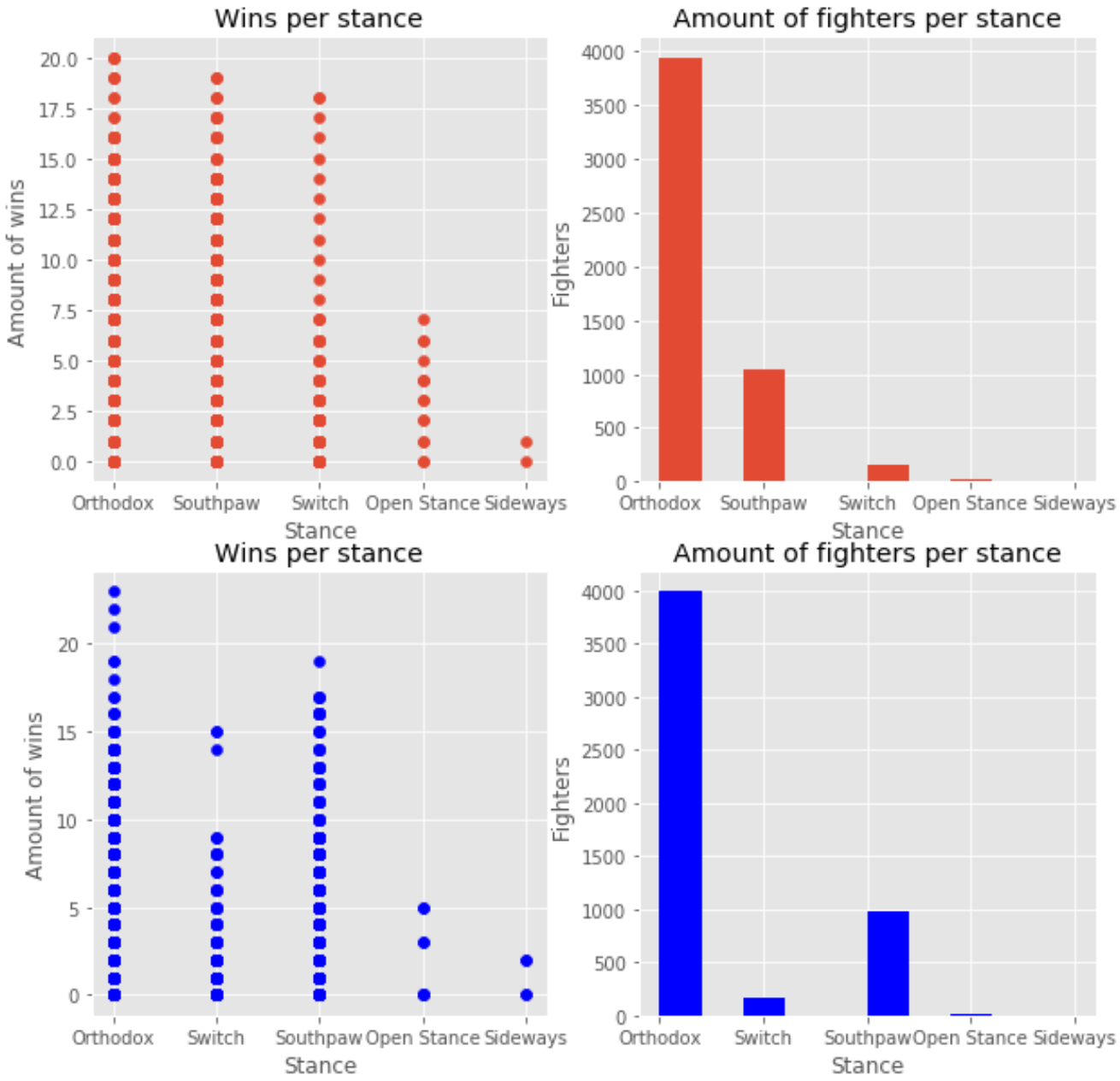

Figure 5 Wins per stance and amount of fighters per stance

At the time I also believed that the amount of wins of the fighter, together with the total rounds spent in the ring would be good predictors. I also saw a correlation between both of these features, which led me to believe that they could be exploited. (Figure 6 More rounds fought means more wins) In truth it would make sense that the more fights a person has, statistically speaking, he would have more wins. That does not mean that he was a better fighter than his opponent. Interestingly enough, the same could be said about the age of the fighter, although that later turned out to be one of the best predictors for my models.

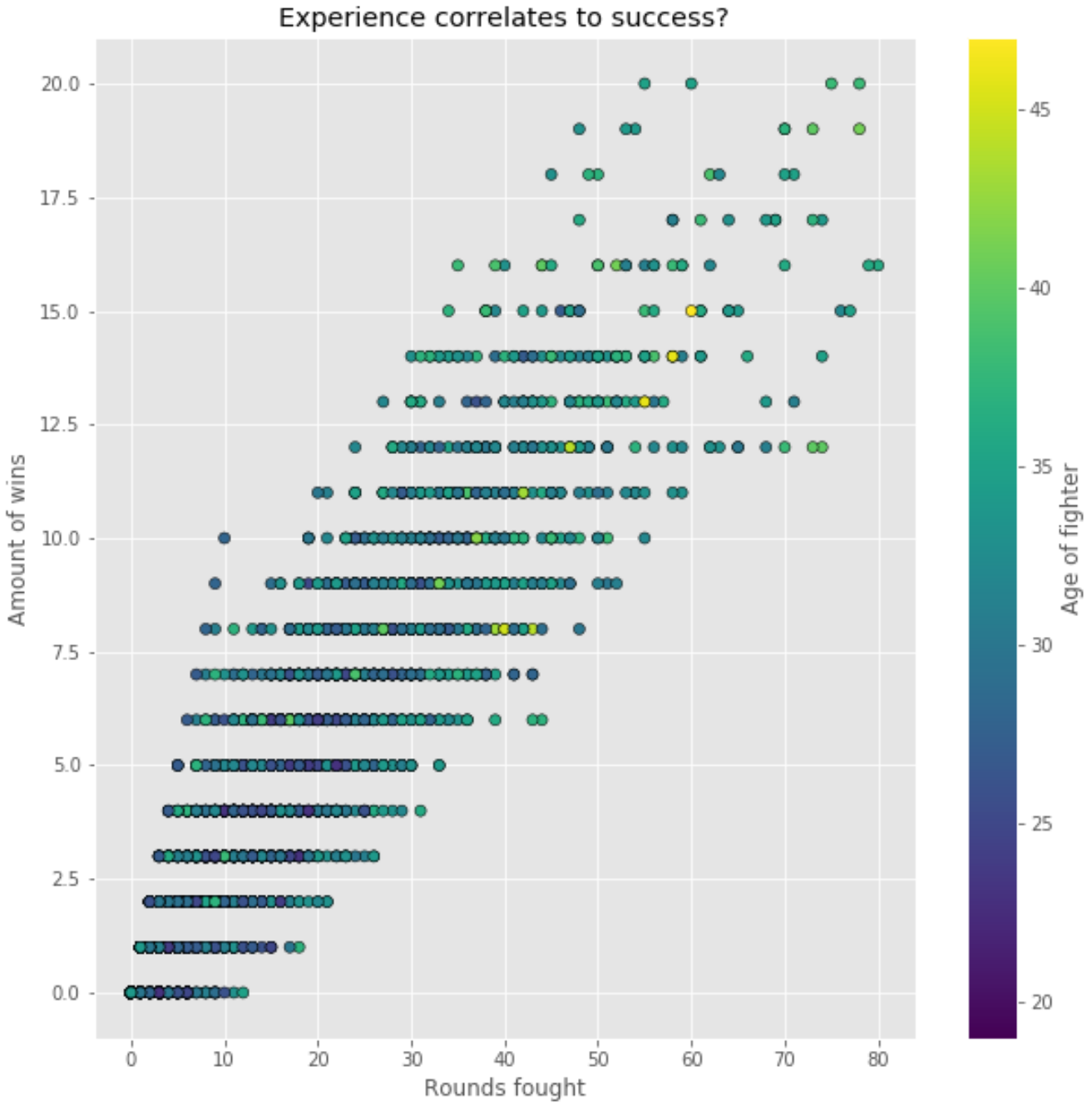

Figure 6 More rounds fought means more wins

Furthermore, I visualized the peculiar finding that the third round bore the most wins for both fighters. (Figure 7 Wins per round) This visualization tells us not only that the third round is usually the fight's conclusion, but how much more red fighters win. This may be due to a myth that the reigning champion or fan-favorite is usually put in the red corner, while the blue corner is assigned to the underdog. Nevertheless, this also tells us that our model will inherently be underfitted, due to low variance.

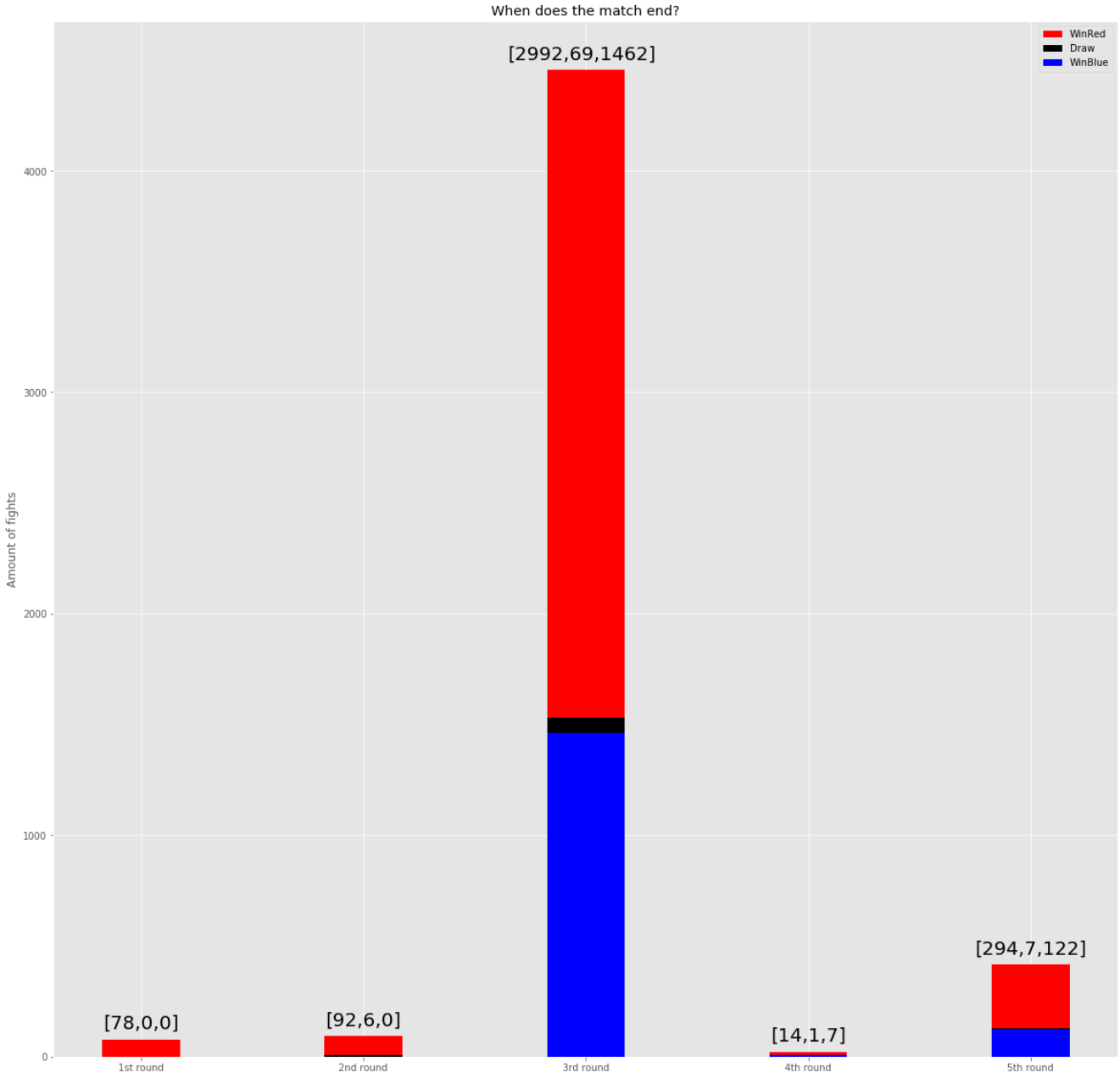

Figure 7 Wins per round

Finally, I decided to use the embedded attribute of the Extra Trees classifier called feature importance, so that I can see which features would best suit my models, even If their correlation isn't that good. I made my train and test sets and used this classifier to get the following result. (Figure 8

Feature importance)

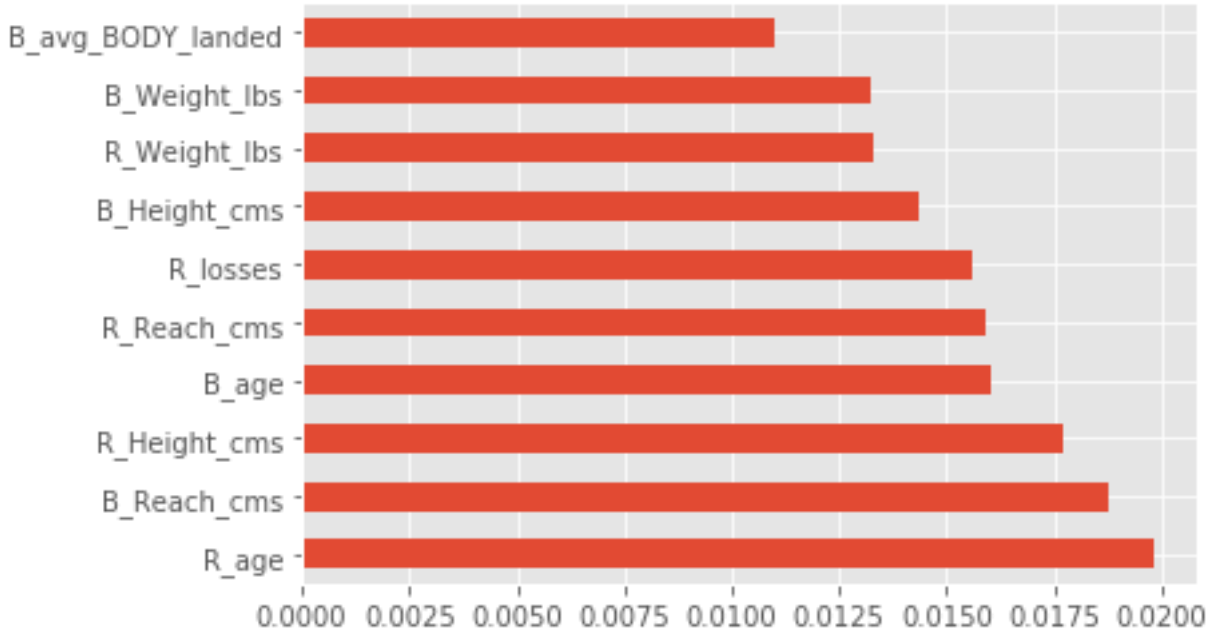

Figure 8 Feature importance

V. Results

My analysis resulted in the best features that could be used for a prediction. By using the sklearn cheat sheet I was able choose 5 models. I already knew I had to use a classification, due to the predicted value being categorical and not continuous but was unsure which one exactly to use. The cheat sheet was very helpful. I have illustrated the road I took to find the best classifier in the supplied picture. (Figure 9 Road to choose classifier)

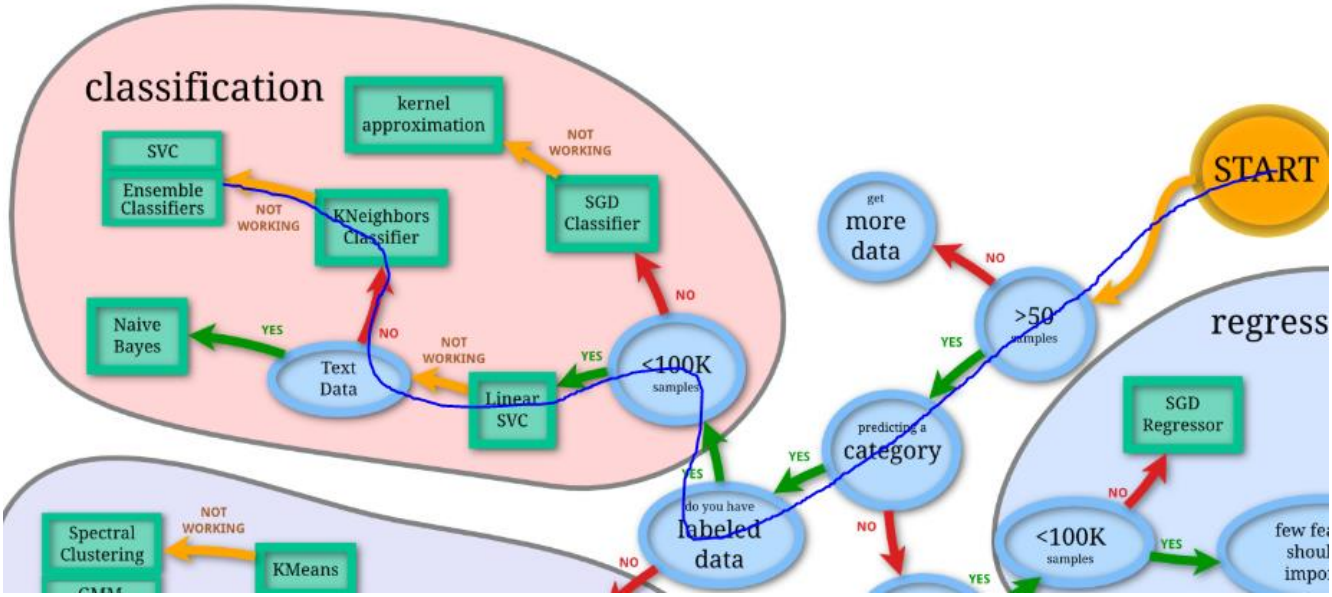

Figure 9 Road to choose classifier

1) Decision trees

In practice, the first model I used was the decision tree, because my intuition told me that it would be a good classifier before I used the cheat sheet. I tried out different parameters and found out that the best is using 'entropy' as a criterion, keeping the max_depth to be 5, as it logically worsens the model when you make this parameter smaller. From all possible parameters none of them improved the prediction, but min_samples_leaf, which I choose to be 10. More did not change the prediction, but less made it weaker. I used the same random state for all classifiers, so that I could be sure that my comparisons would be good. This bore me the accuracy of 68.9%. I used a classification report to find out that this method could not predict draws at all, perhaps due to the fact that they occurred rarely or because they were all put in a test set. This was true for all models.

2) Random Forest

The second classifier used was the random forest. This ensemble method would be the final step in the cheat sheet, but I had already implemented it earlier. I tried using 'gini' for criterion this time. I used 100 estimators, as putting any more trees in the forest did not improve the model. By making the max_depth to be 2 instead of 5 it actually boosted the prediction to 69%. I used min_samples_split instead of leaf, because that seemed to help the model as well.

3) KNearestNeighbor

I skipped the Linear SVC as it gave me very low predictions. KNearestNeighbor, on the other hand, when given 4 neighbours and the brute algorithm would be close to my original prediction. It predicted with 66% confidence.

4) SVC

The support vector classifier was next in the cheat sheet. Sadly, nothing worth noting can be said about it. It performed okay, but no matter what parameter I would give it, it would have similar confidence as the other models. Its accuracy score was 68.7%.

5) ADABoost

Due to ADABoost being an ensemble classifier I decided to use it to try and boost my model, as I had previously tried it in a Udacity course, but the prediction could not get more accurate than 68.3%. I used 100 estimators as in the random forest classifier. I tried changing the algorithm that it uses and change its learning rate, but it either did not improve or worsened.

6) Neural Network

I decided to use a MLP classifier, because I was interested in what would happen. This was not included in the cheat sheet, but I still wanted to try it out. By using a hidden layer of 3x30, a learning rate of 0.01 and a hyperbolic tangent activation I was able to get 68.2% of accuracy.

7) Decision tree on more specific data

Due to the fact that the features that I was using being for example "reach of the fighter" or their height I started to wonder if this was indeed a good way of action, as I imagined that different weight classes would have different predictors. The least of all, I had an intuitive feeling, that I wanted to see through, even if I was wrong. After grouping the data by the weight class, I created an algorithm that would take the best features per class and use them as predictors. Some classes could not be predicted, due to interesting reasons. For example, the Open Weight division had only winners in the red corner, while the Women's Featherweight had only 10 instances, which was not enough to make

a prediction. For this reason, I removed both. The final results were fascinating. Although some weight classes like Bantamweight and Flyweight (Woman's as well), Featherweight and Women's Strawweight did get a reduction in accuracy, Heavyweight, Light Heavyweight and Lightweight received quite the increase. The Middleweight and Welterweight classes stayed pretty much the same. The outlier was the Catch Weight, which received the whopping 83%, but this was due to underfitting, as this class had only 38 occurrences. The accuracies are as follows:

CLASS	*ACCURACY*
Bantamweight	***54%***
Catch Weight	***83%***
Featherweight	***60%***
Flyweight	***52%***
Heavyweight	***73.8%***
Light Heavyweight	***73.5%***
Lightweight	***71%***
Middleweight	***65%***
Welterweight	***66%***
Women's Bantamweight	***52%***
Women's Flyweight	***53%***
Women's Strawweight	***58%***

I used the decision tree classifier on all of these, as it was the classifier that usually had the best accuracy in my previous iterations. Although this was not planned in my business proposal, I am glad that I took the time to explore it, as I like the results that came out of the predictions.

VI. Conclusion

In conclusion, after rigorous research, I can say that there is no certain way to predict the outcome of a UFC match yet, even if you have the most specific stats of a fighter. In my opinion, were more data available the precision of my models could have been increased, but to my regret the current amount of information we have is not enough. Perhaps in a few years when more fights have taken place, and thus more data accumulated there would be a bigger chance of success. The current accuracy of my models varies between 66 and 69 %. I would have considered 90% a success. I found out that I can increase the accuracy by slicing the data per weight class of the fighter. As it was seen this coin is double-sided, as it works only for the Heavyweight, Light-Heavyweight and Lightweight classes. Nevertheless, I am very happy with my results as I learned quite a lot and the percentages themselves are relatively good all things considered. I hope that one day I will be able to come back to this project and perfect it, when more data is available.

VII. Recommendations

If this project is ever to be improved, I believe there are a few steps that can be taken differently. First of all, I would make sure that much more data is used as currently, in my opinion, 5144 rows are not enough, especially if you group the fights by weight class. What is more, if I had more time, I would probably try a different approach as I realized too late that the variance of the data is too low. After all, the winner is from the Red corner in 3470 of the cases. If I predicted that the winner is red all of the time I would still have a pretty high accuracy and this should not be the case. Perhaps a better way would be to extrapolate the fighters and try to predict whether they would be the winner regardless of their corner color. A final note would be that, given I had more time, I would have also liked to visualize my predictions a little bit better, especially the decision tree.

VIII. Appendix

Business Proposal

BusinessProposalChallengeV2.pdf

Jupyter Notebook (reproducible research)

Personal Challenge Filip Georgiev.ipynb

Jupyter Notebook (pdf)

Personal Challenge Filip Georgiev.pdf