UFC Match Predictor

BrainStation Capstone Project
Report By: Aref Jadda

The Problem:

The UFC is the largest mixed martial arts (MMA) promotion company in the world and it hosts some of the highest-level fights with some of the best martial artists on the planet [1]. I have personally been a fan of the company and their fights for years. My goal with this project was to use data science and machine learning methods to explore the data and mainly try to build predictive models that can predict the winner in a fight based on the fighter's statistics and other data available before the fight. I also aimed to find some features that were good predictors of the final outcome.

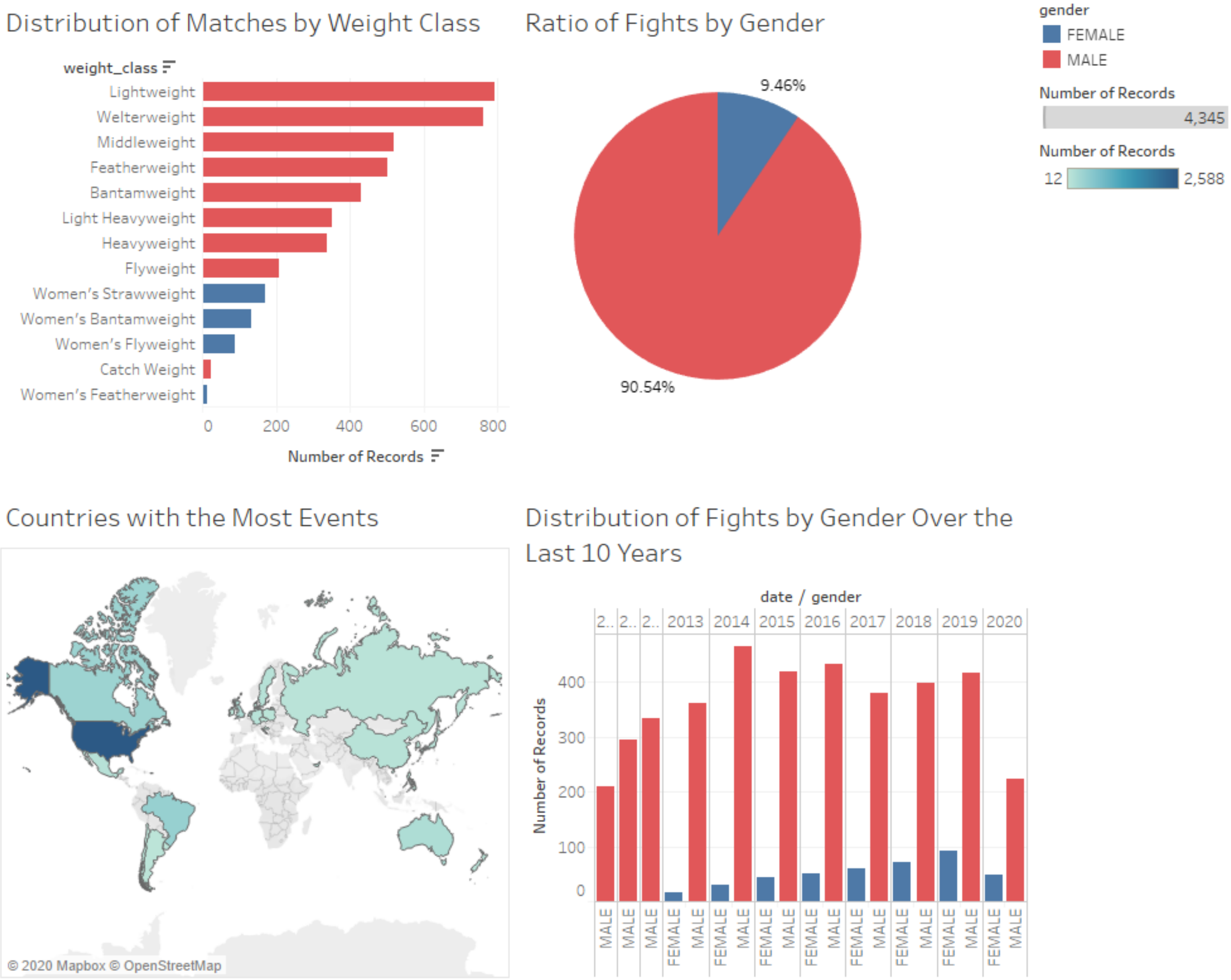
Having years of experience with different martial arts myself, I believe that I can truly understand and analyze the data from an expert perspective. Knowing that fighting and more generally sports outcomes are very difficult to predict, I initially aimed for 75 to 80% accuracy as a target for this project. Fight specialists that have trained and studied martial arts their entire lives have a hard time making good predictions even with having access to a broad range of information before the fight that our machine learning models will not have, so tackling this problem I already understand that models with 90% accuracy are impossible to achieve, and even if that does happen in a case it is purely by chance.

Data Collection:

Initially the idea was to scrape the data *ufcstats.com*. I started this process with the 'Request' and 'Beautiful Soup' packages in python. However, after seeing how many different pages I would need to scrape and merge I decided that I should seek quicker alternatives. Luckily, I found a Kaggle dataset that had done all this scraping for me. The initial data contained 4345 rows and 137 columns which is a small dataset to work with.

Exploratory Data Analysis:

After collecting the data, I first did some exploratory data analysis (EDA) on it with Tableau to see what I was working with, how the data looked, and what features were available. Below is a picture of the dashboard of some of the interesting graphs from the EDA.



The EDA also helped me realize that many columns were redundant and caused collinearity and potentially multicollinearity from my knowledge of MMA, so they would need to be removed. I also figured that the data has some inconsistencies and typos that need to be fixed. In some cases, the same entries were seen as different due to spelling or capitalization errors. With this knowledge of the data I then started the data cleaning process.

Data Preprocessing and Feature Engineering:

The data preprocessing phase was the longest and most stressful part of the project. I needed to ensure that I do not introduce any bias into the data and at the same time make sure I preserve as much rows of data that I possibly can since I already had a very small dataset to begin with. Thankfully, by the end I did manage to keep all rows. You can view the full schema in my 'Data Cleaning' python notebook. Many of the columns are stats on the two fighters in the two corners having a 'R' or 'B' prefix representing the red corner fighter and the blue corner fighter respectively. These corners are selected randomly and the fighters can end up on either side irrespective of their ranking or popularity.

Many Null values were replaced by the mean, median or zero where applicable. These decisions were made based on the feature distributions. 'String' and 'datetime' columns were turned into dummy variables and added as columns to the dataframe for machine learning purposes later on. After all the null values were filled in and the columns were all numeric, I decided to engineer features that can help out my machine learning models to make better predictions. The major change that can be seen is replacing separate features of the red and blue corner fighters with the difference of those values. For example, the 'R_height' and 'B_height' columns were replaced with 'height_dif' which is `B_height – R_height'. This example refers to the difference in height of the two fighters.

I am explaining this process very linearly, however, the final version of the cleaned data is the result of numerous tests of different models on the data to see how it performs. I went back to the data preprocessing phase after testing and changed the data many times over and over. Next, I performed normality tests by just looking at the histogram of the distributions of different columns to get a general overview. It would have been a better idea to use Q-Q plots or other tests to verify normality too, however I had many columns most of which were

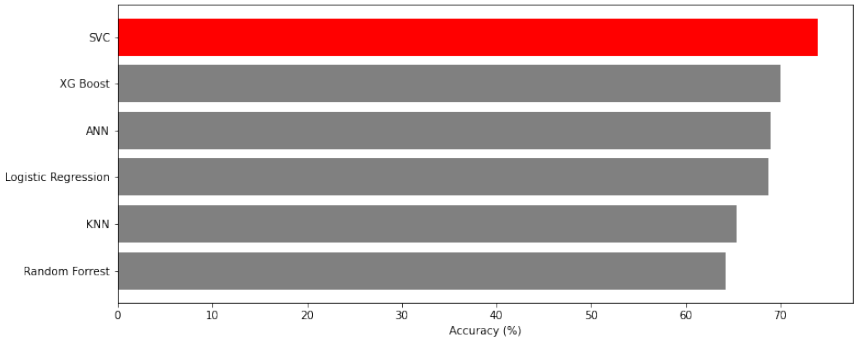
Booleans so I just relied on the quick histogram eye check. I then transformed the columns that I thought would perform better if normalized.

Lastly, I looked at the correlation matrix between the columns and removed columns with collinearity higher than 70% to avoid extreme collinearity. Since I was not looking to perform any statistical regressions, and I was only planning on using machine learning models, I did not worry too much about collinearity under 0.7.

Machine Learning:

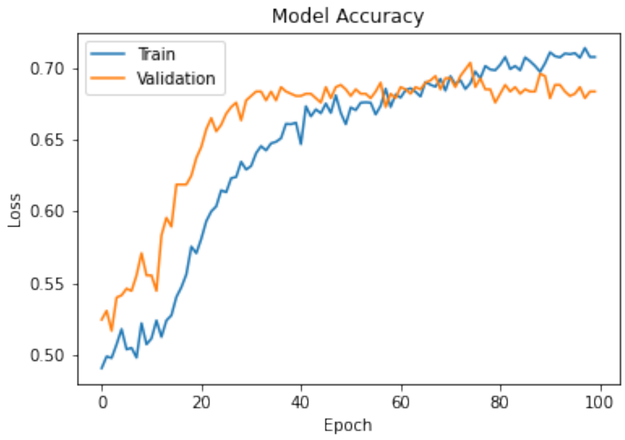
After the data was clean and all our features were numerical, the next step was to run different machine learning models on it and find the best one. This step was done in different ways in different notebooks and only the best one was submitted. There were mainly differences in how the class imbalance in the target variable was addressed and also how the train test validation splits were done which is explained in a bit more depth in the machine learning notebook. Different scalers were also run on the independent variables, however, only the best one was kept. In some cases, the scaler actually lowered the accuracies considerably.

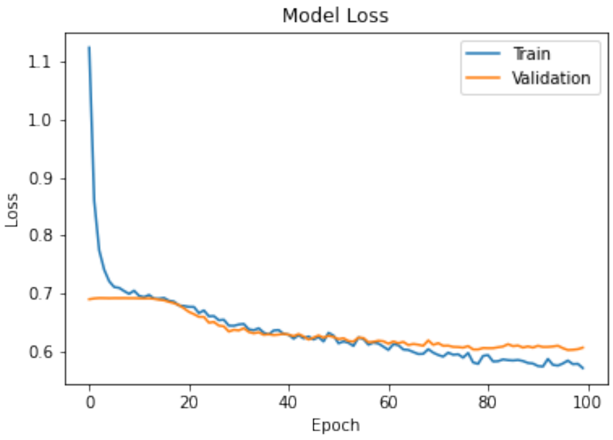
Cross-validation was performed on the data and the test set remained untouched until the final evaluation to avoid introducing bias and actually test the models on unseen data. After performing a grid search on all models to optimize hyperparameters the support vector classifier proved to be the best model. This might be because SVC is usually more effective in higher dimensional spaces, and in cases where the number of samples is not much greater than the number of dimensions. Below you can see the highest test accuracies achieved by different models.



In this project only the accuracy scores were used as an evaluation method. The reason being that this is not really a classification case where the classes are 'different' things. They are both fighters and the corners are chosen randomly so the idea of 'true positives', 'precision' and 'recall' are not necessarily relevant here.

You might notice that the artificial neural network model ranks third among the models. Initially the ANN overfit very quickly after a few epochs. However, after lowering the number of layers (because we have a small dataset to train) and increasing the dropout rate significantly (a dropout rate of 70 in each layer in this case) the models started looking much more reasonable and it no longer overfit. Below you can see the accuracy and loss function graphs from the neural network training.





The GUI for the Model:

I was hoping to create a graphical user interface for my model using Flask or Streamlit where you can select two fighters and some other details on a match and get a prediction of who will be winning the fight by the model. Unfortunately, a series of unexpected events halted my progress in the last week before the deadline of the report. I will be completing this last part of the project in the weeks following the deadline, so you can check my GitHub for the final product later on.

Conclusion and Next Steps:

Sports outcomes are generally very challenging to model and I believe fighting is among the toughest sports to predict. Fighters are always one small mistake away from getting knocked out which finishes the fight immediately. The data we have is such a small part of all the things that can affect the outcome of a fight. Something as simple as a bad weight-cut before the fight can change everything. Our data only has a small peek into these factors in the odds difference column where it sees who was the favorite among the crowd.

To improve these models we need more data, and this data can be collected from other fight organizations too. We can also engineer and collect new features such as the fighters main fighting style (ex. grappler or striker) than can help out.

References:

1. ‘UFC buys rival Strikeforce’. Link: https://www.espn.com/extra/mma/news/story?id=6209923