UFC Fight Predictor

Mixed martial arts is one of the fastest growing sports in recent history and in 2019 it made a 5 year deal with espn to broadcast some of its fights. With any sport of this size people want to know who is gonna win before the fight even happens whether to brag about your knowledge of the sport or make a few safe bets for some extra cash.

1. **Data**

The data was all taken from a very well put together kaggle database. The dataset contained a list of every fight from the ufcs start in 1993 to june of 2019. This data contains all sorts of information about the fight where it took place, who was refereeing, and all sorts of fight statistics.

2. **Data Wrangling**

The cleaning for this data was very easy this dataset is very well put together as I mentioned before but there were a few things I did need to do. The first was to drop all the fights that were before the rules for the ufc were made and used because those were not the same type of fights we see now those were basically anything can go. Then I dropped most of the useless columns there were such as referee, location, and date. I also made sure to drop all the draws that happened as I was going to make this a binary classification problem trying to predict if the red fighter won which would be a 1 and if blue won it would be a 0. Finally after seeing there were still nan values after all this I decided to drop all the rows with nan values since there was no good way to replace that data.

3. EDA [Visualizations]

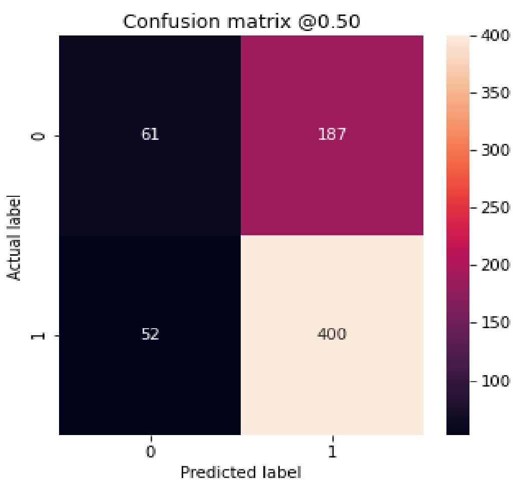
During my eda I was a little nervous at first glance with my heatmap because it seemed there was not a whole lot of correlation with anything and the Winner. I did not understand how that worked because there had to be something that influenced it so I decided to look at an individual fighters heatmap and saw almost an entirely different thing. This made me realize that I should leave in the fighter names and just encode them because each fighter is better than another at some things for instance the fighter I looked at was specifically susceptible to knockdowns which made sense since this fighter was a veteran and has taken a lot of head damage meaning his recovery from such strikes is poorer than say a younger fighter or newer fighter. I also made sure to look at every column since there were a total of 141 in my model most of the columns were all fight related. There wuld be one specifying how many strikes were thrown another for how many landed it also had a wide variety of different strikes and positions whether they were thrown standing up, in the clinch, on the ground, or just specifically the head. There were also many columns talking about what there average opponents looked like I was not sure if I should use these columns as we were just trying to see if it could predict the fight at hand but they did seem to not really affect the model either way so I left them in for my final model.

4. Deep Learning

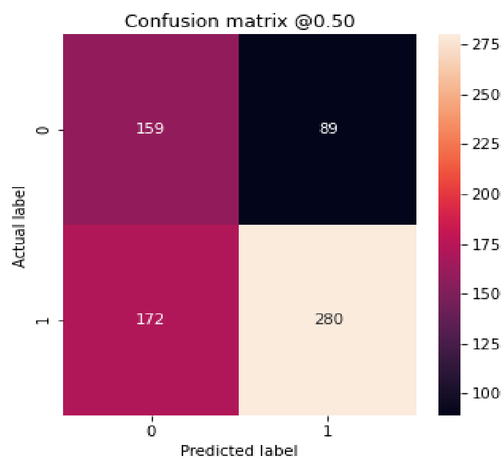
For my deep learning models I used tensorflow to create my prediction model and I tryed 2 different datasets. For the datasets one was 141 columns and the other was 72 columns I found that the model performed a little worse with the 72 column one so I dropped it all together.

For my models on the other hand I used different forms of weights for each model my first was with just initial weights which i found performed better than than the model that had class weights attached to it. That said the class weight model did perform a lot better finding the true negatives than the other model but at a great cost of the true positives.

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense (Dense)                (None, 600)               85200
_________________________________________________________________
dropout (Dropout)            (None, 600)               0
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 601
=================================================================
Total params: 85,801
Trainable params: 85,801
Non-trainable params: 0
_________________________________________________________________
```



Initial weights only

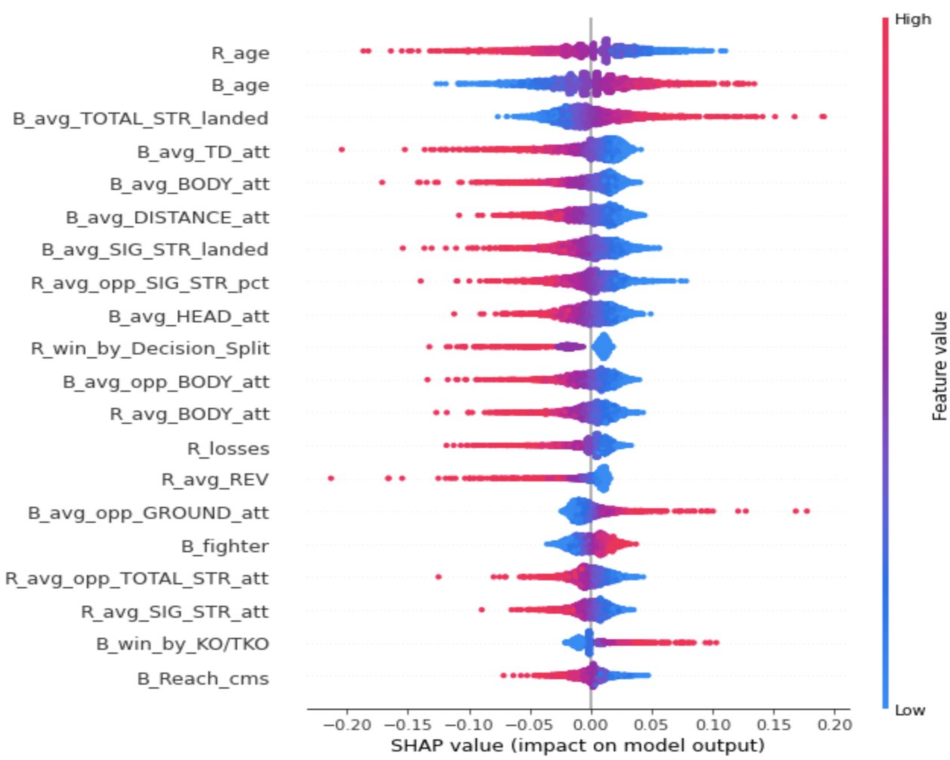


Class + Initial weights

Initial weights only	Class + Initial weights
loss : 0.60981565713	loss: 0.63559824228
accuracy : 0.65857142210	accuracy: 0.62714284658
precision : 0.6814309954	precision: 0.75880759954
recall : 0.88495576381	recall: 0.61946904659
auc : 0.68273621797	auc: 0.69237530231

5. Feature Importance/Conclusions [Extra Visualizations]

While looking at the features it is important to remember our model was trying to predict whether the red fighter would win or not. If you click on the extra visualizations and look at the dependence plot you can see even more in detail on how each feature affected the model. A good example is the B_avg_td_att which is the amount of takedown attacks the blue fighter may have done during the fight which has a clear downward trend meaning that the more takedown attacks from the blue fighter the worsemimpact it had on the model. This makes sense because the model is based on if red won or not and if blue was having a significant amount of takedowns in a fight we know they most likely are going to be winning that fight. After noticing this I realized my model must have a hard time predicting comebacks as any model would because the data will always be heavily skewed in the other direction since a comeback is usually a lucky or well placed strike that knocks there opponent out.



6. **Future Improvements**

Would really like to play around with different layers more I only choose these model metrics because they seemed to work consistently the best. Would like to see how adding some layers with a tanh activation function would change the model outcome. I would also like to try some more different datasets as well because there do tend to be very different style of fighting throughout the different divisions so maybe just using specific divisions when training would help with the prediction outcome at that level.

Thank you to Rahul Sagrolikar best mentor on this project I coulda asked for wouldn't of been able to do it without him.