CMPT-353: Project Report

Mitchell Hole
301198741

Background

In most combat sports athletes are separated by and compete within weight classes. A weight class is a range of weights (i.e 171 lbs – 186 lbs for the middleweight division) in which the athlete must officially weigh in at in order for the fight to be sanctioned by the governing athletic commission of where the fight is taking place. This weigh-in is usually held the day before the contest and is designed to ensure fair competition since being significantly bigger than your opponent is a huge advantage in combat sports.

This had led to a practice called weight cutting. With weight cutting your weight is only actually within your weight class when you officially weigh in. The days preceding this are spent trying to drain your body of as much water as possible until you reach the upper limit of your weight class. After you weigh in you start rehydrating to come into the fight the next day as heavy as possible. This process can result in you being more than 30 lbs heavier than your weight class during the fight.

Needless to say this process is incredibly dangerous. A prisoner's dilemma situation is created in which fighter's have to cut large amounts out of fear that if they don't their opponent will. Weight cutting has also led to fighters missing weight, having to stop their weight cut short of the upper limit due to medical concerns, and weighing in heavier than their classes limit. The goal of this project is to explore under what conditions this might happen and what attributes to look for in a fighter that is likely to miss weight.

Data Gathering

Code: /ETL/getFighterInfo.py

Data gathering began by making a GET request to the fighter URL of UFC's official API, http://ufc-data-api.ufc.com/api/v3/iphone/fighters. This returns a list of every fighter that's ever been contracted in the organization and summary information about them; name, id (appending this to the previous URL gets detailed information), and a link to their profile on the UFC's site. This data was read into a Pandas Dataframe, each row being an individual fighter, and only a few of the original rows kept.

As mentioned appending the fighter's unique id to the above URL returns detailed information about them. Each of these URL's were iterated through to add their DOB and height to the dataframe.

Unfortunately the API didn't have information about the fighter's wingspan (length of outstretched arms) but this was available from their profile in the link previously stored. I iterated through each, scraping the HTML to find the location in the returned DOM. A small fraction of these profile sites didn't contain the fighter's wingspan so I imputed the data by setting it equal to the fighter's height + 1 inch. Wingspan in general is close to one's height (pro athletes tend to be a bit longer) so this seemed like a fair estimate.

The dataframe is finally saved to a JSON file, fighters.json.

Code: /ETL/getFightInfo.py

This program starts by loading in the previously generated JSON file and uses the fighter's ID to make a request to the individual fighter's API URL. The information about every match the fighter has been in is stored here. The fighter's weigh-in weight, the weightclass the fight was at, the date and whether they won are extracted and put into a row of a dataframe. This is done for each fighter and each fight.

Whether the fighter missed weight or not is computed with:
fights['missed_weight'] = fights['weigh_in'] > fights['max_weight']
and the fighter's age at the time of the fight is done by subtracting the event date by the fighter's birthdate and converting the produced timedelta object into years.

The dataframe is finally saved to a JSON file, fights.json.

Problems

1. Does cutting weight get harder as you age? This is a commonly used rhetoric in the community and I'd like to see if the data backs this up. Specifically, I want to test if the average age of fighters that miss weight is different from the average of those that make weight.

Code: /Analysis/missedWeightAge.py

I made two dateframes, made_weight & missed_weight, by partitioning the fights.json file on the ['missed_weight'] column. I then isolated the ['age'] column of both these dataframes as the input for a ttest:

$H_0 : \mu_{missed} = \mu_{made}$ (fighters that miss weight are the same age as ones that make)

H_α: $\mu_{missed} \neq \mu_{made}$ (fighters that miss weight aren't the same age as ones that make)

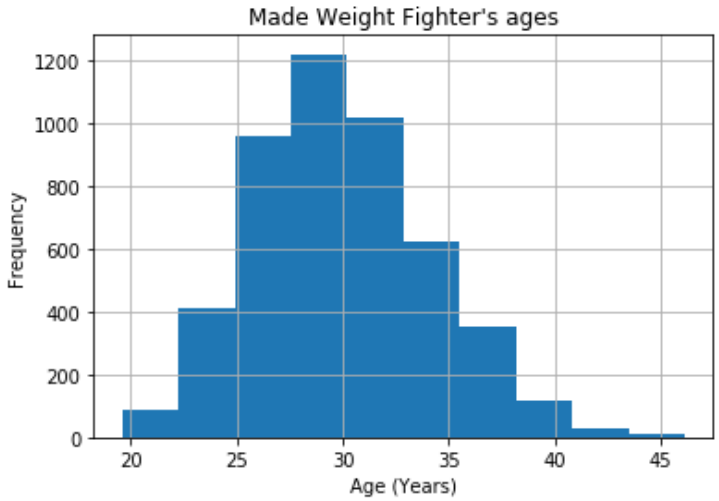
Normality & Variance Tests

```
print(stats.normaltest(missed_weight['age'].values).pvalue)
print(stats.normaltest(made_weight['age'].values).pvalue)
print(stats.levene(missed_weight['age'].values, made_weight['age'].values).pvalue)
```

```
0.21878818100203806
2.8295278132545175e-21
0.7472463405009295
```

Missed weight is normal, variances are equal but made weight is resoundingly not normal…

The histogram suggests otherwise though:



This looks slightly right skewed but very normal to me. Using:

'In practice, if you have ≥40 data points, and a plot shows a not-too-crazy distribution, go ahead.'

from the lecture slides I decided to proceed with the T-Test:

```
ttest = stats.ttest_ind(missed_weight['age'].values, made_weight['age'].values)
```

```
ttest.pvalue
```

```
5.102997732605877e-05
```

and reject the Null Hypotheses. They have different means.

The result wasn't as expected though. The fighters that missed weight are two years younger than those that make it on average.

```
np.mean(missed_weight['age'].values)
```

```
28.01795198363798
```

```
np.mean(made_weight['age'].values)
```

```
29.92999257527085
```

Interpretation: From my domain knowledge it is common to see fighters cut massive amounts of weight, to reach the lowest weight class possible, in the early parts of their career. After missing weight once or twice or having health scares during their cuts they will move up a weightclass. I think these numbers here are showing around when that typically happens.

2. In July of 2015, the UFC hired USADA (United States Anti Doping Agency) to do their banned substance testing. They are renowned for their stringent policies and for performing random drug tests on a '24-7-365' basis. Before this, all testing was done by the state's athletic commission on a predictable schedule. Knowing when a drug test will occur makes cheating it very easy. You just need to know how long the particular drug remains detectable in your body. I suspect that prior to the implementation of USADA several banned substances, such as diuretics, were commonly used to aid in cutting weight and that in their absence fighter's missing weight has increased.

Code: /Analysis/missedWeightTime.py

H_0 : The fighter's missed weight at the same rate before and after USADA testing
H_α: The fighter's missed weight at different rates before and after USADA testing

I made two dataframes, pre_USADA and post_USADA, from the fights.json file, partitioning by rows with 'date' columns before/after July 1, 2015. Each of these dataframes was in turn partitioned on the 'missed_weight' column. The counts of these four dataframes was put into a chi^2 contingency table.

```
pre_missed = pre_USADA[pre_USADA['missed_weight'] == True].shape[0]
pre_made = pre_USADA[pre_USADA['missed_weight'] == False].shape[0]
post_missed = post_USADA[post_USADA['missed_weight'] == True].shape[0]
post_made = post_USADA[post_USADA['missed_weight'] == False].shape[0]
```

```
contingency = [[pre_missed, pre_made], [post_missed, post_made]]
```

The resulting P-value was 0.018 so the Null Hypothesis can be rejected.

```
chi2, p, dof, expected = stats.chi2_contingency(contingency)
print(p)
print(expected)
```

```
0.01773487296575431
[[  40.9450235 2495.0549765]
 [  38.0549765 2318.9450235]]
```

```
contingency
```

```
[[30, 2506], [49, 2308]]
```

Interpretation: The proportions of the two groups has been proven to be different and the info above shows that fighters are missing weight more after USADA drug testing was implemented. This may indicate the use of diuretics and other banned substances to aid in weight cutting were more common prior to USADA.

3. Can information about a fighter be used to figure out at what weight class he/she should fight at? I will attempt to do so with machine learning.

Code: /Analysis/weightClassifier.py

Using the fighters height, reach and age as the input tuples and the weight class they fight at as the output, I tried several different classification models to see which gave the most accurate predictions.

Random Guessing: 12.5% (8 weight classes)

KNN: 25% - 30%

- Any value between 5-10 for nneighbors yielded similar results

SVC: 35% - 40%

- Performs best with a linear kernel and C=10
- Small margin is preferred

Naïve-Bayes: 30% - 40%

Interpretation: The predictions made by the classifiers were 2 – 3 times greater than chance but were still relatively low. This was likely because the classifier wasn't able to take varying body types into account. If features such as body-fat percentage and waist size were available I think that the predictions would have dramatically improved.

Project Experience Summary

- Gathered data through web scraping and making requests to the UFC's official API to begin the ETL pipeline
- Cleaned data by discarding unwanted rows, imputing missing values and converting values to the necessary type so that the data would be in a format useable to the statistical tools
- Performed t-tests and chi-squared tests on the data using the stats.scipy module to make inferences on the data
- Created a classifier that uses data about a fighter's to try to predict his/her weight class