UFC_Weight_Cut

August 1, 2025

[1]:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import openpyxl
import statsmodels.api as sm
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.linear_model import LogisticRegression

import pandoc

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import accuracy_score, roc_auc_score, confusion_matrix

import math

import statsmodels.formula.api as smf

from sklearn.metrics import roc_curve, roc_auc_score
from scipy.stats import chi2
from statsmodels.stats.diagnostic import linear_harvey_collier
from statsmodels.stats.outliers_influence import variance_inflation_factor
from statsmodels.stats.proportion import proportions_chisquare
from statsmodels.stats.api import het_breuschpagan

import statsmodels.api as sm
from statsmodels.stats.diagnostic import acorr_ljungbox
from statsmodels.stats.outliers_influence import summary_table
from statsmodels.stats.diagnostic import linear_reset

from statsmodels.stats.diagnostic import lilliefors
from statsmodels.stats.diagnostic import het_white

from sklearn.metrics import roc_auc_score, roc_curve
from statsmodels.stats.diagnostic import het_breuschpagan
```

```python
from statsmodels.stats.proportion import proportions_chisquare
from statsmodels.stats.api import het_white

from statsmodels.stats.diagnostic import linear_harvey_collier


from statsmodels.stats.diagnostic import het_breuschpagan
from statsmodels.stats.api import het_white

from statsmodels.stats.diagnostic import het_white
import os
print(os.environ['PATH'])
```

```
/Users/caseymoser/opt/anaconda3/envs/UFC_data/bin:/Users/caseymoser/opt/anaconda
3/condabin:/usr/bin:/bin:/usr/sbin:/sbin
```

1 Background

One of the most pervasive controversies in fighting is weight cutting. Fighters are divided into weight classes that they choose to fight in for the sake of fairness. However, fighters have realized they can gain a massive advantadge by reducing their weight through water cutting — the process of temporarily losing weight through sweat and regaining it post weigh-in — to make weight classes that otherwise would be impossible for their body composition. Since weigh-ins are done the day before fights, fighters can then recover the lost water and go back to their "real" weight by fight night. The advantadge of weight cutting, especially against an opponent who cuts less, is drastic. Fighters that weigh more hit harder, can more easily smother their opponents in wrestling, and can have a signficant reach advantadge.

Because of this signficant advantadge from weight cutting, fighting has increasingly become a game of who can cut the most weight. Fighters have been known to regain up to 20 lbs between weigh in and fight night, with the highest recorded regain from Geoff Neal, who gained 30.3 pounds at UFC 298 (https://www.espn.com/mma/story/_/id/39610394/seven-ufc-298-fighters-flagged-rehydration-issue).

On one hand, this phenomenon hurts the sport by making fighting skill less important to the overall equation. Some fighters who may be less skilled are able to win fights because their bodies are naturally adapted to rapidly losing and gaining water weight, a term disparagingly reffered to as "weight bullying." Furthermore, exciting fights can get cancelled from fighters failing to make weight from attempting too large of a weight cut, or even having to pull out due to health complications from bad water cuts (https://www.mmamania.com/2022/9/9/23345292/ufc-279-dana-white-reveals-khamzat-chimaevs-weight-cut-ended-after-locking-and-cramping). Even more concerning is the danger that comes with weight cutting. Fighters have been known to not only have serious health complications from bad weight cuts, but have even died from it (https://www.espn.com/mma/story/_/id/14344041/chinese-mma-fighter-yang-jian-bing-dies-trying-make-weight).

To combat this isue, organizations such as One Championship, a rival organization to the UFC, have implemented measures such as measuring hydration levels post weigh in to ensure fighters are not cutting too much water weight. However the UFC, the biggest and most popular MMA organization in the world, currently has no such measures.

Due to how pervasive weight cutting has become, it is important to investigagte how crucial weight cutting has become to fighting. In this study, I intend to investiage the impact weight cutting has on the odds of winning fights.

2 Explanation of the Data and the Data Sources

The data used for fight results is up to date as of UFC 318 which took place on 7/19/25. The data set can be found here: https://github.com/Greco1899/scrape_ufc_stats

2.0.1 For context, the weight classes in the UFC are as follow:

- Flyweight – up to 125 lb
- Bantamweight – over 125 to 135 lb
- Featherweight – over 135 to 145 lb
- Lightweight – over 145 to 155 lb
- Welterweight – over 155 to 170 lb
- Middleweight – over 170 to 185 lb
- Light Heavyweight – over 185 to 205 lb
- Heavyweight – over 205 to 265 lb

2.0.2 Independent Variable

Weight regain percentage: To measure a fighter's weight cut, I have made a variable called PERCENT_REGAIN. This variable is the percentage change between a fighter's weigh-in weight and their fight weight the next day. This fight-night weight is unfortunately not universal across UFC events because not every region measures it. For example, the California State Athletic Commission (CSAC) measures fight night weight for the purposes of regulating extreme weight cuts. This data has been collected from the dataset linked here: https://www.reddit.com/r/MMA/comments/evbnjd/released_offical_ufc_fight_night_weights/. Weight differences are measured as a percentage to better account for the fact that the higher weight classes have the room to lose more absolute weight compared to the lighter weights.

3 Hypothesis

My hypothesis is that greater percentage of weight regained will be associated with higher odds of winning fights by giving the fighter a height and weight advantadge on fight night.

[2]:
```python
stats_path = '/Users/caseymoser/Desktop/UFC Analysis/UFC/ufc_fight_stats.csv'

# data source: https://www.reddit.com/r/MMA/comments/evbnjd/
↪released_offical_ufc_fight_night_weights/
```

```
weight_path = '/Users/caseymoser/Desktop/UFC Analysis/UFC/UFC Fight Night␣
↪Weights.xlsx'

results_path = '/Users/caseymoser/Desktop/UFC Analysis/UFC/ufc_fight_results.
↪csv'

stats_df = pd.read_csv(stats_path)

weight_df = pd.read_excel(weight_path)

results_df = pd.read_csv(results_path)
```

[3]:
```
# Step 1: Split fighters into separate columns
results_df[['Fighter_1', 'Fighter_2']] = results_df['BOUT'].str.split(' vs. ',␣
↪expand=True)

fighter1_df = results_df.copy()
fighter1_df['FIGHTER'] = fighter1_df['Fighter_1']
fighter1_df['RESULT'] = fighter1_df['OUTCOME'].str[0].map({'W': 'Win', 'L':␣
↪'Loss'})
fighter1_df = fighter1_df.drop(columns=['Fighter_1','Fighter_2'])


fighter2_df = results_df.copy()
fighter2_df['FIGHTER'] = fighter2_df['Fighter_2']
fighter2_df['RESULT'] = fighter2_df['OUTCOME'].str[2].map({'W': 'Win', 'L':␣
↪'Loss'})
fighter2_df = fighter2_df.drop(columns=['Fighter_1','Fighter_2'])


# Note to self, clean this code so that I first drop fighter_1 and fighter_2,␣
↪then combine the data sets together/

# Combine both into a single dataframe
results_df_clean = pd.concat([fighter1_df, fighter2_df])

results_df_clean['UFC_EVENT'] = results_df_clean['EVENT'].str.extract(r'(UFC␣
↪\d+)', expand=False)
```

[4]:
```
weight_df['UFC_EVENT'] = weight_df['EVENT'].str.extract(r'(UFC \d+)',␣
↪expand=False)
weight_df

weight_df['FIGHTER'] = weight_df['FIGHTER'].str.strip().str.lower()
weight_df['UFC_EVENT'] = weight_df['UFC_EVENT'].str.strip().str.upper()
```

```python
results_df_clean['FIGHTER'] = results_df_clean['FIGHTER'].str.strip().str.
    ↪lower()
results_df_clean['UFC_EVENT'] = results_df_clean['UFC_EVENT'].str.strip().str.
    ↪upper()

merged_weight = pd.merge(
    results_df_clean,
    weight_df,
    on=['FIGHTER', 'UFC_EVENT'],
    how='left',
    suffixes=('_result', '_weight')
)
results_df_clean

# Drop rows with any NaN values
merged_weight_clean = merged_weight.dropna(subset=['WEIGH IN WEIGHT (lbs)'])
```

[5]:

```python
win_weight_df = merged_weight_clean

win_weight_df['RESULT'] = win_weight_df['RESULT'].map({'Win': 1, 'Loss': 0})


win_weight_df_clean = win_weight_df.dropna(subset=['RESULT', 'WEIGHT INCREASE␣
    ↪(lbs)'])

win_weight_df_clean['PERCENT_REGAIN'] =win_weight_df_clean['WEIGHT INCREASE␣
    ↪(lbs)']/win_weight_df_clean['WEIGH IN WEIGHT (lbs)']*100
```

```
/var/folders/rc/yw7v7vhj4l3_vjznh0cwm8wr0000gn/T/ipykernel_16016/888630642.py:3:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  win_weight_df['RESULT'] = win_weight_df['RESULT'].map({'Win': 1, 'Loss': 0})
/var/folders/rc/yw7v7vhj4l3_vjznh0cwm8wr0000gn/T/ipykernel_16016/888630642.py:8:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  win_weight_df_clean['PERCENT_REGAIN'] =win_weight_df_clean['WEIGHT INCREASE
(lbs)']/win_weight_df_clean['WEIGH IN WEIGHT (lbs)']*100
```

3.0.1 Choice of Regression

Given that the dependent variable is binary (win or loss), I have chosen to fit a logistic model to predict the log-odds of winning fights based on percentage weight regained.

[6]:
```
#Logit Model

X = win_weight_df_clean[['PERCENT_REGAIN']]


y = win_weight_df_clean['RESULT']

# Add a constant (intercept) to the independent variable
X = sm.add_constant(X)

# Fit the OLS model
model = sm.Logit(y, X)
result = model.fit()
print(result.summary())


math.exp(0.0432)
```

```
Optimization terminated successfully.
         Current function value: 0.684487
         Iterations 4
                           Logit Regression Results
==============================================================================
Dep. Variable:                 RESULT   No. Observations:                  437
Model:                          Logit   Df Residuals:                      435
Method:                           MLE   Df Model:                            1
Date:                Fri, 01 Aug 2025   Pseudo R-squ.:                0.004868
Time:                        08:16:46   Log-Likelihood:                -299.12
converged:                       True   LL-Null:                       -300.58
Covariance Type:            nonrobust   LLR p-value:                   0.08714
================================================================================
==
                     coef    std err          z      P>|z|      [0.025
0.975]
--------------------------------------------------------------------------------
--
const             -0.2334      0.275     -0.849      0.396      -0.772
0.306
PERCENT_REGAIN     0.0432      0.025      1.704      0.088      -0.006
0.093
================================================================================
==
```

[6]: 1.0441467033097327

In this first log-odds model, the coefficient on PERCENT_REGAIN is 0.0432, meaning every 1 percent increase in weight is linked to a 1.044 change in odds of winning. In other words, a 1% increase in weight regained following fight-weigh in is is linked to a 4.4% increase in odds of winning a fight holding all other variables constant.

This model is not statistically significant at alpha = 0.05, but it is significant at alpha =0.10. This means that PERCENT_REGAIN has a statistically signficant impact on winning at a 90% condifidence interval. However, given that the R^2 value is only 0.007, less than 1% of the variation in fight outcome is explained by percentage weight regain. In this way, this model has insufficient predictive power to be useful for fight prediction.

[7]:
```
# testing for causality on randomzied data in logistic model

from sklearn.model_selection import train_test_split

X = win_weight_df_clean[['PERCENT_REGAIN']]

y = win_weight_df_clean['RESULT']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
 ↪random_state=42)

### Logistic Regression
log_model = LogisticRegression()
log_model.fit(X_train, y_train)
y_pred_log = log_model.predict(X_test)

print("Logistic Accuracy:", accuracy_score(y_test, y_pred_log))
print("Logistic AUC:", roc_auc_score(y_test, log_model.predict_proba(X_test)[:,␣
 ↪1]))

### Random Forest Classifier
rf_model = RandomForestClassifier(n_estimators=100, random_state=42)
rf_model.fit(X_train, y_train)
y_pred_rf = rf_model.predict(X_test)

print("\nRF Accuracy:", accuracy_score(y_test, y_pred_rf))
print("RF AUC:", roc_auc_score(y_test, rf_model.predict_proba(X_test)[:, 1]))
```

```
Logistic Accuracy: 0.5568181818181818
Logistic AUC: 0.4728947368421053

RF Accuracy: 0.5227272727272727
RF AUC: 0.4723684210526315
```

In the logistic regression, the performance of the model further confirms that weight regain alone is insufficient to predict fight outcome. Since the AUC values are less than 0.5, the model is worse than random guessing at predicting fight outcomes. In this way, the amount of weight regained by a fighter does not have sufficient predictive power.

4 Regression with Squared Term

In my model, I want to investigate if doing big water cuts has a a signficant fall off past a point. As mentioned earlier, large water cuts can impact the health and performance of fighters, so the purpose of the squared term is to see if there is a point where doing too large of a water-cut negatively impacts the odds of a fighter winning.

[8]:
```
win_weight_df_clean['PERCENT_REGAIN_SQ'] =␣
 ↪win_weight_df_clean['PERCENT_REGAIN'] ** 2


X = win_weight_df_clean[['PERCENT_REGAIN', 'PERCENT_REGAIN_SQ']]
X = sm.add_constant(X)

y = win_weight_df_clean['RESULT']


# Fit logistic regression
logit_model = sm.Logit(y, X)
results = logit_model.fit()

# Print summary
print(results.summary())
```

```
Optimization terminated successfully.
         Current function value: 0.683942
         Iterations 4
                           Logit Regression Results
==============================================================================
Dep. Variable:                 RESULT   No. Observations:                  437
Model:                          Logit   Df Residuals:                      434
Method:                           MLE   Df Model:                            2
Date:                Fri, 01 Aug 2025   Pseudo R-squ.:                0.005661
Time:                        08:16:49   Log-Likelihood:                -298.88
converged:                       True   LL-Null:                       -300.58
Covariance Type:            nonrobust   LLR p-value:                    0.1824
================================================================================
=====
                        coef    std err          z      P>|z|      [0.025
0.975]
--------------------------------------------------------------------------------
-----
const                -0.5447      0.533     -1.022      0.307      -1.589
0.500
```

```
PERCENT_REGAIN        0.1169      0.110      1.058      0.290      -0.100
0.333
PERCENT_REGAIN_SQ    -0.0037      0.005     -0.687      0.492      -0.014
0.007
================================================================================
=====

/var/folders/rc/yw7v7vhj4l3_vjznh0cwm8wr0000gn/T/ipykernel_16016/4279618373.py:3
: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  win_weight_df_clean['PERCENT_REGAIN_SQ'] =
win_weight_df_clean['PERCENT_REGAIN'] ** 2
```

In this third logistics model, the coefficients on PERCENT_REGAIN and PERCENT_REGAIN_SQ are both statsitically insignificant at a 95% and 90% confidence interval, implying that we fail to reject the null hypothesis that regaining weight has a downside at higher levels.

5 Analyzing Impact of Weight Regain on Odds of Winning by Weight Class

Certain weight classes may gain more benefit from weight regains. Based on my intutition, fighers in heavy weight do not have much to gain from drastic weight cuts, whereas the lower classes could see fighters who are a lot heavier do a drastic weight cut to have a signficant size advantadge relative to their competition. To explore this effect, I have done seperate regressions below by weight class.

[14]:
```
import statsmodels.formula.api as smf
import re

# --- STEP 1: Normalize and label weight classes, separating men's and women's␣
↪divisions ---

def clean_weight_class(row):
    wc = row['WEIGHTCLASS']
    sex = row['SEX']

    if pd.isna(wc) or pd.isna(sex):
        return None

    match = re.search(
        ␣
↪r'(Featherweight|Lightweight|Welterweight|Middleweight|Heavyweight|Flyweight|Bantamweight|L
↪Heavyweight|Strawweight|Catchweight)',
        wc
```

```
    )
    base_wc = match.group(1) if match else wc
    return f"Women's {base_wc}" if sex == 'F' else base_wc

# Apply the function row-wise
win_weight_df_clean['WEIGHTCLASS_CLEAN'] = win_weight_df_clean.
 ↪apply(clean_weight_class, axis=1)


# --- STEP 3: Get unique cleaned weight classes ---
weight_classes = win_weight_df_clean['WEIGHTCLASS_CLEAN'].dropna().unique()

# --- STEP 4: Run logistic regression for each weight class ---
results = {}

for wc in weight_classes:
    df_wc = win_weight_df_clean[win_weight_df_clean['WEIGHTCLASS_CLEAN'] == wc].
 ↪copy()

    # Drop missing data
    df_wc = df_wc.dropna(subset=['PERCENT_REGAIN', 'RESULT'])

    # Check data sufficiency
    if len(df_wc) < 10 or df_wc['RESULT'].nunique() < 2:
        print(f"Skipping {wc} - insufficient data or outcome variation.")
        continue

    try:
        # Standardize the predictor
        df_wc['PERCENT_REGAIN_STD'] = (
            df_wc['PERCENT_REGAIN'] - df_wc['PERCENT_REGAIN'].mean()
        ) / df_wc['PERCENT_REGAIN'].std()

        # Run logistic regression
        model = smf.logit("RESULT ~ PERCENT_REGAIN_STD", data=df_wc).fit(disp=0)

        # Store result
        results[wc] = model.summary2().as_text()
        print(f"Model completed for {wc}")

    except Exception as e:
        print(f"  Error in {wc}: {e}")

# --- STEP 5: Print all regression summaries ---
print("\n" + "="*80)
print("LOGISTIC REGRESSION RESULTS BY WEIGHT CLASS")
print("="*80 + "\n")
```

```
for wc, summary in results.items():
    print(f"\n=== {wc.upper()} ===")
    print(summary)
    print("\n" + "-"*80)
```

```
/var/folders/rc/yw7v7vhj4l3_vjznh0cwm8wr0000gn/T/ipykernel_16016/1463475943.py:2
1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  win_weight_df_clean['WEIGHTCLASS_CLEAN'] =
win_weight_df_clean.apply(clean_weight_class, axis=1)
```

```
Model completed for Featherweight
Model completed for Bantamweight
Model completed for Heavyweight
Model completed for Middleweight
Model completed for Women's Strawweight
Model completed for Lightweight
Model completed for Light Heavyweight
Model completed for Women's Bantamweight
Model completed for Flyweight
Model completed for Welterweight
Model completed for Women's Flyweight
Skipping Women's Catch Weight Bout - insufficient data or outcome variation.
Skipping Catch Weight Bout - insufficient data or outcome variation.
Skipping Women's Featherweight - insufficient data or outcome variation.

================================================================================
LOGISTIC REGRESSION RESULTS BY WEIGHT CLASS
================================================================================


=== FEATHERWEIGHT ===
                        Results: Logit
===============================================================
Model:              Logit            Method:           MLE
Dependent Variable: RESULT           Pseudo R-squared: 0.007
Date:               2025-08-01 08:29 AIC:              115.8103
No. Observations:   84               BIC:              120.6719
Df Model:           1                Log-Likelihood:   -55.905
Df Residuals:       82               LL-Null:          -56.281
Converged:          1.0000           LLR p-value:      0.38608
No. Iterations:     4.0000           Scale:            1.0000
---------------------------------------------------------------
```

```
                  Coef.  Std.Err.   z    P>|z|   [0.025 0.975]
---------------------------------------------------------------
Intercept         0.4392   0.2246 1.9555 0.0505 -0.0010 0.8793
PERCENT_REGAIN_STD 0.1946   0.2248 0.8657 0.3866 -0.2460 0.6352
===============================================================


--------------------------------------------------------------------------------


=== BANTAMWEIGHT ===
                        Results: Logit
===============================================================
Model:              Logit            Method:           MLE
Dependent Variable: RESULT           Pseudo R-squared: 0.000
Date:               2025-08-01 08:29 AIC:              97.9949
No. Observations:   68               BIC:              102.4339
Df Model:           1                Log-Likelihood:   -46.997
Df Residuals:       66               LL-Null:          -47.016
Converged:          1.0000           LLR p-value:      0.84600
No. Iterations:     3.0000           Scale:            1.0000
---------------------------------------------------------------
                  Coef.  Std.Err.   z    P>|z|   [0.025 0.975]
---------------------------------------------------------------
Intercept         0.1178   0.2430 0.4849 0.6277 -0.3585 0.5942
PERCENT_REGAIN_STD 0.0475   0.2449 0.1942 0.8460 -0.4324 0.5275
===============================================================


--------------------------------------------------------------------------------


=== HEAVYWEIGHT ===
                        Results: Logit
===============================================================
Model:              Logit            Method:           MLE
Dependent Variable: RESULT           Pseudo R-squared: 0.250
Date:               2025-08-01 08:29 AIC:              20.5553
No. Observations:   17               BIC:              22.2217
Df Model:           1                Log-Likelihood:   -8.2777
Df Residuals:       15               LL-Null:          -11.037
Converged:          1.0000           LLR p-value:      0.018810
No. Iterations:     7.0000           Scale:            1.0000
---------------------------------------------------------------
                  Coef.  Std.Err.   z    P>|z|   [0.025 0.975]
---------------------------------------------------------------
Intercept         1.0070   0.7430 1.3553 0.1753 -0.4493 2.4634
PERCENT_REGAIN_STD 1.6066   0.8972 1.7907 0.0733 -0.1519 3.3651
===============================================================
```

```
--------------------------------------------------------------------------------

=== MIDDLEWEIGHT ===
                        Results: Logit
===============================================================
Model:              Logit            Method:           MLE
Dependent Variable: RESULT           Pseudo R-squared: 0.057
Date:               2025-08-01 08:29 AIC:              97.3205
No. Observations:   72               BIC:              101.8738
Df Model:           1                Log-Likelihood:   -46.660
Df Residuals:       70               LL-Null:          -49.461
Converged:          1.0000           LLR p-value:      0.017940
No. Iterations:     5.0000           Scale:            1.0000
---------------------------------------------------------------
                   Coef.  Std.Err.   z    P>|z|   [0.025 0.975]
---------------------------------------------------------------
Intercept          0.2463   0.2475 0.9950 0.3197 -0.2388 0.7314
PERCENT_REGAIN_STD 0.6016   0.2696 2.2312 0.0257  0.0731 1.1301
===============================================================


--------------------------------------------------------------------------------

=== WOMEN'S STRAWWEIGHT ===
                        Results: Logit
===============================================================
Model:               Logit            Method:           MLE
Dependent Variable:  RESULT           Pseudo R-squared: 0.002
Date:                2025-08-01 08:29 AIC:              66.2260
No. Observations:    46               BIC:              69.8833
Df Model:            1                Log-Likelihood:   -31.113
Df Residuals:        44               LL-Null:          -31.186
Converged:           1.0000           LLR p-value:      0.70322
No. Iterations:      4.0000           Scale:            1.0000
---------------------------------------------------------------
                   Coef.  Std.Err.   z    P>|z|   [0.025 0.975]
---------------------------------------------------------------
Intercept          0.3526   0.3000 1.1755 0.2398 -0.2353 0.9406
PERCENT_REGAIN_STD 0.1163   0.3068 0.3790 0.7047 -0.4851 0.7177
===============================================================


--------------------------------------------------------------------------------

=== LIGHTWEIGHT ===
                        Results: Logit
===============================================================
```

```
Model:              Logit            Method:           MLE
Dependent Variable: RESULT           Pseudo R-squared: 0.110
Date:               2025-08-01 08:29 AIC:              24.9277
No. Observations:   17               BIC:              26.5941
Df Model:           1                Log-Likelihood:   -10.464
Df Residuals:       15               LL-Null:          -11.754
Converged:          1.0000           LLR p-value:      0.10819
No. Iterations:     5.0000           Scale:            1.0000
---------------------------------------------------------------
                   Coef.  Std.Err.   z    P>|z|   [0.025 0.975]
---------------------------------------------------------------
Intercept          0.1161   0.5260 0.2207 0.8253 -0.9149 1.1471
PERCENT_REGAIN_STD 0.8601   0.5730 1.5009 0.1334 -0.2631 1.9832
===============================================================
```

```
--------------------------------------------------------------------------------
```

```
=== LIGHT HEAVYWEIGHT ===
                         Results: Logit
=================================================================
Model:              Logit            Method:            MLE
Dependent Variable: RESULT           Pseudo R-squared:  0.004
Date:               2025-08-01 08:29  AIC:               52.3190
No. Observations:   35               BIC:               55.4297
Df Model:           1                Log-Likelihood:    -24.159
Df Residuals:       33               LL-Null:           -24.246
Converged:          1.0000           LLR p-value:       0.67768
No. Iterations:     4.0000           Scale:             1.0000
-----------------------------------------------------------------
                    Coef.  Std.Err.    z    P>|z|   [0.025 0.975]
-----------------------------------------------------------------
Intercept          -0.0574   0.3390 -0.1694 0.8655 -0.7219 0.6071
PERCENT_REGAIN_STD  0.1434   0.3468  0.4134 0.6793 -0.5363 0.8230
=================================================================
```

```
--------------------------------------------------------------------------------
```

```
=== WOMEN'S BANTAMWEIGHT ===
                        Results: Logit
===============================================================
Model:              Logit            Method:           MLE
Dependent Variable: RESULT           Pseudo R-squared: 0.003
Date:               2025-08-01 08:29 AIC:              26.0112
No. Observations:   17               BIC:              27.6777
Df Model:           1                Log-Likelihood:   -11.006
Df Residuals:       15               LL-Null:          -11.037
```

```
Converged:          1.0000           LLR p-value:      0.80150
No. Iterations:     4.0000           Scale:            1.0000
---------------------------------------------------------------
                   Coef.  Std.Err.   z    P>|z|   [0.025 0.975]
---------------------------------------------------------------
Intercept          0.6086   0.5089 1.1961 0.2317 -0.3887 1.6060
PERCENT_REGAIN_STD 0.1327   0.5305 0.2500 0.8025 -0.9072 1.1725
===============================================================


--------------------------------------------------------------------------------


=== FLYWEIGHT ===
                         Results: Logit
=================================================================
Model:              Logit            Method:           MLE
Dependent Variable: RESULT           Pseudo R-squared: 0.006
Date:               2025-08-01 08:29 AIC:              32.5088
No. Observations:   21               BIC:              34.5978
Df Model:           1                Log-Likelihood:   -14.254
Df Residuals:       19               LL-Null:          -14.341
Converged:          1.0000           LLR p-value:      0.67712
No. Iterations:     4.0000           Scale:            1.0000
-----------------------------------------------------------------
                    Coef.  Std.Err.    z    P>|z|   [0.025 0.975]
-----------------------------------------------------------------
Intercept          -0.2899   0.4429 -0.6546 0.5127 -1.1580 0.5782
PERCENT_REGAIN_STD -0.1882   0.4526 -0.4158 0.6776 -1.0752 0.6989
=================================================================


--------------------------------------------------------------------------------


=== WELTERWEIGHT ===
                         Results: Logit
=================================================================
Model:              Logit            Method:           MLE
Dependent Variable: RESULT           Pseudo R-squared: 0.010
Date:               2025-08-01 08:29 AIC:              40.1483
No. Observations:   27               BIC:              42.7400
Df Model:           1                Log-Likelihood:   -18.074
Df Residuals:       25               LL-Null:          -18.249
Converged:          1.0000           LLR p-value:      0.55392
No. Iterations:     4.0000           Scale:            1.0000
-----------------------------------------------------------------
                    Coef.  Std.Err.    z    P>|z|   [0.025 0.975]
-----------------------------------------------------------------
Intercept          -0.3782   0.3944 -0.9591 0.3375 -1.1512 0.3947
```

```
PERCENT_REGAIN_STD  0.2353   0.3998  0.5884 0.5562 -0.5483 1.0189
=================================================================


--------------------------------------------------------------------------------


=== WOMEN'S FLYWEIGHT ===
                        Results: Logit
===============================================================
Model:                Logit            Method:            MLE
Dependent Variable:   RESULT           Pseudo R-squared: 0.010
Date:                 2025-08-01 08:29 AIC:               36.2799
No. Observations:     24               BIC:               38.6361
Df Model:             1                Log-Likelihood:    -16.140
Df Residuals:         22               LL-Null:           -16.301
Converged:            1.0000           LLR p-value:       0.57081
No. Iterations:       4.0000           Scale:             1.0000
---------------------------------------------------------------
                   Coef.  Std.Err.   z    P>|z|   [0.025 0.975]
---------------------------------------------------------------
Intercept          0.3419   0.4173 0.8194 0.4126 -0.4760 1.1598
PERCENT_REGAIN_STD 0.2446   0.4380 0.5584 0.5766 -0.6139 1.1031
===============================================================


--------------------------------------------------------------------------------
```

[10]:
```python
print(math.exp(0.6016))

print(math.exp(1.6066))
```

```
1.8250365240275959
4.985830553163939
```

- For fighters in the middleweight division, every 1 percent increase in weight is linked to a 1.83 change in odds of winning. In other words, a 1% increase in weight regained following fight-weigh in is is linked to a 83% increase in odds of winning a fight holding all other variables constant. This result is statistically signficant at a 95% confidence level. Intutitively, this increase in odds makes sense because of the middleweight division's relative position compared to other weight classes. Many of the competitive middleweights, such as Alex Pereira who is 6"4 and walks around at 220+ pounds, are big enough to fight in the light-heavyweight division. These fighters that make the cut to middleweight are signficantly bigger than their competitors at middleweight. On the otherhand, there are also fighters in the middleweight division that are too heavy to fight in welterweight so they end up moving to middleweight to be more competitive, such as Robert Whittiker or Sean Strickland, both of whom were former champions at middleweight but were uncompetitive at welterweight. Since middleweight exists in an akward position between welterweight and light-heavyweight, it could be the case that the middle weights that have bigger frames (in other words the ones that can cut more weight) have a statistically significant advantadge over the middleweights that have smaller frames and are unable to cut as much weight.

- For fighters in the heavyweight division, every 1 percent increase in weight is linked to a 4.99 change in odds of winning. In other words, a 1% increase in weight regained following fight-weigh in is is linked to an approximate 500% increase in odds of winning a fight holding all other variables constant. This result is statistically signficant at a 90% confidence level. Since this model has a pseudo R^2 value of 0.25, there is a strong relationship between percentage of weight regained and the odds of winning for heavyweight fighters.

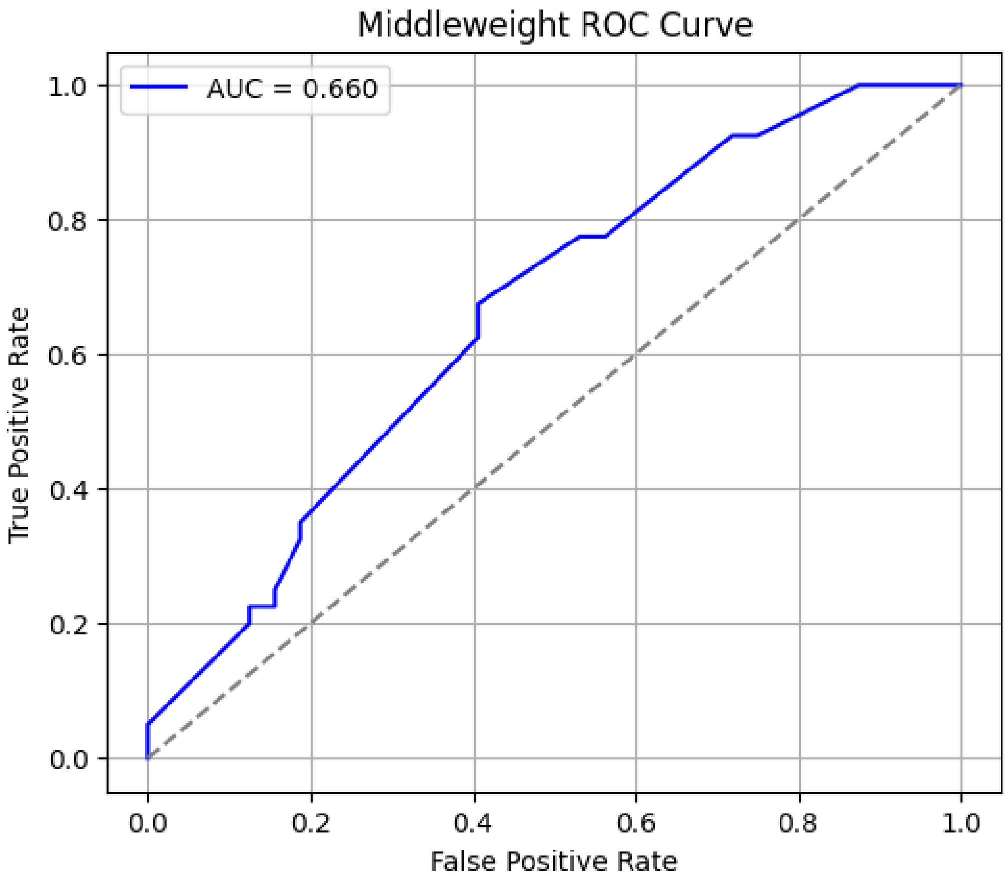
- The result for heavyweight went against my initial intution because I assumed that heavyweights have the least to gain from cutting weight, since they are already in the highest weight class. However, the implication for heavyweights who cut is that they are in better physical conditioning and lose water weight so they can pack on more muscle mass prior to their fight. Typically weight cutting is not as prevalent at heavyweight because of the fact it is the max weight class, leading to many in the division having poorer conditioning relative to other weight classes. In this way, the heavyweight fighters that do cut weight unlike their competitors have a signficant advantadge. This can be seen in fighters like Francis Ngannou who are in heavyweight but are in considerably better physical shape compared to their competitors like Derrick Lewis or Daniel Cormier.

[15]:
```
def likelihood_ratio_test(full_model):
    llf_full = full_model.llf
    llf_null = full_model.llnull
    df_full = full_model.df_model
    df_null = 0  # Null model has only intercept

    lr_stat = 2 * (llf_full - llf_null)
    p_value = chi2.sf(lr_stat, df=df_full)

    return {"LR statistic": lr_stat, "df": df_full, "p-value": p_value}

def plot_roc_auc(model, data, title="ROC Curve"):
    data = data.copy()
    y_true = data['RESULT']
    y_scores = model.predict()

    fpr, tpr, thresholds = roc_curve(y_true, y_scores)
    auc = roc_auc_score(y_true, y_scores)

    plt.figure(figsize=(6, 5))
    plt.plot(fpr, tpr, label=f"AUC = {auc:.3f}", color='blue')
    plt.plot([0, 1], [0, 1], '--', color='gray')
    plt.xlabel("False Positive Rate")
    plt.ylabel("True Positive Rate")
    plt.title(title)
    plt.legend()
    plt.grid(True)
```

```
    plt.show()

    return auc
```

[16]:
```
df_middleweight = win_weight_df_clean[win_weight_df_clean['WEIGHTCLASS_CLEAN']␣
 ↪== 'Middleweight'].dropna(subset=['RESULT', 'PERCENT_REGAIN'])

# Standardize predictor
df_middleweight['PERCENT_REGAIN_STD'] = (
    df_middleweight['PERCENT_REGAIN'] - df_middleweight['PERCENT_REGAIN'].mean()
) / df_middleweight['PERCENT_REGAIN'].std()

# Fit model
model_middle = smf.logit("RESULT ~ PERCENT_REGAIN_STD", data=df_middleweight).
 ↪fit(disp=0)

# Likelihood Ratio Test
print("Likelihood Ratio Test:")
print(likelihood_ratio_test(model_middle))


# ROC Curve & AUC
print("\nROC Curve and AUC:")
auc = plot_roc_auc(model_middle, df_middleweight, title="Middleweight ROC␣
 ↪Curve")
print(f"AUC: {auc:.3f}")
```

```
Likelihood Ratio Test:
{'LR statistic': 5.602016889689978, 'df': 1.0, 'p-value': 0.01793981346892276}

ROC Curve and AUC:
```




```
AUC: 0.660
```

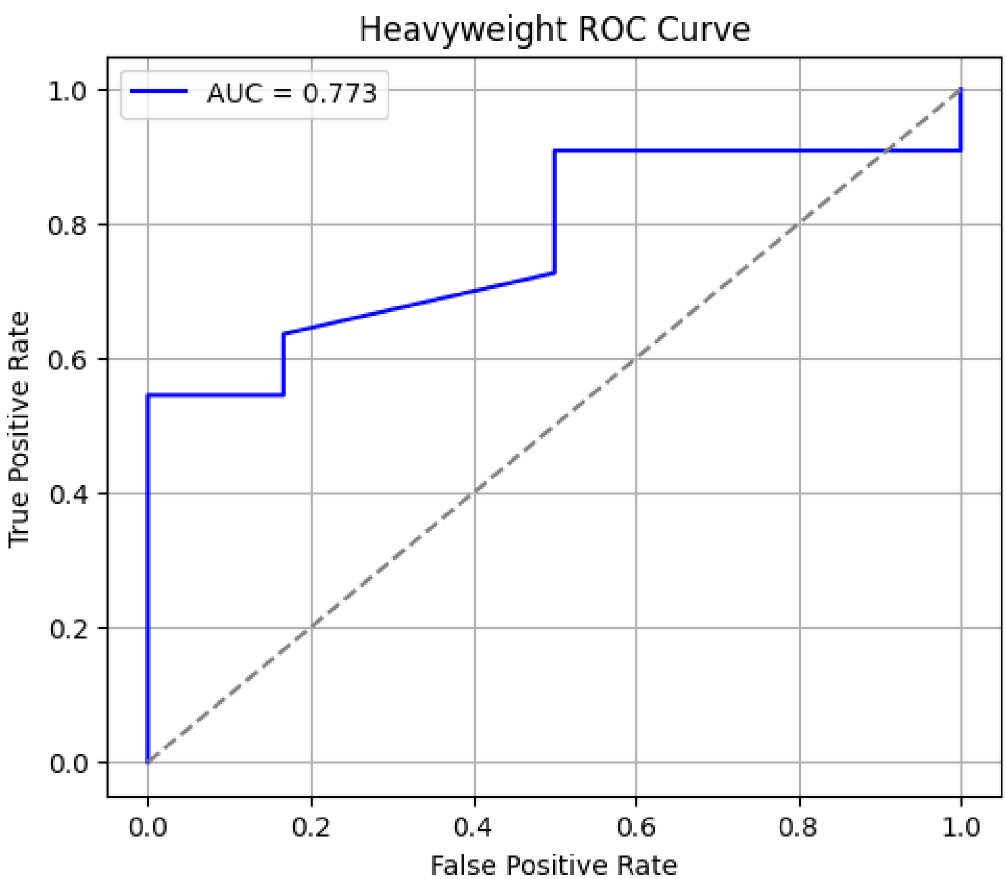
The AUC value for the model for the Middleweight division is 0.66. This value suggests that the model is better at predicting which fighter wins than random guessing, but is not a perfect predictor since it is closer to 0.5 than 1.

Additionally, the likelihood ratio statistic is statistically signficant at a 95% confidence interval, meaning that middleweight fighters that regain a greater percentage of weight between weigh-in and fight night are signficantly increasing their odds of winning the fight with all other variables being held constant.

[13]:
```
# Note to self: do above analysis for Heavyweight
df_heavyweight= win_weight_df_clean[win_weight_df_clean['WEIGHTCLASS_CLEAN'] ==␣
 ↪'Heavyweight'].dropna(subset=['RESULT', 'PERCENT_REGAIN'])

# Standardize predictor
df_heavyweight['PERCENT_REGAIN_STD'] = (
    df_heavyweight['PERCENT_REGAIN'] - df_heavyweight['PERCENT_REGAIN'].mean()
) / df_heavyweight['PERCENT_REGAIN'].std()
```

```
# Fit model
model_heavy = smf.logit("RESULT ~ PERCENT_REGAIN_STD", data=df_heavyweight).
  ↪fit(disp=0)

# Likelihood Ratio Test
print("Likelihood Ratio Test:")
print(likelihood_ratio_test(model_heavy))


# ROC Curve & AUC
print("\nROC Curve and AUC:")
auc = plot_roc_auc(model_heavy, df_heavyweight, title="Heavyweight ROC Curve")
print(f"AUC: {auc:.3f}")
```

```
Likelihood Ratio Test:
{'LR statistic': 5.519138799373273, 'df': 1.0, 'p-value': 0.01880951646509554}

ROC Curve and AUC:
```



```
AUC: 0.773
```

The AUC value for the model for the Heavyweight division is 0.773. This value suggests that the model is better at predicting which fighter wins than random guessing. Relative to the model for the middleweight division, the Heavyweight model has better predictive power because the AUC value is much closer to 1.

Furthermore, the likelihood ratio statistic is statistically signficant at a 95% confidence interval, meaning that heavyweight fighters that regain a greater percentage of weight between weigh-in and fight night are signficantly increasing their odds of winning the fight with all other variables being held constant.

6 Limitation to Analysis of Weight Regain Models and Room to Expand

The biggest limitation to this analysis is the data set. Not all regions require fighters to publicize their fight night weight, meaning a lot of weight regains are not capture in my analysis. Having more post fight weigh in data could let us more accurately assess if there is a limitation to the benefit of cutting and regaining weight.

Furthermore, although the model for the heavyweight divsion yielded strong results, the sample size for the heavyweight division was small at n=17. Therefore, my analysis could be made more robust with a larger sample of heavyweight fights.

[ ]: