Data Vis Final Project

January 15, 2020

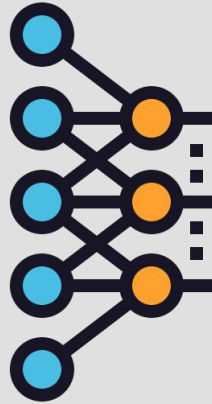
Quy, Justin, Rafael, Sunny

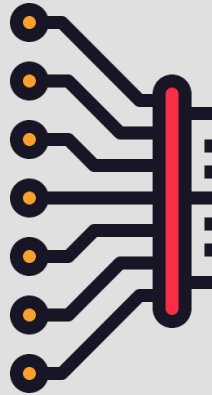
The Team

Project objective:

Utilize machine learning library to create a tool to predict outcome of UFC fights

Tools Used:
SciKit-Learn, Pandas, & Matplotlib





Approach

Step 1

kaggle UFC

Clean dataset from Kaggle to convert all strings

Eliminate unnecessary variables & organize df

Step 2

scikit learn

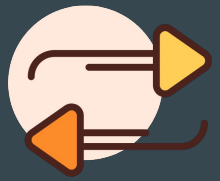
Identify relevant features to be used for ML training

Build via various models: Logistic Regression at first, then we explored Decision Tree & Random Forest

Step 3

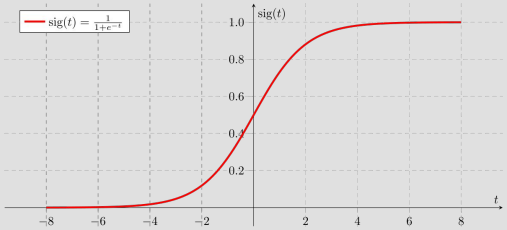
Ensure fit & accuracy of model by comparing predictions vs. actual results (test)

Step 4



Rinse & Repeat Step 2 & Step 3 until accuracy is at an acceptable rate

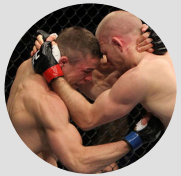
Our Model's Features



Match Data

Head Strikes

Body Strikes

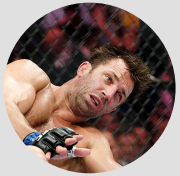


Clinches

Leg Strikes

Ground & Pounds

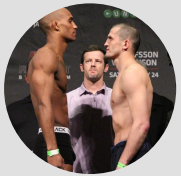
Submissions



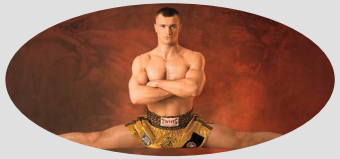
Knockdowns

Takedowns

Fighter Data



Height



Reach

Age

Calculating Accuracy

```
In [18]: predictions = classifier.predict(X_test)
         print(f"First 10 Predictions:   {predictions[:10]}")
         print(f"First 10 Actual labels: {y_test[:10].values.tolist()}")

         First 10 Predictions:   ['1' '0' '0' '0' '0' '0' '1' '1' '1' '0']
         First 10 Actual labels: [['1'], ['1'], ['1'], ['0'], ['1'], ['0'], ['1'], ['1'], ['0'], ['0']]

In [19]: accuracy = classifier.score(X_test, y_test)
         print('The accuracy is: ' + str(accuracy *100) + '%')

         The accuracy is: 80.95854922279793%

In [20]: pd.DataFrame({"Prediction": predictions, "Actual": y_test['new_column']}).reset_index(drop=True)
```

Out[20]:

	Prediction	Actual
0	1	1
1	0	1
2	0	1
3	0	0
4	0	1
...	...	...
1539	0	0
1540	0	1
1541	0	0
1542	0	0
1543	1	1

1544 rows × 2 columns

*We converted results to represent:
1 = Red Side Win, 0 = Blue Side Win

80% accuracy using Logistic Regression...

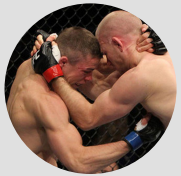
Can we do better?

Our New Model's Features

Match Data

Head Strikes DIFFERENCE

Body Strikes DIFFERENCE

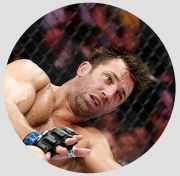


Clinches DIFFERENCE

Leg Strikes DIFFERENCE

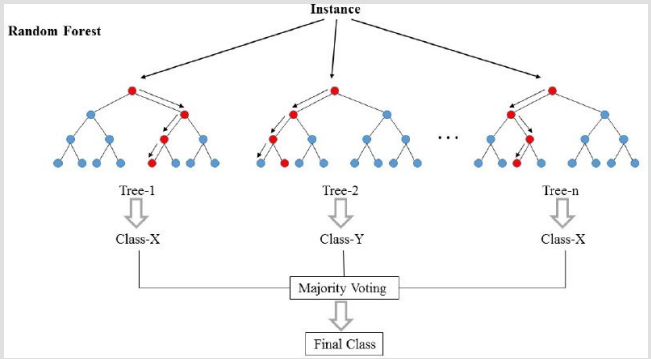
Ground & Pounds DIFFERENCE

Submissions DIFFERENCE



Knockdowns DIFFERENCE

Takedowns DIFFERENCE



Final Accuracy

```
In [81]: diff_df = pd.DataFrame(columns = [diff_df_cols])
         for x in range (len(diff_df_cols)):
             diff_df[diff_df_cols[x]] = R[R_COL[x]] - B[B_COL[x]]
         #diff_df

         diff_train, diff_test, diff_y_train, diff_y_test = train_test_split(diff_df, y2, test_size=0.3)

In [82]: from sklearn.ensemble import RandomForestClassifier
         from sklearn.datasets import make_classification

In [83]: random_forest = RandomForestClassifier(n_estimators=100,max_depth=6,random_state=1)
         random_forest.fit(diff_train,diff_y_train)

         C:\Users\Public\Anaconda3\lib\site-packages\ipykernel_launcher.py:2: DataConversionWarning: A column-vector y was pas
         sed when a 1d array was expected. Please change the shape of y to (n_samples,), for example using ravel().

Out[83]: RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini',
                                 max_depth=6, max_features='auto', max_leaf_nodes=None,
                                 min_impurity_decrease=0.0, min_impurity_split=None,
                                 min_samples_leaf=1, min_samples_split=2,
                                 min_weight_fraction_leaf=0.0, n_estimators=100,
                                 n_jobs=None, oob_score=False, random_state=1, verbose=0,
                                 warm_start=False)

In [84]: diff_test_results = random_forest.predict(diff_test)
         print(f"Training Data Score: {random_forest.score(diff_train, diff_y_train)}")
         print(f"Testing Data Score: {random_forest.score(diff_test, diff_y_test)}")

         Training Data Score: 0.83515731874145
         Testing Data Score: 0.8014354066985646
```

83%

Training Data

81%

Test Data

accuracy using “difference” features & Random Forest

“Future” Predictions

Due to our Kaggle dataset ending on June of 2019, we decided to leverage our model to predict 2H 2019 matches...

Korean Zombie v. Frankie Edgar
UFC Fight Night 165

Our Model Prediction	Vegas Odds	Actual Result
Korean Zombie	Zombie -170 Edgar +140	Korean Zombie (R1 KO/KTO)

Kamaru Usman v. Colby Covington
UFC 245

Our Model Prediction	Vegas Odds	Actual Result
Usman	Usman -175 Covington +145	Usman (R5 KO/TKO)

Thank You!