Hero or Villain?
Predicting Superhero Alignment

By Jason Masurovsky

GWU Data Analytics and Visualization Bootcamp

January 2021

Topic Choice

- Comics have been around since 1930s
- In the past decade, DC and Marvel have skyrocketed film adaptations
- Recent years, datasets on comic books and characters have been uploaded and open to the public
- Few data viz projects have been made and not many have focused on using machine learning

Research Questions

- Can we classify a superhero as good or evil based off their physical characteristics, superpowers, and abilities?
- Is there a bias in the dataset?
- Which features are most influential?

Dataset

- All the datasets were found on Kaggle
- Kaggle is webportal for the data science community
- Hosts a wide range of datasets and tooltips as open data
- Source data: https://www.kaggle.com/dannielr/marvel-superheroes
- "Characters_stats.csv", "marvel_characters_info.csv", and "superheroes_power_matrix.csv"

Data Cleaning

- Missing values
  - Alignment
- Duplicates
  - Names
- Convert height, weight to feet and lbs.
  - Create a BMI field
- Delete any unnecessary fields
- 517 characters in final dataset

Data Analysis

- Completed in jupyter notebooks using Python and in Tableau
- Target variable: Alignment (good or evil)
- Distribution count of alignment
  - Grouped by gender, race, superpower
- Average ability stats and gen label
- Pearson Correlation on character stats such as strength, power, durability, combat, speed, intelligence
  - Which are correlated?

Data Analysis

Alightment Count
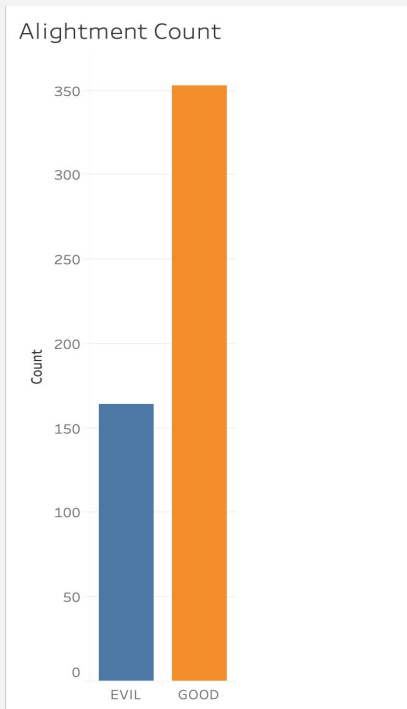
Breakdown of character alignment

- 353 good characters
- 164 evil characters

Data Analysis

Breakdown of character alignment by gender

- Good: 243 male, 110 female
- Evil: 136 male, 28 female

Data Analysis

- Breakdown of good and evil characters by their race
- Good chunk of data is Unknown, Human, and Mutant
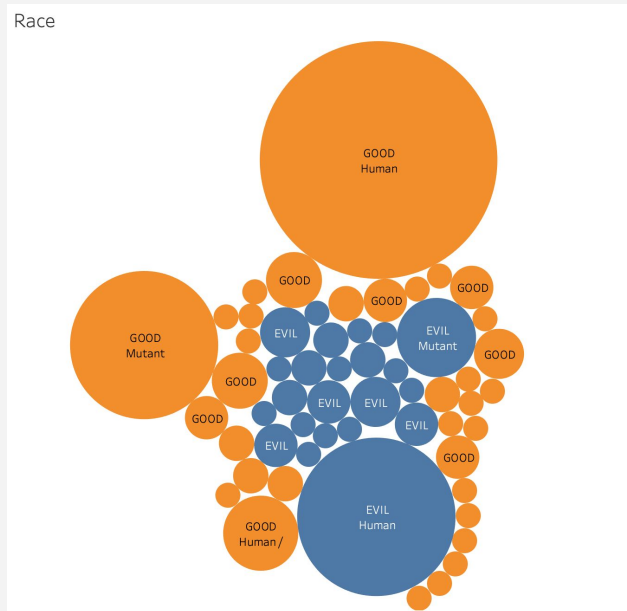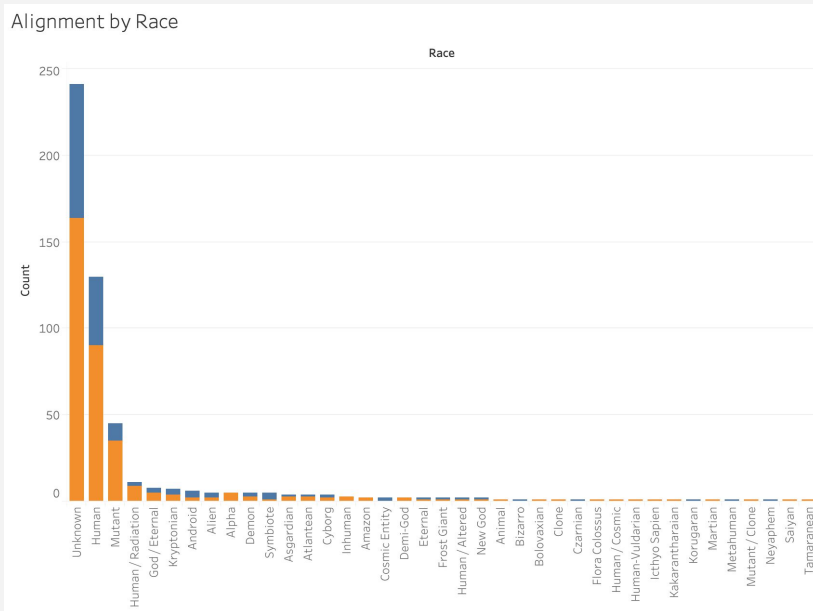
Data Analysis

Average Abilities by Alignment
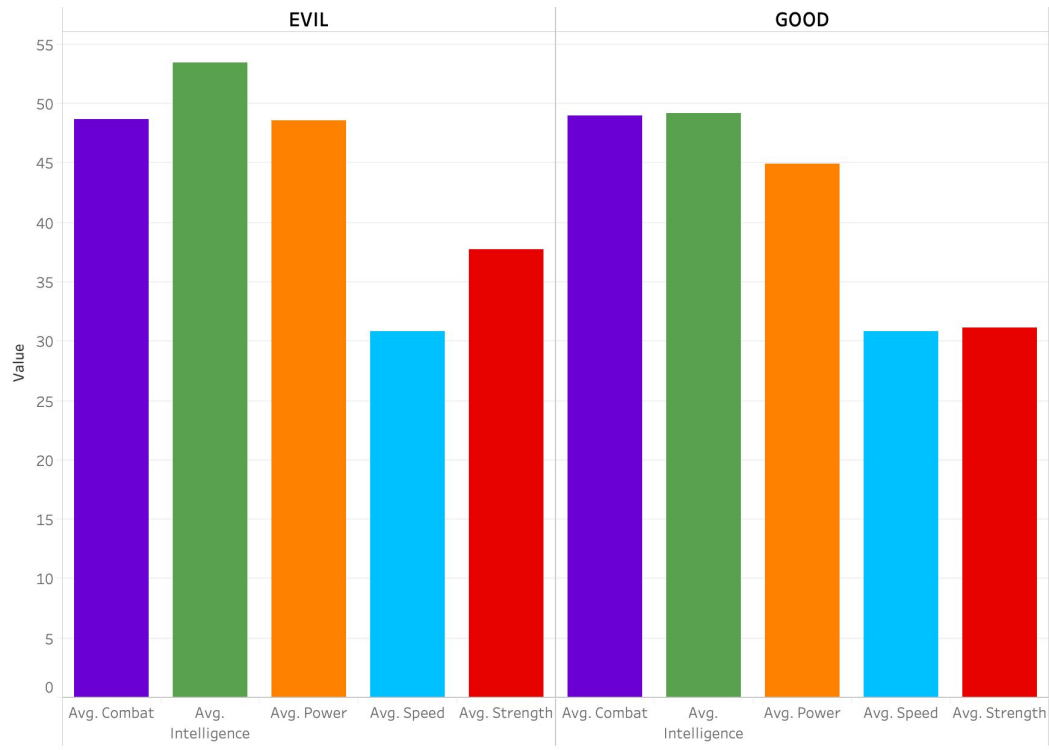
- For all characters.. intelligence, power, and combat have the highest values

Data Analysis

- Pearson correlation matrix of character abilities
- Strength and Durability
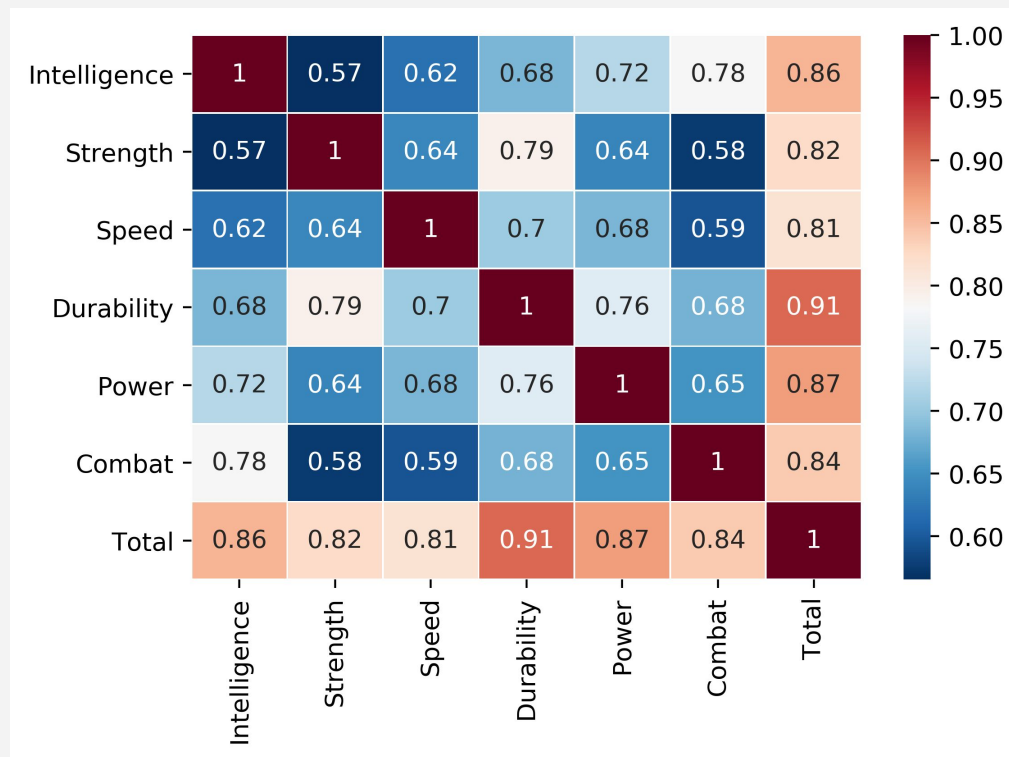
Machine Learning Models

- Classifying a character as good or evil using "Alignment" as target variable
    - Logistic Regression
    - Random Forest Classifier
    - Decision Tree
    - Support Vector Machine (SVM)
- Compare models

ML data pre-processing

- Remove highly correlated features
- Convert 'Alignment' and 'Gender' to binary variables represented as 1 and 0
- Get only true dummy values for race, hair color, eye color and of superhero abilities (agility, invisibility, web creation, and so on)
- Convert superhero matrix (strength, power, intelligence, etc..) as weights by label encoding using adaptive binning (range from 0-3 using IQRs)
- ROC (receiver operating characteristic) curve
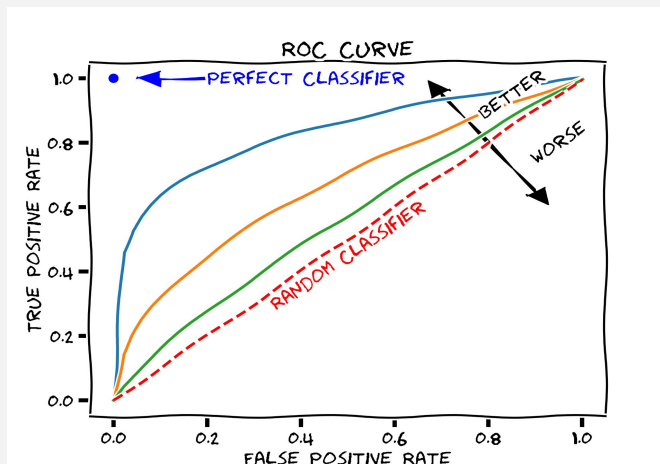
Logistic Regression Results

```python
from sklearn.metrics import accuracy_score
print(accuracy_score(y_test, y_pred))
```

```
0.7307692307692307
```

```python
matrix = confusion_matrix(y_test, y_pred)
print(matrix)
```

```
[[17 24]
 [11 78]]
```

```python
report = classification_report(y_test, y_pred)
print(report)
```

```
              precision    recall  f1-score   support

           0       0.61      0.41      0.49        41
           1       0.76      0.88      0.82        89

    accuracy                           0.73       130
   macro avg       0.69      0.65      0.65       130
weighted avg       0.72      0.73      0.71       130
```


Receiver operating characteristic — Logistic Regression (area = 0.65)
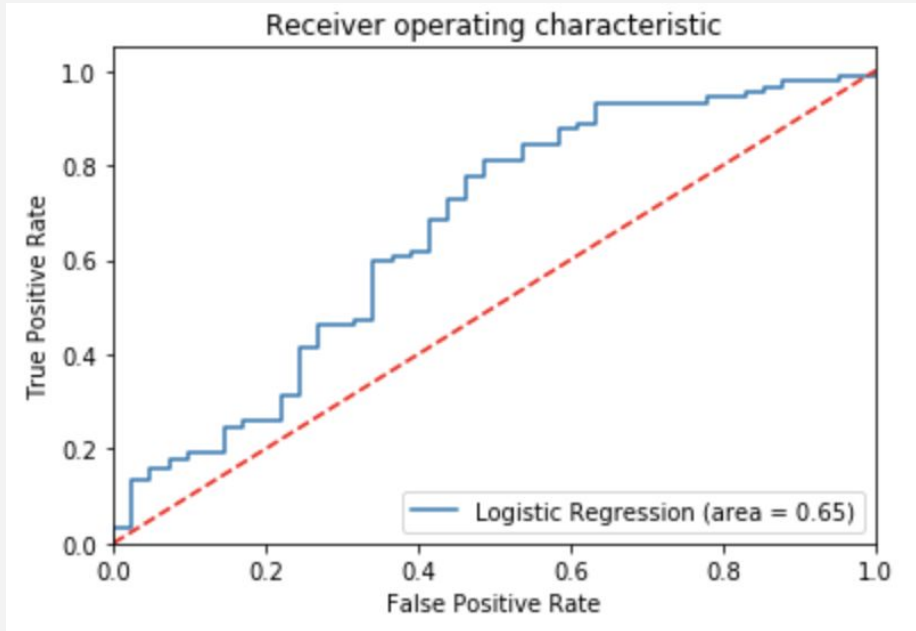
Random Forest Classifier Results

```
Confusion Matrix

            Predicted 0   Predicted 1
Actual 0        12            27
Actual 1         5            86

Accuracy Score : 0.7538461538461538
Classification Report
               precision    recall  f1-score   support

           0       0.71      0.31      0.43        39
           1       0.76      0.95      0.84        91

    accuracy                           0.75       130
   macro avg       0.73      0.63      0.64       130
weighted avg       0.74      0.75      0.72       130
```

```python
# We can sort the features by their importance.
sorted(zip(rf_model.feature_importances_, X.columns), reverse=True)

[(0.033722566224920196, 'Intelligence_Gen_Label'),
 (0.03041071215605239, 'Speed_Gen_Label'),
 (0.025588502516218917, 'Publisher_Image Comics'),
 (0.0244352535936276, 'Power_Gen_Label'),
 (0.02273016367930574, 'Combat_Gen_Label'),
 (0.02265772044235813, 'Strength_Gen_Label'),
 (0.020032354416435135, 'Publisher_Marvel Comics'),
 (0.019841312393854484, 'HairColor_Black'),
 (0.017236903491103277, 'EyeColor_blue'),
 (0.016585731633076202, 'HairColor_Blond'),
 (0.016231862025821584, 'Gender'),
 (0.016000038814074413, 'Flight_True'),
 (0.015234202957567125, 'Durability_y_True'),
 (0.015206137168497837, 'Super Strength_True'),
 (0.014924949088187301, 'Stamina_True'),
 (0.014632487961048291, 'Publisher_DC Comics'),
 (0.014419925708228903, 'Energy Blasts_True'),
 (0.014441265992446499, 'HairColor_No Hair'),
```


Receiver operating characteristic

Random Forest (area = 0.63)

Random Forest Classifier Results

Feature Importance

Feature	Importance
Intelligence_Gen_Label	0.0338
Speed_Gen_Label	0.0305
Publisher_Image Comics	0.0256
Power_Gen_Label	0.0245
Combat_Gen_Label	0.0225
Strength_Gen_Label	0.0225
Publisher_Marvel Comics	0.0200
HairColor_Black	0.0197
EyeColor_blue	0.0173
HairColor_Blond	0.0164
Gender	0.0162
Flight_True	0.0160
Durability_y_True	0.0152
Super Strength_True	0.0152
Stamina_True	0.0148
Publisher_DC Comics	0.0146
Energy Blasts_True	0.0145
HairColor_No Hair	0.0144
Race_-	0.0142
Weapon-based Powers_Tr..	0.0130
Reflexes_True	0.0124
Race_Human	0.0123
HairColor_Brown	0.0116
EyeColor_brown	0.0108
Natural Weapons_True	0.0108
Super Speed_True	0.0104
Intelligence_y_True	0.0100

Decision Tree Results

```
Confusion Matrix


              Predicted 0   Predicted 1

 Actual 0         20            27

 Actual 1         26            83


Accuracy Score : 0.6602564102564102
Classification Report
              precision    recall   f1-score    support

         0       0.43       0.43       0.43         47
         1       0.75       0.76       0.76        109

  accuracy                             0.66        156
 macro avg       0.59       0.59       0.59        156
weighted avg     0.66       0.66       0.66        156
```
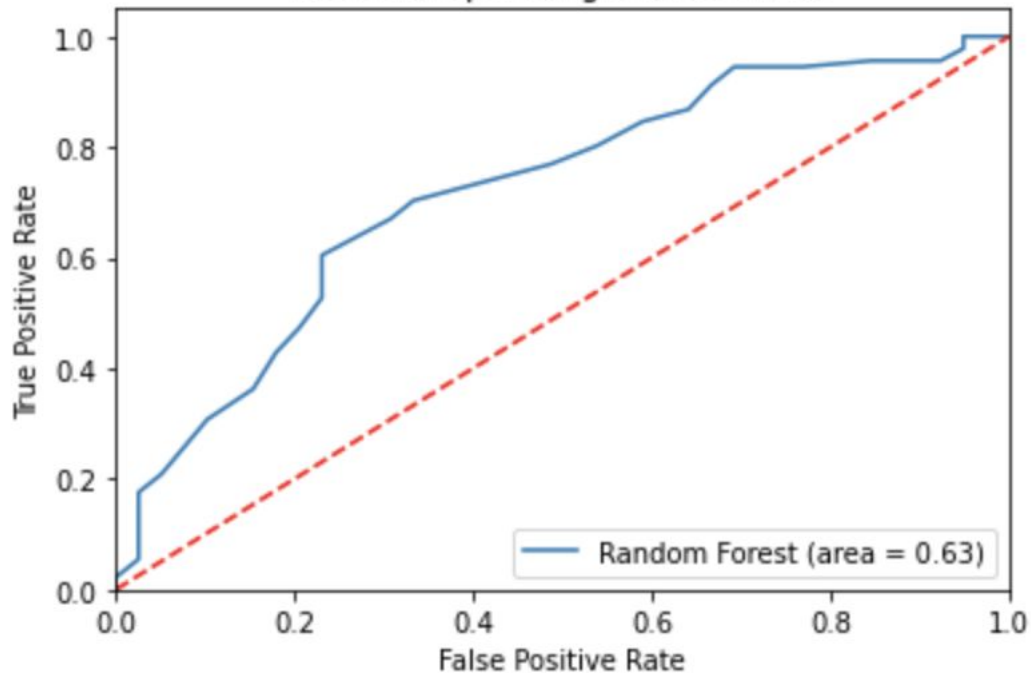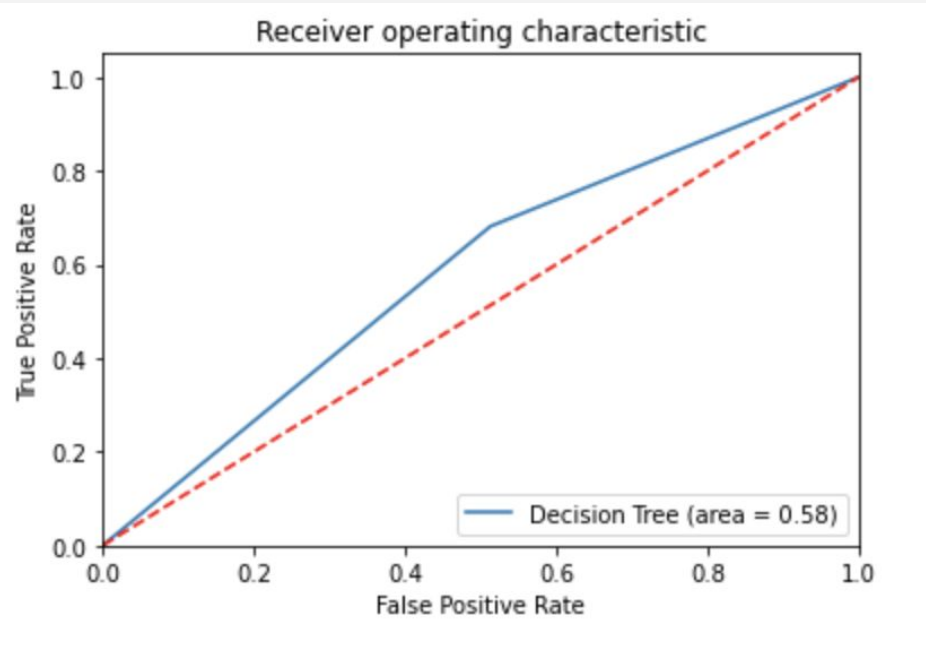
Support Vector Machine Results

```python
from sklearn.metrics import accuracy_score
accuracy_score(y_test, y_pred)
```

```
0.6384615384615384
```

```python
from sklearn.metrics import confusion_matrix
confusion_matrix(y_test, y_pred)
```
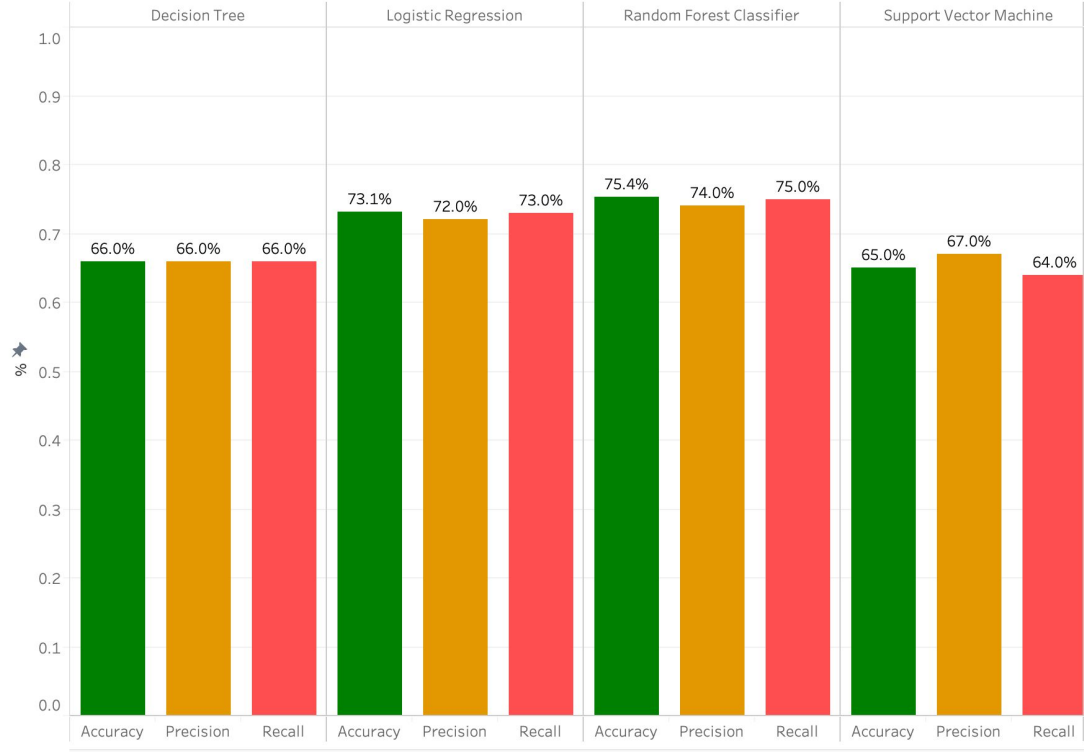
```
array([[24, 17],
       [30, 59]])
```

```python
from sklearn.metrics import classification_report
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       0.44      0.59      0.51        41
           1       0.78      0.66      0.72        89

    accuracy                           0.64       130
   macro avg       0.61      0.62      0.61       130
weighted avg       0.67      0.64      0.65       130
```

Model Results

Model Results

	Decision Tree	Logistic Regression	Random Forest Classifier	Support Vector Machine

Future Work

- Increase # of characters
- Create tiers based off of character ability stats
- Investigate similarities using k-means clustering
- Relationship between ability stats using linear regression

Tableau Dashboard

[Link to dashboard](#)