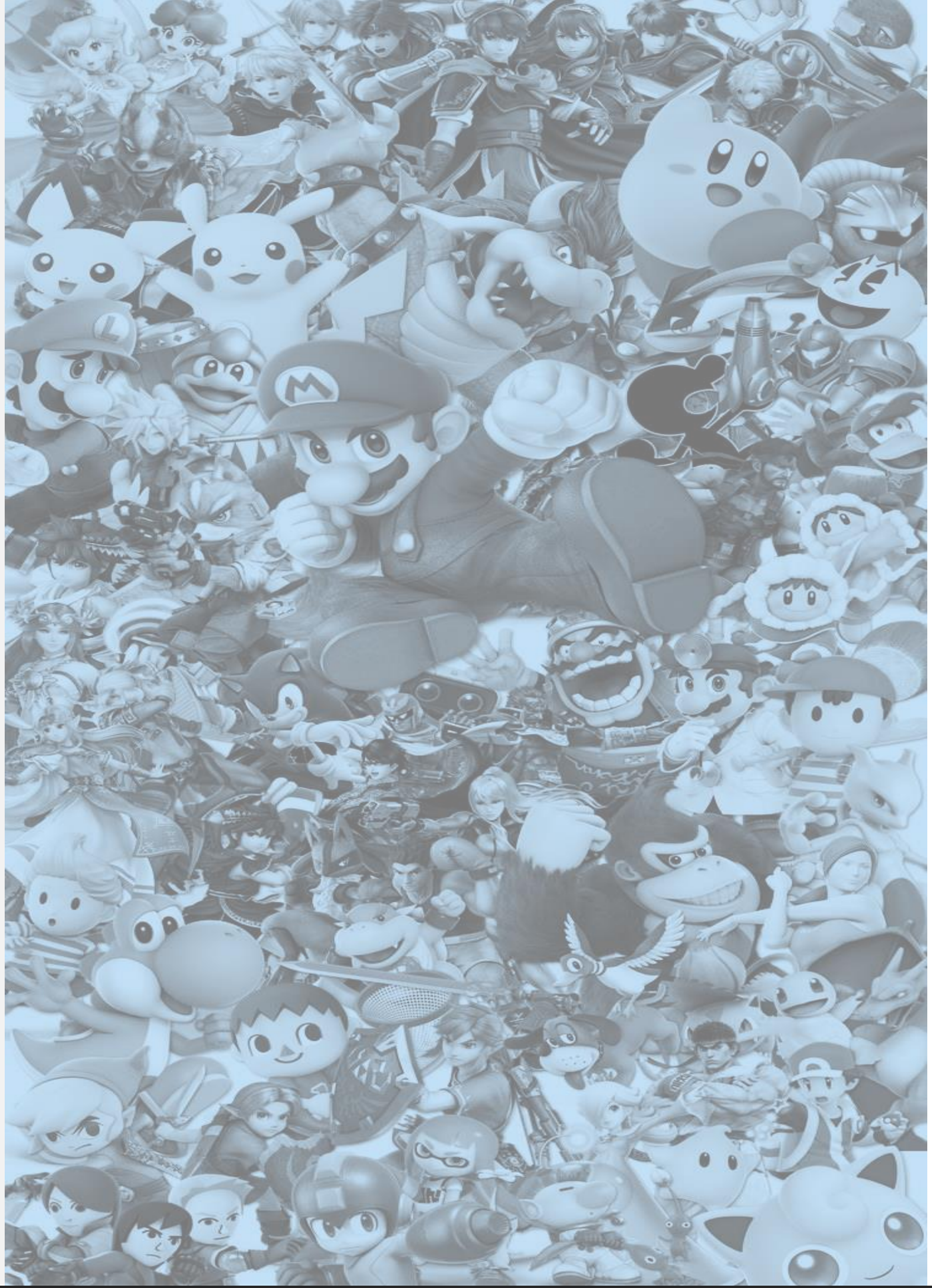
Project #4: Marvel / DC

Christina Chau
Nitin Dhiman
Robert Harvey
Justin Kiang
Xander Polny

Presentation Layout

- Project Focus
- Where we started, and how we landed on this topic — Videogames, Movie Franchise, etc.
- Data sources
- Notebook — Jupyter, Tableau, Google Collab
- Visualizations/Models
- Results/So what?
- Limitations
- Questions



Where We Started

Initially started with looking at Videogames

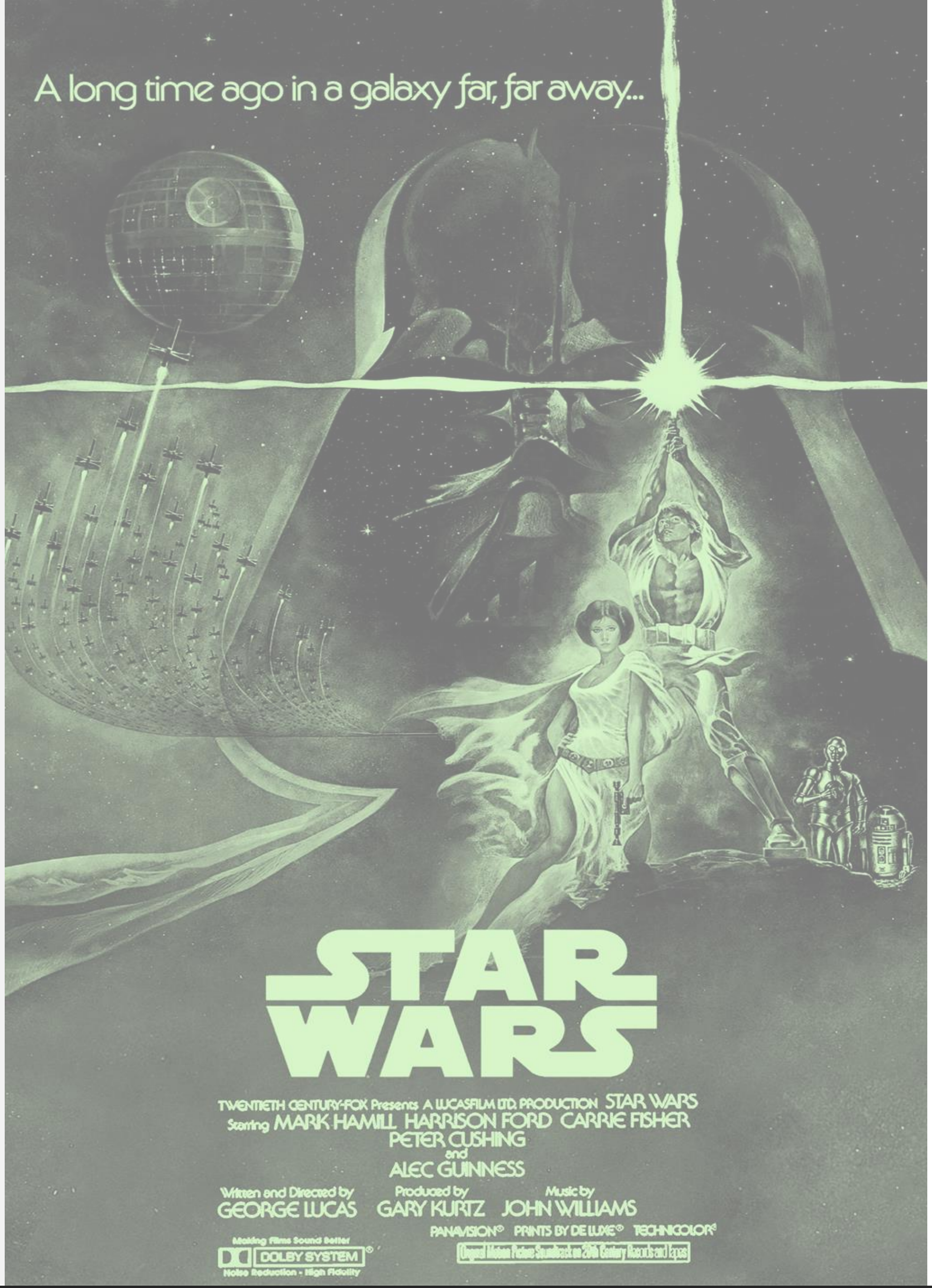
- Developer/Publisher/Genre

Then we began exploring potential movie franchises (Fast and Furious, James Bond, Star Wars, etc.)

- Franchise performance

Finally, we landed on Marvel/DC – Good/Bad

- Characteristics define morality?



Where We Started

Initially started with looking at Videogames

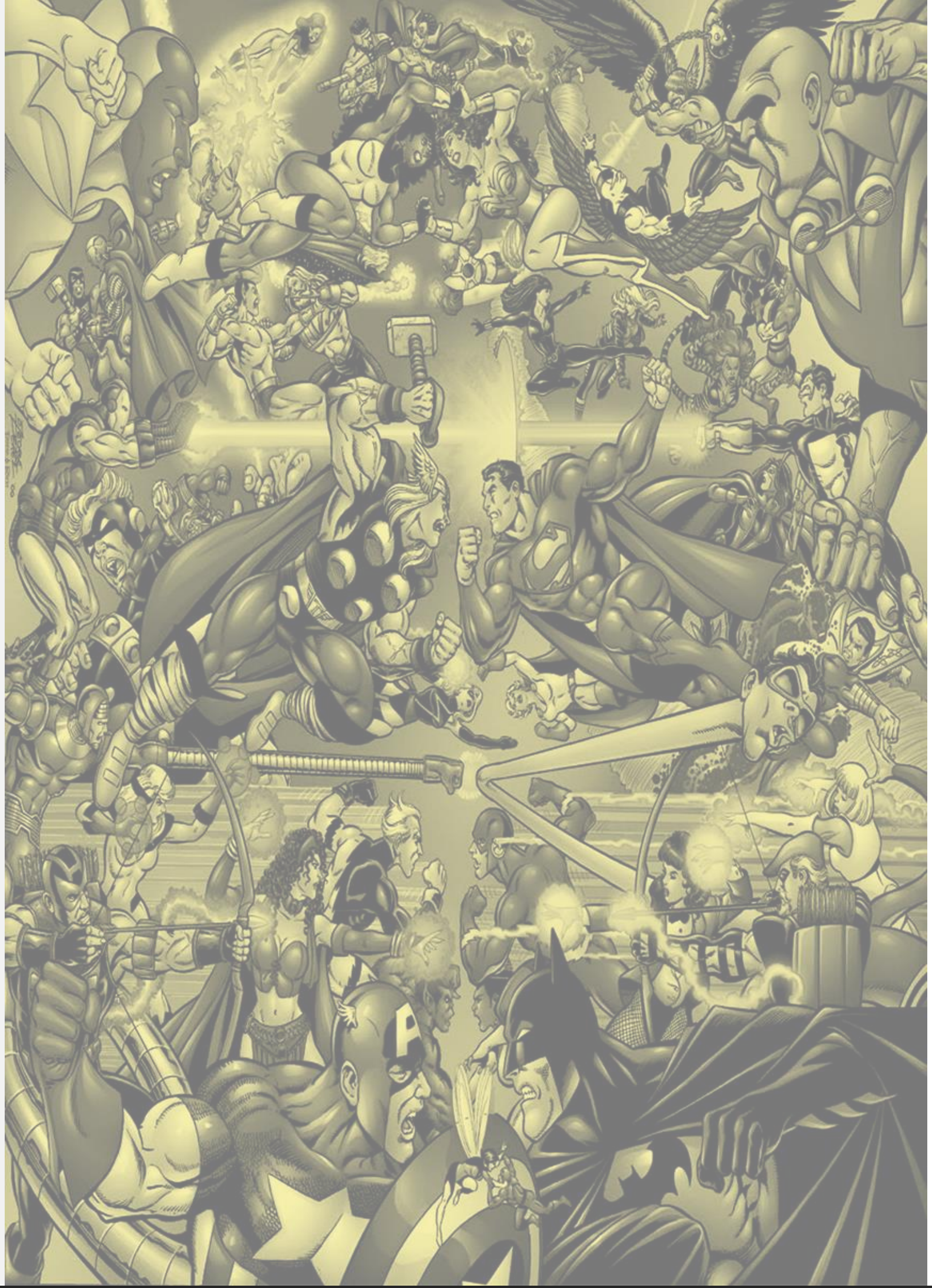
- Developer/Publisher/Genre

Then we began exploring potential movie franchises (Fast and Furious, James Bond, Star Wars, etc.)

- Franchise performance

Finally, we landed on Marvel/DC – Good/Bad

- Characteristics define morality?



Where We Started

Initially started with looking at Videogames

- Developer/Publisher/Genre

Then we began exploring potential movie franchises (Fast and Furious, James Bond, Star Wars, etc.)

- Franchise performance

Finally, we landed on Marvel/DC – Good/Bad

- Characteristics define morality?

Project Focus

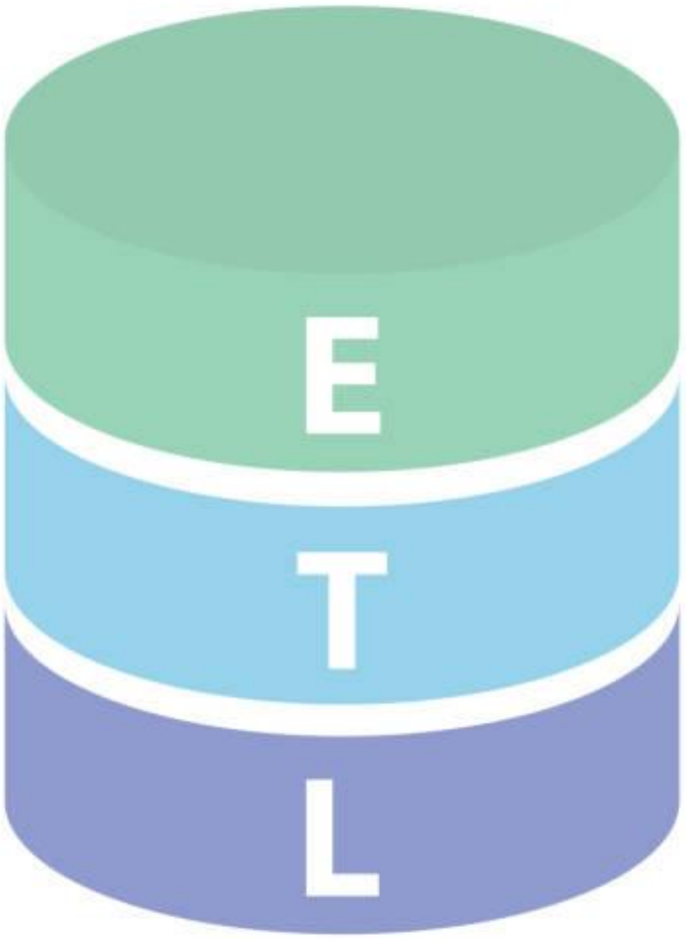
Can character traits be an indicator of whether a character is morally “Good” or “Bad”?

- Identity
- Hair Color
- Sex (biological)
- Gender/Sexual Preference
- Eye Color
- Studio

Data Sources

We utilized 2 datasets found on Kaggle

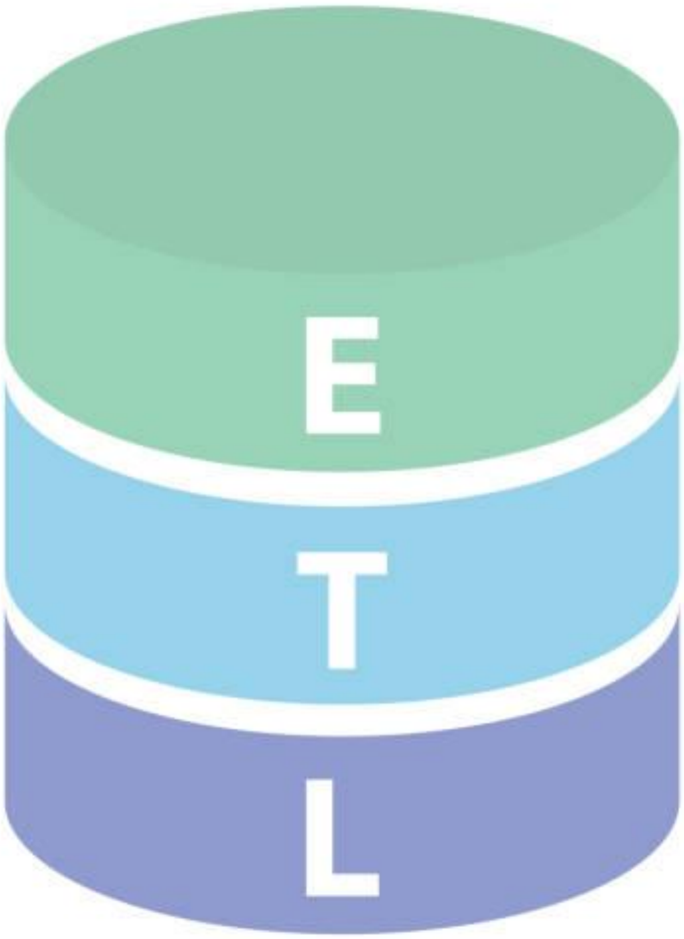
- Marvel-wikia-data.csv
- DC-wikia-data.csv



Extract, Load, Transform

Jupyter Notebook

1. Loaded Marvel and DC Kaggle CSVs
2. Cleaned datasets
3. Added Studio column to differentiate between Marvel vs DC datasets
4. Concatenated both datasets into one
5. Created SQLAlchemy engine to link pandas dataframe with PostGreSQL



Extract, Load, Transform

Tableau

1. Loaded Cleaned CSV
2. Edited some of the dimensions
   1. Groups, Sets, Re-naming, etc.
3. Created Visualizations
4. Developed three Dashboards
5. Constructed a Story

Data Model Implementation

- Used Google Collab script to:
  - read CSVs
  - convert categorical data to numeric
  - develop ML models
    - Target was 75%
    - Started at 50/50
    - Ended up with ~68%

Data Model Optimization

Predictive Model underperformed on our first attempt.

Predicting Good, Bad and Neutral Characters			
ML Models	**Score**	**Scaled**	**Score2**
Logistic Regression Training	56.22%	Logistic Regression Training Scaled	56.36%
Logistic Regression Test	54.99%	Logistic Regression Test Scaled	54.97%
Random Forest Training	59.60%	Random Forest Training Scaled	59.60%
Random Forest Test	54.90%	Random Forest Test Scaled	54.90%
Decision Tree Training	59.60%	Decision Tree Training Scaled	59.60%
Decision Tree Test	54.62%	Decision Tree Test Scaled	54.62%
Extra Trees Training	59.60%	Extra Trees Training Scaled	59.60%
Extra Trees Test	54.75%	Extra Trees Test Scaled	54.75%
Ada Boost Training	47.86%	Ada Boost Training Scaled	47.86%
Ada Boost Test	46.87%	Ada Boost Test Scaled	46.87%

Data Model Optimization

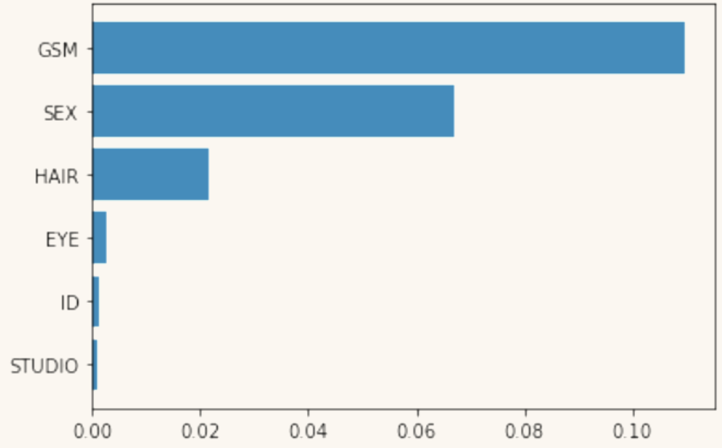
On our second attempt, the model provided much stronger Predictive Validity.

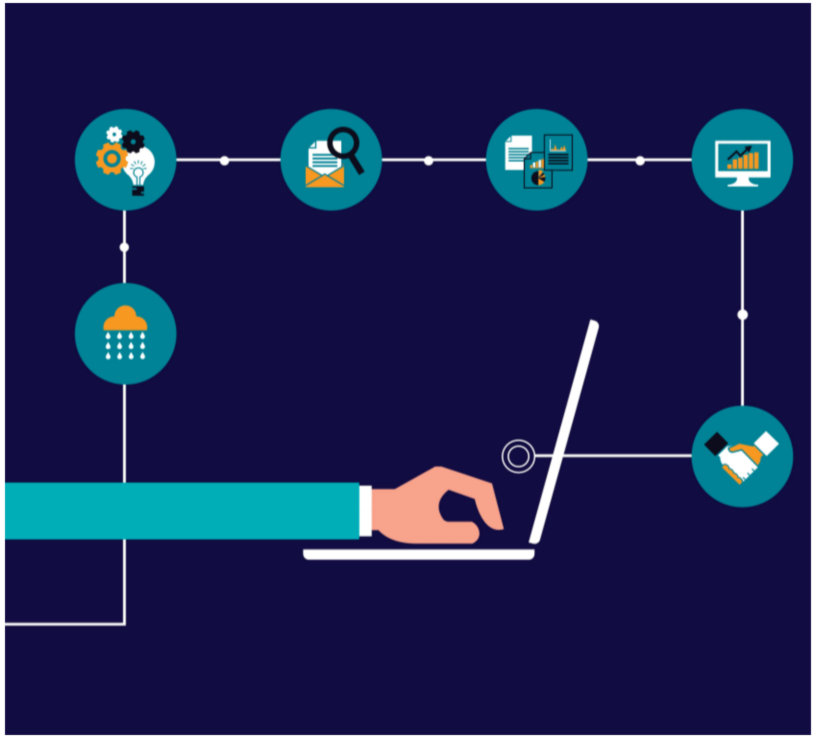
Predicting Only Good and Bad Characters			
ML Models	**Score**	**Scaled**	**Score2**
Logistic Regression Training	65.18%	Logistic Regression Training Scaled	65.19%
Logistic Regression Test	64.60%	Logistic Regression Test Scaled	64.57%
Random Forest Training	68.22%	Random Forest Training Scaled	68.22%
Random Forest Test	63.96%	Random Forest Test Scaled	63.96%
Decision Tree Training	68.22%	Decision Tree Training Scaled	68.22%
Decision Tree Test	63.98%	Decision Tree Test Scaled	63.98%
Extra Trees Training	68.22%	Extra Trees Training Scaled	68.22%
Extra Trees Test	63.85%	Extra Trees Test Scaled	63.85%
Ada Boost Training	59.00%	Ada Boost Training Scaled	59.00%
Ada Boost Test	58.52%	Ada Boost Test Scaled	58.52%

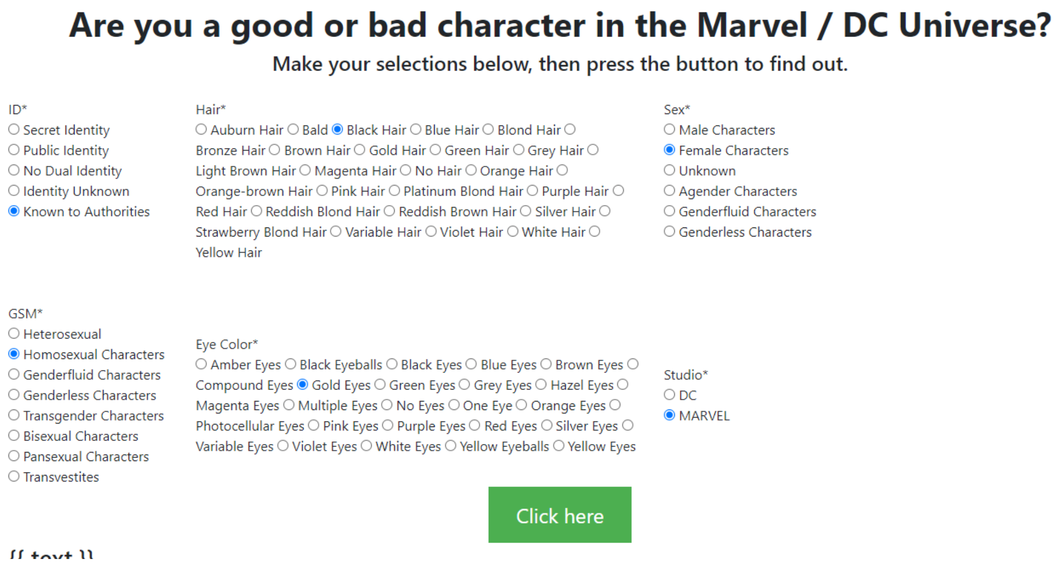
Results

- We can conclude that certain characteristics are somewhat (68%) predictive of a comic-book character's moral compass
- Specific traits lean one way or another
- GSM was the most important feature within the Random Forest model

```
[ ] features = sorted(zip(X.columns, rfclf.feature_importances_), key = lambda x: x[1])
    cols = [f[0] for f in features]
    width = [f[1] for f in features]

    fig, ax = plt.subplots()

    ax.barh(y=cols, width=width)

    plt.show()
```







Web Form Using Flask

- Create API from the Postgres database
- Pass form inputs through app using for loop
- Pickle ML model

Limitations

TIME

ACCESS TO SPECIFIC DATASETS

PREDICTIVE MODELLING BASED ON STRING DATA PROVIDES ADDITIONAL CHALLENGES

NEW SOFTWARE/PACKAGES WE WERE PREVIOUSLY UNAWARE OF (.PKL)

Questions?