Anime Recommendation System

Matt Fournier, Christie Hui

MyAnimeList

Description

The purpose of this project is to create an anime recommendation system for users based on existing user data on myanimelist.net. Existing user data consists of anime titles, genre, types, ratings, and amount of users who have watched a particular anime.

Questions sought to answer

Some interesting questions we intend to answer are:

- Which genres of anime contribute to ambiguous users' decisions?
- How much do aspects such as rating, episode count, and type influence a user's choices?
- Is popularity of an anime or similarity to other users more significant in impacting users' preferences?

Datasets

- Anime Recommendations Database: https://www.kaggle.com/CooperUnion/anime-recommendations-database
- Found on Kaggle

Content

Anime.csv

- anime_id - myanimelist.net's unique id identifying an anime.
- name - full name of anime.
- genre - comma separated list of genres for this anime.
- type - movie, TV, OVA, etc.
- episodes - how many episodes in this show. (1 if movie).
- rating - average rating out of 10 for this anime.
- members - number of community members that are in this anime's "group".

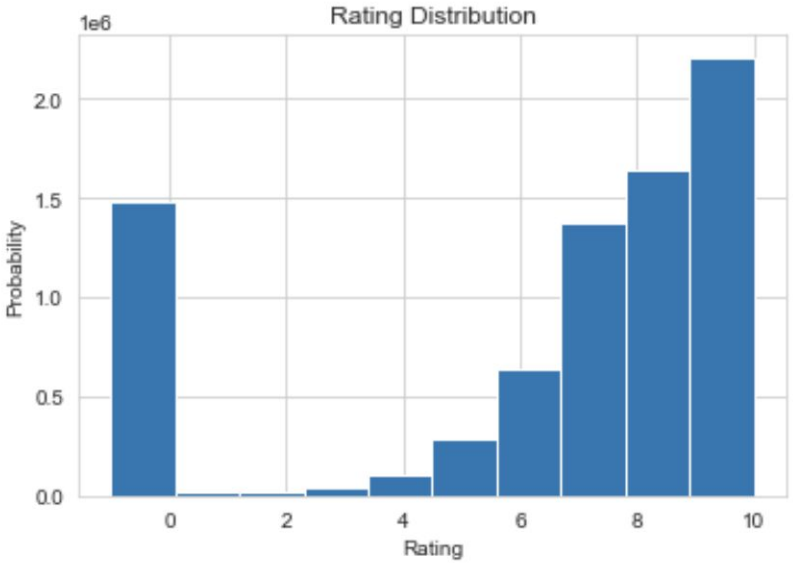
Rating.csv

- user_id - non identifiable randomly generated user id.
- anime_id - the anime that this user has rated.
- rating - rating out of 10 this user has assigned (-1 if the user watched it but didn't assign a rating).

List of tool(s) used

- Jupyter Notebook
- Pandas, numpy, sklearn, scikitlearn (for calculations, EDA, data processing, analysis)
- matplotlib, seaborn (for visualizations)

Data Preprocessing




```
]: rated_anime = pd.merge(users,anime[['anime_id','name']], l
```

```
]: rated_anime.head(10)
```

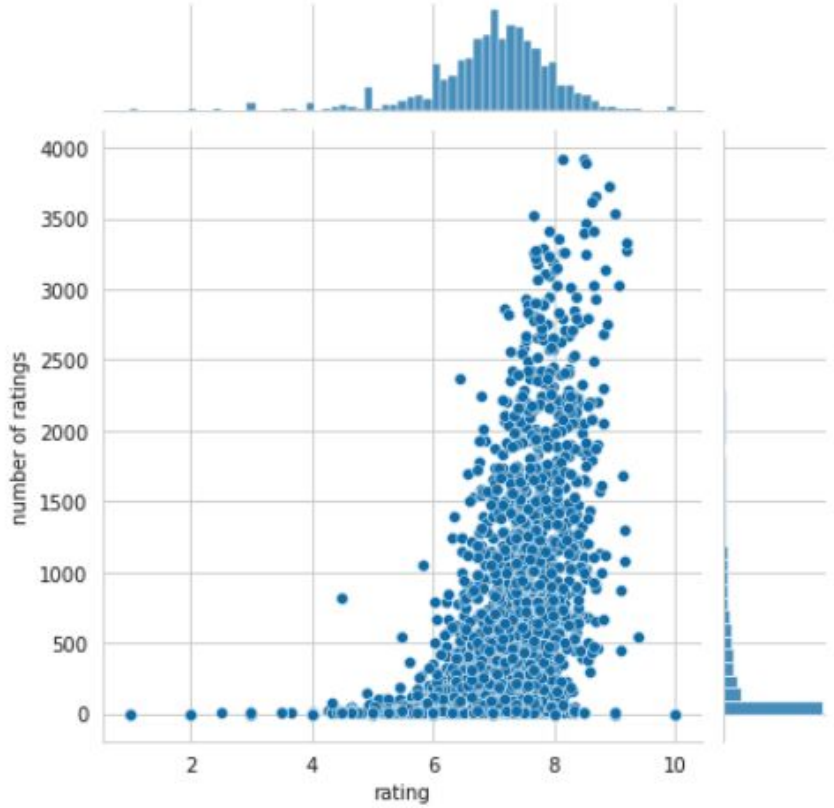
]:

	user_id	rating	name
0	1	10	Highschool of the Dead
1	3	6	Highschool of the Dead
2	5	2	Highschool of the Dead
3	12	6	Highschool of the Dead
4	14	6	Highschool of the Dead
5	17	7	Highschool of the Dead
6	24	7	Highschool of the Dead
7	27	9	Highschool of the Dead
8	29	2	Highschool of the Dead
9	30	8	Highschool of the Dead

Exploratory Data Analysis

```
42635    1953
57620    1485
59643    1442
45659    1315
7345     1182
         ...
39065       1
18558       1
18373       1
33195       1
66215       1
Name: user_id, Length: 68929,
```

```
42635    1953
57620    1485
59643    1442
45659    1315
7345     1182
         ...
61438     200
16375     200
62483     200
20100     200
24074     200
Name: user_id, Length
```



Simple Recommendation System with Correlation

```
# create sparse matrix with user_id as rows and the titles of the anime as columns
# each cell contains the rating given by the user for the anime

df_rec = anime_rec.pivot_table(index='user_id',columns=['name'],values='rating').fillna(0)
df_rec.head()
```

name	07-Ghost	11eyes	Aa! Megami-sama! (TV)	Absolute Duo	Accel World	Acchi Kocchi (TV)	Afro Samurai	Air	Air Gear	Akagami no Shirayuki-hime	...	Zero no Tsukaima: Futatsuki no Kishi	Zero no Tsukaima: Princesses no Rondo	Zetman	Zets Tem
user_id															
5	0.0	0.0	0.0	2.0	3.0	0.0	0.0	0.0	0.0	0.0	...	1.0	1.0	0.0	
7	0.0	0.0	0.0	8.0	8.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	
17	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	
38	0.0	1.0	0.0	0.0	8.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	
43	0.0	9.0	0.0	7.0	8.0	0.0	7.0	0.0	8.0	0.0	...	6.0	0.0	0.0	

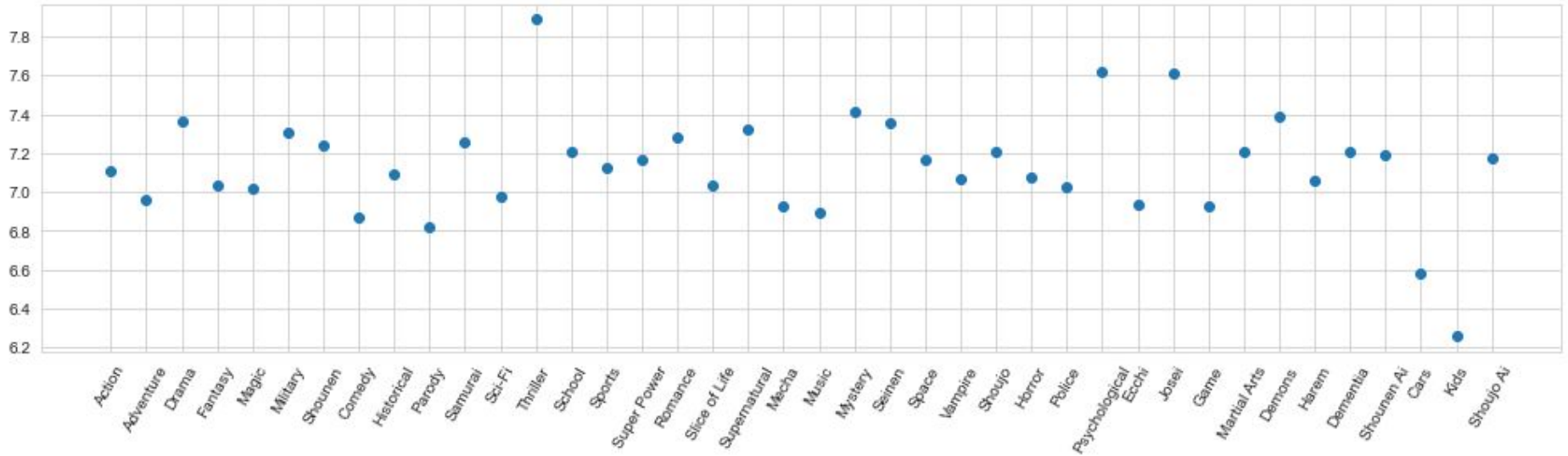
5 rows × 532 columns

```
# function to find correlation of anime with others

def find_correlation(df, name):
    similar = df.corrwith(df[name])
    similar = pd.DataFrame(similar,columns=['Correlation'])
    similar = similar.sort_values(by = 'Correlation', ascending=False)

    return similar
```

```
anime1 = 'Naruto'

# find recommendation for user input of Naruto
find_correlation(df_rec, anime1).head(10)
```

name	Correlation
Naruto	1.000000
Bleach	0.426035
Fairy Tail	0.307742
Katekyo Hitman Reborn!	0.280213
Dragon Ball Z	0.263396
D.Gray-man	0.249084
Dragon Ball	0.244696
Dragon Ball GT	0.235358
Shijou Saikyou no Deshi Kenichi	0.221060
Ao no Exorcist	0.217335

Naive Feature Importance Prediction - Genres



Model Fields

	episodes	members	Action	Adventure	Cars	Comedy	Dementia	Demons	Drama	Ecchi	...	Shoujo Ai	Shounen	Shounen Ai	Slice of Life	Space	Sports	Super Power	Supernatural	Thriller	Vampire
0	12	535892	1	0	0	0	0	0	0	1	...	0	0	0	0	0	0	0	1	0	0
1	12	535892	1	0	0	0	0	0	0	1	...	0	0	0	0	0	0	0	1	0	0
2	12	535892	1	0	0	0	0	0	0	1	...	0	0	0	0	0	0	0	1	0	0
3	12	535892	1	0	0	0	0	0	0	1	...	0	0	0	0	0	0	0	1	0	0
4	12	535892	1	0	0	0	0	0	0	1	...	0	0	0	0	0	0	0	1	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
4364288	13	260	0	0	0	1	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
4364289	151	536	0	0	0	1	0	0	0	0	...	0	0	0	1	0	0	0	0	0	0
4364290	25	199	1	1	0	0	0	0	0	0	...	0	0	0	0	1	0	0	0	0	0
4364291	20	151	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
4364292	39	314	0	0	0	1	0	0	0	0	...	0	0	0	0	1	0	0	0	0	0

4364293 rows × 42 columns

Episodes, Members, Action, Adventure, Cars, Comedy, Dementia, Demons, Drama, Ecchi, Fantasy, Game, Harem, Historical, Horror, Josei, Kids, Magic, Martial Arts, Mecha, Military, Music, Mystery, Parody, Police, Psychological, Romance, Samurai, School, Sci-Fi, Seinen, Shoujo, Shoujo Ai, Shounen, Shounen Ai, Slice of Life, Space, Sports, Super Power, Supernatural, Thriller, Vampire

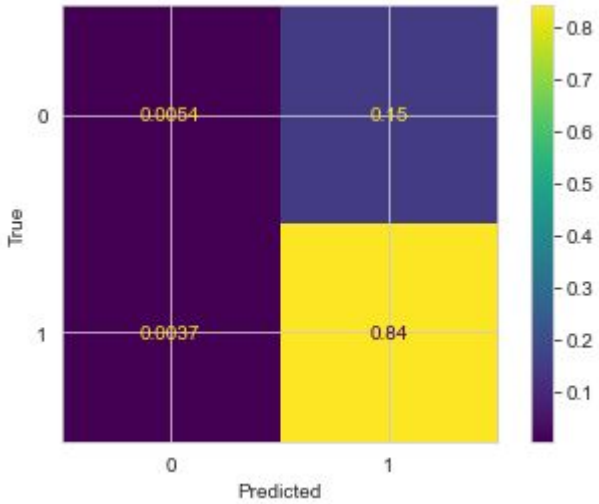
Model Goal Field - Rating Threshold

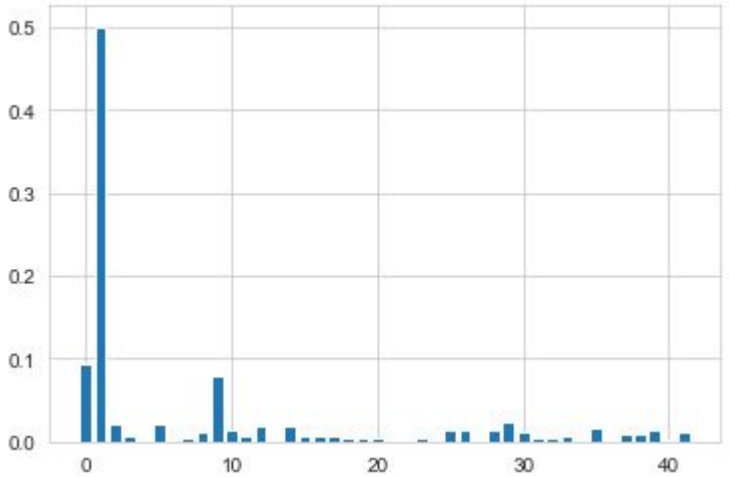
```
0          1
1          0
2          0
3          0
4          0
          ..
4364288    0
4364289    0
4364290    0
4364291    0
4364292    0
Name: recommendation, Length: 4364293, dtype: int64
```

Hot Encoding Function

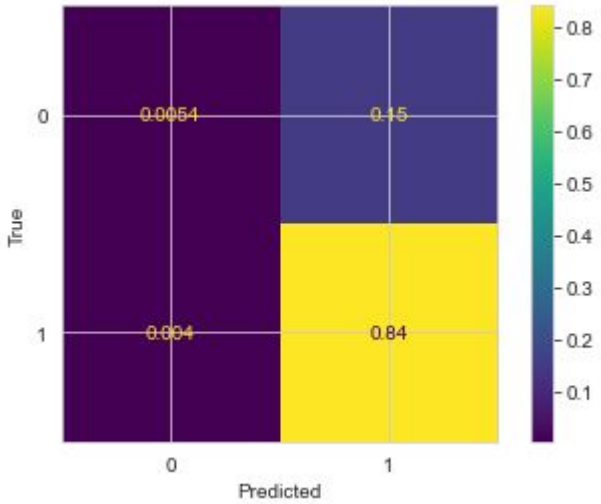
```
def calc(row):
    if row['rating'] >=7:
        return 1
    else:
        return 0
```

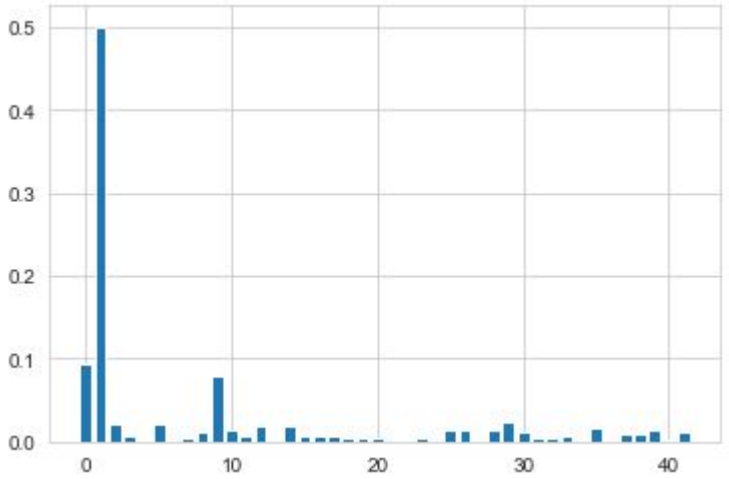
Decision Tree Classification Model



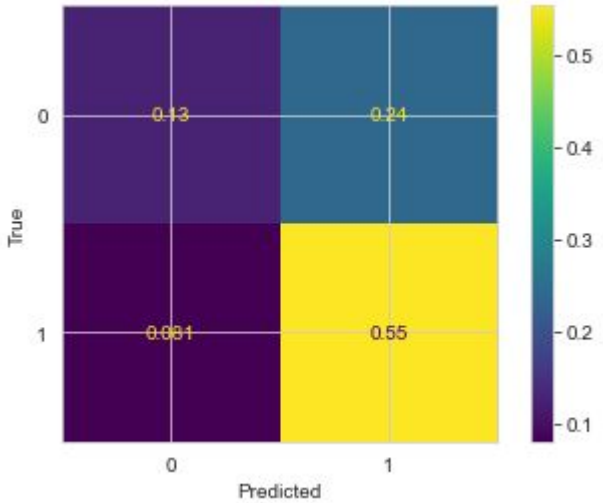


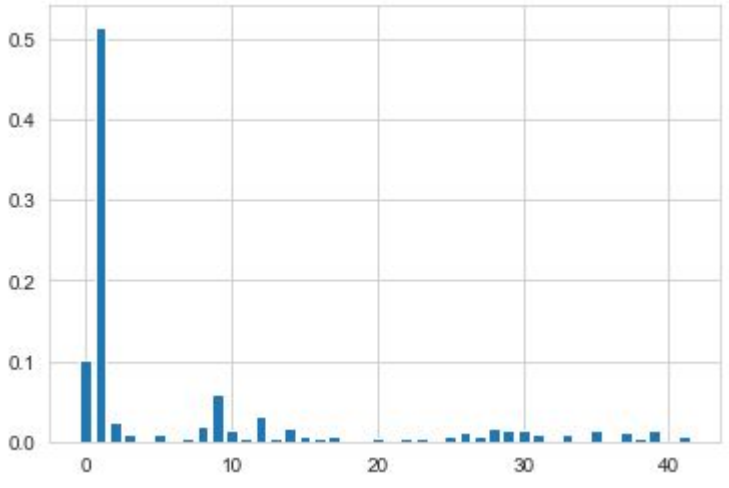
Random Forest Classification Model



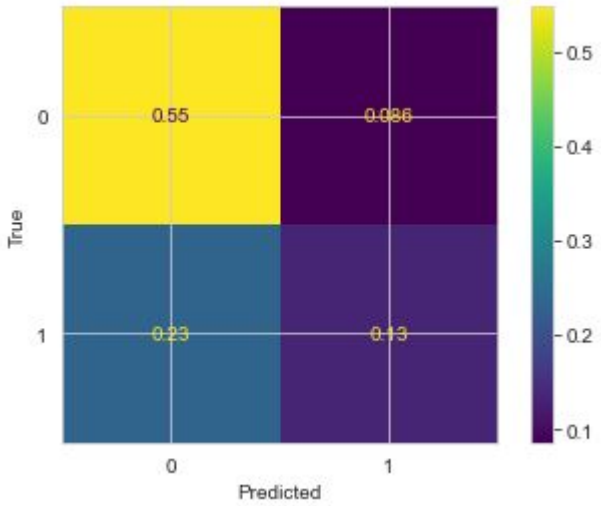


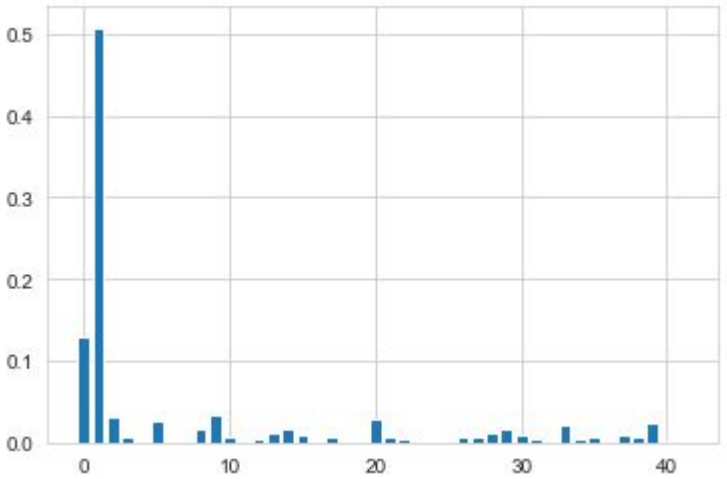
Increase Predicted Rating (Results)- 8



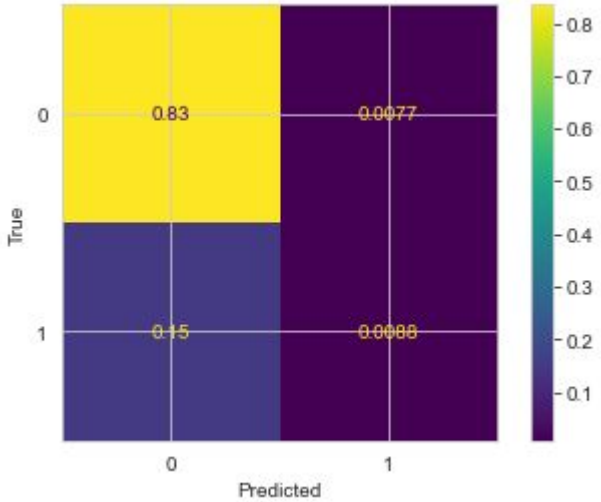


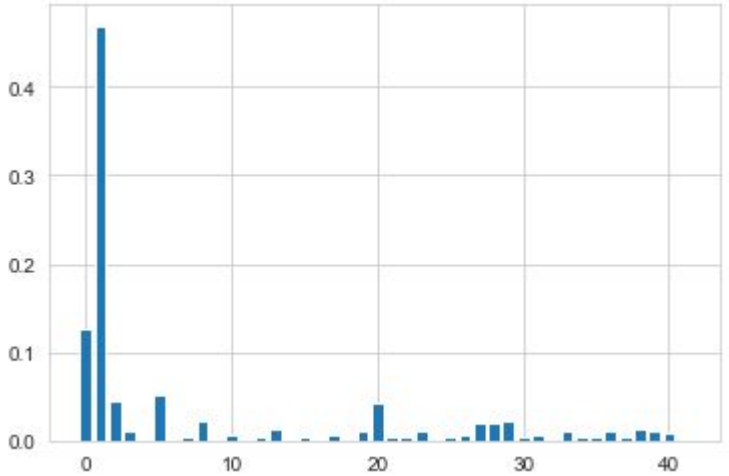
Increase Predicted Rating (Results)- 9





Increase Predicted Rating (Results)- 10





Knowledge Gained and Applications

- Data preprocessing and cleaning/prep for modeling
- Big Data analysis and exploration
- Machine Learning implementations and analysis
- Optimum Satisfaction Rating of 7 for our Prediction Model

Thank You!