Content-Based Anime Recommender System

Process

- Explode genre columns into numerous encoded features
- Remove ratings of -1
- Normalize features
- Create a profile for a user with cosine similarities
- Find most similar animes based on all cosine similarities

For a collaborative filtering method, check here

In [1]:

```
import pandas as pd
import numpy as np
from sklearn.metrics.pairwise import cosine_similarity
import kaggle
```

In [3]:

```
#load in data
#individual anime data

kaggle.api.authenticate()
kaggle.api.dataset_download_files('maulipatel18/anime-content-based-recommendation-
anime_df = pd.read_csv('../data/anime.csv')
anime_df.head()
```

```
Dataset URL: https://www.kaggle.com/datasets/maulipatel18/anime-content-based-recomm
endation-system-datasets
```

In [119…

```
#user data

rating_df = pd.read_csv('../data/rating.csv')
rating_df.head()
```

Out[119…

	user_id	anime_id	rating
0	1	20	-1
1	1	24	-1
2	1	79	-1
3	1	226	-1
4	1	241	-1

In [120…

```
# #expand genres column

anime_df['genre'] = anime_df['genre'].str.split(',')
```

```python
data_exploded = anime_df.explode('genre')

genre_dummies = pd.get_dummies(data_exploded['genre'], prefix='genre')
genre_dummies = genre_dummies.groupby(level=0).sum()
anime_df = pd.concat([anime_df, genre_dummies], axis=1)
anime_df = anime_df.drop('genre', axis=1)

anime_df.head()
```

Out[120…

	anime_id	name	type	episodes	rating	members	genre_ Adventure	genre_ Cars	geni Come
0	32281	Kimi no Na wa.	Movie	1	9.37	200630	0	0	
1	5114	Fullmetal Alchemist: Brotherhood	TV	64	9.26	793665	1	0	
2	28977	Gintama°	TV	51	9.25	114262	0	0	
3	9253	Steins;Gate	TV	24	9.17	673572	0	0	
4	9969	Gintama&#039;	TV	51	9.16	151266	0	0	

5 rows × 88 columns

In [121…

```python
#remove NaN values and duplicates from anime df

anime_df = anime_df.dropna().drop_duplicates()
```

In [122…

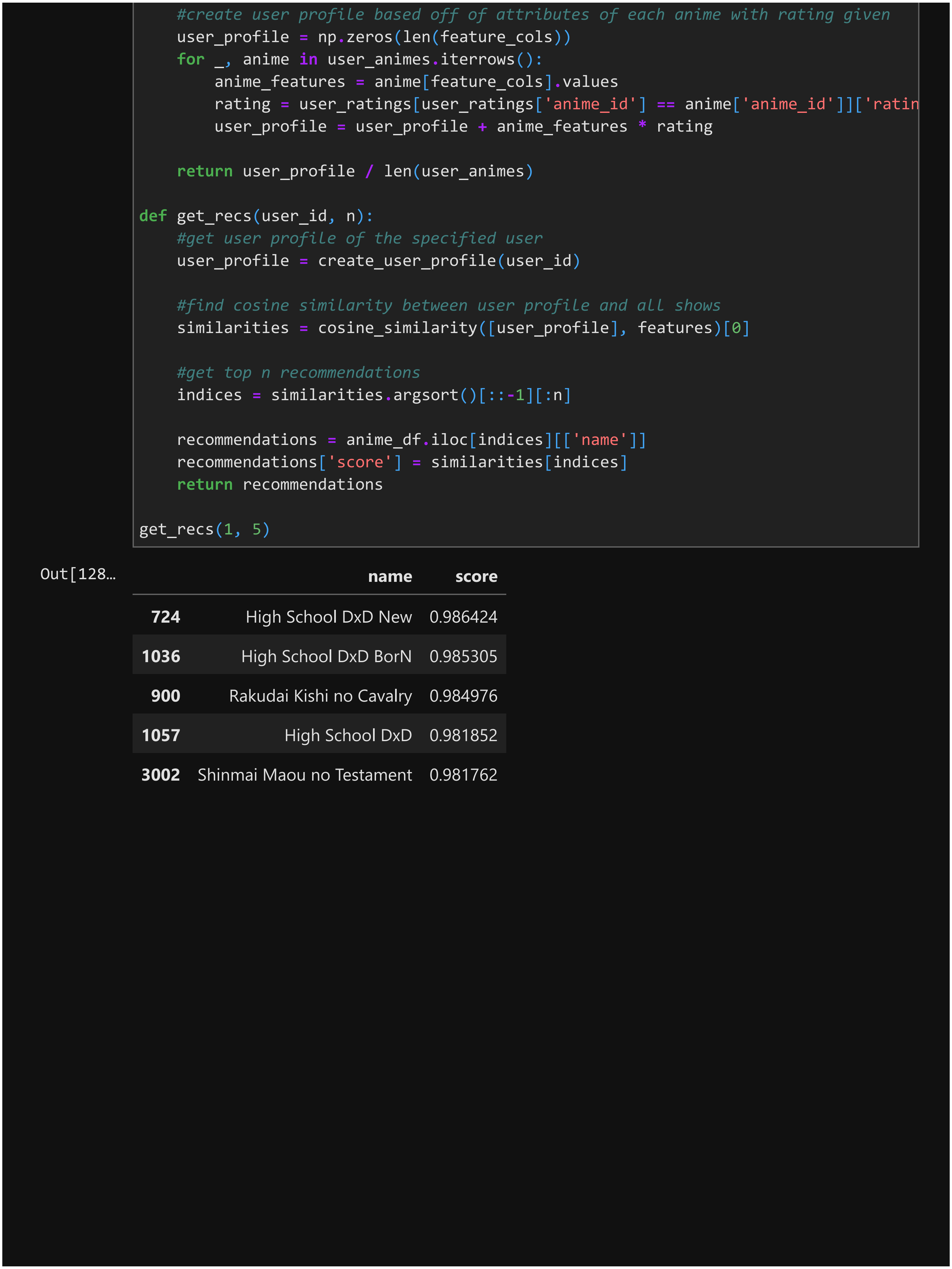
```python
#remove episodes column (largely irrelevant)

anime_df = anime_df.drop('episodes', axis=1)

#remove -1 values from ratings, cannot easily be solved otherwise
rating_df = rating_df[rating_df['rating'] != -1]
```

In [123…

```python
#normalize continuous features

anime_df['norm_members'] = (anime_df['members'] - anime_df['members'].min()) / (ani

#get all feature columns
anime_df = pd.get_dummies(anime_df, columns=['type'], drop_first=False)
feature_cols = [col for col in anime_df.columns if col.startswith('genre_')] + ['no
features = anime_df[feature_cols]

features = features.values
```

In [128…

```python
# create a "profile" for a user

def create_user_profile(user_id):
    #get user ratings and user watched animes
    user_ratings = rating_df[rating_df['user_id'] == user_id]
    user_animes = anime_df[anime_df['anime_id'].isin(user_ratings['anime_id'])]
```



```
    #create user profile based off of attributes of each anime with rating given
    user_profile = np.zeros(len(feature_cols))
    for _, anime in user_animes.iterrows():
        anime_features = anime[feature_cols].values
        rating = user_ratings[user_ratings['anime_id'] == anime['anime_id']]['ratin
        user_profile = user_profile + anime_features * rating

    return user_profile / len(user_animes)

def get_recs(user_id, n):
    #get user profile of the specified user
    user_profile = create_user_profile(user_id)

    #find cosine similarity between user profile and all shows
    similarities = cosine_similarity([user_profile], features)[0]

    #get top n recommendations
    indices = similarities.argsort()[::-1][:n]

    recommendations = anime_df.iloc[indices][['name']]
    recommendations['score'] = similarities[indices]
    return recommendations

get_recs(1, 5)
```

Out[128…

	name	score
724	High School DxD New	0.986424
1036	High School DxD BorN	0.985305
900	Rakudai Kishi no Cavalry	0.984976
1057	High School DxD	0.981852
3002	Shinmai Maou no Testament	0.981762