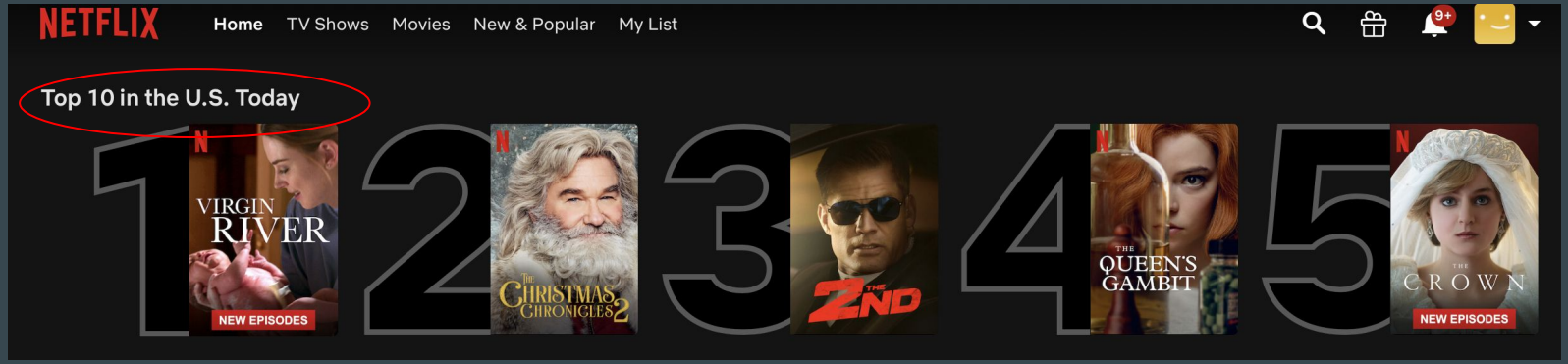
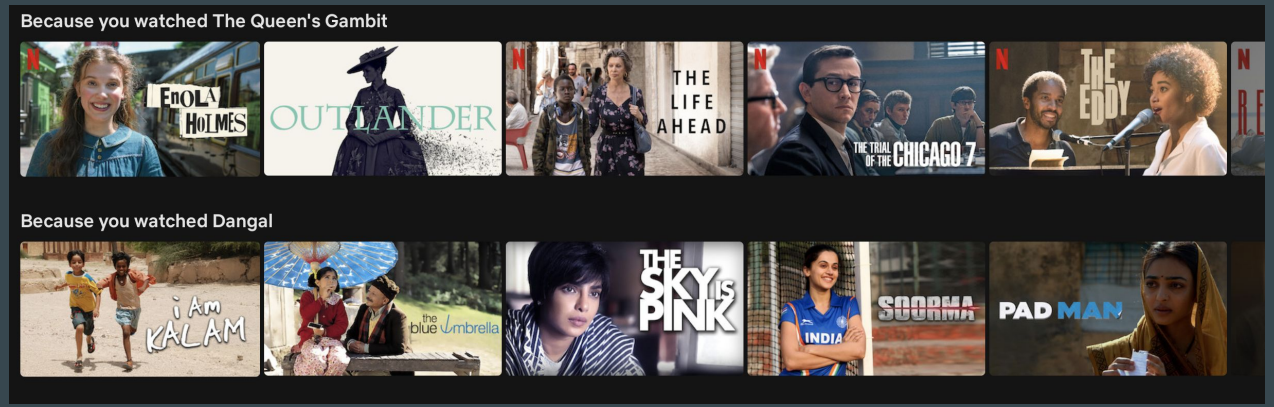
Outline

- Why we chose a recommendation system ?
- Similar real world applications (non-music related)
- Methods to build a recommendation system
  - Popularity based
  - Classification
  - Collaborative filtering
- Our data and model
  - How can we make it better?

Real world applications - Movie/Television

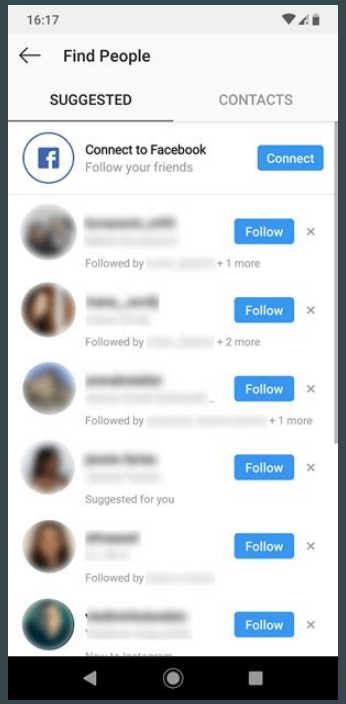




Social Media Recommendation

Instagram

TIKTOK



Problem

- Can we suggest a song to a user such that they do not need to skip ?
  - New Artists/Songs/Genres that are tailored to a specific users taste

Description/Solutions

- Using various datasets, and algorithms (classifications) we will use collaborative filtering as the basis of our model, which will then also to accurately be able to suggest songs to users based on their music taste
- Suggestions are made based on the users likes/dislikes
- Suggestion are made from careful analysis of past data of the user to get the most optimum result.

Popularity based recommendation system

- Based on popularity find these songs under “Trending”, “Top Charts”
- Find these songs under “Trending”, “Top Charts”

Use the popularity model to make some predictions

```
[64]:
1 user_id = users[6]
2 model.recommend(user_id)
```

[64]:

	user_id	song	score	Rank
3194	e006b1a48f466bf59feefed32bec6494495a4436	Sehr kosmisch - Harmonia	37	1.0
4083	e006b1a48f466bf59feefed32bec6494495a4436	Undo - Björk	27	2.0
931	e006b1a48f466bf59feefed32bec6494495a4436	Dog Days Are Over (Radio Edit) - Florence + Th...	24	3.0
4443	e006b1a48f466bf59feefed32bec6494495a4436	You're The One - Dwight Yoakam	24	4.0
3034	e006b1a48f466bf59feefed32bec6494495a4436	Revelry - Kings Of Leon	21	5.0
3189	e006b1a48f466bf59feefed32bec6494495a4436	Secrets - OneRepublic	21	6.0
4112	e006b1a48f466bf59feefed32bec6494495a4436	Use Somebody - Kings Of Leon	21	7.0
1207	e006b1a48f466bf59feefed32bec6494495a4436	Fireflies - Charttraxx Karaoke	20	8.0
1577	e006b1a48f466bf59feefed32bec6494495a4436	Hey_ Soul Sister - Train	19	9.0
1626	e006b1a48f466bf59feefed32bec6494495a4436	Horn Concerto No. 4 in E flat K495: II. Romanc...	19	10.0

[17]:

	song	listen_count	percentage
3660	Sehr kosmisch - Harmonia	45	0.45
4678	Undo - Björk	32	0.32
5105	You're The One - Dwight Yoakam	32	0.32
1071	Dog Days Are Over (Radio Edit) - Florence + Th...	28	0.28
3655	Secrets - OneRepublic	28	0.28
...	...	...	...
5139	high fives - Four Tet	1	0.01
5140	in white rooms - Booka Shade	1	0.01
5143	paranoid android - Christopher O'Riley	1	0.01
5149	¿Lo Ves? [Piano Y Voz] - Alejandro Sanz	1	0.01
5150	Época - Gotan Project	1	0.01

5151 rows × 3 columns

User Based Collaborative Filtering (Implicit)

Find users who have similar taste in music (not based on popularity)
Similarity is based upon co-occurence of of songs listened to
Steps Taken:

- Created dummy data
- Used a co-occurrence matrix to determine which songs to recommend

We used an Implicit Collaborative Filtering Approach

Example Output:

```
[47]:
1 song = 'U Smile - Justin Bieber'
2 model.get_similar_items([song])
```

```
no. of unique songs in the training set: 4483
Non zero values in cooccurence_matrix :271
```

[47]:

	user_id	song	score	rank
0		Somebody To Love - Justin Bieber	0.428571	1
1		Bad Company - Five Finger Death Punch	0.375000	2
2		Love Me - Justin Bieber	0.333333	3
3		One Time - Justin Bieber	0.333333	4
4		Here Without You - 3 Doors Down	0.333333	5
5		Stuck In The Moment - Justin Bieber	0.333333	6
6		Teach Me How To Dougie - California Swag District	0.333333	7
7		Paper Planes - M.I.A.	0.333333	8
8		Already Gone - Kelly Clarkson	0.333333	9
9		The Funeral (Album Version) - Band Of Horses	0.300000	10

```
[50]:
1 song = 'Somebody To Love - Justin Bieber'
2 model.get_similar_items([song])
```

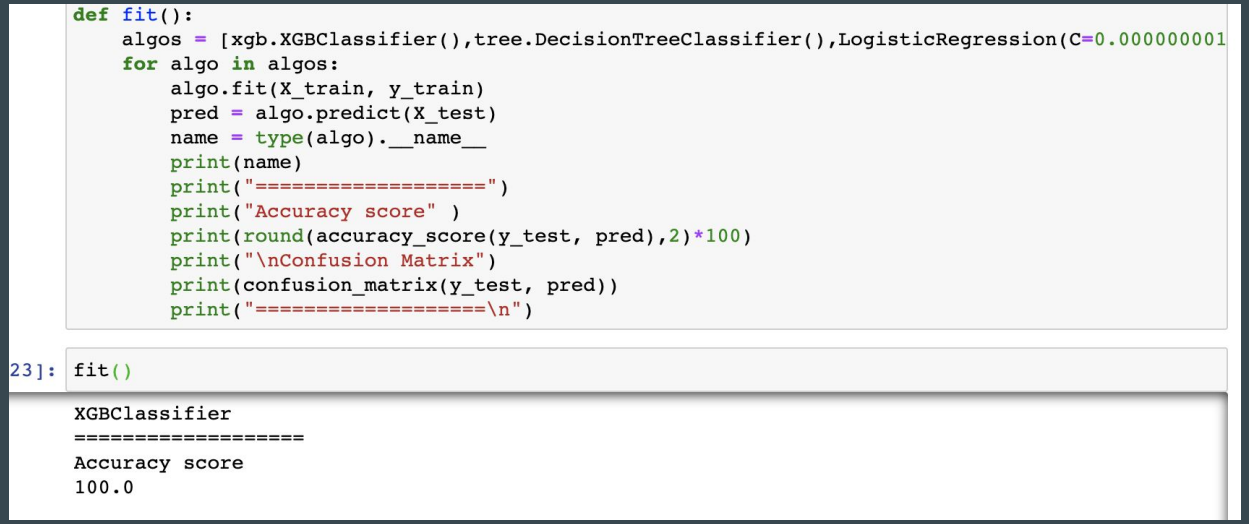
```
no. of unique songs in the training set: 4483
Non zero values in cooccurence_matrix :453
```

[50]:

	user_id	song	score	rank
0		U Smile - Justin Bieber	0.428571	1
1		Lucky (Album Version) - Jason Mraz & Colbie Ca...	0.304348	2
2		Heartbreak Warfare - John Mayer	0.294118	3
3		Marry Me - Train	0.291667	4
4		Party In The U.S.A. - Miley Cyrus	0.285714	5
5		Pursuit Of Happiness (nightmare) - Kid Cudi / ...	0.277778	6
6		Eenie Meenie - Sean Kingston and Justin Bieber	0.266667	7
7		Already Gone - Kelly Clarkson	0.250000	8
8		Bleed It Out [Live At Milton Keynes] - Linkin ...	0.250000	9
9		Monster - Lady GaGa	0.235294	10

System based on Classification

- Problems with this
  - More Data
  - User like and dislikes



```
def fit():
    algos = [xgb.XGBClassifier(),tree.DecisionTreeClassifier(),LogisticRegression(C=0.000000001
    for algo in algos:
        algo.fit(X_train, y_train)
        pred = algo.predict(X_test)
        name = type(algo).__name__
        print(name)
        print("====================")
        print("Accuracy score" )
        print(round(accuracy_score(y_test, pred),2)*100)
        print("\nConfusion Matrix")
        print(confusion_matrix(y_test, pred))
        print("====================\n")
```

```
23]: fit()
```

```
XGBClassifier
====================
Accuracy score
100.0
```

What's Next?

- Test out different ways of filtering music
  - Implement NLP
  - Test against top songs
  - Bigger data set (maybe real data lol)
  - Explicit Collaborative Filtering