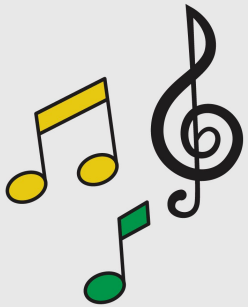
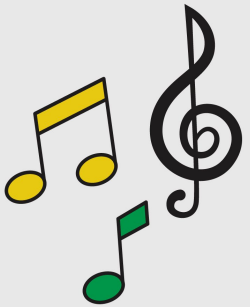
Music Recommender System

Presented by: Eddie Amaitum





Problem Definition

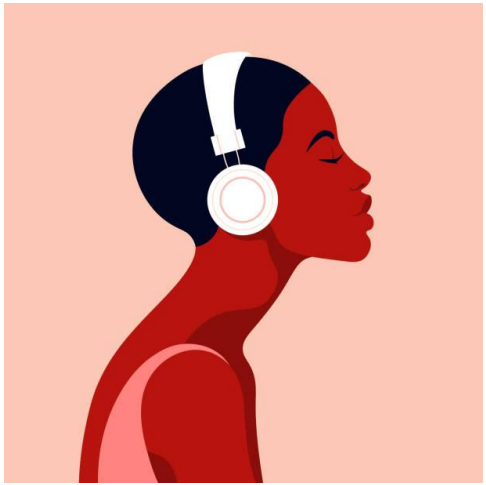
Today we live in a world of rapid technological advancements

Result: Distractions hence limited time to consume good content

Platforms rely on recommender systems to retain user attention

The challenge of predicting top_n songs is easy to understand for a non technical audience

Objective



Build a recommendation system to predict the top_n songs for a user based on the likelihood of listening to those songs

Showcase my ability to develop ML tools and lay foundation for deploying and end - to - end ML process

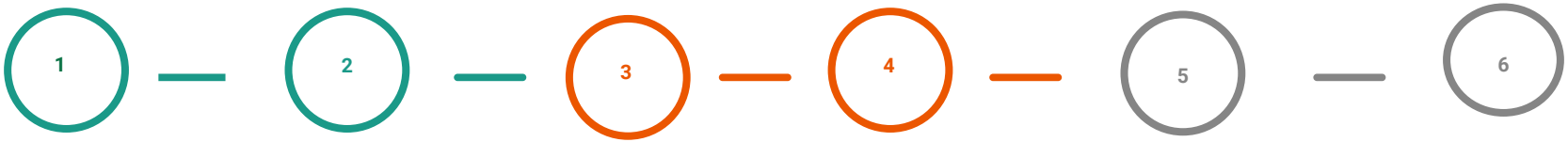
Data:

- Taste profile subset by Echo Nest: millionsongdataset.com
- Song data (song_id, title, release,artist_name, year) ⇨ 1,000,000 records
- Count data (user_id, song_id, play_count) ⇨ 2,000,000 records
- It is freely available to the public

Solution Approach

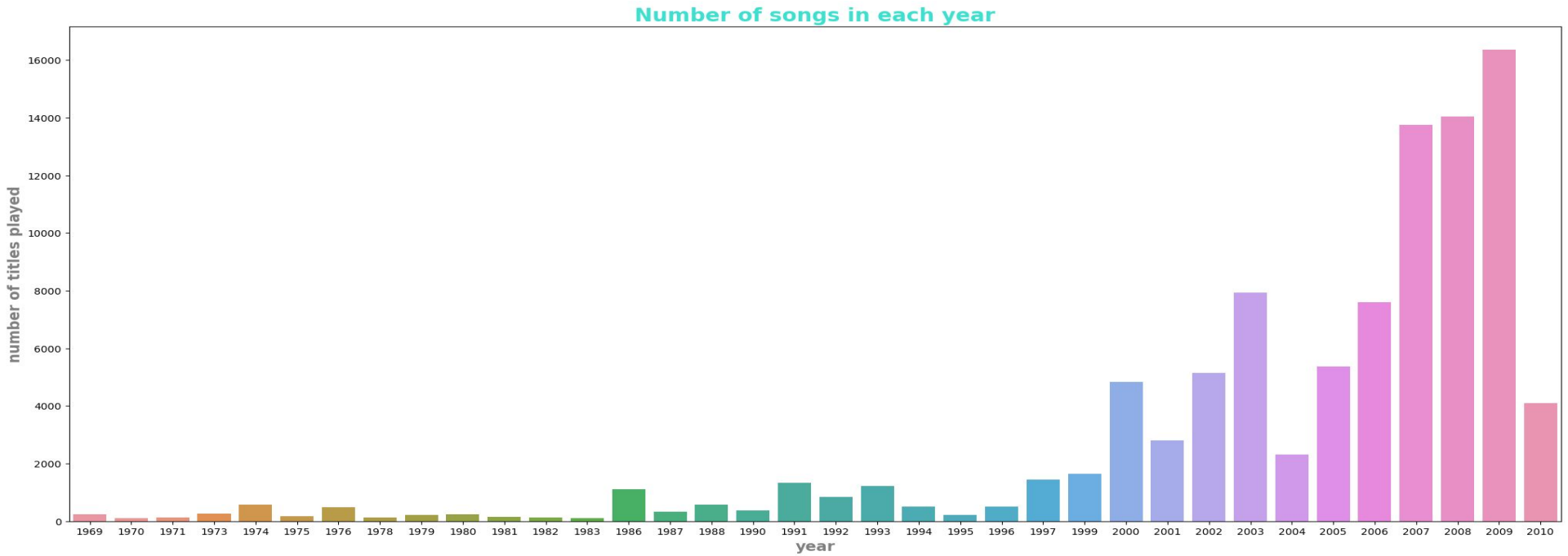
Data cleaning + Feature engineering

Exploratory data analysis



1. Rank / Popularity based recommendation system
2. User - User similarity based collaborative filtering
3. Item - Item similarity based collaborative filtering
4. Matrix factorization
5. Clustering - based recommendation system
6. Content based recommendation system

Distribution of songs



Key observations & insights:

- The final data was sparse
- Data filtered to retain < = 5 song play_count
- "Corrected_ ratings" utilized
- Hyperparameters tuned to improve models

Solution Approach

Performance Metrics:

- Precision @ k : The fraction of recommended items that are relevant in top k predictions
- Recall @ k: The fraction of relevant items that are recommended to the user in top k predictions
- RMSE: Checks how far the overall predicted ratings are from the actual ratings
- F1_Score @ k: The harmonic mean of Precision @ k and Recall @ k

Solution Approach

User - User Similarity

RMSE: 1.0521
Precision: 0.413
Recall: 0.721
F_1 score: 0.525

Item - Item Similarity

RMSE: 1.0328
Precision: 0.408
Recall: 0.665
F_1 score: 0.506

Model based/Matrix factorization

RMSE: 1.0141
Precision: 0.415
Recall: 0.635
F_1 score: 0.502

Cluster based

RMSE: 1.0654
Precision: 0.394
Recall: 0.566
F_1 score: 0.465

Sample recommendation

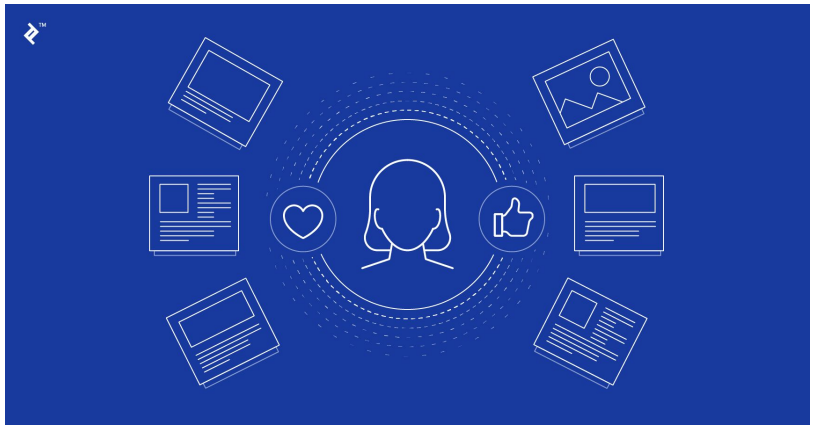
```
# Making the recommendation for the song with title 'Learn To Fly'
recommendations('Learn To Fly', similar_songs)
```

✓ 0.0s

```
[509, 234, 423, 345, 394, 370, 371, 372, 373, 375]

['Everlong',
 'The Pretender',
 'Nothing Better (Album)',
 'From Left To Right',
 'Lifespan Of A Fly',
 'Under The Gun',
 'I Need A Dollar',
 'Feel The Love',
 'All The Pretty Faces',
 'Bones']
```

Proposal for future solution design and outlook



- A robust hybrid recommendation system will be used
- I will build an interactive tool showcasing the end to end machine learning process
- Hyperparameter tuning will be done to improve model performance
- Continuous training on new data to improve model


THANK YOU!