Music Recommender System

Aditya Maheshwari, Animesh Sharma, Fan Shen, Raju Datla, Toshitt Ahuja, Vipul Gharde, Wanying Mo, Yaniv Bronshtein

The Problem

The Problem: Can We Predict What a New User Would Listen To?

- **Genre?**

  Rock, Hip Hop, Classical, Pop, ...
- **Artist?**

  Led Zeppelin, Eminem, Mozart, Avril Lavigne, ...
- **Track (Song)?**

  Stairway to Heaven, ‘Till I Collapse, Symphony No. 40, Complicated, ...
- **Playlist?**

The Data

Dataset 1

Source:
https://www.kaggle.com/iqbalbasyar/spotify-genre-classification/data?select=SpotifyFeatures.csv

Size: (232725,18)
Key Base Features: Genre, artist_name, track_name, popularity, and various acoustic features
Problem? Too generalized

Dataset 2

Source:
https://www.kaggle.com/andrewmvd/spotify-playlists

Size: (~12.9M,4)
Key Base Features: user_id, artistname, trackname, playlistname
Problem? Large dataset, but few features to train on

Planning

- Idea 1

Join Dataset 1 and 2 by creating Artist-Track feature and then creating fuzzy match using Levenshtein distances

  - Ran for hours and failed to finish basic join. Exact match produced around 30K observations

- Idea 2

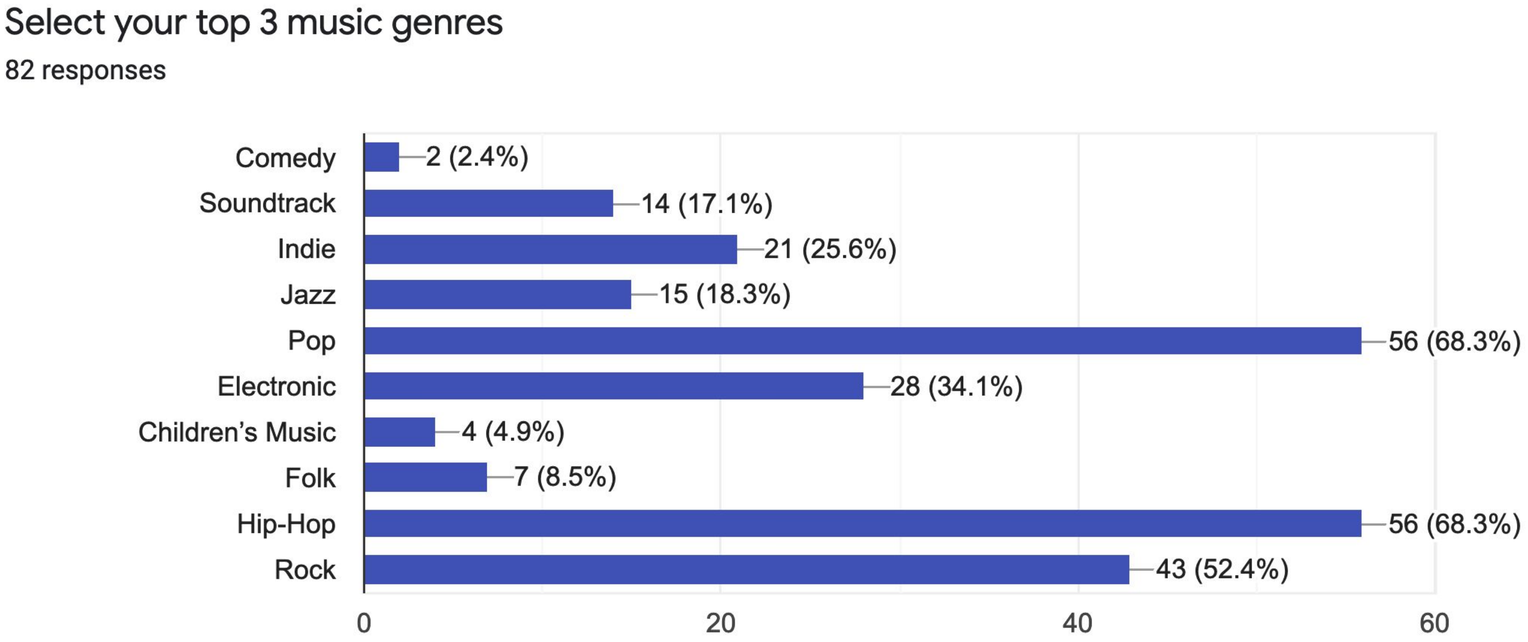
Utilize Dataset 1 and 2 in two **separate** DL/ML Tasks

Validation Using Data from Google Form Survey

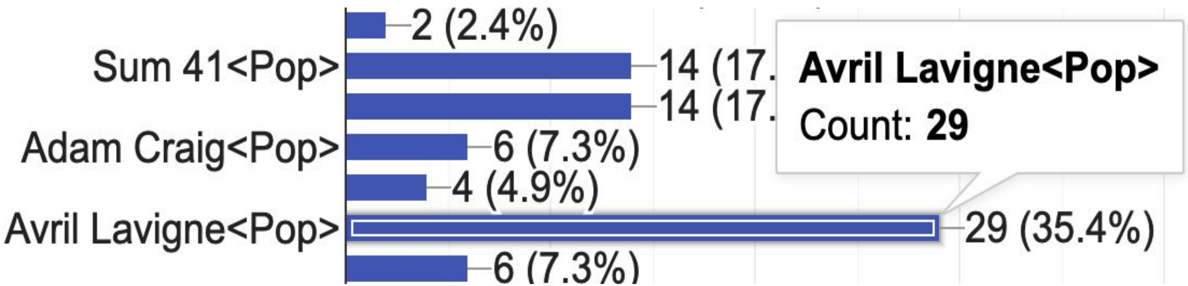
Question 1: Pick Top 3 music genres (out of 10)

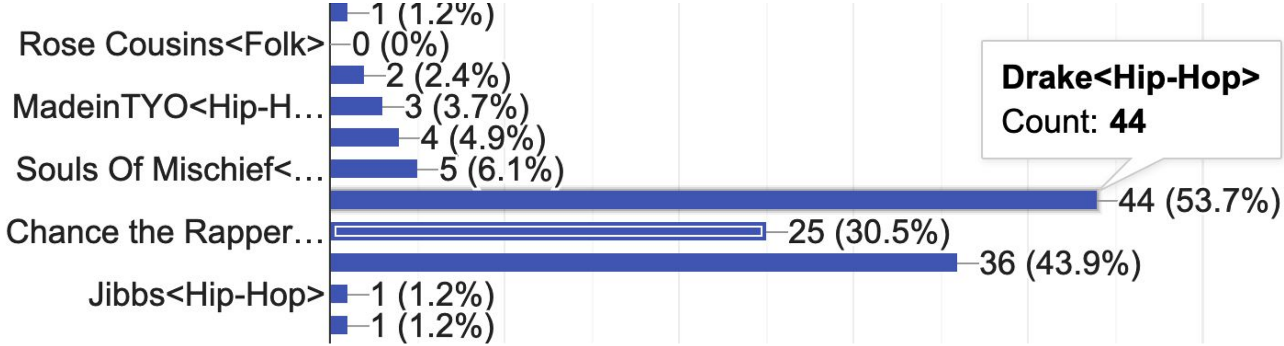
Question 2: Pick Top 5 artists based on genres selected (out of 100 **randomly selected)**

Question 1: Genre Distribution

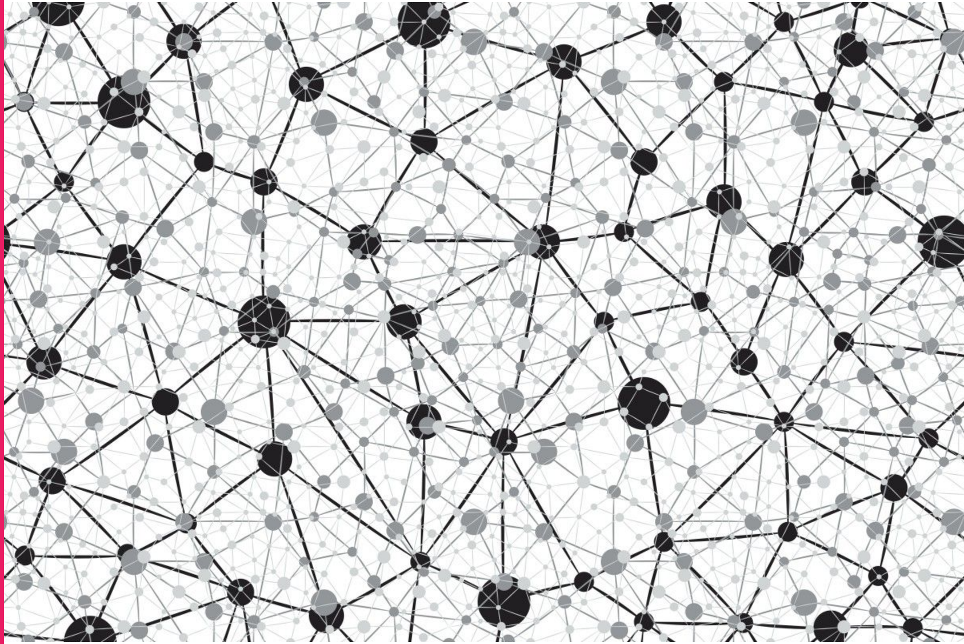


Question 2: Artist Preference Distribution





Idea 1: Feedward Neural Net



Libraries Used

- numpy
- pandas
- tensorflow
- keras
- scikit-learn

Preprocessing of Dataset

- Read Dataset 1 file using pandas
- Clean the dataset
  - Skip bad rows, NA rows
- Use a Hash function to generate user_id for Dataset 1 to mimic the user_id in Dataset 2
- Create artist-genre, artist-track feature and serialized both
- Create an input matrix

Preprocessing Flow

	07e61e82a152	3e7e8e1a0d94	1da541ba91a8
HenriSalvadorMovie	NaN	NaN	NaN
Martin&lesféesMovie	NaN	NaN	NaN
JosephWilliamsMovie	NaN	NaN	NaN

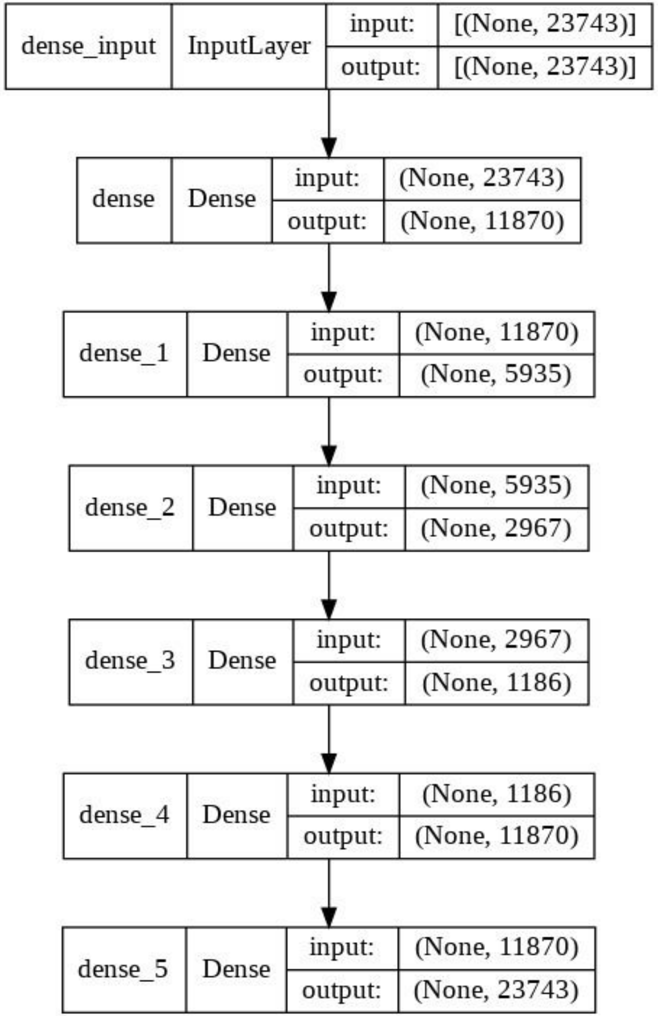
	7f03dcad408b	1ced8f5be2db	33ddec12969e
HenriSalvadorMovie	0.0	0.0	0.0
Martin&lesféesMovie	0.0	0.0	0.0
JosephWilliamsMovie	0.0	0.0	0.0

	7f03dcad408b	1ced8f5be2db	33ddec12969e
HenriSalvadorMovie	2.0	0.0	7.0
Martin&lesféesMovie	2.0	2.0	1.0
JosephWilliamsMovie	0.0	1.0	2.0

The Input Matrix

	d2aef613e249	98beaab316b0	77e0a2775da3
HenriSalvadorMovie	0.003504	0.0	0.003584
Martin&lesféesMovie	0.000000	0.0	0.000000
JosephWilliamsMovie	0.000292	0.0	0.000000

Our NN Architecture



Training our model

- **Train-Test Split:** 75% train, 25% test
- **Optimizer:** Adam
- **Epochs:** 200
- **Activation:** Leaky ReLU for hidden layers, softmax for output layer
- **Loss Function:** Categorical Cross Entropy
- **Metric:** Accuracy

Neural Network Results: An Abysmal Idea

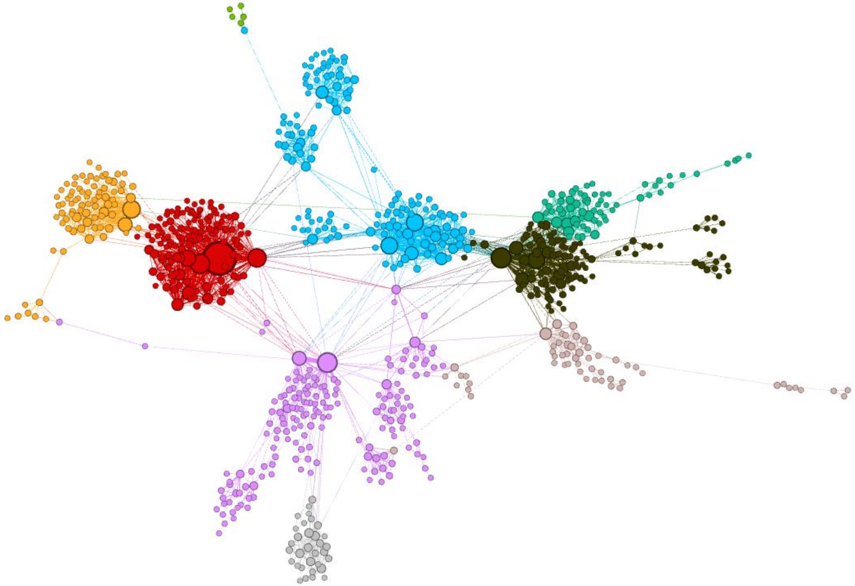
Tried the following to fix the issue:

- Dropout of 0.2 at various hidden layers
- Laplace Smoothing of input matrix

It only made it worse and below was our star performer!

```
Epoch 198/200
5/5 [==============================] - 35s 7s/step - loss: 5.1691 - accuracy: 0.6076 - val_loss: 9.8555 - val_accuracy: 0.1698
Epoch 199/200
5/5 [==============================] - 36s 7s/step - loss: 5.1671 - accuracy: 0.6266 - val_loss: 9.8574 - val_accuracy: 0.1509
Epoch 200/200
5/5 [==============================] - 36s 7s/step - loss: 5.1676 - accuracy: 0.6392 - val_loss: 9.8339 - val_accuracy: 0.2075
```

Idea 2: Unsupervised Clustering Methods



Libraries Used

- scikit-learn
- scikit-surprise
- pandas
- numpy
- NLTK
- PySpark
- Unidecode

Preprocessing

- Use python re library and NLTK to remove featured artists, punctuation, bad characters, and text after ‘&’
- Use Unidecode to convert non-english characters such as ‘ä’ to ‘a’
- Lowercase all strings
- Remove all whitespaces

Ásgeir Trausti -> asgeirtrausti

Lifelike feat. A-Trak -> lifelike

Simon & Garfunkel -> simongarfunkel

Training

- Used **Matrix Factorization**, a collaborative filtering based technique that takes a User-Item rating matrix as input:

	Artist 1	Artist 2	Artist 3	Artist 4
User 1	1.5	?	3.6	4.4
User 2	2.9	1.9	?	2.2
User 3	?	3.3	4.8	?

Training (continued)

- Generated the rating matrix from the dataset by using the frequency of each user - artist interaction (how often a user listens to an artist)
- Scaled frequency using a MinMaxScaler between a range (1, 5).

Training (continued)

- Even with all the preprocessing, we had a 15914 x 289821 ratings matrix, which contained a total of 4,612,211,394 interactions
- Used the **Singular Value Decomposition (SVD) algorithm** from the scikit-surprise Python library to train our model for 25 epochs

Results

- **Epochs:** 25
- **Metric:** Root Mean Square Error (RMSE)
- For 2-fold cross validation:

```
Processing epoch 21
Processing epoch 22
Processing epoch 23
Processing epoch 24
RMSE: 0.0299
```

```
Processing epoch 21
Processing epoch 22
Processing epoch 23
Processing epoch 24
RMSE: 0.0298
```

Sample Prediction

Input uid: 00055176fea2346e027cd3302289378b

Output artists and songs:

```
final = {}
for artist in res.keys():
    final[artist] = df[df['artist'] == artist]['track'].value_counts()[0:5].index
final = pd.DataFrame(final)
final.T
```

	0	1	2	3	4
Vitamin String Quartet	Snow (Hey Oh) - Tribute to Red Hot Chili Peppers	Poker Face	Rolling in the Deep	Rebellion (Lies)	Starlight
Grateful Dead	Touch Of Grey	Friend Of The Devil - Remastered Version	Casey Jones (Remastered Album Version)	Truckin' (Remastered Album Version)	Ripple
Frank Zappa	Bobby Brown Goes Down	Peaches En Regalia	Joe's Garage	Watermelon In Easter Hay	Don't Eat The Yellow Snow
Wolfgang Amadeus Mozart	Piano Sonata No. 11 in A major, K. 331: III. R...	Requiem in D Minor, K. 626: Lacrimosa	Serenade No. 13 in G Major, K. 525 Eine Kleine...	Symphony No. 40 in G Minor, KV. 550: I: Molto ...	Sonata No. 16 in C Major for Piano, K. 545, So...
Girl Talk	Play Your Part (Pt. 1)	Shut The Club Down	Still Here	Hands In The Air	Set It Off

Future Scope

- Faster Processing - Parallelizing Queries
- Additional features for model (E.g. Danceability) - Better predictions
- Incorporating user responses and creating a pipeline to process new data dynamically

Conclusion

- Artists and Tracks for a user are one of the most influencing factors for deciding a new track to listen
- Users rate their songs based on Artists and listen more frequently to popular artists
- Several Music Streaming services use these metrics for recommendations (Spotify utilizes it the best)