Spotify NLP & Recommendation system Project Presentation

Shang-Shiun (Brian) Tsai

Northeastern University, Seattle

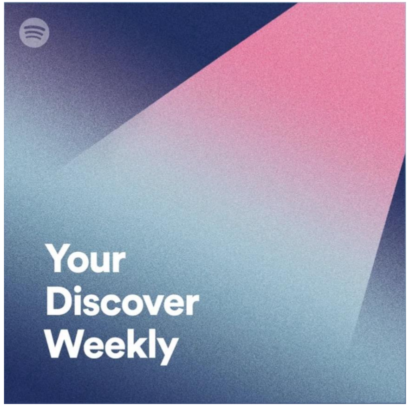
ALY 6040 Data Mining Applications

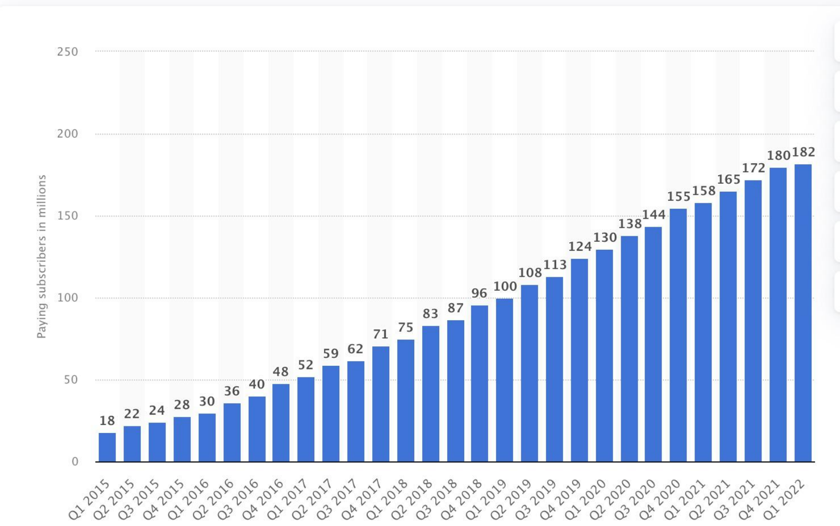
Shanu Sushmita

May 16, 2022

Part 1

- **NLP: Detecting Explicit Content in Music Lyrics**
- **Topic Modeling and Text classification**





Data Overview

The initial dataset, sourced from Kaggle, comprised a collection of 57,650 songs accompanied by lyrics from a diverse range of 643 artists. However, one crucial piece of information was missing: the explicit labels for these songs. To address this gap, we leveraged the Spotify API to assign explicit labels to each song, resulting in three distinct labels: 'True', 'False', and 'No match'. The 'No match' labels, indicating a failure to find an explicit tag, were subsequently removed from the dataset, leaving us with a refined set of 24,676 rows for further analysis.

	artist	song	text	explicit_label
1	ABBA	Andante, Andante	Take it easy with me, please Touch me gently...	False
2	ABBA	As Good As New	I'll never know why I had to go Why I had to...	False
4	ABBA	Bang-A-Boomerang	Making somebody happy is a question of give an...	False
7	ABBA	Chiquitita	Chiquitita, tell me what's wrong You're ench...	False
11	ABBA	Dancing Queen	You can dance, you can jive, having the time o...	False
...	...	...	...	...
57593	Zao	To Think Of You Is To Treasure An Absent Memory	When you shut your eyes and fell asleep Dark...	False
57605	Zebra	As I Said Before	And I said before I don't want no more And...	False
57608	Zebra	Hard Living Without You	Nothing to say no place to hide I can't find...	False
57609	Zebra	When You Get There	You wake up in the morning And you're not fe...	False
57612	Zebra	You're Only Losing Your Heart	Don't' do anything I wouldn't do Pass me any...	False

24676 rows × 4 columns

Getting labels via Spotify API

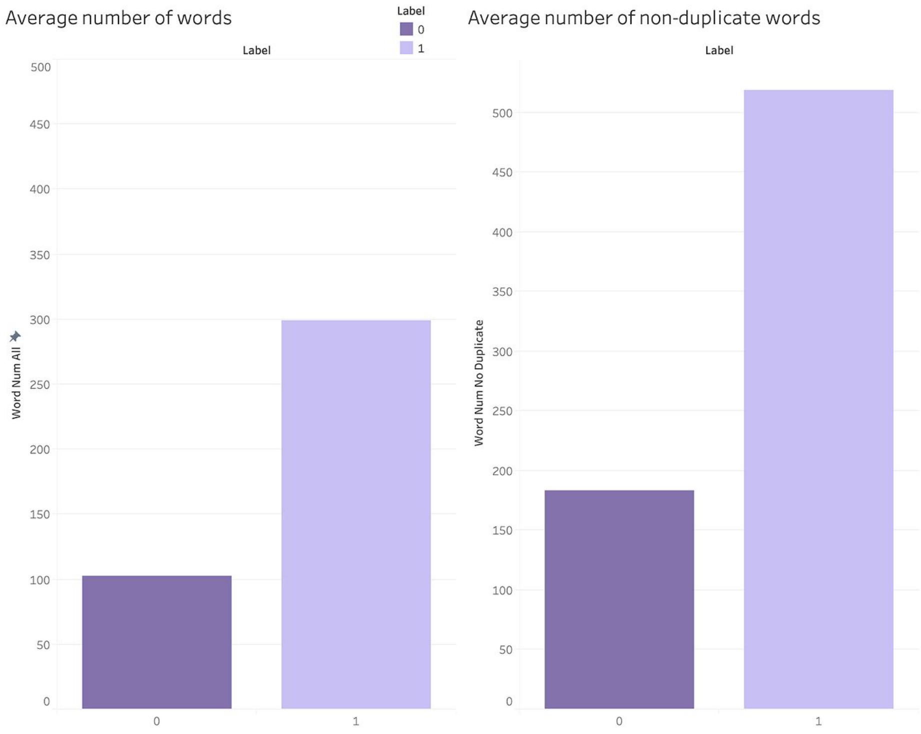
Through the utilization of the Spotify API, we successfully matched the names and artists of the songs in our dataset, thereby obtaining explicit labels for each song.

Here's how the explicit labels were determined:

- If a song contained explicit content, the Spotify API returned the label 'True' to indicate its explicit nature.
- Conversely, if a song was deemed free from explicit content, the API assigned the label 'False'.
- In cases where no match was found in the Spotify database, we assigned the label 'no match' to denote the absence of a conclusive result. Subsequently, we removed songs with this label from our dataset.

By leveraging the Spotify API, we were able to assign explicit labels to the songs in our dataset, enabling us to differentiate between explicit and non-explicit content accurately.

Exploratory Data Analysis (EDA)



word clouds


love
know
like
heart
want
way
time
say
one
life
tell
baby
chorus
take
world
day
still
make
go
think
feel


nigga
shit
that
bitch
ain't
fuck
love
never
chorus
oh
ooh
can't
money
wanna
fucking

Topic Modelling – Latent Dirichlet Allocation (LDA)

In order to gain a deeper insight into the content of the lyrics, we employed Topic Modeling techniques to uncover the underlying themes within the song texts. Specifically, we utilized the Python gensim package to implement Latent Dirichlet Allocation (LDA), a popular algorithm for topic modeling.

Text processing: lemmatization & stemming

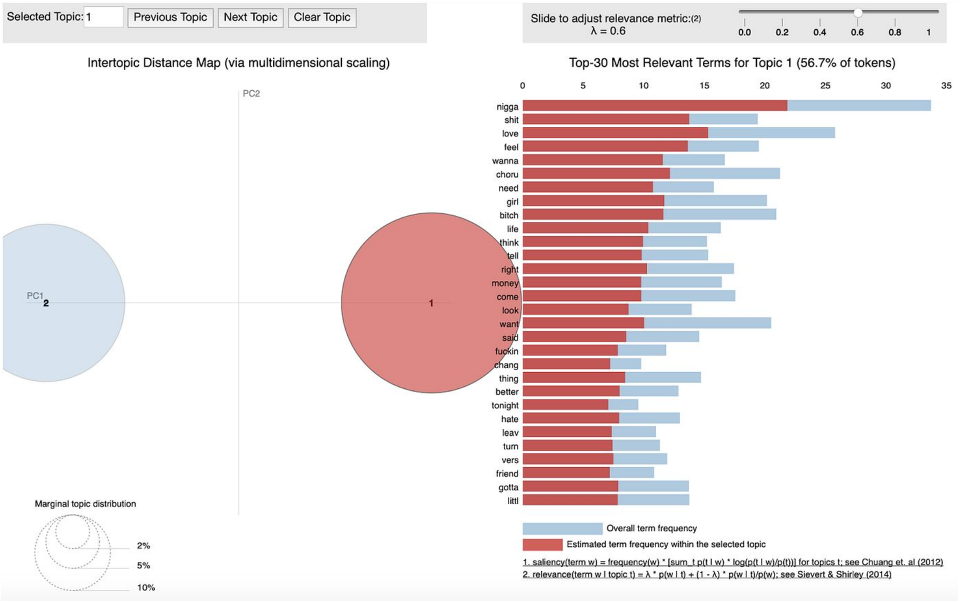
```
len(STOPWORDS)
stopwords = set(STOPWORDS)
stopwords.update(["yeah", "oh", "ya", "let"])
len(stopwords)


def lemmatization_stemming(text):
    wnl = WordNetLemmatizer()
    lemmatization = wnl.lemmatize(text, pos='n')

    stemmer = PorterStemmer()
    stemming = stemmer.stem(lemmatization)
    return stemming


def text_preprocess(text):
    result = []
    for token in simple_preprocess(text):
        if token not in stopwords and len(token) > 3:
            result.append(lemmatization_stemming(token))
    return result


processed_lyrics = song_data_1['text'].map(text_preprocess)


dictionary = Dictionary(processed_lyrics)
dictionary.filter_extremes()


bow_corpus = [dictionary.doc2bow(doc) for doc in processed_lyrics]

tfidf_model = TfidfModel(bow_corpus)
tfidf_corpus = tfidf_model[bow_corpus]
```

LDA Visualization

Two visualizations were generated using pvLDAvis to enhance our understanding of the topic modeling results.

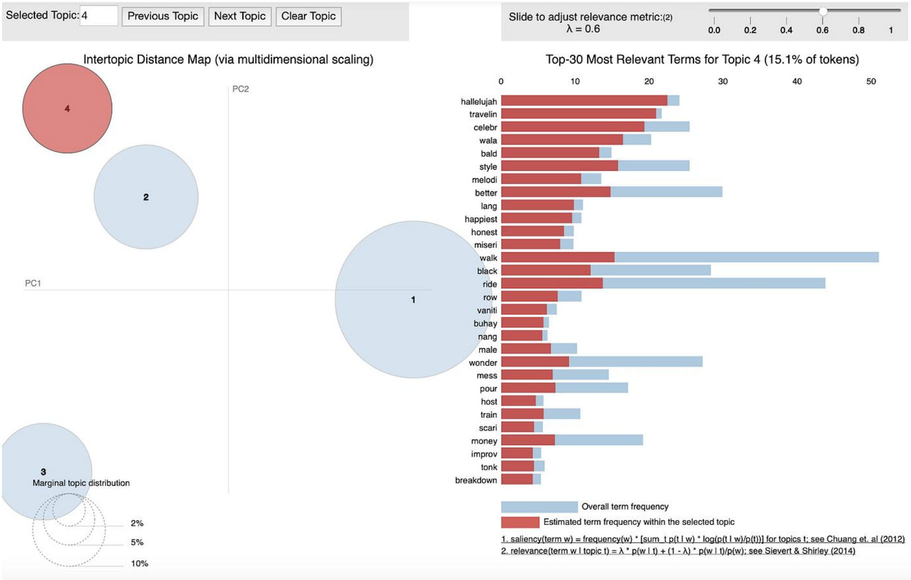
In the left-side plot, each circle represents a topic, with the size of the circle indicating the importance of the topic within the entire corpus. The distance between the centers of two circles indicates the similarity between the corresponding topics. This visualization allows us to grasp the relative significance and relationships between different topics.

Upon selecting a specific topic, the right-side histogram presents the top 30 most relevant terms associated with that topic. The blue bars represent the overall term frequency across the entire corpus, while the red bars represent the estimated term frequency specifically for the selected topic. This histogram provides insights into the importance and prevalence of certain terms within the chosen topic.



For explicit lyrics

Through the application of LDA on non-explicit lyrics, we were able to extract four primary topics. As we delve into each topic, we observe a predominant association of words with specific themes or characters. For instance, let's consider topic 4 as an example. Within this topic, we find highly relevant terms such as 'hallelujah' and 'celebr', indicating a strong likelihood that topic 4 pertains to religious themes.



For non-explicit lyrics

Custom Features

1.Number of words:

As previously mentioned, we observed that explicit songs tend to have a higher average word count compared to non-explicit songs. This led us to hypothesize that the number of words could potentially serve as a distinguishing feature between the two song types.

```
def get_num_words(review):
  threshold = 0
  words = review.split(' ')
  count = len(list(words))
  return count > threshold
```

2.Number of bad words

We obtained an Offensive/Profane Word List from Luis von Ahn’s Research Group at CMU, comprising more than 1,300 English terms that are considered offensive. We utilized this list to count the occurrence of these words within each song, considering it as one of the features in our analysis.

```
def get_bad_words(review):
  target_word = bad_words
  count = 0
  threshold = 0
  for t in target_word:
        if review.find(t) != -1:
            count += 1
  return count > threshold
```

3.Number of keywords from LDA

We extracted keywords from explicit songs using LDA, and we believe that these topic words could serve as valuable features in the classification process.

```
def get_lda_words(review):
  target_word = ['chorus','girl','money','baby','nigga','bitch','want','love','wanna','g
  count = 0
  threshold = 0
  for t in target_word:
        if review.find(t) != -1:
            count += 1
  return count > threshold
```

Text Classification

Rebalancing the dataset

In our original dataset, the distribution of 'True' labels to 'False' labels was highly imbalanced, with a ratio of 1:17. To address this issue of imbalanced data, we randomly extracted 5424 rows from the dataset, ensuring a new ratio of 'True' to 'False' labels at 1:3, before proceeding with the modeling process.

Model performance

Then we ran 5 classification models

Model	Parameters		Precision	Recall	F1-score
Logistic Regression	C=50	0	0.89	0.97	0.88
		1	0.87	0.64	
Decision Tree	criterion='gini', min_samples_split=0.4, max_depth=77	0	0.94	0.92	0.90
		1	0.76	0.83	
Random Forest	n_estimator=110, max_depth=140, min_samples_split=30	0	0.90	0.98	0.89
		1	0.92	0.64	
SVM	C = 10000, kernel = 'rbf'	0	0.88	0.98	0.87
		1	0.92	0.57	
KNN	n_neighbors=10	0	0.83	0.99	0.81
		1	0.89	0.37	

Challenges

The initial challenge we encountered was the limited information provided by the lyrics due to their inherent nature, particularly when it came to topic modeling. For instance, the presence of interjection words like 'yeah' and 'oh,' as well as the repetition of sentences, posed difficulties in extracting meaningful insights. Additionally, detecting nuances within languages, such as metaphors or subtle indications of violence or sexuality in lyrics, proved challenging without the necessary contextual information. Analyzing explicit metaphors or hints without proper context presented considerable difficulties.

Conclusion

The Decision Tree Model demonstrated the highest accuracy in detecting explicit content in English lyrics, as evidenced by its superior F-1 score compared to other models.

Furthermore, the versatility of the model extends beyond English-language content, as it can effectively identify profanity in other languages given the availability of similar data.

In various industry applications, this model proves invaluable in the following key scenarios:

1. Musicians and music publishers can proactively assess their songs using the model before undergoing official screening reviews, ensuring adherence to content guidelines.

2. Music streaming services, such as Spotify, can seamlessly implement the model to automatically assign "parent advisory" tags to music content, enabling users to make informed choices about their listening preferences.

3. Families who share music libraries, such as their iPod playlists, can utilize the model to screen and filter out inappropriate content, safeguarding younger audiences from exposure to explicit material.

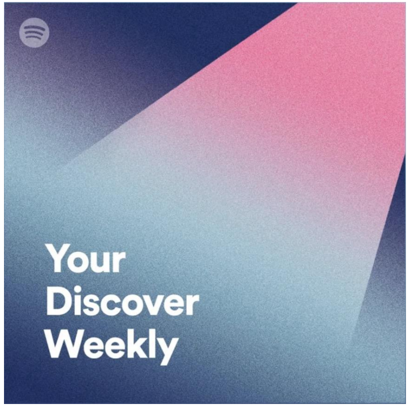
4. Government entities can leverage the model to effectively censor musical content in specific situations, aligning with political and regulatory considerations.

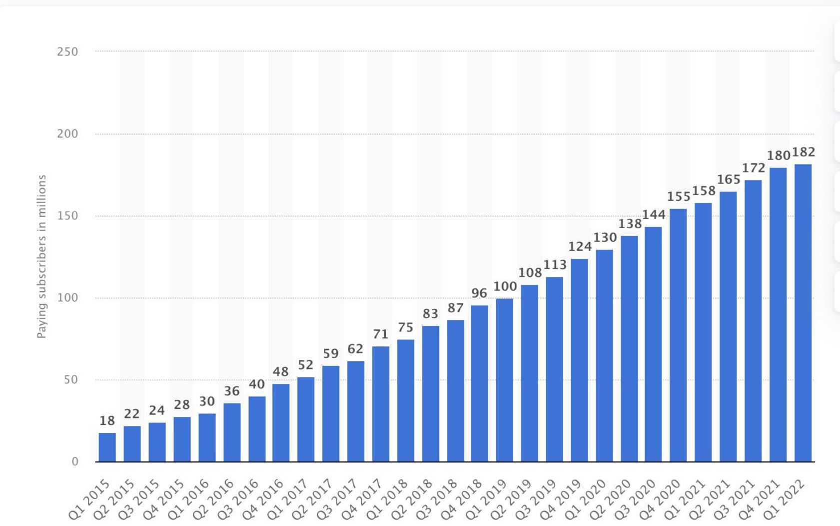
5. Educational institutions can equip their computer systems with the model, enabling them to prevent underage students from accessing and being exposed to unsuitable content.

Part 2

- **Recommendation System by Genre**

- **Popularity Prediction**





EDA

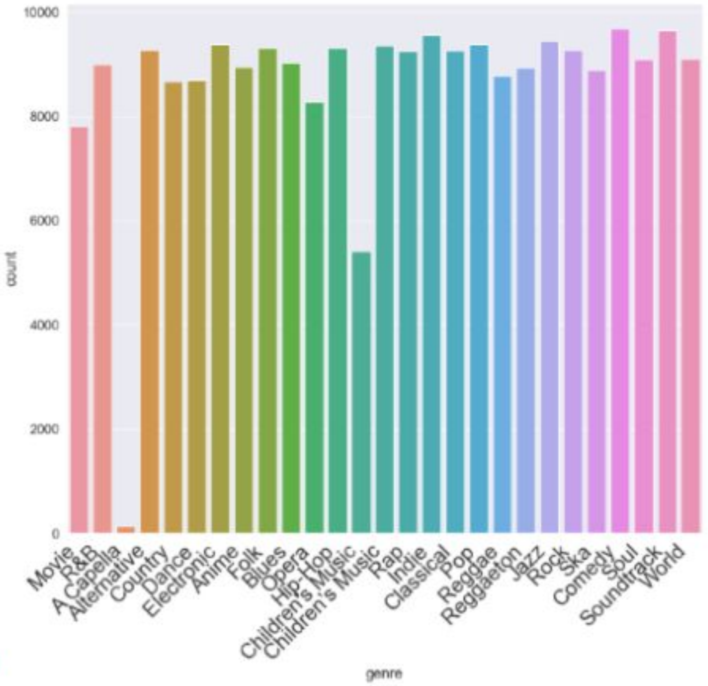
```
df.info()
```

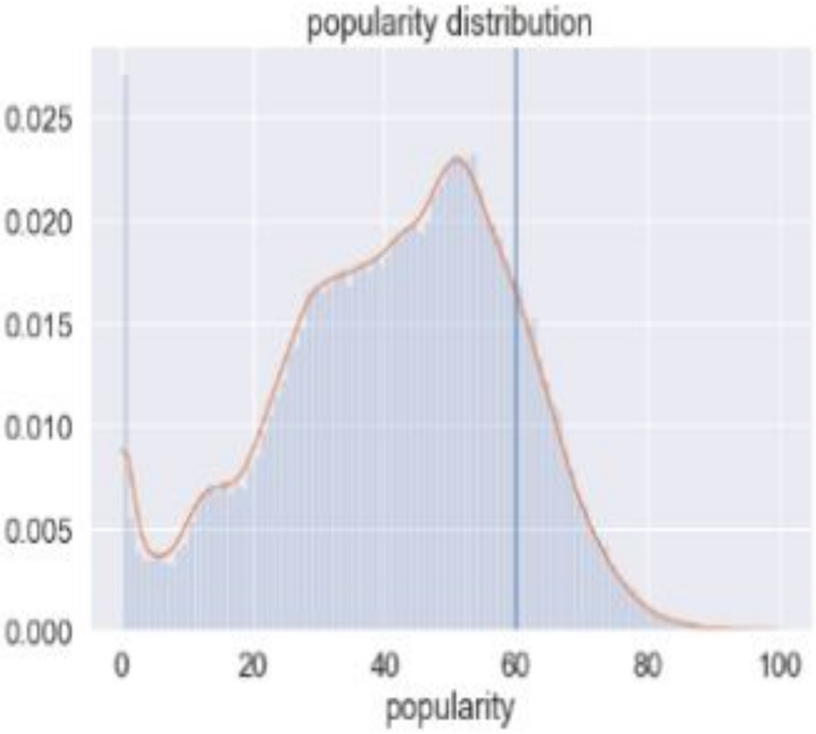
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 232725 entries, 0 to 232724
Data columns (total 18 columns):
 #   Column            Non-Null Count   Dtype  
---  ------            --------------   -----  
 0   genre             232725 non-null  object 
 1   artist_name       232725 non-null  object 
 2   track_name        232725 non-null  object 
 3   track_id          232725 non-null  object 
 4   popularity        232725 non-null  int64  
 5   acousticness      232725 non-null  float64
 6   danceability      232725 non-null  float64
 7   duration_ms       232725 non-null  int64  
 8   energy            232725 non-null  float64
 9   instrumentalness  232725 non-null  float64
 10  key               232725 non-null  object 
 11  liveness          232725 non-null  float64
 12  loudness          232725 non-null  float64
 13  mode              232725 non-null  object 
 14  speechiness       232725 non-null  float64
 15  tempo             232725 non-null  float64
 16  time_signature    232725 non-null  object 
 17  valence           232725 non-null  float64
dtypes: float64(9), int64(2), object(7)
memory usage: 32.0+ MB
```

	genre	artist_name	track_name	track_id	popularity	acousticness	danc
0	Movie	Henri Salvador	C'est beau de faire un Show	0BRjO6ga9RKCKjfDqeFgWV	0	0.61100	
1	Movie	Martin & les fées	Perdu d'avance (par Gad Elmaleh)	0BjC1NfoEOOusryehmNudP	1	0.24600	
2	Movie	Joseph Williams	Don't Let Me Be Lonely Tonight	0CoSDzoNIKCRs124s9uTVy	3	0.95200	
3	Movie	Henri Salvador	Dis-moi Monsieur Gordon Cooper	0Gc6TVm52BwZD07Ki6tIvf	0	0.70300	
4	Movie	Fabien Nataf	Ouverture	0IuslXpMROHdEPvSl1fTQK	4	0.95000	
...	...	...	...	...	...	...	
232720	Soul	Slave	Son Of Slide	2XGLdVl7IGeq8ksM6Al7jT	39	0.00384	
232721	Soul	Jr Thomas & The Volcanos	Burning Fire	1qWZdkBI4UVPj9IK6HuuFM	38	0.03290	
232722	Soul	Muddy Waters	(I'm Your) Hoochie Coochie Man	2ziWXUmQLrXTiYjCg2fZ2t	47	0.90100	
232723	Soul	R.LUM.R	With My Words	6EFsue2YbIG4Qkq8Zr9Rir	44	0.26200	
232724	Soul	Mint Condition	You Don't Have To Hurt No More	34XO9RwPMKjbvRry54QzWn	35	0.09730	

232725 rows × 18 columns

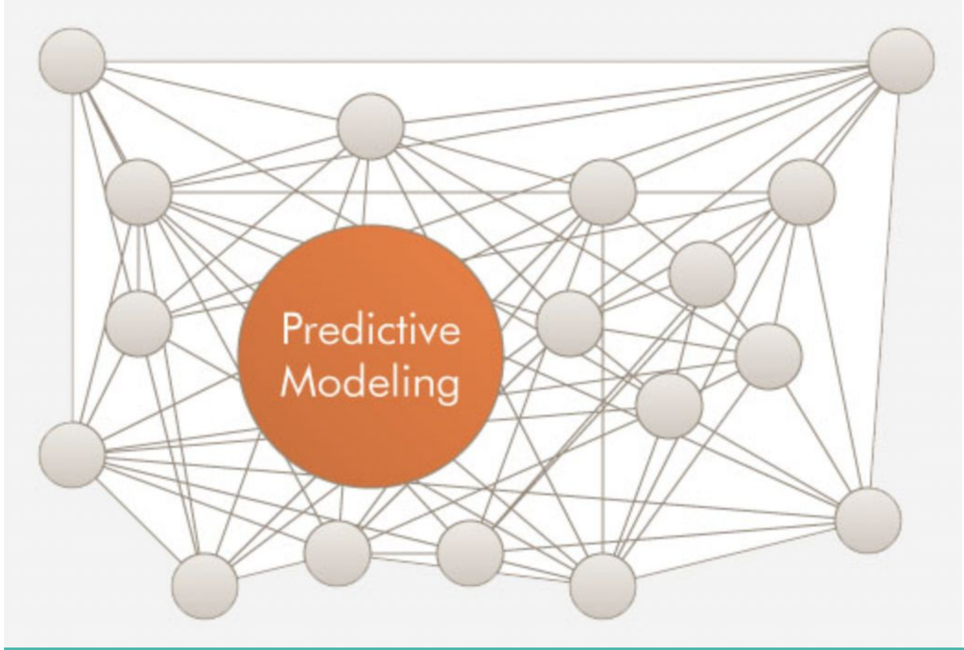
EDA


10000
8000
6000
4000
2000
0
count
Movie
R&B
A Capella
Alternative
Country
Dance
Electronic
Anime
Folk
Blues
Opera
Hip-Hop
Children's Music
Children’s Music
Rap
Indie
Classical
Pop
Reggae
Reggaeton
Jazz
Rock
Ska
Comedy
Soul
Soundtrack
World
genre


popularity distribution
0.025
0.020
0.015
0.010
0.005
0.000
0
20
40
60
80
100
popularity

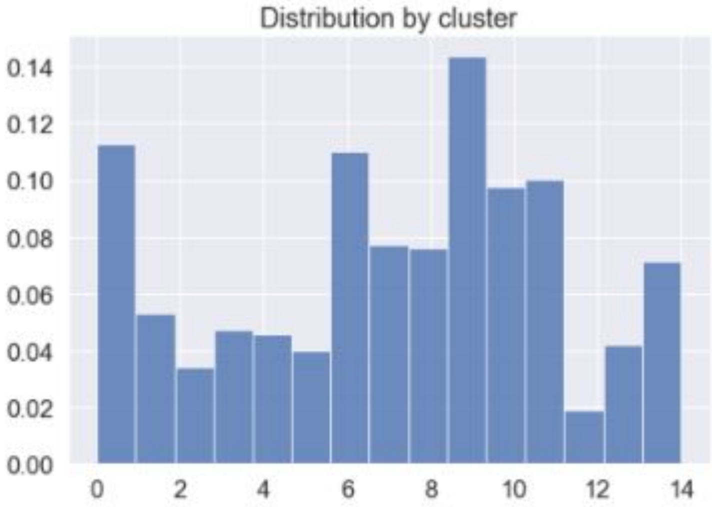
Recommendation System by Genre

Cosine Similarity
Random Forest
XGBoost



Recommendation

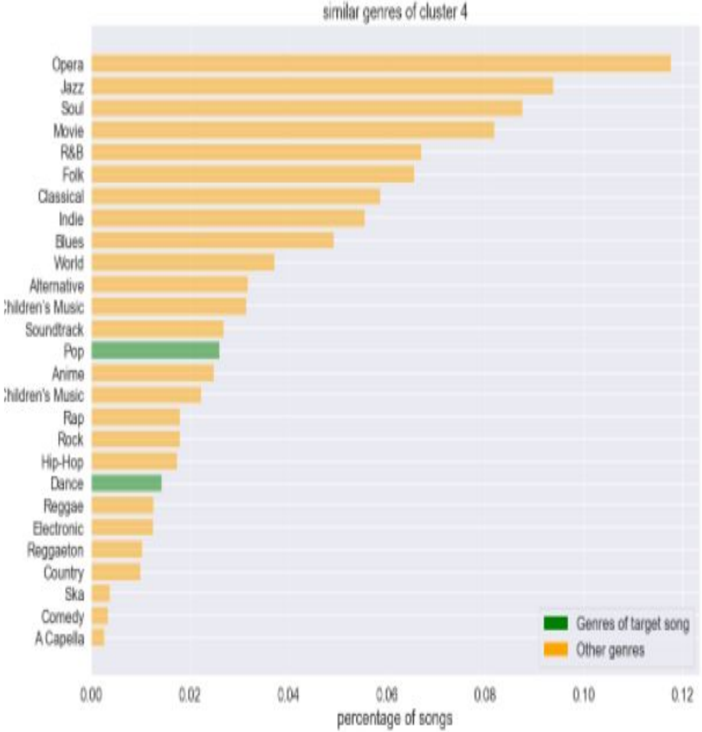
Cluster K=15

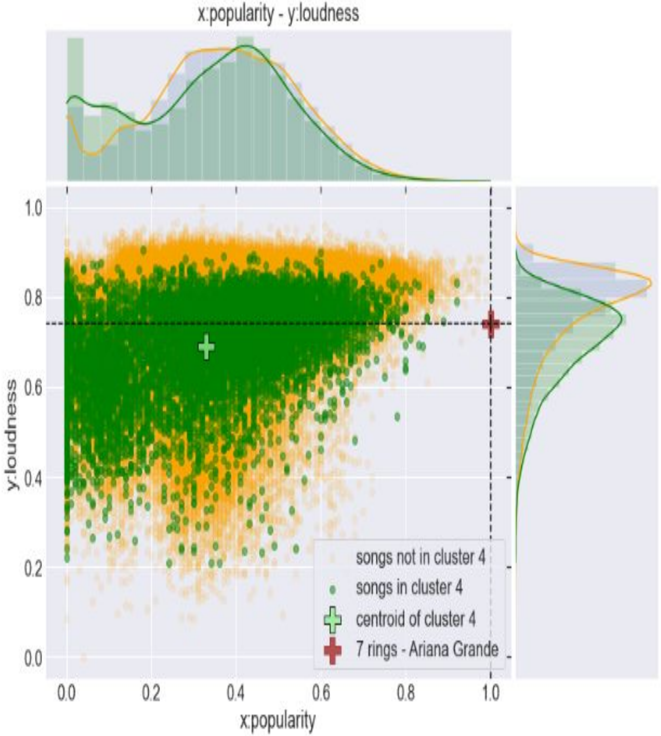


Ariana Grande's “7 rings”

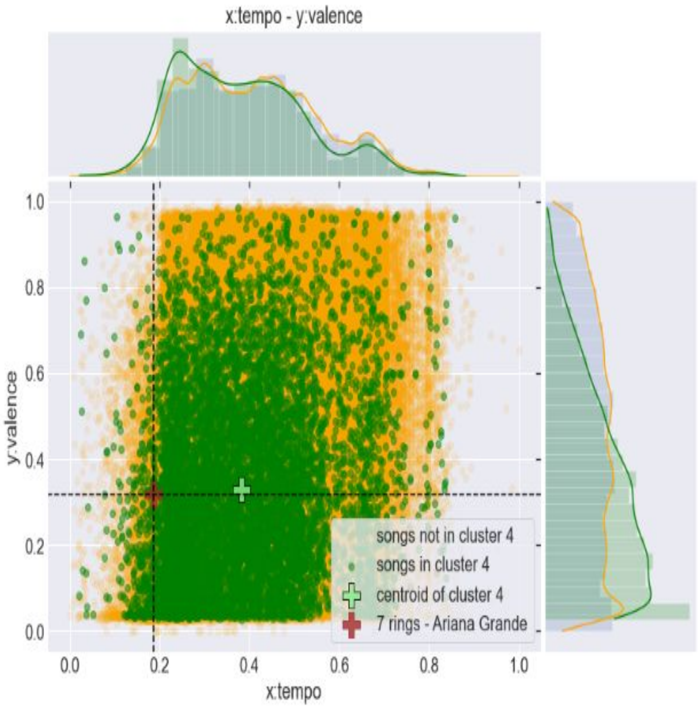
```
most similar songs:
    Ariana Grande - 7 rings
    Metro Boomin - 10 Freaky Girls (with 21 Savage)
    Ariana Grande - 7 rings (feat. 2 Chainz) - Remix
    gnash - i hate u, i love u (feat. olivia o'brien)
    H.E.R. - Could've Been (feat. Bryson Tiller)
    Drake - From Time
    Camila Cabello - All These Years
    Daniel Caesar - Who Hurt You?
    Daniel Caesar - Get You (feat. Kali Uchis)
    Metro Boomin - Lesbian (feat. Gunna & Young Thug)
```

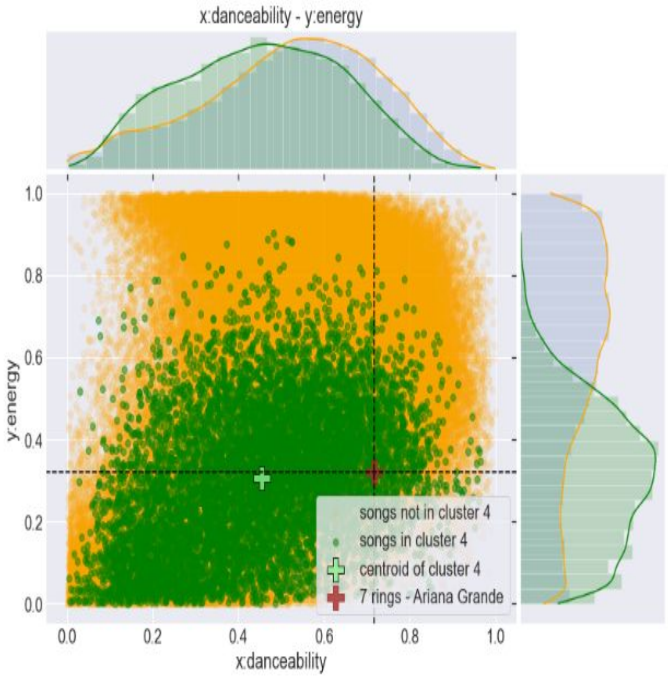
Cosine Similarity





Cosine Similarity





Random Forest & XGBoost

- Accuracy score is pretty low and underfitting the model

Random Forest

```
In [68]: clf_en = DecisionTreeClassifier(criterion='gini', max_depth=3, random_state=0)
         clf_en.fit(X_train, y_train)
         y_pred_en = clf_en.predict(X_test)
         print('Model accuracy score with criterion entropy: {0:0.4f}'. format(accuracy_score(y_test, y_pred_en)))
         clf.fit(X_train, y_train)
         y_pred = clf.predict(X_test)

Out[68]: DecisionTreeClassifier(max_depth=3, random_state=0)
```

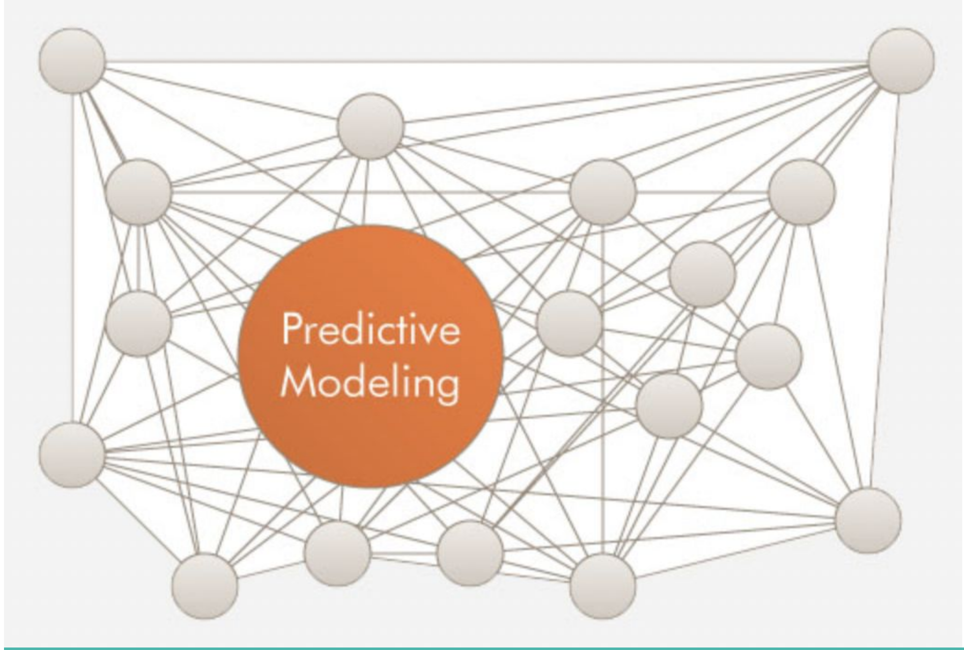
```
In [61]: Accuracy = accuracy_score(y_test, y_pred)
         print("Accuracy: " + str(Accuracy))

         Accuracy: 0.1161671500698249
```

XGBoost

```
In [16]: XGB_Model = XGBClassifier(objective = "binary:logistic", n_estimators = 10, seed = 123)
         XGB_Model.fit(X_train, y_train)
         XGB_Predict = XGB_Model.predict(X_test)
         XGB_Accuracy = accuracy_score(y_test, XGB_Predict)
         print("Accuracy: " + str(XGB_Accuracy))

         XGB_AUC = roc_auc_score(y_test, XGB_Predict)
         print("AUC: " + str(XGB_AUC))

         Accuracy: 0.14287248898915028
```

Popularity Prediction

Decision Tree
Logistic Regression
Random Forest

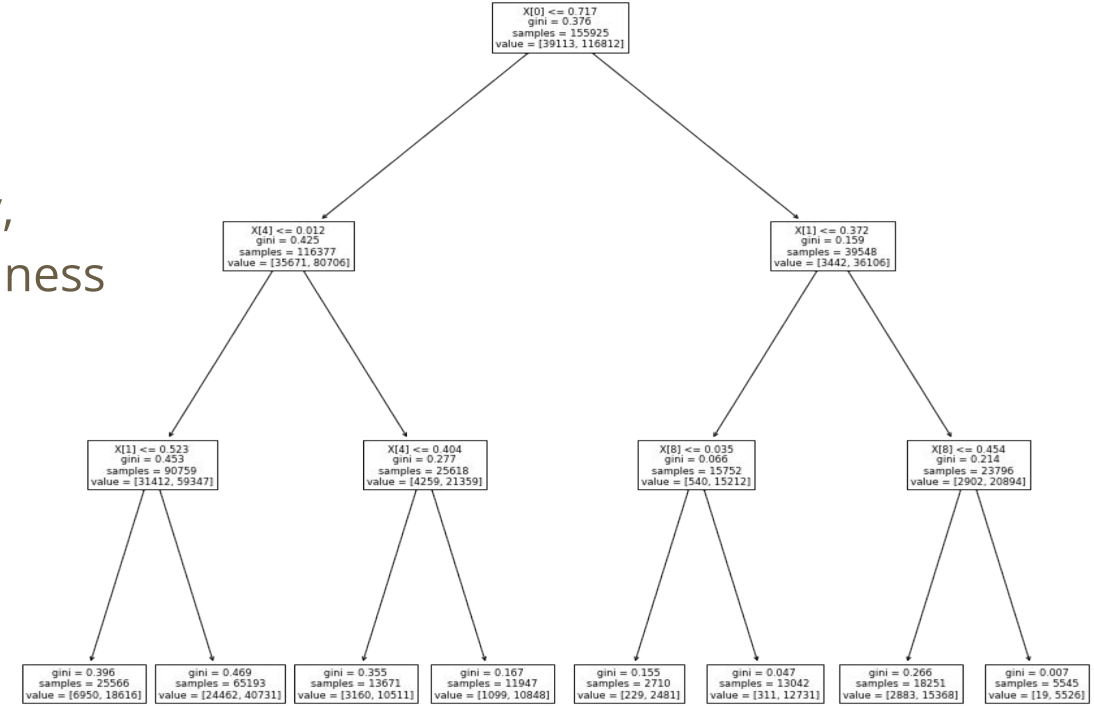


Popularity Prediction

- 0 to 100
- 55 → 75 percentile in popularity column
- Features: Acoustiness, Danceability, Duration, Energy, Instrumentalness, Key, Liveness, Mode, Speechiness, Tempo, Time signature, Valence
- Decision Trees
- Logistics Regression
- Random Forest

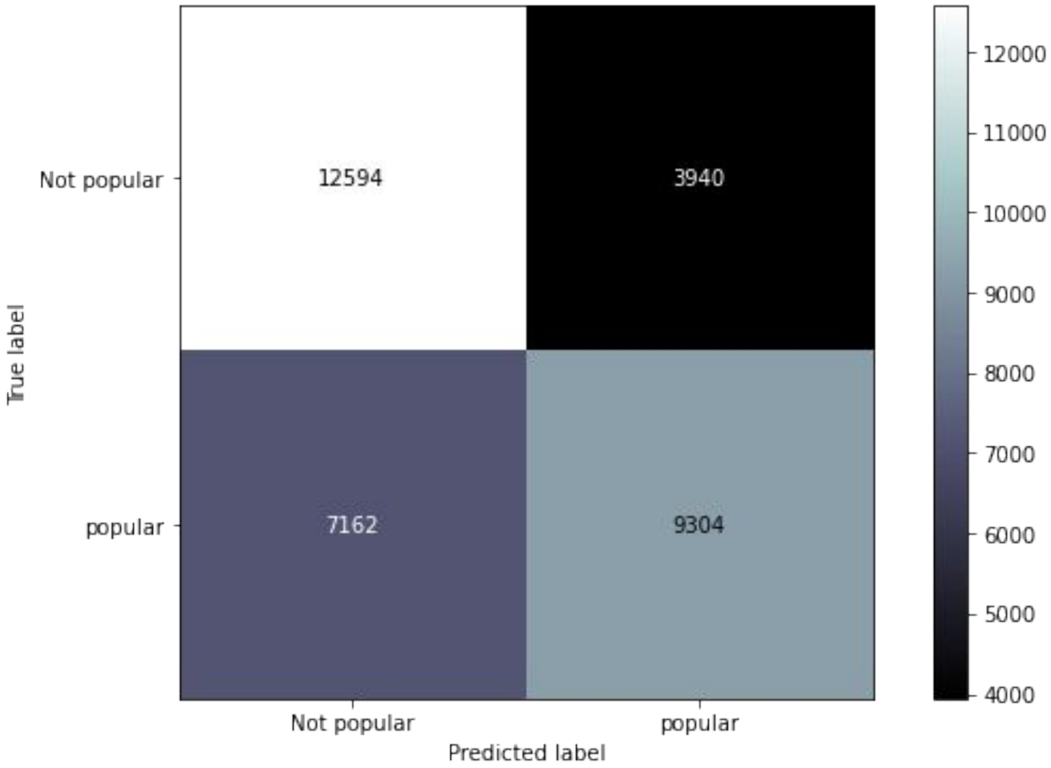
Decision Tree Classifier

- Accuracy score: 75% (no overfitting)
- Acousticness, Danceability, Instrumentalness, Speechiness



Logistic Regression

- 50,000 samples
- Accuracy score: 50%
- Precision: 50%
- F1 score: 66%



Random Forest

- Accuracy score is 0.90 without parameters.
- Overfitting issue persists when not resampling the data
- Best performance even there is overfitting issue compared to decision tree and logistic regression.

Random Forest

In [36]:
```python
from sklearn.ensemble import RandomForestClassifier

RFC_Model = RandomForestClassifier()
RFC_Model.fit(X_train, y_train)
RFC_Predict = RFC_Model.predict(X_test)
RFC_Accuracy = accuracy_score(y_test, RFC_Predict)
print("Accuracy: " + str(RFC_Accuracy))

from sklearn.metrics import make_scorer, accuracy_score, roc_auc_score

RFC_AUC = roc_auc_score(y_test, RFC_Predict)
print("AUC: " + str(RFC_AUC))
```

```
Accuracy: 0.9070833333333334
AUC: 0.8377863627833014
```

In [42]:
```python
# check for overfitting or underfitting by printing accuracy score for train and test data

print('Training set score: {0:0.4f}'. format(RFC_Model.score(X_train, y_train)))
print('Test set score: {0:0.4f}'. format(RFC_Model.score(X_test, y_test)))
```

```
Training set score: 0.9919
Test set score: 0.9071
```

Conclusion & Limitations

Conclusion

- Genre result
  - Important features for building recommendation system: danceability, tempo
- Popularity result
  - Random forest is the final selected model with best accuracy score and less overfitting issue when removing classifier parameters

Limitations

- The accuracy score is low or overfitted - genre
- Missing data points (ID & Personal Information) to build personalized music recommendation system for each individual
- Trend might change over time
- Popularity prediction isn’t entirely based on the variables we currently have

Thank you

Q&A