Project Report

Music Recommendation System for Spotify Playlists

Prasanna Vengatesh Venkataraman

Shwetha Raj

Department of Information and Decision Sciences - University of Illinois at Chicago

IDS561 - Analytics for Big Data

Prof. Yuheng Yu

December 9th, 2020

What is Recommender System?

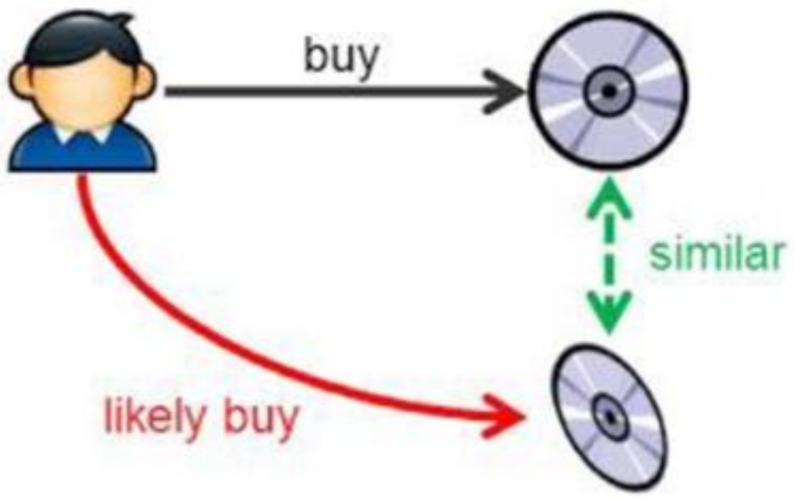
In the modern age of internet, people are left with many options to choose from abundant items to use. A recommender system is a system that enables the user to choose from the top items based on certain set of items that is of the user's interest. In the past, people were left with only limited options to choose from as most of the items picked were from a physical store. Nowadays users are not limited to a physical store, they can choose from thousands of options running throughout the internet. With the increase in the availability of information, there is a new problem where people have a hard time selecting the items they want to see. This problem is solved using the recommender system. There are 2 ways for building this system – Collaborative filtering and Content- based recommendation. For our project we are building a Content-based recommendation system for the Spotify dataset where the playlist information saved by the user will help in solving the problem by finding the right music as per the user needs.

Content-based recommendation system

Collaborative filtering-based recommender systems have been proven useful, but there is space left for improvements by adapting Content-based filtering approach to better fit to the music recommendations problem. In this project, context-based information about the music consumption is incorporated into the recommendation process. This information is extracted from playlist names, which are analyzed and aggregated into so-called contextual clusters. We find that the listening context plays an important role and thus allows for providing improved recommendations in comparison to the traditional approaches.

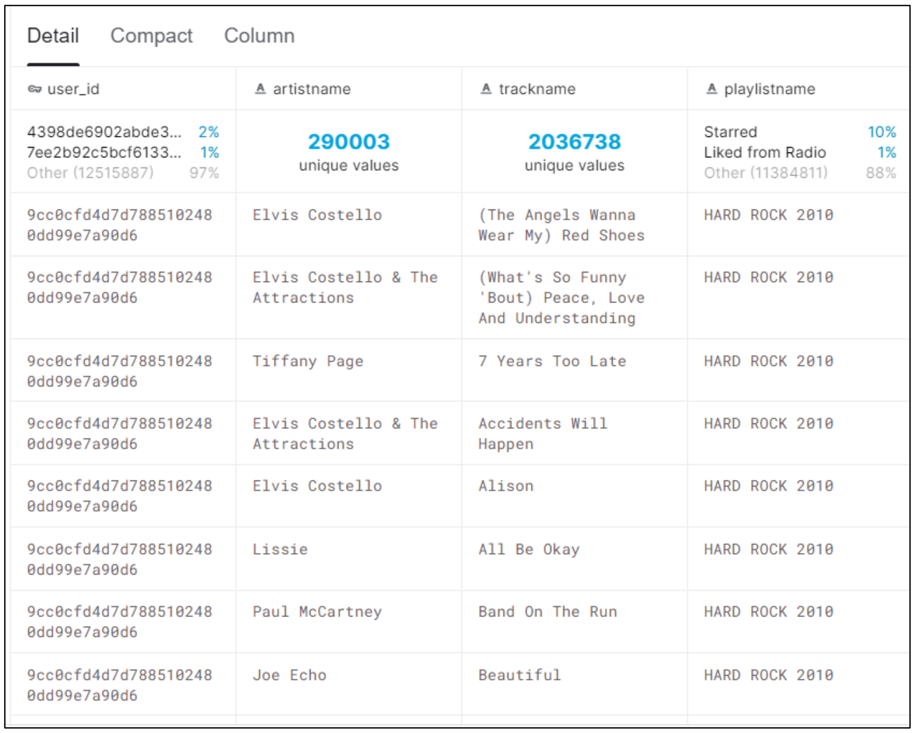
The basic idea of this method is each item has few features x which holds a score for every item in the users list. For each user, determined by a parameter, the scores of the features are consolidated to check their preference. For example, if a dataset has scores for certain playlists like rock, pop and blues and the user holds a score parameter [1, 0, 0] it means that user's interests lie in rock and the system will recommend playlists that has genre rock.

While building the model, the system finds the similarity between all pairs of items, then it uses the most similar items to a user's items that are already rated or scored to generate a list of recommendations in recommendation stage.



Spotify Data

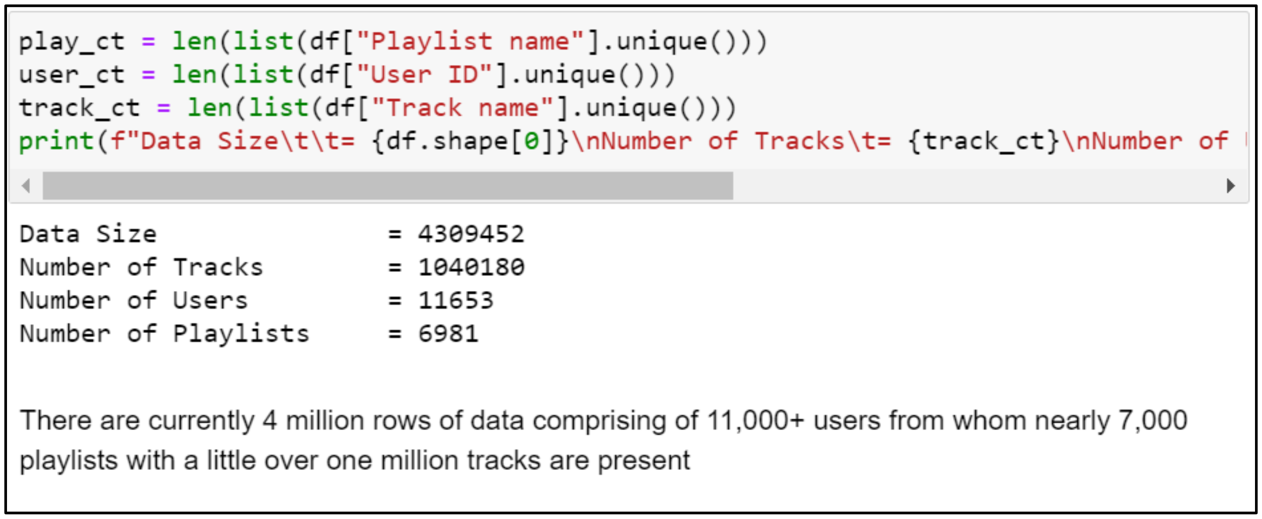
Data is retrieved from Kaggle which used the #nowplaying dataset by Zangerle with playlist names to enrich the Spotify users in the dataset.



Detail Compact Column

user_id	artistname	trackname	playlistname
4398de6902abde3... 2% 7ee2b92c5bcf6133... 1% Other (12515887) 97%	290003 unique values	2036738 unique values	Starred 10% Liked from Radio 1% Other (11384811) 88%
9cc0cfd4d7d788510248 0dd99e7a90d6	Elvis Costello	(The Angels Wanna Wear My) Red Shoes	HARD ROCK 2010
9cc0cfd4d7d788510248 0dd99e7a90d6	Elvis Costello & The Attractions	(What's So Funny 'Bout) Peace, Love And Understanding	HARD ROCK 2010
9cc0cfd4d7d788510248 0dd99e7a90d6	Tiffany Page	7 Years Too Late	HARD ROCK 2010
9cc0cfd4d7d788510248 0dd99e7a90d6	Elvis Costello & The Attractions	Accidents Will Happen	HARD ROCK 2010
9cc0cfd4d7d788510248 0dd99e7a90d6	Elvis Costello	Alison	HARD ROCK 2010
9cc0cfd4d7d788510248 0dd99e7a90d6	Lissie	All Be Okay	HARD ROCK 2010
9cc0cfd4d7d788510248 0dd99e7a90d6	Paul McCartney	Band On The Run	HARD ROCK 2010
9cc0cfd4d7d788510248 0dd99e7a90d6	Joe Echo	Beautiful	HARD ROCK 2010

Raw dataset from Kaggle



```
play_ct = len(list(df["Playlist name"].unique()))
user_ct = len(list(df["User ID"].unique()))
track_ct = len(list(df["Track name"].unique()))
print(f"Data Size\t\t= {df.shape[0]}\nNumber of Tracks\t= {track_ct}\nNumber of
```

```
Data Size               = 4309452
Number of Tracks        = 1040180
Number of Users         = 11653
Number of Playlists     = 6981
```

There are currently 4 million rows of data comprising of 11,000+ users from whom nearly 7,000 playlists with a little over one million tracks are present

Processed dataset with details

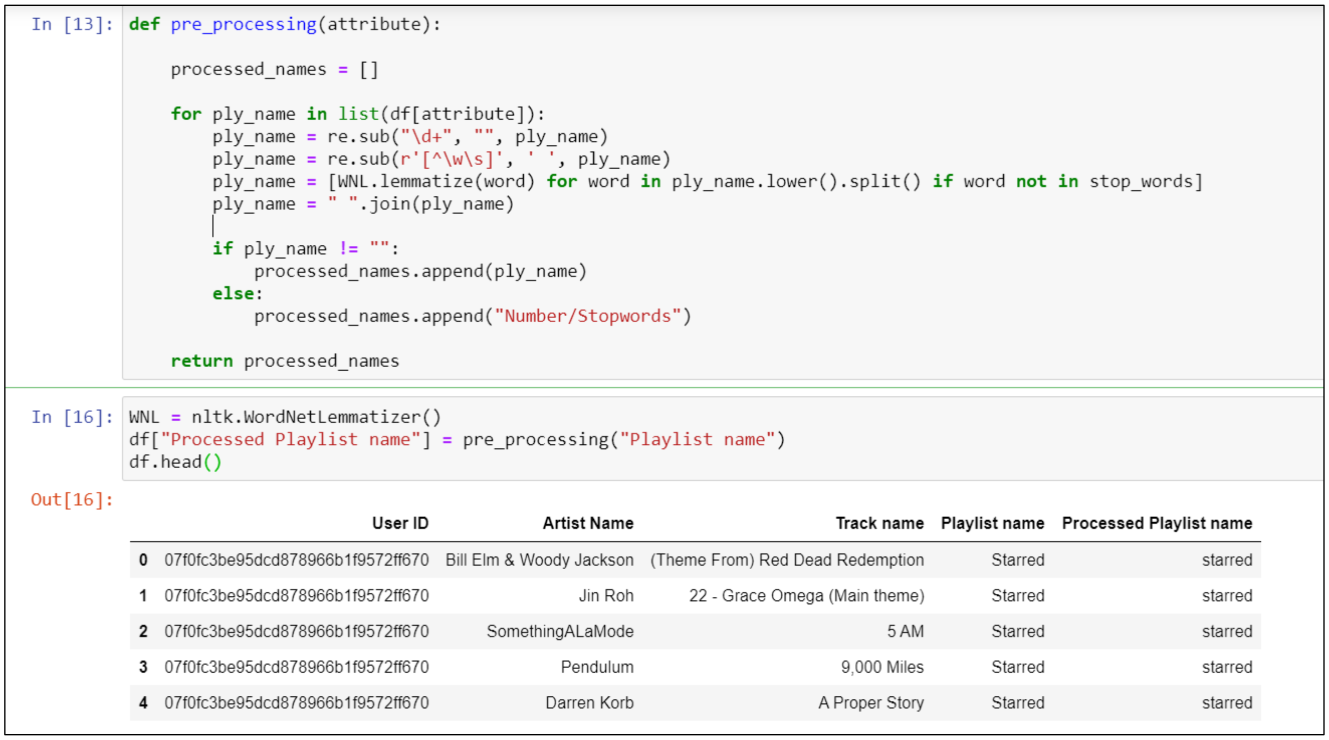
Techniques and tools

Tools: Python and Pyspark for cleaning data and building model, Tableau for visualization

Technique:

1. **Packages used:**
   - Pandas – data manipulation and analysis
   - IPython.display – used HTML package for displaying the dataframe without index
   - nltk.corpus – uses the stopwords package for removing stopwords from the column
   - sklearn.decomposition – uses PCA for dimensionality reduction
   - Pyspark for spark dataframe API
   - pyspark.ml – used kmeans from clustering for model building

2. **Data Cleaning**: Used pandas, WordNetLemmatizer from nltk to process the Playlist names



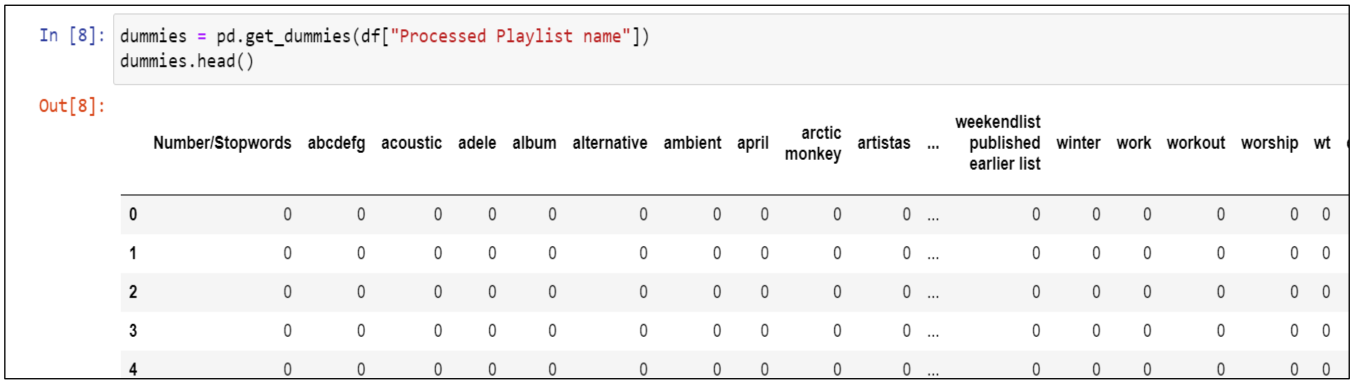
```
In [13]: def pre_processing(attribute):

             processed_names = []

             for ply_name in list(df[attribute]):
                 ply_name = re.sub("\d+", "", ply_name)
                 ply_name = re.sub(r'[^\w\s]', ' ', ply_name)
                 ply_name = [WNL.lemmatize(word) for word in ply_name.lower().split() if word not in stop_words]
                 ply_name = " ".join(ply_name)

                 if ply_name != "":
                     processed_names.append(ply_name)
                 else:
                     processed_names.append("Number/Stopwords")

             return processed_names
```

```
In [16]: WNL = nltk.WordNetLemmatizer()
         df["Processed Playlist name"] = pre_processing("Playlist name")
         df.head()
```

Out[16]:

	User ID	Artist Name	Track name	Playlist name	Processed Playlist name
0	07f0fc3be95dcd878966b1f9572ff670	Bill Elm & Woody Jackson	(Theme From) Red Dead Redemption	Starred	starred
1	07f0fc3be95dcd878966b1f9572ff670	Jin Roh	22 - Grace Omega (Main theme)	Starred	starred
2	07f0fc3be95dcd878966b1f9572ff670	SomethingALaMode	5 AM	Starred	starred
3	07f0fc3be95dcd878966b1f9572ff670	Pendulum	9,000 Miles	Starred	starred
4	07f0fc3be95dcd878966b1f9572ff670	Darren Korb	A Proper Story	Starred	starred

3. **One- hot encoding** – used get_dummies function from Pandas to perform one hot encoding for the 250 unique playlist names which converts it into binary form

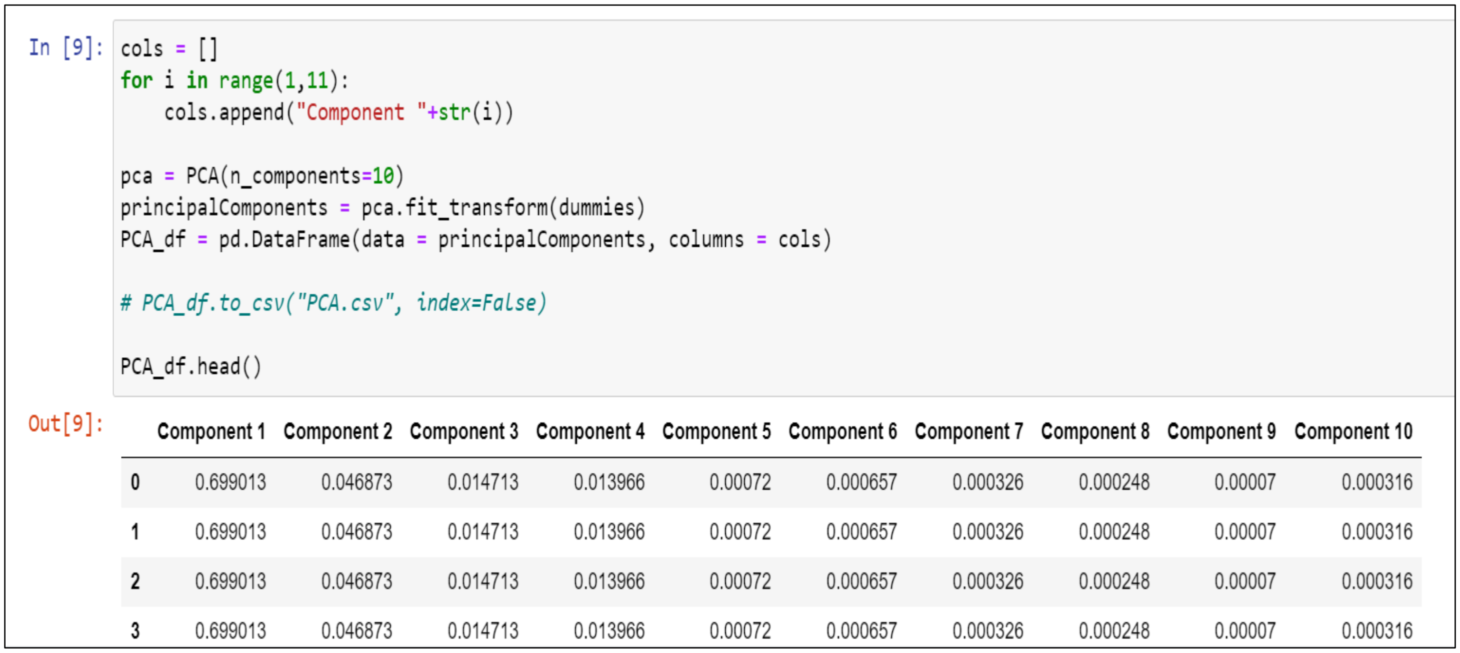


```
In [8]: dummies = pd.get_dummies(df["Processed Playlist name"])
        dummies.head()
```

Out[8]:

	Number/Stopwords	abcdefg	acoustic	adele	album	alternative	ambient	april	arctic monkey	artistas	...	weekendlist published earlier list	winter	work	workout	worship	wt
0	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0

4. **Principle Component Analysis** – used sklearn.decomposition to perform PCA on the 250 measures in order to reduce the dimensions to 10 columns and preserve the originality so that any computations on the data remains unaffected

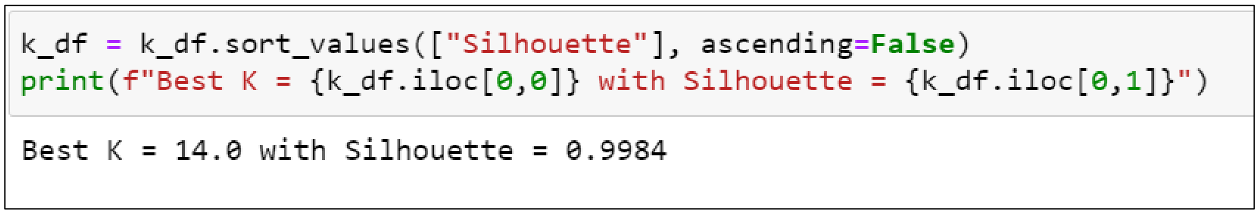


```
In [9]: cols = []
        for i in range(1,11):
            cols.append("Component "+str(i))

        pca = PCA(n_components=10)
        principalComponents = pca.fit_transform(dummies)
        PCA_df = pd.DataFrame(data = principalComponents, columns = cols)

        # PCA_df.to_csv("PCA.csv", index=False)

        PCA_df.head()
```

Out[9]:

	Component 1	Component 2	Component 3	Component 4	Component 5	Component 6	Component 7	Component 8	Component 9	Component 10
0	0.699013	0.046873	0.014713	0.013966	0.00072	0.000657	0.000326	0.000248	0.00007	0.000316
1	0.699013	0.046873	0.014713	0.013966	0.00072	0.000657	0.000326	0.000248	0.00007	0.000316
2	0.699013	0.046873	0.014713	0.013966	0.00072	0.000657	0.000326	0.000248	0.00007	0.000316
3	0.699013	0.046873	0.014713	0.013966	0.00072	0.000657	0.000326	0.000248	0.00007	0.000316

5. **K-Means Clustering**: For clustering the common playlists, we are going to implement K-means clustering. Before building the model, we use the ClusteringEvaluator package from pyspark.ml.evaluation module to evaluate the best Silhouette measure to build the model.



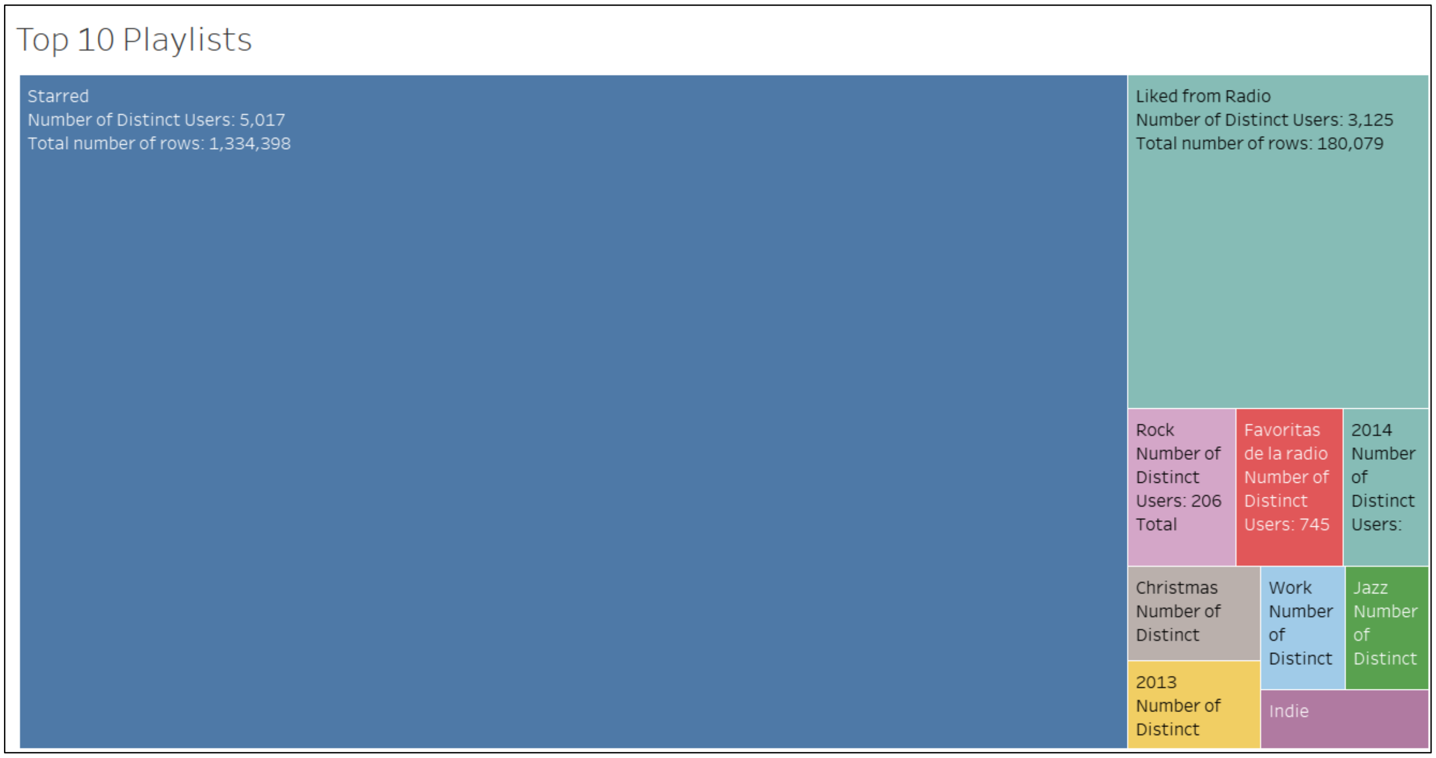
```
k_df = k_df.sort_values(["Silhouette"], ascending=False)
print(f"Best K = {k_df.iloc[0,0]} with Silhouette = {k_df.iloc[0,1]}")
```

```
Best K = 14.0 with Silhouette = 0.9984
```

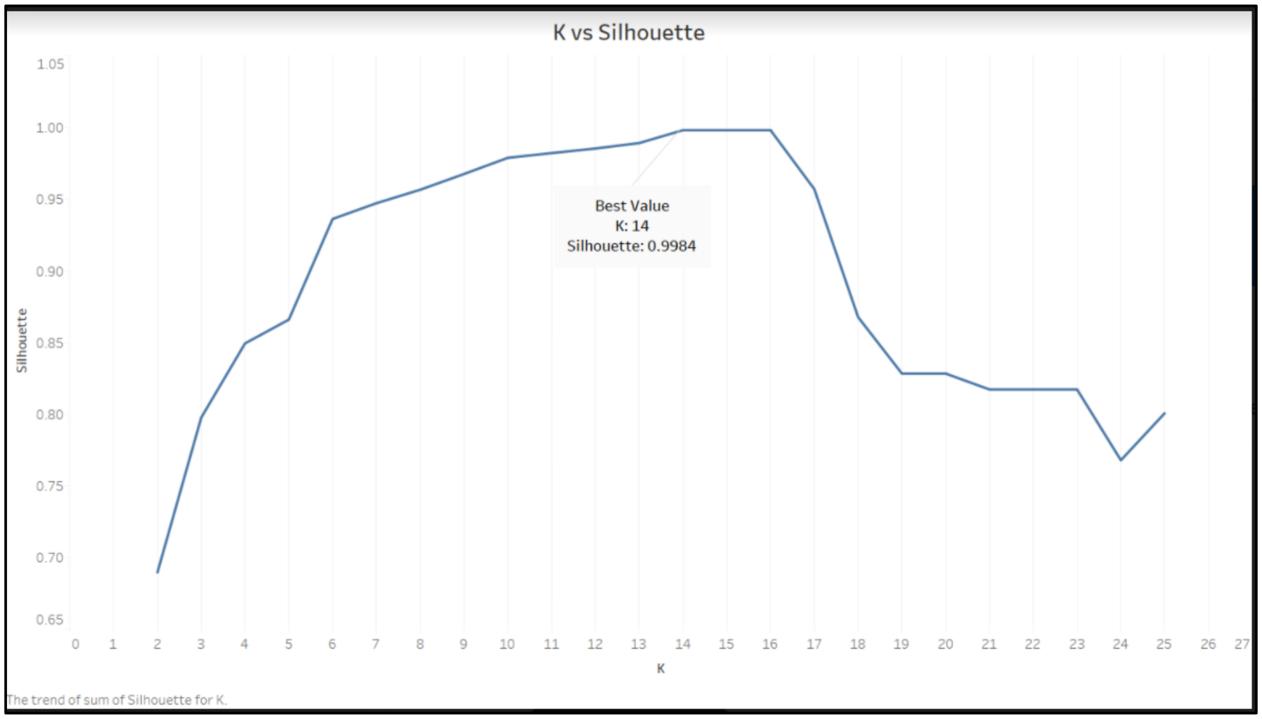
6. **Recommendation system:** A function which determines similarity between users in the same cluster using Jaccard index (Number of common tracks between users / Total number of unique tracks between users). Recommends 3 songs which are not present in the user's current playlist from the playlist of other users in the order of highest similarity.

Visualizations

1. **Top 10 Playlists**

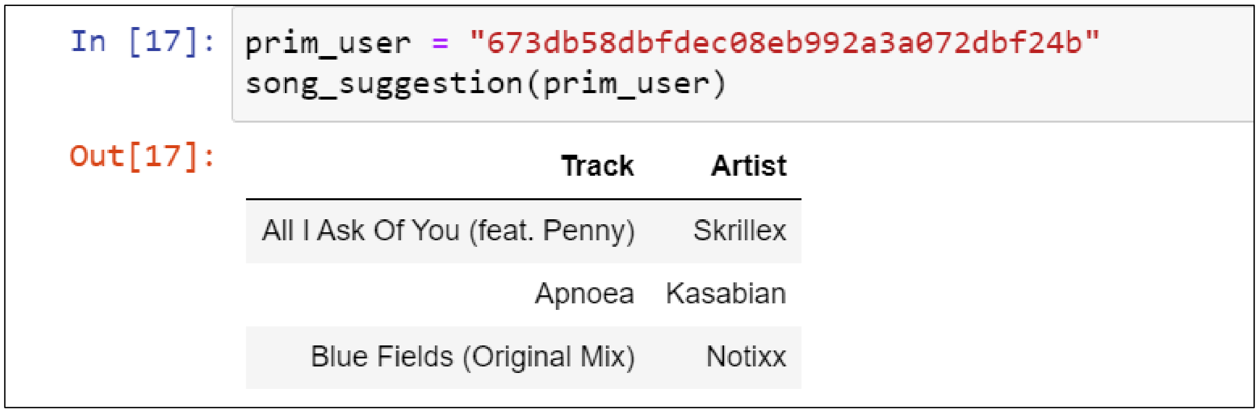


2. Best Silhouette Measure



Results

- **Expected Result**: Suggest songs for a user based on tracks of a user's playlist of his interest.
- **Solution**: A new method that extracts contextual information from playlist names and integrates into the recommendation process for improving music recommendations. Creating contextual clusters and incorporating them into the recommendation approach is a valuable and promising way for building better music recommender systems.



```
In [17]: prim_user = "673db58dbfdec08eb992a3a072dbf24b"
         song_suggestion(prim_user)

Out[17]:
```

Track	Artist
All I Ask Of You (feat. Penny)	Skrillex
Apnoea	Kasabian
Blue Fields (Original Mix)	Notixx

Role of team members

- Prasanna Vengatesh Venkataraman – Efficiently build the clustering model with all the necessary functions. Helped in preparing the presentation and delivered it to the class effectively
- Shwetha Raj – Collected the dataset, analyzed the possibility of implementation. Cleaned the data for building the model and functions. Consolidated the report together for submission

References

Pandey, Parul. (2019, 20th February). *Word Clouds in Tableau: Quick & Easy.* Retrieved from https://towardsdatascience.com/word-clouds-in-tableau-quick-easy-e71519cf507a

Mall, Rishabh. (2019, 7th January). *Recommender System.* Retrieved from https://towardsdatascience.com/recommender-system

Sadmine, Asif. Spotify Playlists Dataset. Retrieved from https://www.kaggle.com/asifsadmine/spotify-playlists-dataset