Music Recommendation System Using Spotify Data

Course:

BIA 678 Big Data Technologies

Under the guidance of Distinguished professor:

David Balenger

Team No:

09

Team Members:

Rishab Kasat

Bhuvana Vellanki

Rashmi Swaroop

Sandeep Kumar Moparthy

Contents

Introduction

68 percent of Americans report listening to music every day, and music streaming services like Spotify are the primary means people listen to music. As such, there is an enormous and incredibly consistent market for music. Considering how low investment listening to a song is in the era of streaming, people are very willing to try, often looking for, new music. Recommendation systems are critical in today's information and product rich world, but they are especially critical to music. With enough music to last a lifetime, a good recommendation system can identify music that a listener is most likely to enjoy. Improving recommendation systems when it comes to music has the potential to improve satisfaction of an enormous number of customers.

Problem statement

Spotify is an audio streaming device that is all about providing audio content to users and keeping users active on the app by recommending the next possible music or podcast. Since Spotify has over 286 million active users and about 44% of them use it daily, there is a large amount of data that Spotify analyzes to provide user recommendation. Spotify does a fairly decent job in recommending users' music but we believe we can use their dataset and some Machine Learning algorithms to improve the recommendations.

Objective:

1. Develop a personalized song list for users based on their past listening history.

Constraints:

1. Some form of interpretability.

For a given user we need to predict the next song he would like to listen to. The given problem is a Recommendation problem.

Data overview:

Source:https://github.com/ramyananth/Music-Recommender-System-using-ALS-Algorithm-with-Apache-Spark-and-Python/tree/master/data_raw

The artist_data_small and user_artist_data_small files are used for the analysis. The two files consist of the following fields.

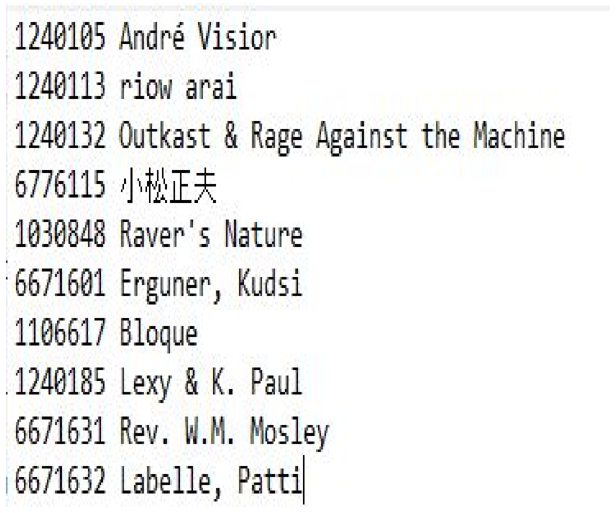
UserId

ArtistId :There are 586 unique artists

PlayCount: The frequency(number of times) of the songs played

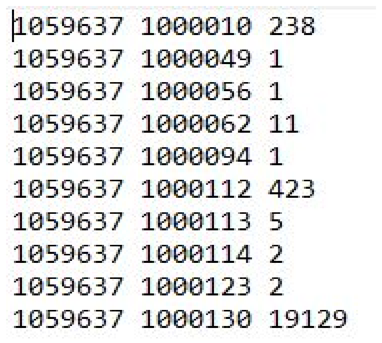
Artist_Name

A snapshot of the raw data in the artist_data_small.txt is as shown below:



```
1240105 André Visior
1240113 riow arai
1240132 Outkast & Rage Against the Machine
6776115 小松正夫
1030848 Raver's Nature
6671601 Erguner, Kudsi
1106617 Bloque
1240185 Lexy & K. Paul
6671631 Rev. W.M. Mosley
6671632 Labelle, Patti
```

A snapshot of the raw data of user_artist_data_small is as shown below:

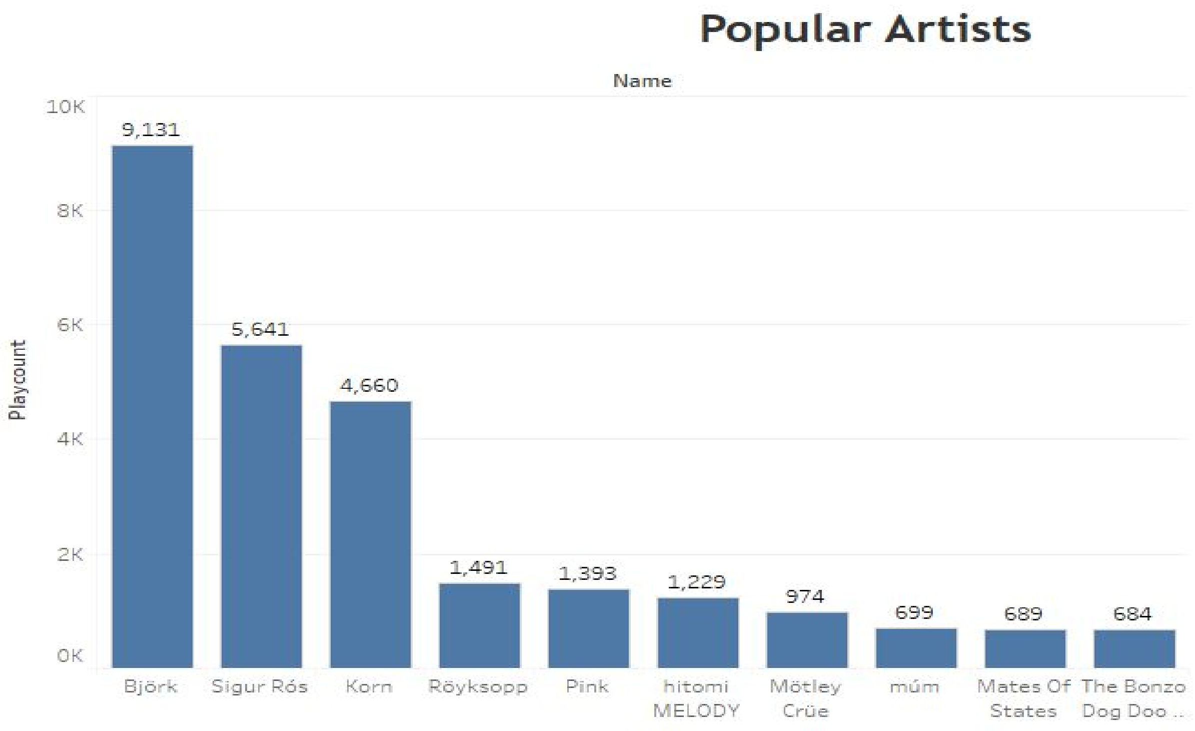


```
1059637 1000010 238
1059637 1000049 1
1059637 1000056 1
1059637 1000062 11
1059637 1000094 1
1059637 1000112 423
1059637 1000113 5
1059637 1000114 2
1059637 1000123 2
1059637 1000130 19129
```

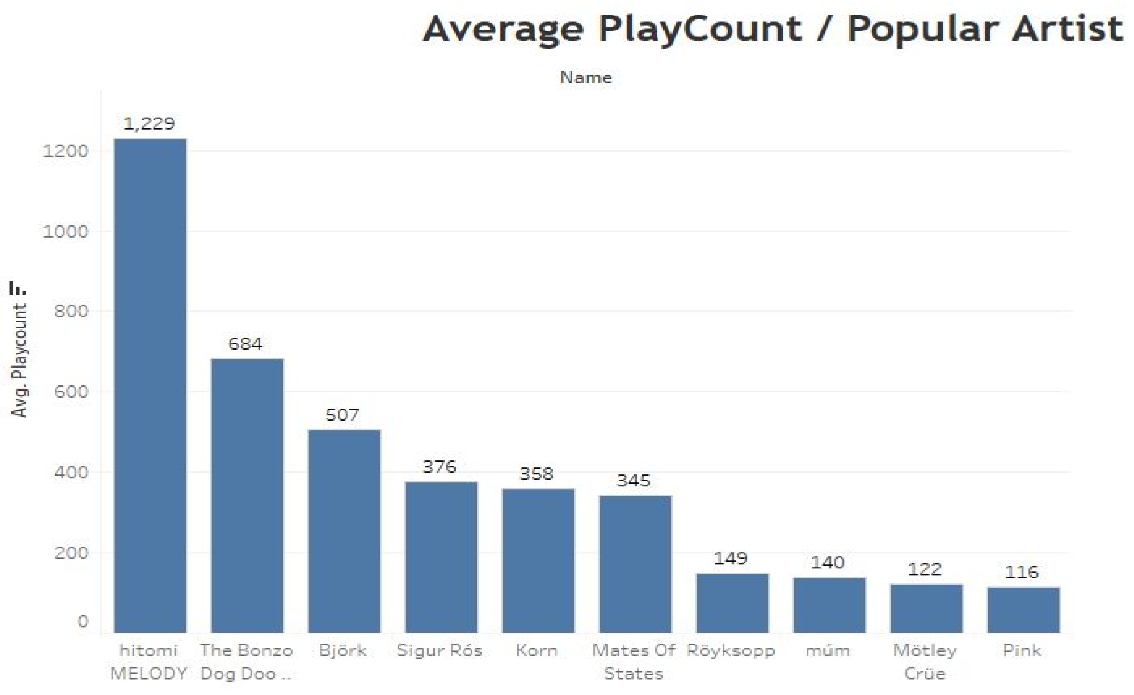
Exploratory Data analysis:

The two files artist_data_small.txt and user_artist_data_small.txt were merged for the analysis.

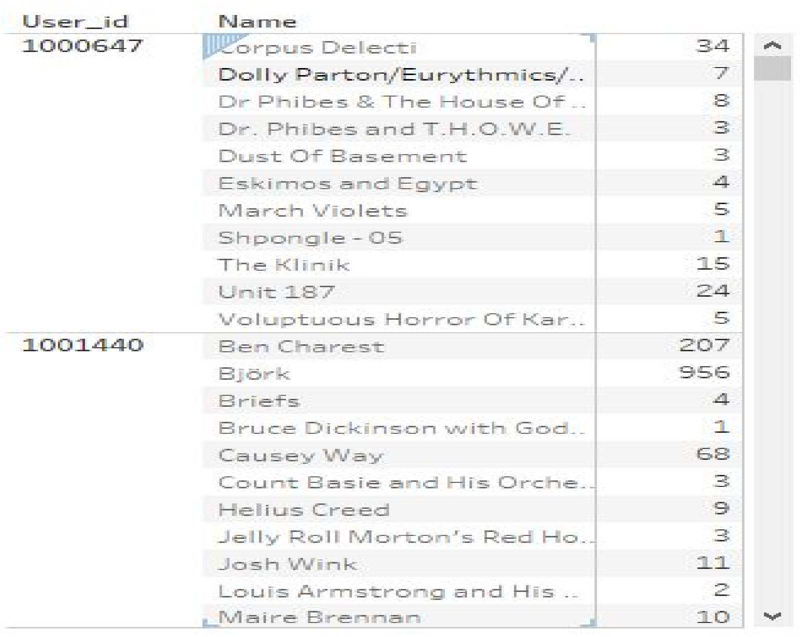
The figure below shows the 10 most popular artists that users listen to.



Below is the average playcount time for the top ten popular artists

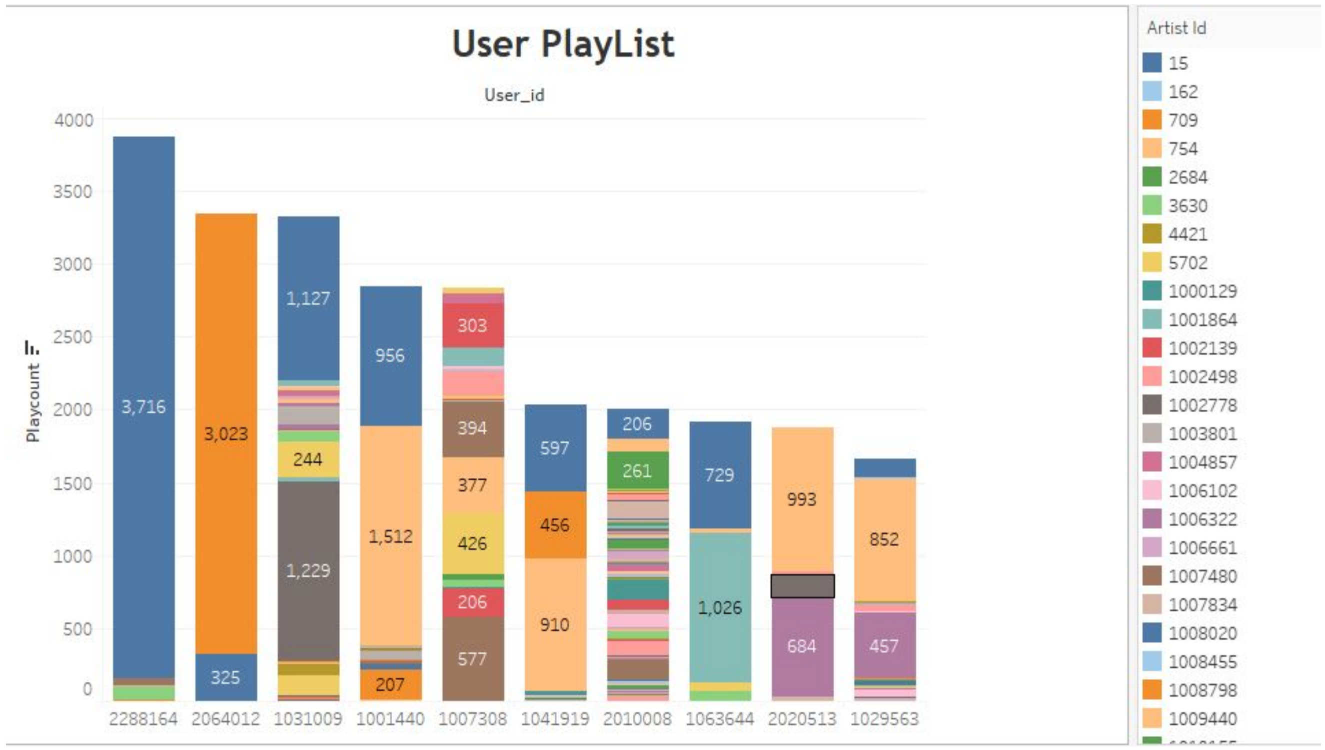


Below is the playlist for each user and the frequency of how many times the artist is listened to

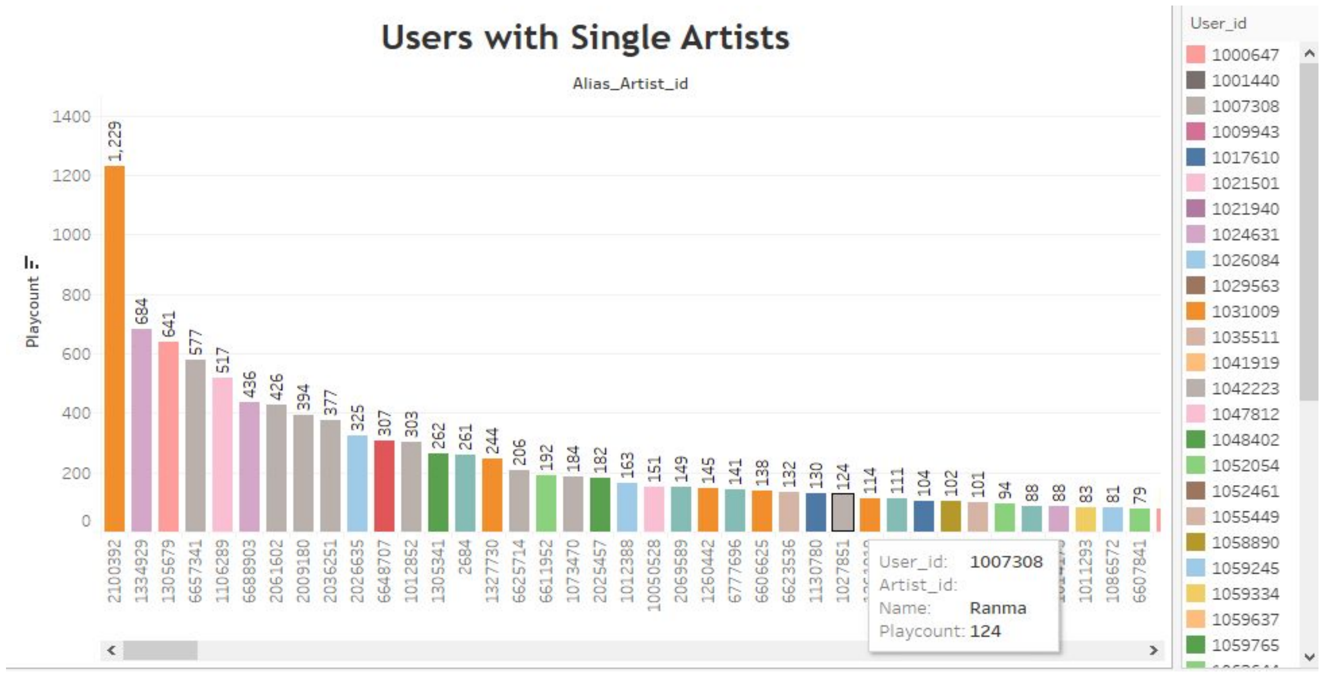


User_id	Name	
1000647	Corpus Delecti	34
	Dolly Parton/Eurythmics/..	7
	Dr Phibes & The House Of ..	8
	Dr. Phibes and T.H.O.W.E.	3
	Dust Of Basement	3
	Eskimos and Egypt	4
	March Violets	5
	Shpongle - 05	1
	The Klinik	15
	Unit 187	24
	Voluptuous Horror Of Kar..	5
1001440	Ben Charest	207
	Björk	956
	Briefs	4
	Bruce Dickinson with God..	1
	Causey Way	68
	Count Basie and His Orche..	3
	Helius Creed	9
	Jelly Roll Morton's Red Ho..	3
	Josh Wink	11
	Louis Armstrong and His ..	2
	Maire Brennan	10

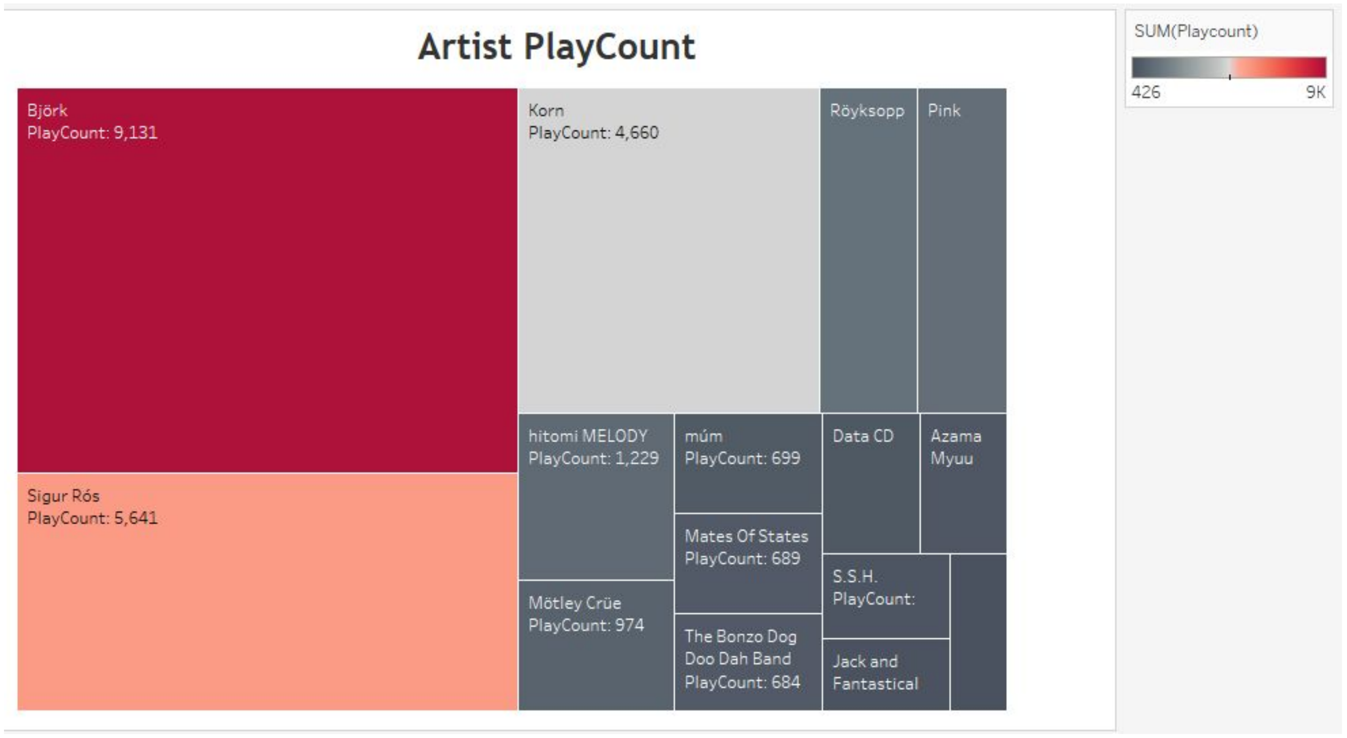
Below is the playlist for each user. This allows us to analyze the listening habits of each user.



Below shows the users that listen to a single artist



The treemap below shows the playcount for the artists:



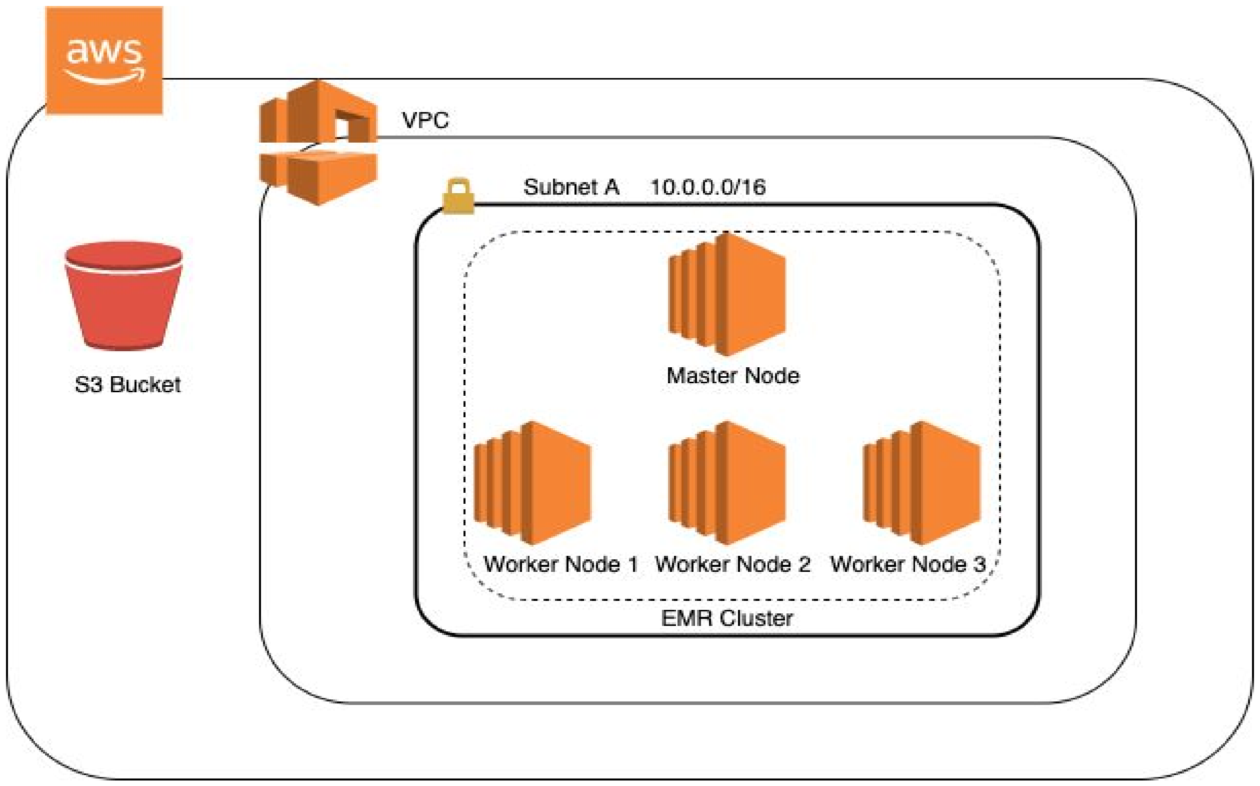
Literature review

- **Auralist: Introducing Serendipity into Music Recommendation** talks about Recommendation systems exist to assist users with finding content in a huge group of things. An ideal Recommendation system should copy the activities of a confided in friend or expert, delivering a customized assortment of proposals that balance between the ideal accuracy, diversity, novelty and serendipity. The Auralist Recommendation system, a framework that endeavors to adjust and improve every one of the four factors all the while. Utilizing an assortment of novel calculations inspired by standards of serendipitous discovery. An exhibit of a strategy for effectively infusing good serendipity, novelty and diversity into suggestions while restricting the effect on accuracy is made. Assessment of Auralist quantitatively over an expansive set of measurements and, with a user concentrate on music suggestion, show that Auralist's accentuation on serendipity in reality improves user fulfillment. Further two novels presented serendipity-enhancing techniques that can emerge with existing ranking methods through “hybridization”. Both Community-Aware Auralist and the Bubble-Aware Auralist prove to effectively boost novelty, diversity and serendipity scores, with the latter offering a better trade-off with regards to accuracy. Although only two serendipity-enhancing methods are investigated, additional techniques can easily be introduced to achieve other performance goals. Of particular interest then would be a framework that allows explicit user feedback to shape the algorithm interpolation for individual users, allowing the system to adapt to the adventurousness and mood of different personalities. [1]

- Context-based music recommendation is one of rapidly emerging applications in the era. And this paper follows a novel context aware music recommendation system which detects human emotion and predictions along with multidisciplinary efforts including low level feature extraction and music classification. So firstly, established ESTM (emotion state transition model) represents human’s emotion state in a mathematical framework. And emotion becomes measurable and analyzable. Secondly, proposed COMUS (context-based music recommendation) for evaluating the user’s desired emotion based on the user’s preferences. Based on current and desired emotions, the system selects appropriate music from the database. Thirdly, to achieve aforementioned music recommendation they proposed a novel music classification based on low level features and extracted various acoustic features like AE, SC, SF, SS, SFX, and MFCC. To avoid the curse of dimensionality problem, NMF was used for dimensionality reduction with low information loss and collected various scenarios which describe the user's situation and preferences. Lastly data was collected for user emotion state through web-based questions. With the support of COMUS ontology and ESTM ground-truth evaluated emotion state transitions from the user's context information. [2]

- A Comparative Study of Collaborative Filtering Algorithms: Collaborative filtering is increasingly gaining interests with the researchers. Its two most important methods are memory-based filtering and user-based collaborative filtering. Some new methods devised revolve around matrix-factorization which include SVD and non-negative matrix factorization. It's being widely debated which method is the best, as many methods' performances are problem parameter oriented. Based on the comparative study done by the authors of this paper methods involving matrix-factorization happen to have been setting benchmarks in prediction accuracy. The prediction accuracy is dependent on the number of users, items and density, and the nature and degree of these dependencies vary from one algorithm to another. Choosing the right recommendation system asks for the understanding of the complex relationship between prediction accuracy, its variance, computation time and memory consumption. Rating Matrix approach and user and item information approach tends to improve prediction accuracy over content-based and collaborative filtering systems. In general, SVD outperforms most other algorithms. The two subgroups of CF systems are memory-based (memorize the rating matrix of the queried user and issue relevant recommendations) and modal-based (fits a parametrized model to a given rating matrix). Modal-based methods have cluster-based filtering, Bayesian classifiers and regression-based methods. The Non-negative matrix factorization method constrains the low rank matrices to form factorization to have non-negative entries. Evaluation measures for predicting accuracy are Mean Absolute Errors and Root Mean Square Error. However, in practical cases asymmetric loss functions provision as a better loss function. One such formula is of Half Life Utility which assumes an exponential decay in the list positioning. [3]

AWS Architecture



Cluster Configuration

● 4 Node EMR cluster with a replication factor of 3

● 1 Master Node and 3 Worker Node

● Master Node - m5.xlarge instance type

o 4 cores and 16 GB RAM

● Worker Node - m5.2xlarge instance type

o 8 cores and 32 GB RAM

● Root Volume - 100 GB

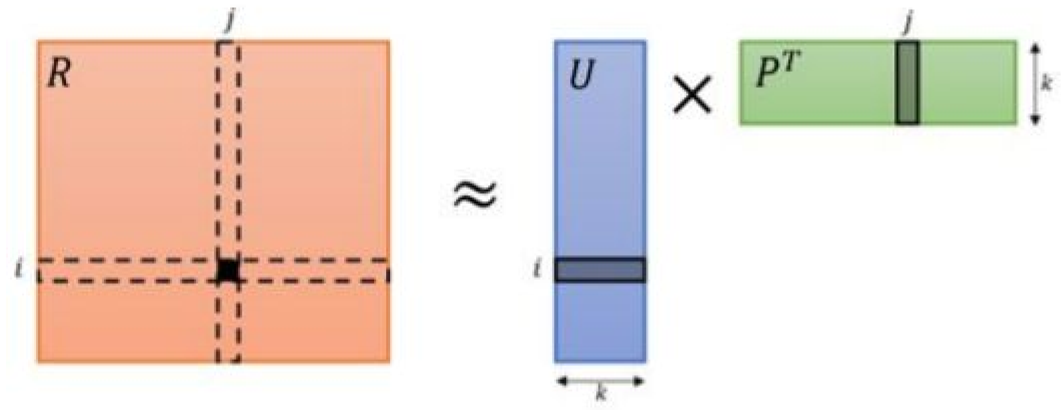
- Data Volume - 250 GB
- Build Steps - Increase the livy server timeout
- Pre-Installed Services - Spark, Zookeeper, Tez, Gangalia
- Password less authentication between nodes
- Key based authentication for user-based access

ALS Model

Recommender systems can be loosely broken down into three categories: content based systems, collaborative filtering systems, and hybrid systems (which use a combination of the other two).

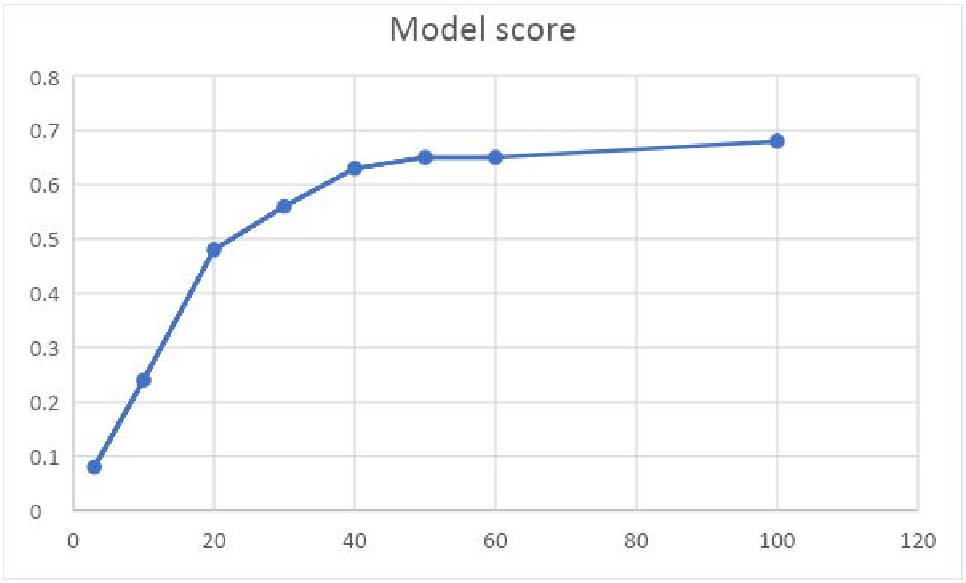
One of the algorithms which uses collaborative filtering is Alternating Least squares algorithm. Alternating Least squares is an algorithm which is a Matrix factorization algorithm and is implemented in Apache Spark. It is used for large-scale collaborative filtering problems. ALS will take this matrix and will factorize it in two smaller matrices U and P such that if multiplied together they will produce an approximation of the original matrix R.

The idea is basically to take a large (or potentially huge) matrix and factor it into some smaller representation of the original matrix through alternating least squares. We end up with two or more lower dimensional matrices whose product equals the original one.



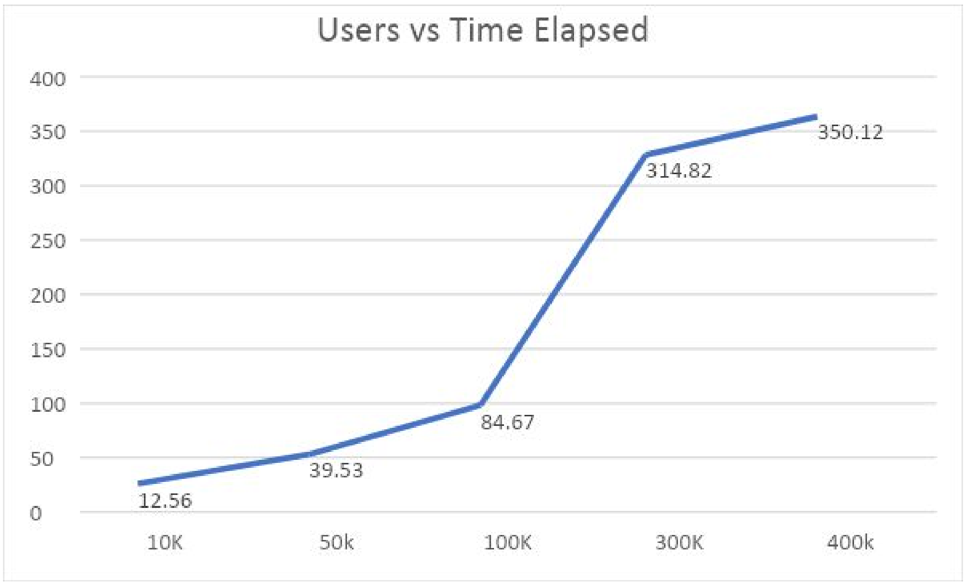
Result Scaling

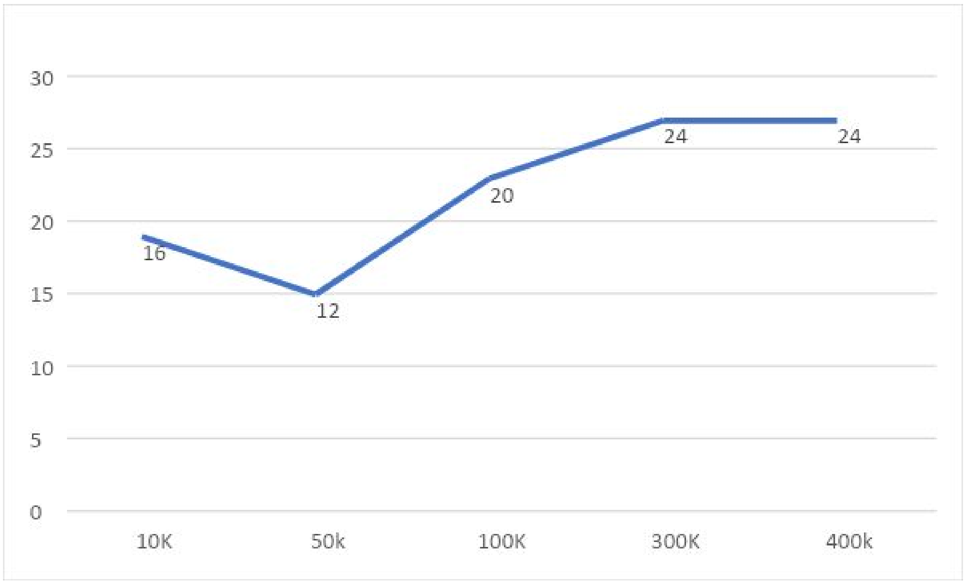
The ALS model was used to predict the ratings and calculate the model score. The aim was to evaluate how data size affects the model score for test data. We tested our model on four different data sets where we scaled the number of users in every dataset. The four sets had 10k, 20k, 100k and 300k users respectively.We ran the algorithm for the datasets to calculate model score for different levels on points to be predicted. As per the result we found that the accuracy increased as the number of users increased and as we predicted more next songs.



Spark Cluster Performance

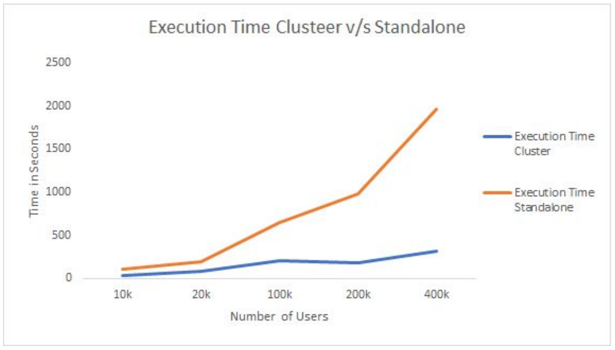
The graph below shows the performance of Spark cluster with respect to scale of the input data. We can observe that the graph is non-linear. The cluster automatically allocates and uses the available cores from the slave nodes to optimally perform based on the incoming data volume. This results in execution time reduction as the volume of data increases





Comparing Spark Cluster vs Standalone Spark

Comparing the cluster's performance against a standalone spark running on a single machine of 8 core and 16Gigs. As the data size increases the execution time in the stand alone and cluster changes significantly. The execution time on a standalone cluster seems to increase exponentially as the volume of data increases while the execution time on a cluster is fairly consistent.



Recommendations

The recommendations generated are for the top 5 music titles which were filtered. Given that the cluster utilized the resources available, the whole process of recommendation did not take more than 5 minutes. The summary of the model recommendation are shown below:

Artist 0: Something Corporate

Artist 1: My Chemical Romance

Artist 2: Counting Crows

Artist 3: U2

Artist 4: Green Day

Conclusion:

- ALS algorithm gives far better performance on Cluster than stand- alone system
- Managing Resource and scalability is easier with cloud computing
- There is a non- linear relation between data size, model performance and execution time. As the data size increases the model performance improves but caps ast 65% .

References

[1] - http://researchswinger.org/publications/zhang12auralist.pdf

[2]https://www.researchgate.net/profile/Seungmin_Rho/publication/229031003_Music_emotion_classification_and_context-based_music_recommendation/links/0fcfd50d1bd3359267000000/Music-emotion-classification-and-context-based-music-recommendation.pdf

[3] - https://arxiv.org/pdf/1205.3193.pdf

[4]https://github.com/ramyananth/Music-Recommender-System-using-ALS-Algorithm-with-Apache-Spark-and-Python/tree/master/data_raw