Project Report on big data visualization

Submitted By: Deepak Gautam and Ayush Raj Aryal

Introduction

With the transfer of music into online media, lots of user data about their music preferences has been collected. The recommendation of artist and songs according to the user's listening habit is the most common feature of the online music systems these days. Most of these system are based on categorization of the music the users have listened in past, and serve with the artists or songs to the users that belong the similar category to those the user likes. Though the personalized music recommendation began with Pandora in 2000, many other services have been launched since then. One such service, Last.fm, provides the anonymized data about the users and the number of plays of artists they like.

In this project, we explored the two aspects of the data: categorization of the similar artists based on their listening communities and the prediction of age group based on what artist the users like. In first approach, We have made a graph structure by taking the n most popular artists as nodes and the total number plays by common users between artists in question as edge weight. The categorization is accomplished by Louvain Modularity algorithm for detecting community from a graph. The recommendation is based on this category.

The second approach is about learning the user's listening habits to predict the age groups. We have build a classifier model out of certain split of the available data and trained with the user's preferred artists, the model is used for predicting age group of the users. The whole point of this prediction of age group is that we wanted to see if the artists they listen says something about the age group.

Dataset Preparation

Last.fm provides an API for the anonymized user datasets for non-commercial use. We collected the datasets of 360K users pulled using the API call *user.getTopArtists()*. The dataset consisted the number of plays of the artists and the personal information about the user such as country, age, age of the sign up. Due to the limitation of the resource for running the algorithms, we had to truncate the datasets to just 10,000 users. The total number of rows of artist plays of those 10,000 users is about 500,000 implying about 50 unique artist plays on average per user. We have selected top 50 artists from those 500,000 plays to build the graph structure and subsequently categorize them based on their listening communities. We have used the same 10,000 users dataset for prediction algorithms as well.

Apart from selecting this subset, the preprocessing included removing rows having their age, gender or or date of signup as NaN. As most of the classification algorithms require the rows to have numeric values, we had to convert date strings into some number of months to represent the date. The gender field was translated into two numbers, -1 and 1, to represent male and female. The artist names were present as strings in the dataset, so we had to apply vectorization technique so that the classification algorithms accept the input dataset.
For machine learning algorithms, little more effort was needed because of the textual nature of the datasets. For prediction via machine learning, new matrix is formed in such a way that each row is represented as a unique user. We have concatenated the names of different artists a user listens, and aggregated the plays. The ‘artist_name’ and ‘country’ field on each row are text values and need proper vectorization techniques because the most of the classification algorithms take numbers as input to the algorithm. We removed english stop-words from the artist names and calculated TF-IDF values of each of the tokens. A list of artists that any arbitrary user listens is considered as document and the whole set of artists as the corpus for the calculation of TF-IDF values for the artists. Similarly, A country name is considered as document and all the countries in the world is considered as corpus for TF-IDF of the field ‘country’. The age field is labeled into classes such as 0-10 as children class, 10-20 teen class and so on for classification purpose.

Graph Analysis

After preprocessing step on original datasets, we selected top 30 most listened artists by those selected 10,000 users. This is again due to lack of resources to process the rows to build graph. The graph is constructed by considering each artist as a node of a graph. The node weight is the normalized value of the total number of plays that artist has. We also tried normalization on artist plays per user so that the number of plays won’t affect the grouping of the artists, but that didn’t work well with the data we had. Similarly, we connected those artists each other by initializing edges between them. The edge weight is calculated by counting total number of plays by the common users between these two artists. We also tried variation of edge weight calculation such as the ratio of the plays between common users, difference of the plays etc. The best result, by looking at the artist names in each community, were seen with the first approach. We applied threshold of 3 artist to form a group, otherwise we consider them as ‘uncategorized’ group of artists.

Thus constructed graph has nodes equal to 30, the number of top artists that we’ve chosen, and 435 edges. Though this graph seems smaller, it took considerable time to build this because of

the repetitive joins and filtering of the users from bigger matrix. We passed this graph to partition it by Louvain Modularity algorithm. In our case, the graph was partitioned into three sub-graphs that we call community of the related artists. The artists that were categorized via this method seemed to have some similarity in their genre or in the time they were popular around the world.
The three groups are listed below:

1. rammstein,red hot chili peppers,linkin park,nine inch nails,rise against,iron maiden,nightwish,tool,system of a down,nirvana,metallica,in flames
2. bob dylan, queen, the beatles, u2, pink floyd, david bowie, led zeppelin
3. depeche mode, death cab for cutie,radiohead,coldplay,muse,the cure,the killers,arctic monkeys,placebo

It is clearly seen from the above groups that the artists within each of these groups seem to be related in different aspects. For example, the first group has the bands that mostly follow hard rock or metal genre of the music, whereas the second group is completely different in terms of when the bands were popular. The most of the bands/artists, if not all, of group 2 have been popular in seventies and eighties. Similarly, the third group seems to have the artists who play soft acoustic/rock music.

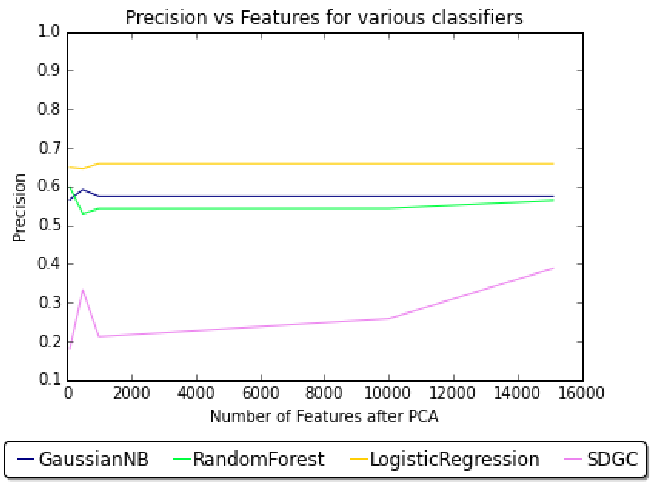
Thus, the users can be recommended with the artists that are in same group as the user's favorite artists. if a user listens 'pink floyd' a lot, then the user can be recommended to listen 'led zeppelin' or 'queen'. This would basically complete a simple recommendation system.

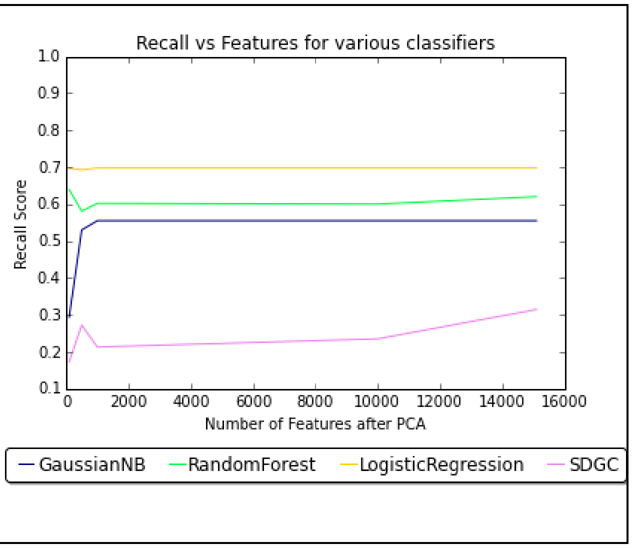
Age Prediction

The initial idea was to cluster the group related artists via user's listening habits but later on we came to explore the possible relationship between age group and their favorite artists.

We ran various classification algorithms such as NaiveBayes Classifier, RandomForest Classifier, Logistic Regression and SGDClassifier with the user's data. We have used TF-IDF values of user's most played artists and their country along with their gender and time of signup
As feature vectors. We have age group of the user as a label for these classifiers because we wanted to see if there's any relation between the artist users play and their age group.

As expected, different classification algorithm performed differently to predict the age group of the dataset under test. We have used K-Fold evaluation technique to evaluate the accuracy of the classifiers with 10 folds. The accuracy figures for each classifier are as follows:





Tools and Libraries Used

We analyzed the dataset in python program with the help of various supporting libraries. The supporting libraries that we used are listed below:

1. Scikit-Learn : For Machine Learning and Vectorization
2. Networkx : For graphs structure implementation
3. Community : Community Detection API that works with networkx.
4. Pandas : Python data wrangling library
5. Numpy : Python fundamental package for scientific computing
6. Vincent : Python Plotting Library
7. Gephi : Graph Visualization Program
8. Jupyter Notebook : Interactive Python Programming server and user interface.

Conclusion

From the graph analysis we can conclude that the artists can be grouped based on the similarity in their user base. This is the root of our recommendation system. We also can conclude that different age groups have different habit of listening artists. This is concluded from the fact that given the artist, the number of plays and the other user attributes, we predicted their age group almost 70% of the time correctly with about 65% precision.