**Group 336: Applying Data Mining Techniques on the Spotify dataset**

|  |  |  |
| --- | --- | --- |
| **First Name** | **Last Name** | **Email-ID** |
| Twinkle | Joshi | tjoshi3@hawk.iit.edu |
| Nikhitha | Kamath | nkamath1@hawk.iit.edu |
| Diksha | Biradar | dbiradar@hawk.iit.edu |

Table of Contents

[**1. Introduction** 2](#_Toc56029461)

[**2. Data** 2](#_Toc56029462)

[**3. Problems and Solutions** 3](#_Toc56029463)

[**4. KDD** 3](#_Toc56029464)

[4.1. Data Processing 3](#_Toc56029465)

[4.2. Data Mining Methods and Processes 5](#_Toc56029466)

[**5. Evaluations and Results** 6](#_Toc56029467)

[5.1. Evaluation Methods 6](#_Toc56029468)

[5.2. Results and Findings 9](#_Toc56029469)

[**6. Conclusions and Future Work** 10](#_Toc56029470)

[6.1. Conclusions 10](#_Toc56029471)

[6.2. Limitations 10](#_Toc56029472)

[6.3. Potential Improvements or Future Work 10](#_Toc56029473)

6.4 References……………………………………………………………………………………………………………………………………11

# **1. Introduction**

The aim of this project is to discover various trends in the dataset and to understand the musical behavior of several generations. Music is always energizing, and we would love to dig through Spotify's music data to learn not just about the genre's trends, but also predict the popularity of the songs based on their genre. We thought this would be a fun project to work on, because music is such an important part of several generation and see how music has changed over time in terms of audio features such as loudness, danceability, energy, liveness, and other musical characteristics.

# **2. Data**

We present a dataset to aid audio analysis of musical performances. We obtained the data set from Kaggle <https://www.kaggle.com/yamaerenay/spotify-dataset-19212020-160k-tracks>

For this project, we used the aforementioned dataset, which contains over 174k records with attributes such as ‘acousticness’, ‘artists’, ‘danceability’, ‘duration\_ms’, ‘energy’, ‘explicit’, ‘id’, ‘instrumentalness’, ‘key’, ‘liveness’, ‘loudness, ‘mode’, ‘name’, ‘popularity’, ‘release’, ‘date’, ‘speechiness’, ‘tempo’, ‘valence’, ‘year’.

The structure of the dataset is as follows:

**Numerical**:

* acousticness (Ranges from 0 to 1)
* danceability (Ranges from 0 to 1)
* energy (Ranges from 0 to 1)
* duration\_ms (Integer typically ranging from 200k to 300k)
* instrumentalness (Ranges from 0 to 1)
* valence (Ranges from 0 to 1)
* popularity (Ranges from 0 to 100)
* tempo (Float typically ranging from 50 to 150)
* liveness (Ranges from 0 to 1)
* loudness (Float typically ranging from -60 to 0)
* speechiness (Ranges from 0 to 1)

**Dummy**:

* mode (0 = Minor, 1 = Major)
* explicit (0 = No explicit content, 1 = Explicit content)

**Categorical**:

* artists (List of artists mentioned)
* artists (IDs of artists)
* release\_date (Date of release mostly in yyyy-mm-dd format)
* name (Name of the song)

# **3. Problems and Solutions**

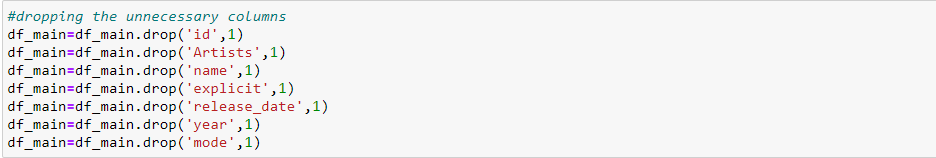
The problems that we would be solving in this project are, firstly we will be analyzing different musical feature and find how music has evolved in the last 10 decades and also to get insights about the different genres of music. We have used the spotify data from year 1922 to 2021. We would be using clustering to group our data based on the musical feature and at last we aim to predict the popularity of the songs using three classification models. To finally find the accuracy of models, we will be using accuracy and confusion matrix as our evaluation metrics.

# **4. KDD**

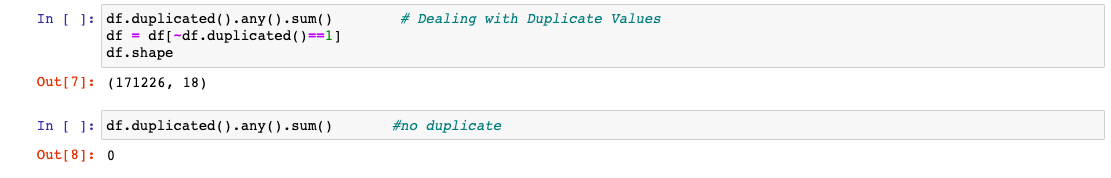
## 4.1. Data Processing

Data processing and Data cleaning are most important steps when applying data mining techniques. It helps us remove the redundant values from our data set, normalizing the data to run different algorithms and it makes our dataset look clean and organized.

1. To drop the unnecessary columns



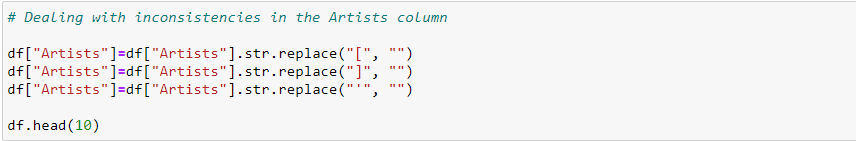
2. Dealing with duplicate values



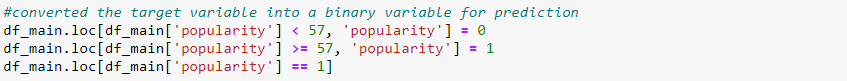
3. To check if we have any missing values



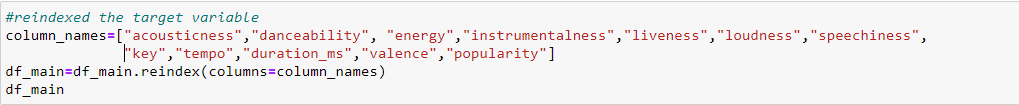
4. Dealing with inconsistencies in the Artist column



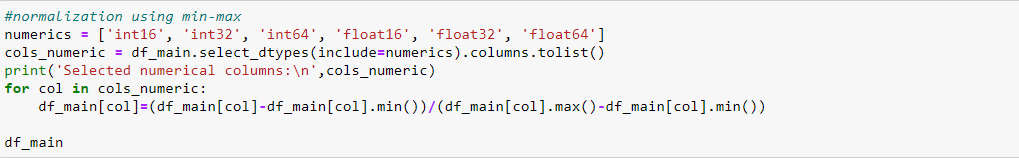
5. Converted target variable into binary variable



6. Re-indexed the target variable



7. Normalization of the data



## 4.2. Data Mining Methods and Processes

We have used four data mining methods in this project and they are:

1. K-Nearest Neighbor
2. Logistic Regression
3. Neural Network (Revised- used Neural Network instead Decision Tree)
4. K-Means Clustering

**K-Nearest Neighbor**

KNN algorithm is supervised machine learning algorithm. This algorithm takes numerical data since it is a distance based learning algorithm, i.e. it calculates the distances between the data points and assigns the value based on its closest data point value. We select the K value and matches how many data point is near to the label we’re finding out. The distances we use can be Euclidean or Manhattan distance to calculate the distance between data points. We have used KNN since we had numerical data to deal with and it is easy to use and implement on such type of data sets.

**Logistic Regression**

Logistic regression is another classification algorithm that we have used in this project. In this project it basically predicts the popularity of songs. We have used this algorithm since we have binary dependent variable in our data set i.e. popularity. It is useful in predicting the probability of dependent variable.

**Neural Network**

Neural network is a quite popular algorithm nowadays. It basically works like how human brain works. They are very effective to use on large data set and provides high accuracy than other classification models. It consists of three main layers, input layer, hidden layer (one or more hidden layer) and output layer.

**Process for Classification**

1. Load the data set
2. Data Preprocessing and Normalizing
3. Splitting the data set into training and testing sets
4. Fit the classification models
5. And at last compute the accuracies

**K Means Clustering**

K-Means is a Machine Learning algorithm that is unsupervised. Unsupervised algorithms make inferences from datasets based solely on input vectors, with no reference to known or labeled outcomes. It is an algorithm for discovering groups of categories in our dataset that would have been very time consuming to categorize considering the number of records in the dataset. It reduces the total work time and gives answers faster. The steps followed are:

1. Choose the Number of K Clusters.
2. Select random K points, the centroids.
3. Assign each data point to the closest centroid.
4. Compute and place the new centroid of each cluster.
5. Reassign each data point to the closest centroid.

# **5. Evaluations and Results**

## 5.1. Evaluation Methods

The evaluation metrics that we have used in all our three models, KNN, Logistic Regression and Neural Network is Accuracy. Accuracy is a percentage of total predictions divided by the total number of instances. We have calculated the accuracy by two methods hold-out evaluation (Splitting the dataset into training and testing) and N-fold cross validation (Randomly splitting data in K group) for the data mining task. Hold out evaluation is good to use when the data set is very large. For KNN Algorithm, we have found other evaluation metrics like Recall value and Precision Value. Also, we have used confusion matrix to better understand the accuracy of our results. (Revised- used hold out evaluation and confusion matrix)

**For KNN**

Table

Description automatically generated with medium confidence

The Confusion Matrix

Text, letter

Description automatically generated

**For Logistic Regression**



The Confusion Matrix

Text

Description automatically generated with medium confidence

**For Neural Network**





The Confusion Matrix

A picture containing text

Description automatically generated

**K Means Clustering**

The Elbow Method to find the optimal number of clusters.

This method results in the number of optimal number of clusters into which the data can be clustered. The ideal number of clusters into which the information can be clustered could be a significant step in any unsupervised calculation. One of the foremost common strategies for deciding the ideal value of k is the Elbow Strategy.

The underneath code snippet points to play down the squared Euclidean distances between the centroid and the perception of the cluster to which it has a place.

Let us consider the K-means clustering feature, Within Cluster Sum of Squares (WCSS). It is the sum of squares of distance between data points and respective centroid of cluster to which the data point belongs in K-means clustering. With the number of iterations, we anticipate WCSS to decrease.

Text

Description automatically generated

The graph of the aforementioned technique is shown below:

Chart, line chart

Description automatically generated

Applying K means clustering algorithm on the preprocessed dataset is done in the following manner. Since we know the optimal number of clusters to achieve a clustering result is 4, we pass that numerical value through the n\_clusters parameter. (Revised- number of clusters taken 4 instead of 2)

The K-means algorithm has the drawback of being vulnerable to the initialization of the centroids or mean points. As a result, if a centroid is set to be a far off point, it may end up with no points associated with it, whereas more than one cluster may be connected to a single centroid. Similarly, if more than one centroid is initialized into the same cluster, that would just result in poor clustering.

We use K-means++ to solve the flaw. This algorithm makes sure that the centroids are correctly initialized, and that the clustering efficiency is improved. The rest of the algorithm is identical to the regular K-means algorithm. K-means++ is the regular K-means algorithm with smarter centroids initialization.

Calendar

Description automatically generated

## 5.2. Results and Findings

|  |  |  |
| --- | --- | --- |
| **Model** | **Accuracy (N-hold out)** | **Accuracy(N-fold cross)** |
| Logistic Regression | 1.0 | 0.9012 |
| Neural Network | 1.0 | 0.9028 |
| KNN Algorithm | K=9, 0.8971 | K=9, 0.8969 |

Since we are using Accuracy as our performance metric to get the best results. We have used three algorithms to get the best accuracy value. And Neural Network gives us the best accuracy with 1.0 value for N-hold out evaluation.

K-Means ClusteringGraphical user interface, text, application

Description automatically generated

# **6. Conclusions and Future Work**

## 6.1. Conclusions

Through the course of this project there are numerous lessons that we learnt. We learnt that data cleaning is a very important step in the process of retrieving a dataset than we first anticipated. We also learnt that no dataset would be readily available, a lot of data preprocessing had to take place.

Throughout multiple steps in the process, there were a lot of duplicate information and unnecessary columns that needed to be filtered out or removed.

It was important to make sure the data preprocessing was done correctly to make sure the desired results were returned. We made to use only variables those were important and necessary. In this project we used classification methods using popularity as the label. We performed EDA and K Means Clustering for analyzing the data. We used to EDA show various trends in the music over the years. We were able to predict the popularity of songs using three classification model having the highest accuracy rate of 1.0 given by Neural Network.

## 6.2. Limitations

One of the limitations we faced was while applying the classification models, we were not able to apply Support Vector Machine because it is not suitable for large datasets as it takes lot of time to run the code which makes it less effective algorithm.

Also, we tried performing Hierarchical clustering but due to the large dataset and it is not so good computational cost made it unreal and not efficient with the dataset we used.

## 6.3. Potential Improvements or Future Work

Context-Aware Audio Recommendation Based on Session Progress Prediction

Session progress is very similar to the task of movie rating prediction in traditional movie recommendation system. Context-sensitive recommender systems tailor their recommendations to additional information that defines the specific situation under which recommendations are made.

Using the same conceptual thought process, we can build a recommendation system that tracks the user’s likability of songs and suggest the user similar types of songs based on the duration the user listened to the song. The server tracks quartile progress (0%, 25%, 50%, 75%, 100%) of each video session using the formula, Session progress = (Duration of the song played by the user / Total time of the song) × 100%.

**References**