MUSIC ANALYSIS FOR GENRE CLASSIFICATION & SONG RECOMMENDATION

Group 8

Aditya Uday Tornekar
Nisha Rangnani
Rishabh Mahesh Upadhye
Shreyas Raghavan Sadagopan

TABLE OF CONTENTS

MUSIC ANALYSIS FOR GENRE CLASSIFICATION & SONG RECOMMENDATION

1. ABSTRACT:

Innovative companies like Spotify and Shazam leverage music data in a very clever way to provide amazing services to their users. They use recommendation algorithms and automatic genre classification which greatly contributes to increasing user experience. From this project, we aim to perform such tasks of genre classification and music recommendation when musical features are provided. We basically aim to create a music recommender system and a playlist generator for companies like Spotify and Pandora.

Inference of musical genre, whilst seemingly innate to the human mind, remains a challenging task for the machine learning community. We used various machine learning algorithms to achieve our goal. We made use of classification algorithms such as Logistic Regression, Naive Bayes Classifier, Neural Networks and Random Forest Classifier to identify genre of the music track. We also applied K-means clustering algorithm to create song clusters and recommend a song which the user is most likely to enjoy.

2. DATA COLLECTION/CLEANING/EXPLORATION (ASSIMILATION AND UNDERSTANDING):

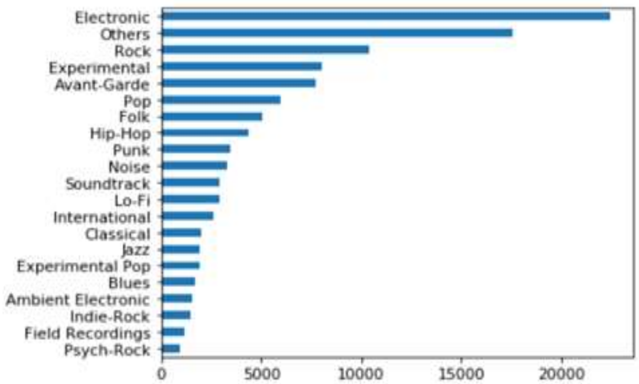
- Our dataset features 106,574 music tracks and their 518 attributes, which contains details about their discographic and technographic details. The dataset was obtained from UCI Machine Learning and can be found on FMA: A Dataset For Music Analysis Data Set.
- The dataset zip folder consists of four different CSV files, containing information such as metadata per music track (title, artist, tags, etc.), genre (genre name, parent genre), features (common features extracted), and audio features provided by Spotify for a subset of tracks.

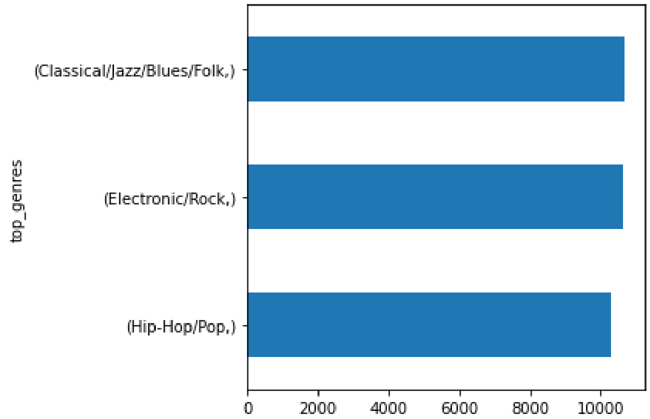
- We determined the genre of the music track when we acquired other attributes of the tracks. We also used song attributes to provide recommendations of songs a user might like. We inferred the kind of genres that are most popular, the artists that are listened to the most and the trends of tracks release and duration.

100%	track_id	100%	title	93%	number
2%	information	14%	language_code	100%	license
4%	composer	1%	publisher	1%	lyricist
98%	genres	98%	genres_all	98%	genres_top
100%	duration	100%	bit_rate	100%	interest
100%	#listens	2%	#comments	61%	#favorites
100%	date_created	6%	date_recorded	22%	tags
100%	album_id	100%	title		
94%	type	96%	#tracks		
76%	information	16%	engineer	18%	producer
97%	#listens	12%	#comments	38%	#favorites
97%	date_created	64%	date_released	18%	tags
100%	artist_id	100%	name	25%	members
38%	bio	5%	associated_labels		
43%	website	2%	wikipedia_page		
		5%	related_projects		
37%	location	23%	longitude	23%	latitude
11%	#comments	48%	#favorites	10%	tags
99%	date_created	8%	active_year_begin		
		2%	active_year_end		

Metadata with the %Rows Populated

- TARGET VARIABLE/CLASS IMBALANCE:
    - The target variable ‘Genre’ was heavily imbalanced, it had more than 20 individual levels with varying populations.
    - In order to perform classification tasks, we need to bring down the number of levels and make the distribution more evened out.
    - To achieve this, we restricted ourselves to the 8 topmost listened to genres and clubbed similar genres together into 3 different buckets. We decided similarities based on the target audience group, tonal features of the songs and the interest shown in each of the genres.
        - Bucket 1: Classical, Jazz, Folk and Blues music
        - Bucket 2: Electronic and Rock music
        - Bucket 3: Hip-Hop and Pop music





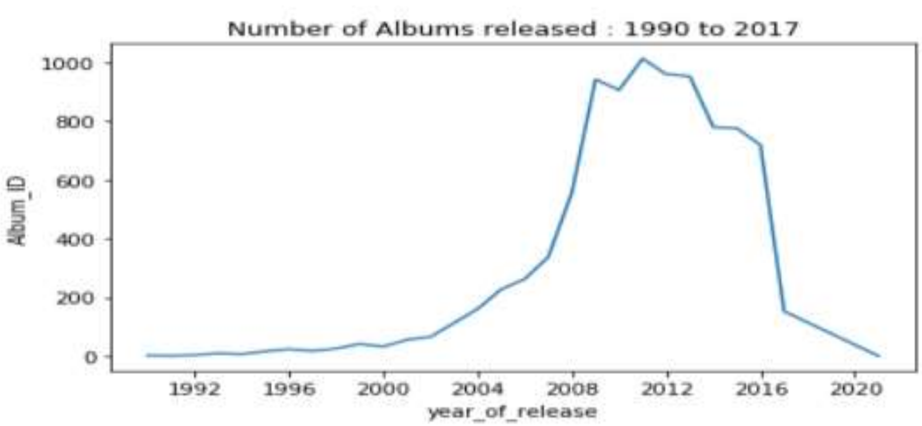
Initial Y Variable Distribution After Sampling

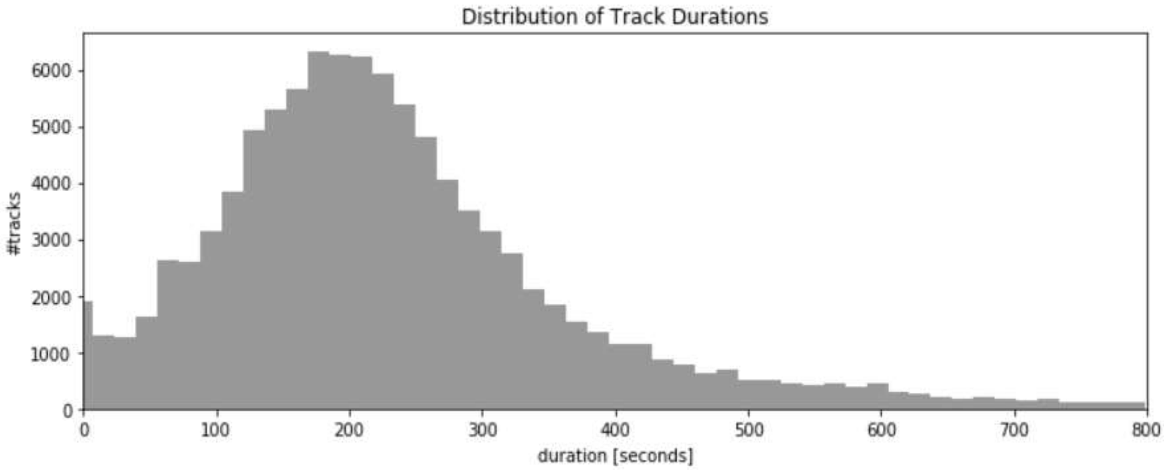
2.1 EXPLORATORY DATA ANALYSIS:

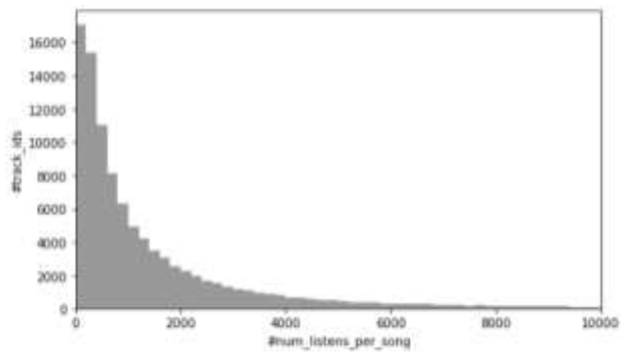
A basic data exploration helped us understand how the variables are behaving and their relationships. We performed an univariates test. A sample result is shown below.

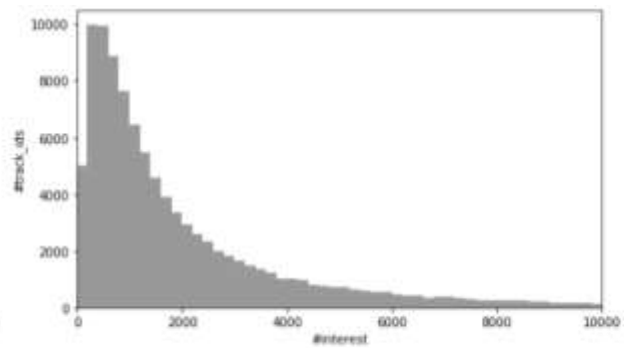
	MissingValue	Missing_perc	mean
num_tracks_in_album	5647	5.167317881	21.68023
num_listens_per_album	6346	5.80694161	27698.33
track_favorites	5981	5.472946387	67.82775
album_favourites	7427	6.796116505	33.89692
duration	6478	5.927728924	269.4263
interest	6400	5.856354602	3504.589

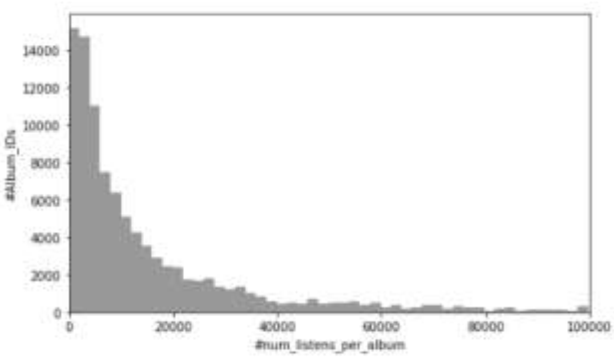
Electronic and rock are more preferred over other genres. We also explored other variables such as track duration, year of release, interest, number of listens per song, etc.

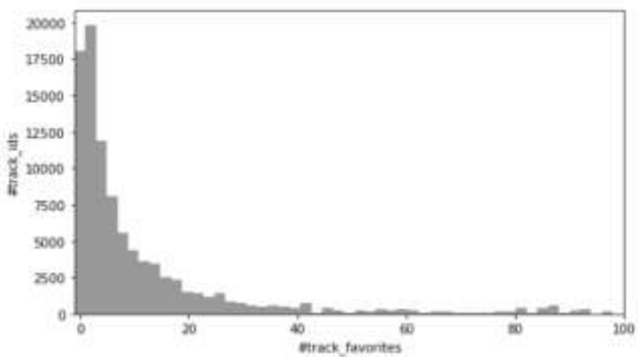


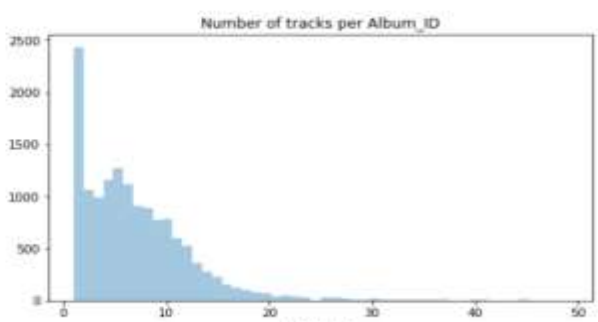

Distribution of Track Durations
#tracks
duration [seconds]

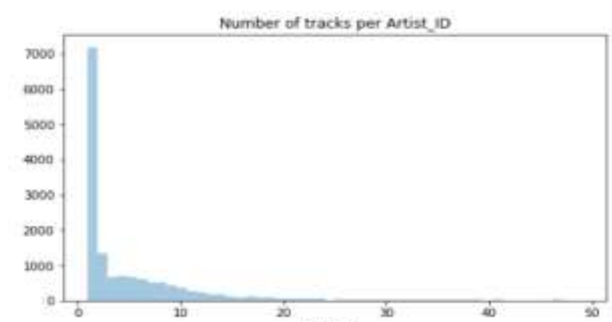

#track_ids
#num_listens_per_song


#track_ids
#interest


#Album_IDs
#num_listens_per_album


#track_ids
#track_favorites


Number of tracks per Album_ID
Album_ID


Number of tracks per Artist_ID
Artist_ID

The EDA process revealed the following:

- Most albums were released between 2011 and 2013.
- Track IDs 1 to 2000 were listened to more than 2000 times, Track IDs 1 to 2000 also have an interest score of more than 3000.
- Tracks are of about 269 seconds on average, up to a maximum of 18,350 seconds.
- We could see that, as the duration of the song increases over 200 seconds, the number of tracks keeps decreasing.

TOTAL NUMBER OF X VARIABLES: 31

2.2 DATA PREPROCESSING AND MANIPULATION:

After exploring the initial data, we dropped the irrelevant features from our business case. We assigned unique track ids to each track and removed IDs, Dates, Titles, and other Multilevel categorical features.

- **MISSING VALUES IMPUTATION:**
  - We will be removing the IDS, Date columns at a later stage before modelling.
  - We checked our dataset for missing data and obtained a total count. A univariate test was performed and achieve the following:
    - Identified the missing percentages in all variables
    - Studied the data types, ranges, and patterns of the explanatory variables
    - Skew, Kurtosis and Quartile distributions were determined
  - Missing values in ‘Duration’ and ‘Year of Release’ features which was a total of 17 were obtained from the internet
  - Imputed all the continuous NA’s by their means and categorical variables by their modes
  - Bucketed the continuous variables using their quartile distributions. 5 levels with an equal distribution per level was maintained

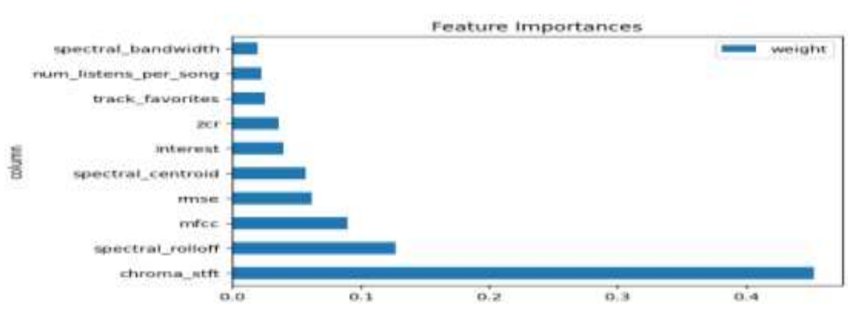
- **OUTLIER TREATMENTS:**
  - All the outliers in the continuous columns have been identified and capped using a 5% window on either side (0.05 and 0.95 percentile). This step is done specifically before handling the NAs to not miss the essence of the raw and miscalculate the interpolations.

2.3 FEATURE ENGINEERING AND SELECTION:

- **VARIABLE REDUCTION:**
  - There are a whopping 33 variables in our dataset. There is no way we would be able to use all of these variables for our predictive models.
  - Missing columns
    - Removed all the variables which have more than 50 % missing values.
  - Dropping variables with very less variance
    - Used a 10 % threshold for variance.
  - Dropping variables using correlation
    - Dropped variables which have a correlation coefficient of more than 0.7
  - Dropped a total of 11 variables

In order to identify the most important predictors, we built a Random Forest model to remove the weakest features. Features are ranked by the model's coef_ or feature_importances_ attributes, and by recursively eliminating a small number of features per loop, this model attempts to eliminate dependencies and collinearity that may exist in the model.

The dataset was split into three parts: Training, Validation and Testing set.



Feature Importance

3. METHODOLOGY

We use the above processed data in 4 machine learning models. We have a multi-class classification problem for classifying the music genres correctly using the 11 important features extracted using the Random Forest feature importance module.

Total of four classification models were used to tackle this problem. For each model, a base classification model was built and after recursive hyper parameter tuning, we built the final genre classification model. Also, we used K-Means clustering technique for song recommendations. Further, the model details are covered in depth:

4. MODELS (CLASSIFICATION)

- **LOGISTIC REGRESSION**

The logistic regression uses SoftMax function for handling multi class classification problem in Spark. The vector sums up to 1 with each co-efficient associated to a feature, giving the probability for different classes, in this case 3 genre classes.

a. Logistic Regression model was used as a base model because of better interpretability

b. After hyperparameter tuning across values of alpha and lambda, best model had alpha = 0.1 and lambda = 0.01

c. The best model after hyperparameter tuning had 50.71% test accuracy in classifying the genre

```
*********************Logistic Regression Final Model Training Results**************
Training Accuracy: 0.5009742172277856
FPR: 0.24984800576218513
TPR: 0.5009742172277856
F-measure: 0.49508724914008095
Precision: 0.49463105992872936
Recall: 0.5009742172277856
*********************Logistic Regression Final Model Test Results*****************
Test Accuracy: 0.5071327794325129
f1: 0.5021853852683762
Precision: 0.5015849089419002
Recall: 0.5071327794325129
Coefficients:
DenseMatrix([[-0.17735075, -0.29157616, -0.27936485, -0.07376576, -0.09299404,
               0.01343181, -0.2055332 ,  0.08912988,  0.00759815,  0.04373767,
               0.06369213],
             [ 0.18202009,  0.06451993,  0.20717105,  0.04900363,  0.04651343,
              -0.08936255,  0.09368024, -0.10031781, -0.11470049,  0.04517138,
              -0.08618205],
             [ 0.00085014,  0.1568319 ,  0.04787519,  0.0164566 ,  0.06676071,
               0.09453651,  0.07772273,  0.05400754,  0.0459435 , -0.0681021 ,
               0.01593675]])
```

- **NAÏVE BAYES CLASSIFIER:**

As Naive Bayes assumes features are independent in nature, it took substantially less compute time than other models. As Spark does not give flexibility in implementation and tuning of Naive Bayes classification model, we were not able to tune any hyper parameters.

Naive Bayes performs better on complex data and this might be the reason for model not performing as well as other models.

```
*********************Naive Bayes Final Model Test Results***************************
Test Accuracy: 0.4847154726446151
f1: 0.47453797349937316
Precision: 0.4767692516804818
Recall: 0.4847154726446151
```

- **NEURAL NETWORK(MLPC):**

Multilayer perceptron classifiers use Feedforward artificial neural networks with sigmoid activation function for intermediate layers and SoftMax function for the final layer.

a. Neural Networks (MLPC) model performed the second best

b. As we had 11 important features and 3 target classes, different number of hidden layers and neurons were experimented, with layers [11,22,9,3] giving the best results

c. Due to Spark not having flexibility in activation function & optimizer, the model did not perform the best, but was the second best in the 4 models that were performed as part of this project

d. Other hyper parameters included convergence tolerance, learning rate and block size. A total of 500 models were checked with all possible combinations of layers, convergence tolerance, learning rate and batch size and best model accuracy achieved was 52.92%

```
*********************Neural Networks Final Model Test Results***********
Test Accuracy: 0.5292365574541464
f1: 0.527265823838639
Precision: 0.5275082733892166
Recall: 0.5292365574541464
```

- **RANDOM FOREST CLASSIFIER:**

Random forest classifiers work on the concept of combining decisions by grouping and combining multiple decision trees.

a. After hyperparameter tuning across number of trees, max depth, and max bins; the best model had 20 trees (Number of trees in the forest), depth = 15(max number of levels in each decision tree) and bins = 6

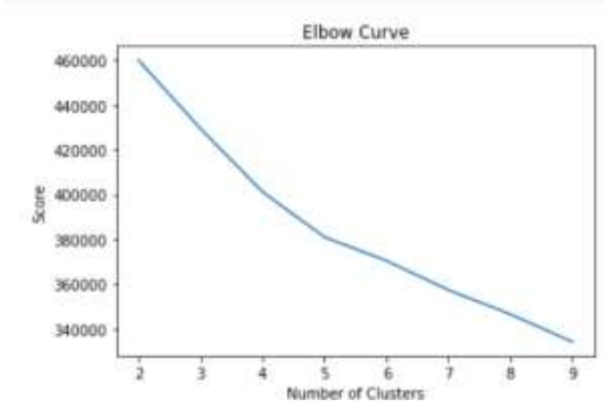
b. The best model gave a test accuracy of 56.67% while classifying the genres out of the 47 models that were tested as part of this module

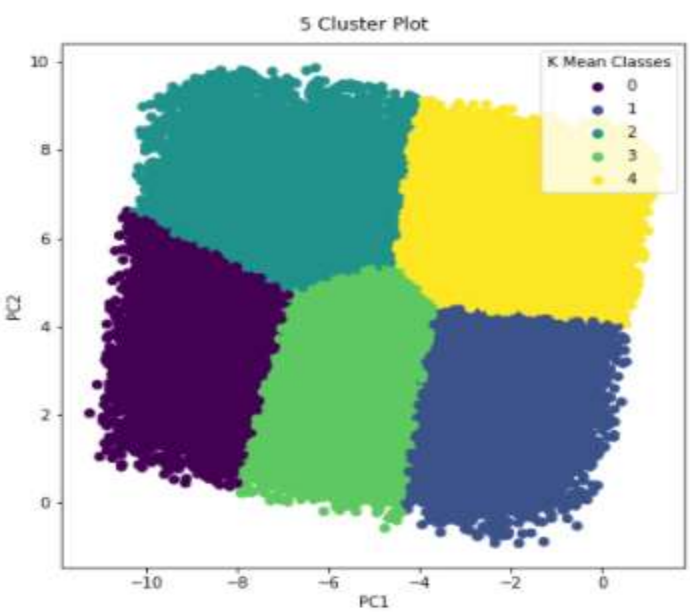
```
*********************Random Forest Final Model Test Results*****************
Test Accuracy: 0.5667032450227308
f1: 0.566255764340597
Precision: 0.5667580496118576
Recall: 0.5667032450227308
```

5. SONG RECOMMENDATION SYSTEM:

- **K-MEANS CLUSTERING:**

  - We wanted to develop an algorithm that, when recommending new songs, would emphasize the qualities of the music users enjoy listening to more than the artist singing the song
  - We wanted to recommend songs that were “similar” to a user’s listening history by clustering songs in the dataset to create our own genres. The method we implemented was a K-Means clustering algorithm because of its simplicity and short training time
  - Identified the number of optimal clusters using the Elbow plot. K=5



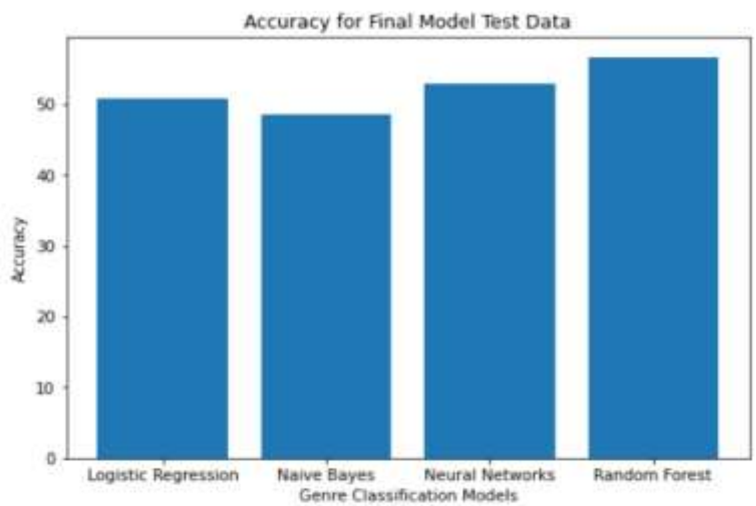


- Analyzing the clusters, we can see that they make sense, as we had bucketed each of our x variables to 5 levels
- Our recommendations would be based on the inter-cluster and intra-cluster arrangements and profiles. We feel this is a comprehensive system as we also consider the characteristics of the songs rather than just their discographic features

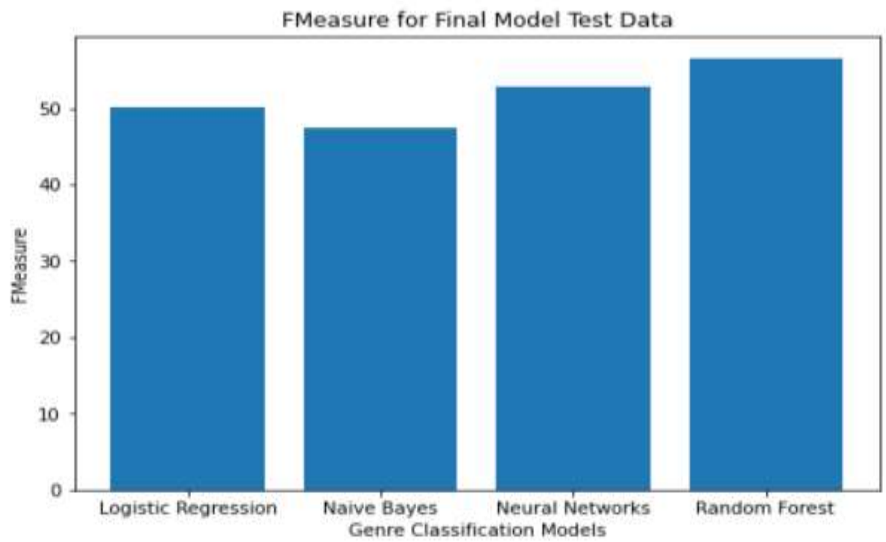
6. STATISTICAL RESULTS - INFERENCE:

After the comparative analysis of different classification models, we find the following results:

- The Random Forest model performed the best with 56.67% test accuracy, when compared to Logistic Regression, Naive Bayes & Neural Networks (MLP) models
- The AUC scores comparison could be done using One Vs All technique as the problem is a multi-class classification problem
- The below bar graph shows the accuracy comparison of different models that were used to classify the genres



- The F-Measure was considered the best evaluator for the genre classification problem
- The Random Forest performed the best with 56.62 F-1 score, followed by Neural networks



7. BUSINESS IMPLICATIONS AND INFERENCE:

Music recommender systems have experienced a boom in recent years, thanks to the emergence and success of online streaming services, which nowadays make available almost all music in the world at the user's fingertip. While today's MRSs considerably help users to find interesting music in these huge catalogs, MRS research is still facing substantial challenges. Through this project, we have attempted to answer the above problems through supervised and unsupervised machine learning techniques.

a. We extracted 11 important features from 30+ features, and performed genre classification task using various models such as Logistic Regression, Naive Bayes, Random Forest, and Neural Networks (MLP)

b. We were able to achieve close to 56.67% accuracy while classifying genres using Random Forest, with MFCC being the most important feature, which makes sense as this feature is generally used for speech recognition system and music information retrieval applications such as genre classification. Ultimately, our final model was fairly accurate based on the metrics we defined for measuring success.

c. Song Recommendation: We were able to successfully recommend songs using inter-cluster arrangements and intra-cluster distances. Our final recommendations took users' listening histories and returned songs from the clusters that were closest to the most concentrated areas of songs from those histories.

8. CONCLUSION:

In this project we have tried to attempt the classification of music into 3 major categories. We achieved reasonable success, while restricting the number of classes to 3. Our approach has raised many interesting questions on which future work can be done. In the future we could try and increase the number of classes to try and figure out the characteristics of individual genres. We would also have liked to try other classification techniques and try to fit different models to the data. Based on our analyses, we can suggest for future research to add other music features in order to improve the accuracy of the recommender system, such as using tempo gram for capturing local tempo at a certain time. We can collect more informative data about the songs, such as instrumentation, form, and Fourier Transform Spectrums. Additionally, adding more “current” songs to the MSD using the Echo Nest API would help make the project’s final model more relevant to today’s trends and popular songs. Knowing how long a user listened to a song or whether they replayed it would provide extremely useful information about weighting input songs.

9. REFERENCES

- Stafford SA. Music in the Digital Age: The Emergence of Digital Music and Its Repercussions on the Music Industry. The Elon Journal of Undergraduate Research in Communications Vol. 1 No. 2. 2010.
- Aggarwal CC. Recommender Systems: The Textbook. 1st ed.: Springer; 2016.
- Oord Avd, Dieleman S, Schrauwen B. Deep content-based music recommendation. Advances in Neural Information Processing Systems 26 (NIPS 2013). 2013.
- O'Bryant J. A survey of music recommendation and possible improvements. In ; 2017. Adiyansjah et al. / Procedia Computer Science 157 (2019) 99–10
- Choi K, Fazekas G, Sandler M. Automatic Tagging using Deep Convolutional Neural Network. arXiv eprints arXiv:1606.00298. 2016
- Choi K, Fazekas G, Sandler M. Convolutional Recurrent Neural Networks For Music Classification. 2017
- Davis J, Goadrich M. The Relationship Between Precision-Recall and ROC Curves. In ICML'06 Proceedings of the 23rd international conference on Machine learning; 2006. p. 233-240.
- Wang X, Wang Y. Improving Content-based and Hybrid Music Recommendation using Deep Learning. In Proceedings of the 22nd ACM international conference on Multimedia; 2014; Orlando. p. 627-636.