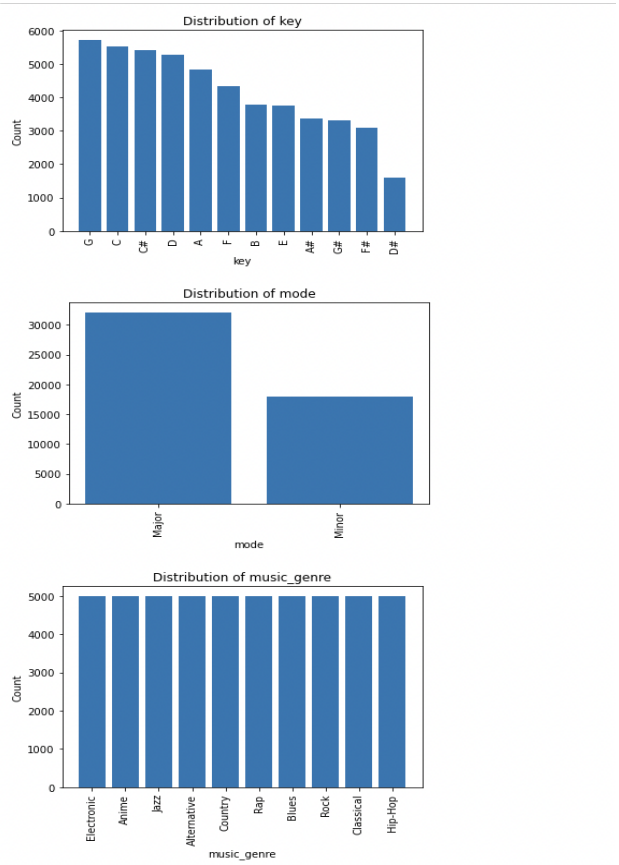
Fundamental to Machine Learning
Yen Le
Capstone Project - Classification Project
Preface :

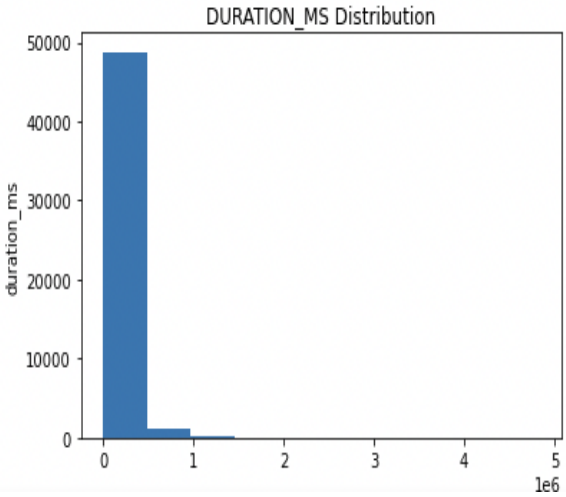
- This capstone project is based on the audio features of Spotify's song released in the Spotify API. In this project, I used features of 50K randomly picked songs to predict the genre that the song belongs to.
- In this project, I perform a visual analysis of the data, perform EDA and visualize data distribution; through plotting bar graphs of each feature in our data. After that, I start to process and clean our data, through handling missing values, outliers, dropping identification data and process categorical variables.
- I then split my data into a train and test set; with the instructions given. Here, the train set size is 45000; and the test set size is 5000 with an equal distribution of music genres in each set.
- After this, I performed feature engineering by applying dimension reduction analysis on our model and then used the reduced data for Classification Models. For our dimension reduction, I used PCA. And for our Classification Models, I will be using Random Forest Classifier and Logistic Regression; then evaluate the two models using AUC score and along with accuracy score.

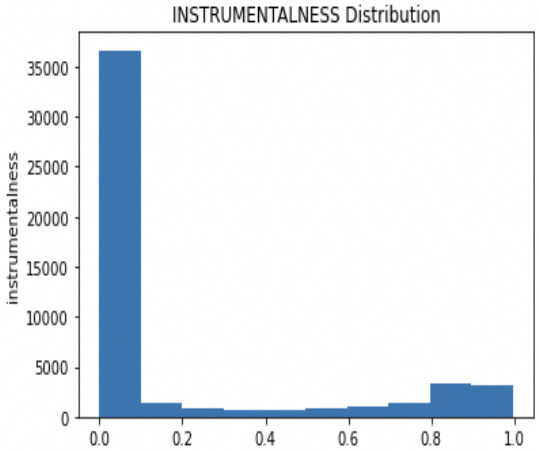
1. EDA :
    - I first look at the type of data and missing values in our data to forecast categorical variables and remove missing values if needed. In our dataset, most variables are in numerical form with exception to identification data as well as tempo, key and artist name.

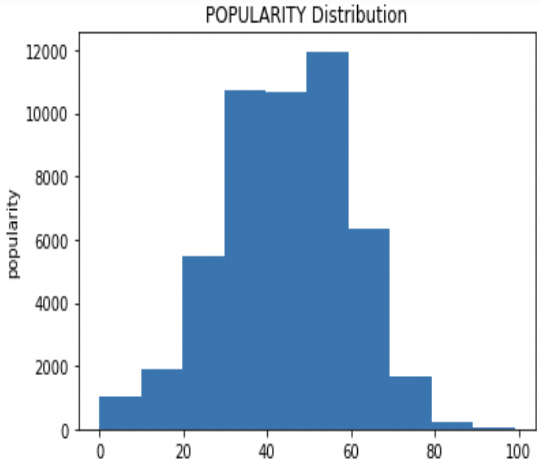
    - I then look at the total of null values in our dataset and locate the missing values. From our EDA, we have 5 missing data in every column; and further data analysis shows that we are missing the whole row at index from 10000-10004. Since all data in a row is missing, I decide to drop that row.
2. Visualizing data :
    - Apart from this, I also visualize data using bar graphs.
    - For categorical data, the distribution of frequency are as follows :

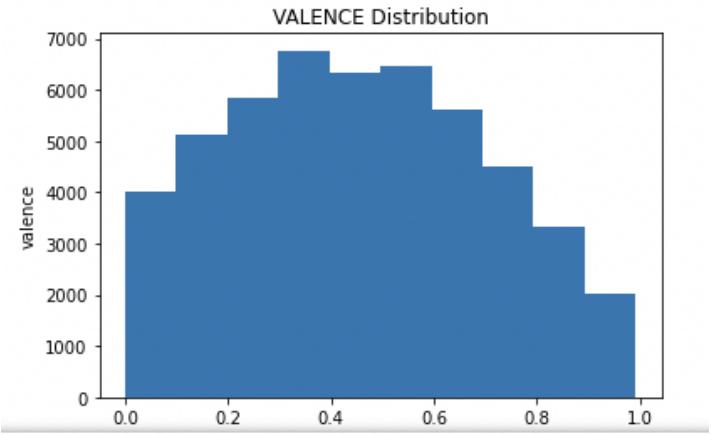
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50005 entries, 0 to 50004
Data columns (total 18 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   instance_id       50000 non-null  float64
 1   artist_name       50000 non-null  object
 2   track_name        50000 non-null  object
 3   popularity        50000 non-null  float64
 4   acousticness      50000 non-null  float64
 5   danceability      50000 non-null  float64
 6   duration_ms       50000 non-null  float64
 7   energy            50000 non-null  float64
 8   instrumentalness  50000 non-null  float64
 9   key               50000 non-null  object
 10  liveness          50000 non-null  float64
 11  loudness          50000 non-null  float64
 12  mode              50000 non-null  object
 13  speechiness       50000 non-null  float64
 14  tempo             50000 non-null  object
 15  obtained_date     50000 non-null  object
 16  valence           50000 non-null  float64
 17  music_genre       50000 non-null  object
dtypes: float64(11), object(7)
memory usage: 6.9+ MB
Null values and data type:  None
```

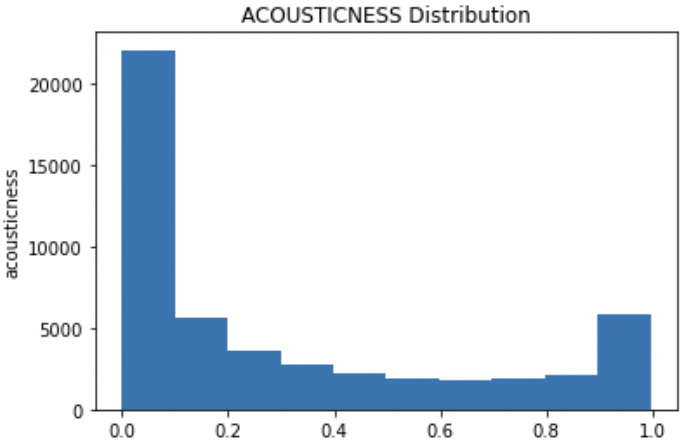












- From our collective graphs, we can see that the distribution of valence and popularity are quite normal; while the other features are very skewed, like acousticness, instrumental or duration_ms.

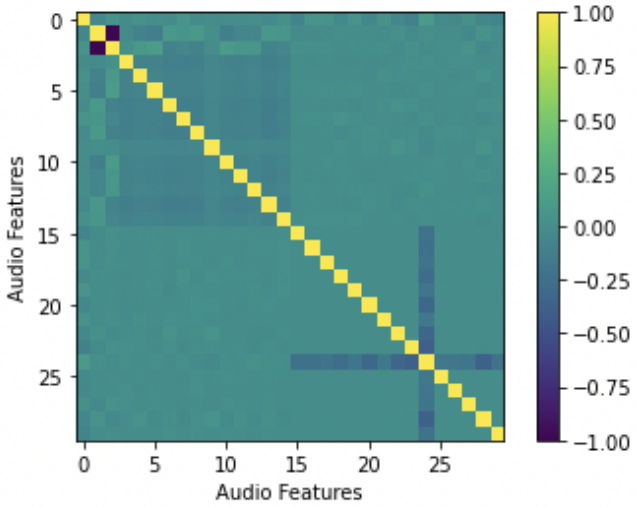
3. Data processing :
    - I drop all the identification data columns that are unrelated to the data analysis such as instance_id, track_name and obtained_date.
    - I then process the artist_name by replacing the missing data labeled as “empty_fields” by others.
    - I also transform value (-1) in duration_ms to Nans values and then imputed by its aggregated median (as introduced in our labs)
    - Next, I transform ‘tempo’ data to categorical by turning “?” data to Nan and then also impute it to my aggregate median. In both cases of duration_ms and tempo, I visualize the data to see if imputing by median would be a good way to handle missing values. From the two graphs, we can see that the mean and median lies close to each other, so it was fair to impute by its aggregated median.
    - I then one-hot encoded other categorical variables such as “key”, “mode”, “artist_name”.
    - I then label encode my genre to numerical variables from 0 to 10.
4. Splitting data :
    - To split my dataset, I group data by genre and randomly select 500 songs from each genre for our test set; and the remaining 45000 songs each genre for our train set.

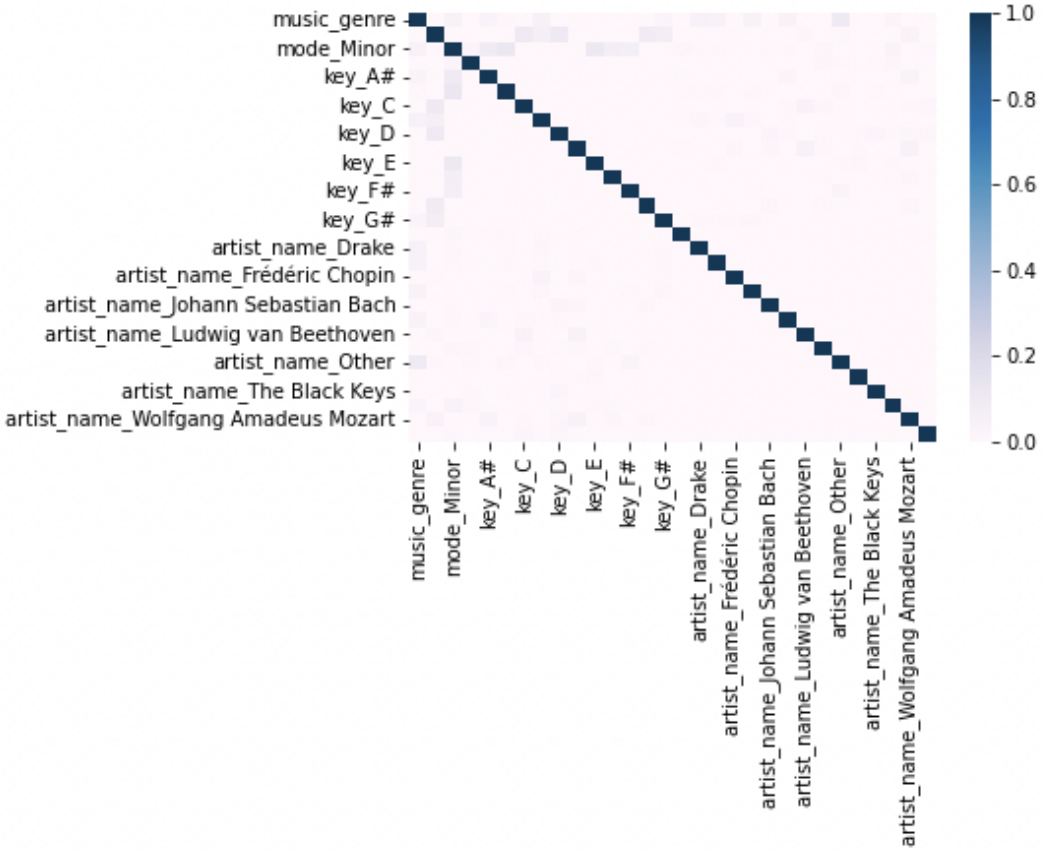
```
Train set size: 45000
Test set size: 5000
```

    -

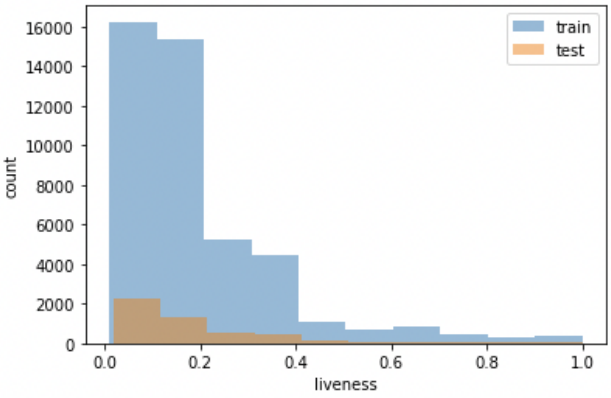
5. Dimension Reduction :

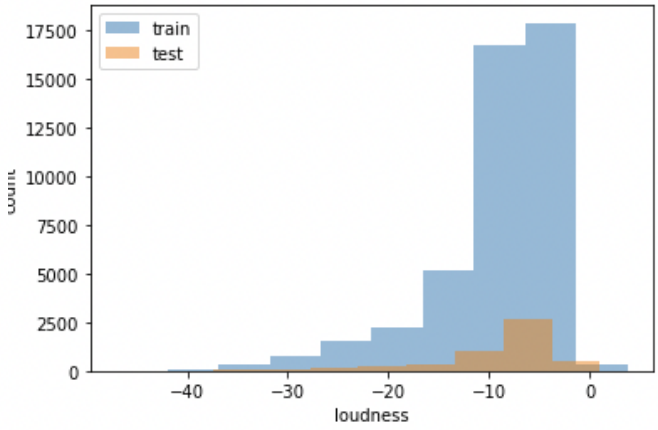
- I first visualize our audio features, and we can see the correlation between our data. Since we have a lot of variables encoded; the correlation is not very clear. However, we can see places where correlations are darker; meaning that our data is somewhat correlated.
- Due to this high dimensionality; we must perform dimension reduction before putting our data in our model to avoid overfitting.

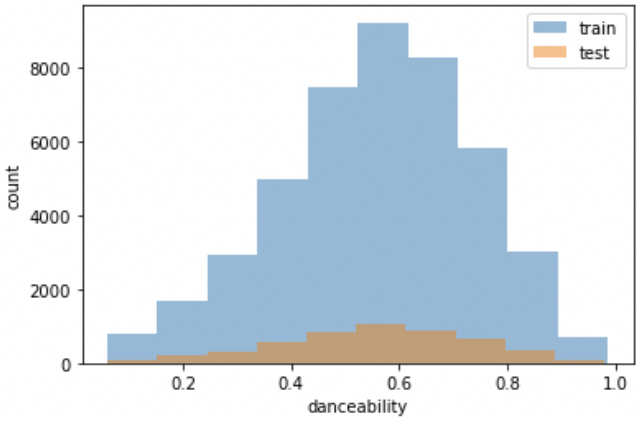


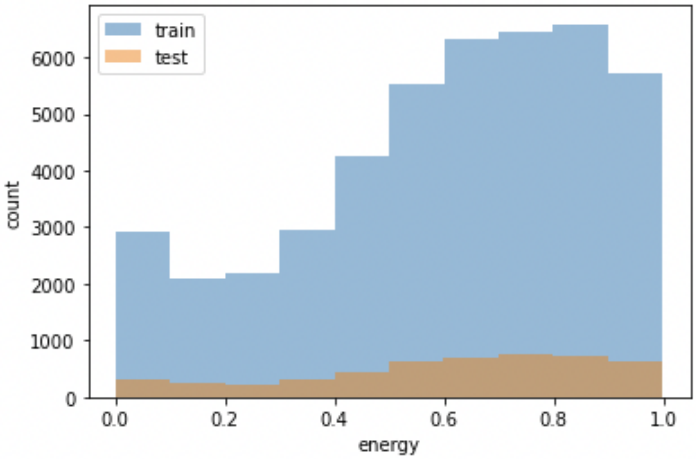


- I then check for covariate imbalance, by plotting the data distribution between our train and test ; and from our visualization we can see that there is a similar distribution across all features. Meaning that there is not covariate imbalance in our data.

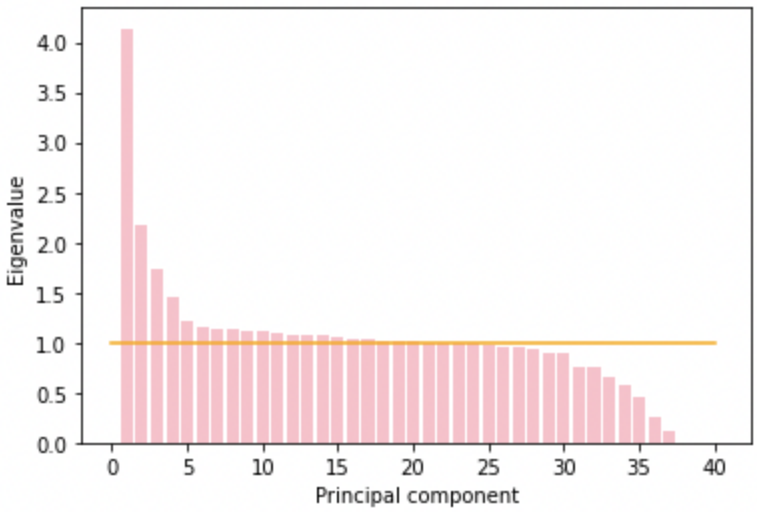




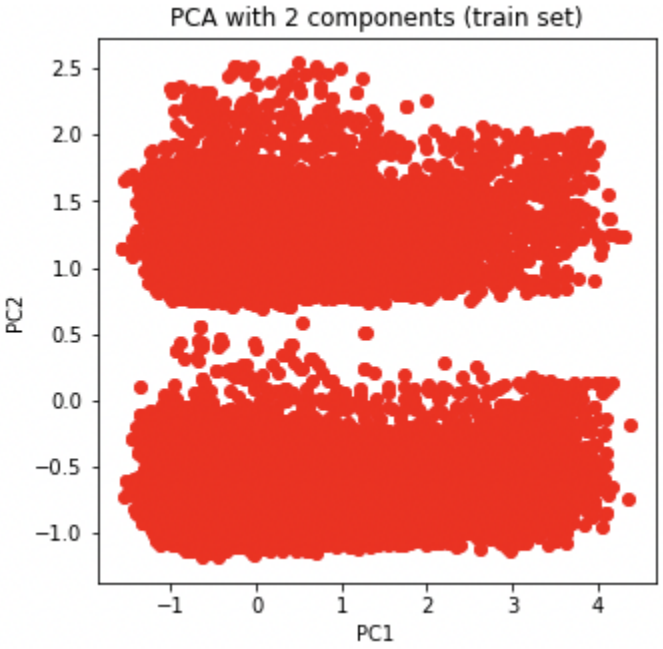




4. Dimension reduction :
   - I use PCA for dimension reduction. Using PCA, I first Z-scored my data using Standard Scaler. I then plot the ScreePlot to visualize the principal component. In this case, I found 22 principle components, as decided by the Kaiser criterion.

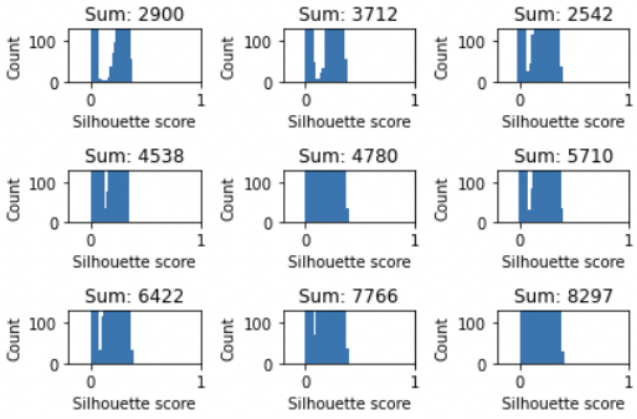


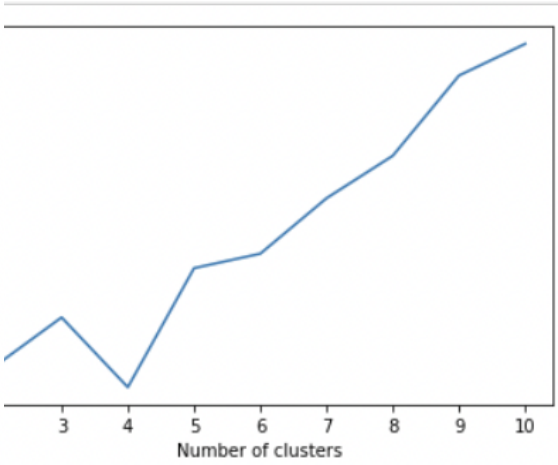
-

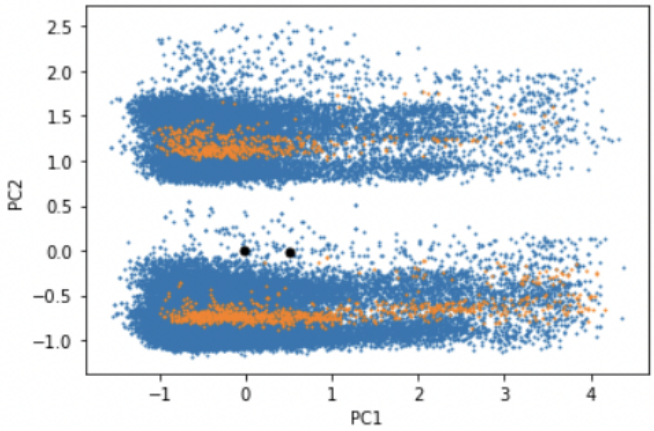


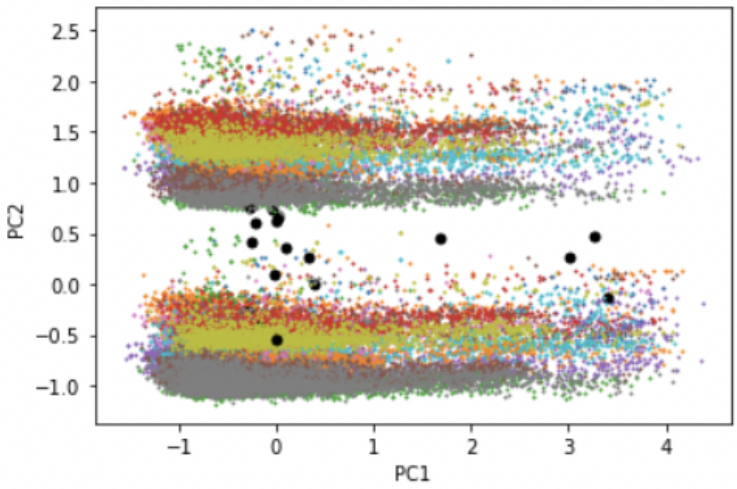
-

- I then calculated the silhouette score and attempted to plot the clusters from our PCA. And plot the data against 2 clusters.



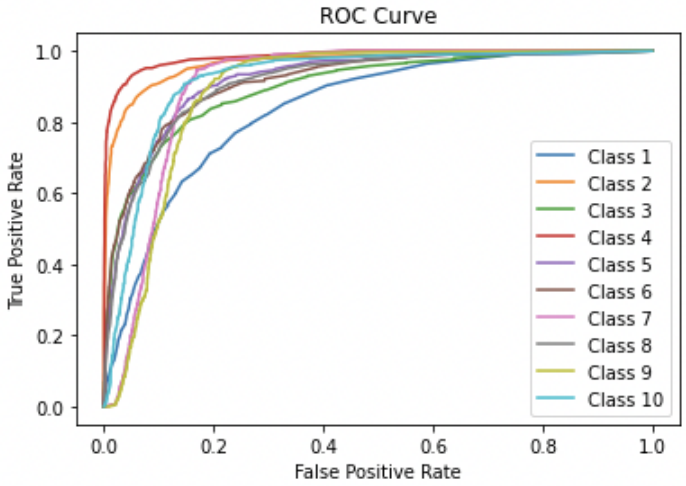






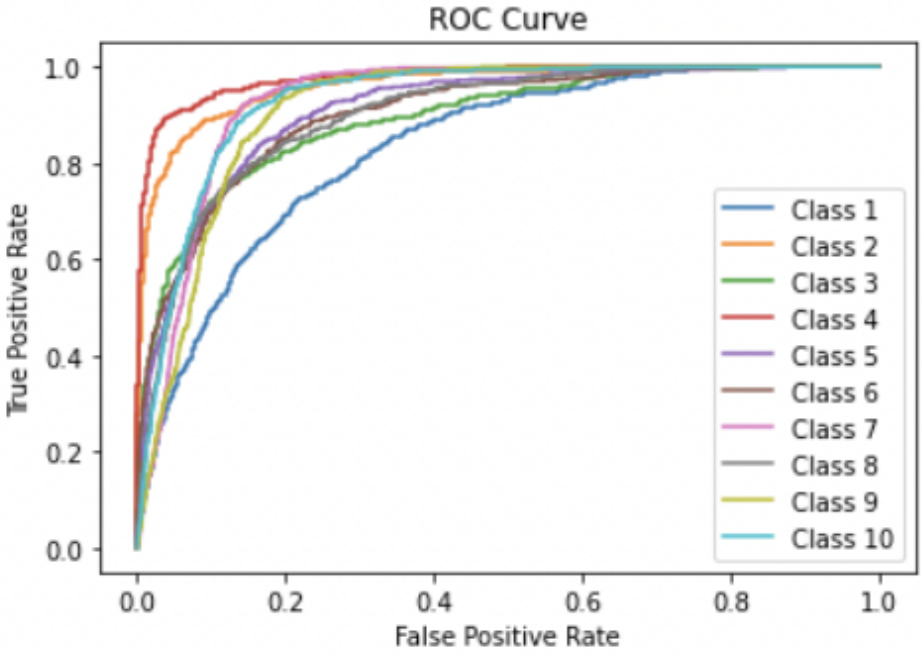
5. Training Model
   - Random Forest :
   - I trained a Random Forest Classifier with 100 trees and used the ‘gini’ criterion. From my model, I achieved an AUC score of 0.916. In this Random Forest Classifier, I also specify ‘multi_class’ = ‘ovr’ to be able to print out the ROC curve of a multiple classifier.

```
AUC score: 0.9160810222222222
```



- I also implemented hyper parameter tuning; using Grid Search to find the best number of trees and the maximum depth of my tree. And my result was a max_depth of 8 and n_estimator of 500. This contrast in number suggests me to perform this at a deeper depth.

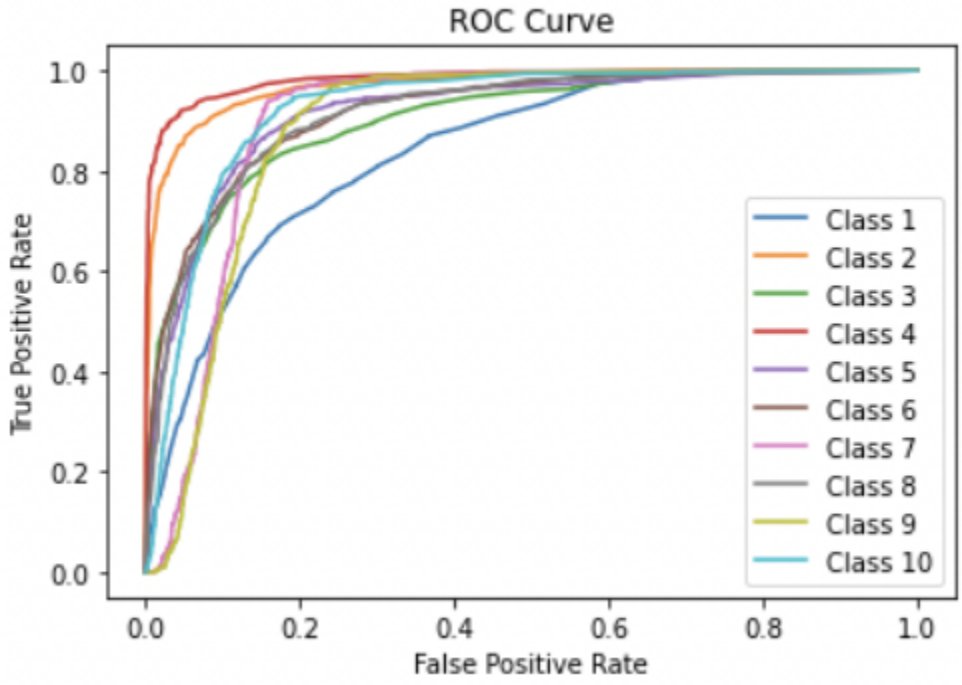
```
Best Parameters: {'max_depth': 8, 'n_estimators': 500}
Best Score: 0.5679555555555555
AUC score: 0.9160062666666666
```



- Logistic Regression ;

- In this model, I also use multi class Logistic Regression to fit my model and find 3 AUC scores for each class, as well as the micro-average and macro-average AUC score.

```
AUC score: 0.9172732
```



-

```
AUC Scores for Each Class: [0.80362178 0.93671289 0.86718533 0.97397956 0.882958   0.88933889
 0.92283267 0.86045889 0.91224956 0.91726778]
Micro-Average AUC Score: 0.9036649022222223
Macro-Average AUC Score: 0.8966605333333334
```

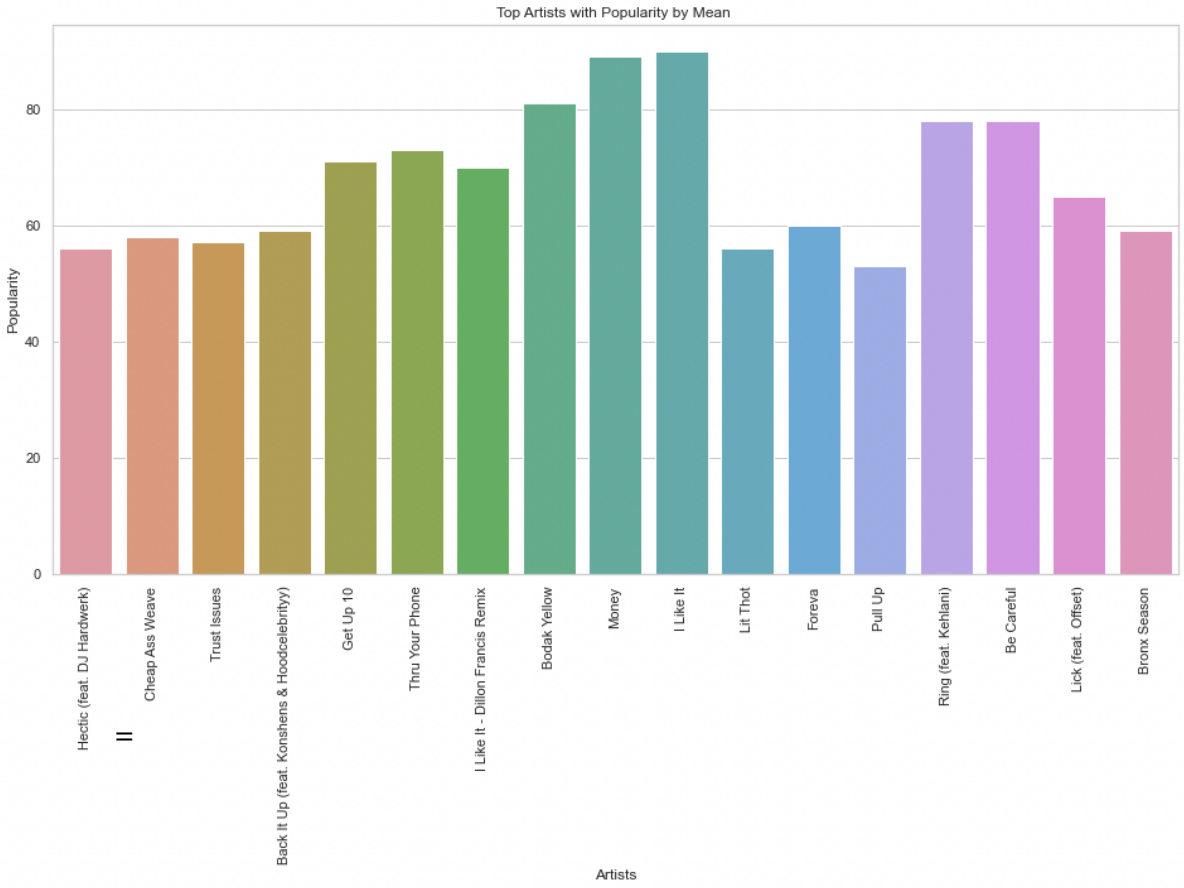
6. Acknowledgement :
   - I want to acknowledge that my findings have some issues, and I would love to continue to continue fixing this project and continue to work on it until I find out what was happening.
   - From my model, although my AUC score was high, my accuracy score was; however; low. I achieved an accuracy score of 0.56 which is not high at all. However, I believe that there are 2 problems with this accuracy score that I can improve if given more time. (As the professor said, to do this project in the midst of final week).
     - a) I use label encode for my music genre.
       - Though I understand that label encode is quite popular to encode categorical data; label encode strips out the mathematical distance of our data. Thus; making it nonsense. What I mean is; say Rock is 1

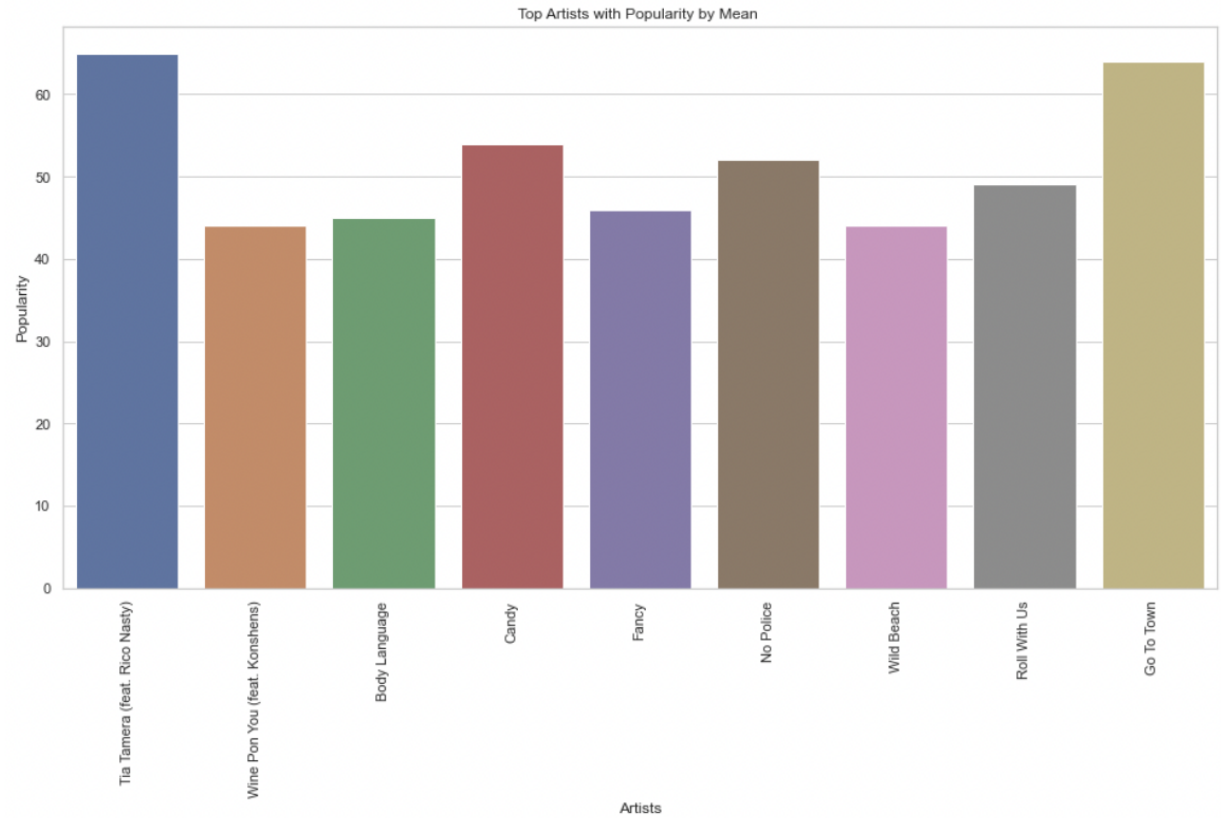
and Hip-hop is 2 and Ballad is 3. That means that Ballad = Rock + Hip-hop; which is not true at all.
- Thus, using this encoding has introduced some wrong notation in my data and made the result not as good as wanted.
- I could change this; yes, but I do not have enough time to because my method of splitting data depends on group by, which requires music_genre to be encoded like that.

b) Dimension Reduction :
- I do believe that my dimension reduction can be improved greatly if I had used other methods such as t-SNE or LDA.
- This is because my PCA now has 22 principal components, which is still a lot !.

7. Regards :
- As I am closing up my capstone report, I want to truly say thank you for everyone who is a part of this class and is making it better. Writing this in a report might not be appropriate but this might be the only time I can say thanks to all those who have helped me in my report. This class and this semester as a whole has taught me so much that I cannot imagine and has helped me grow more than what I can ask for.
- Ending this class is very emotional to me because it has been a big part of this semester. Again, thank you for grading this report and reading through our reports!

Extra Credit :
- My two favorite artists are Cardi B and Doja Cat so using this data; I want to visualize their popular songs on charts and the result is quite surprising too !!.
- I expect Cardi B's most popular song to be Money because the song made her famous. But for Doja Cat I was expecting Say So, but did not see it in the chart (i know that this is a subset data but still :p)


Top Artists with Popularity by Mean
Popularity
0
20
40
60
80
Hectic (feat. DJ Hardwerk)
Cheap Ass Weave
Trust Issues
Back It Up (feat. Konshens & Hoodcelebrityy)
Get Up 10
Thru Your Phone
I Like It - Dillon Francis Remix
Bodak Yellow
Money
I Like It
Lit Thot
Foreva
Pull Up
Ring (feat. Kehlani)
Be Careful
Lick (feat. Offset)
Bronx Season
Artists


Top Artists with Popularity by Mean
Popularity
0
10
20
30
40
50
60
Tia Tamera (feat. Rico Nasty)
Wine Pon You (feat. Konshens)
Body Language
Candy
Fancy
No Police
Wild Beach
Roll With Us
Go To Town
Artists