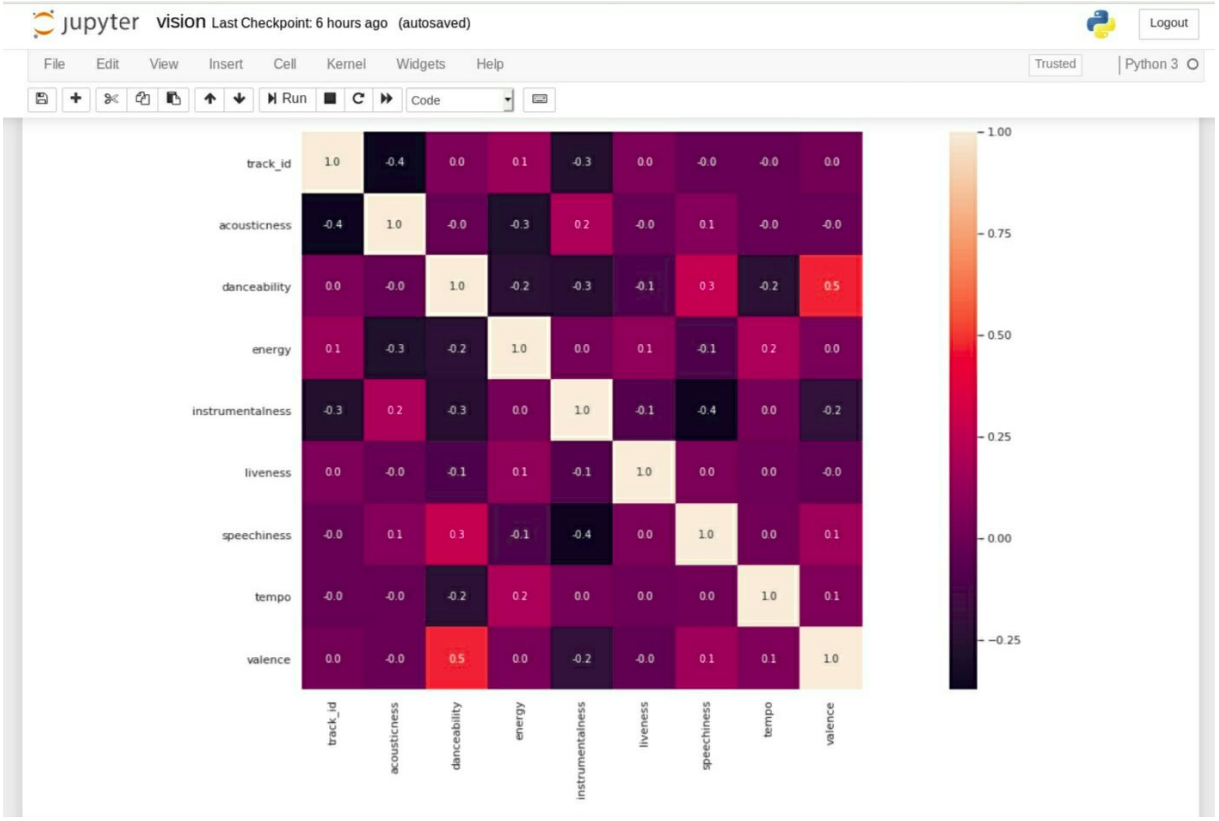
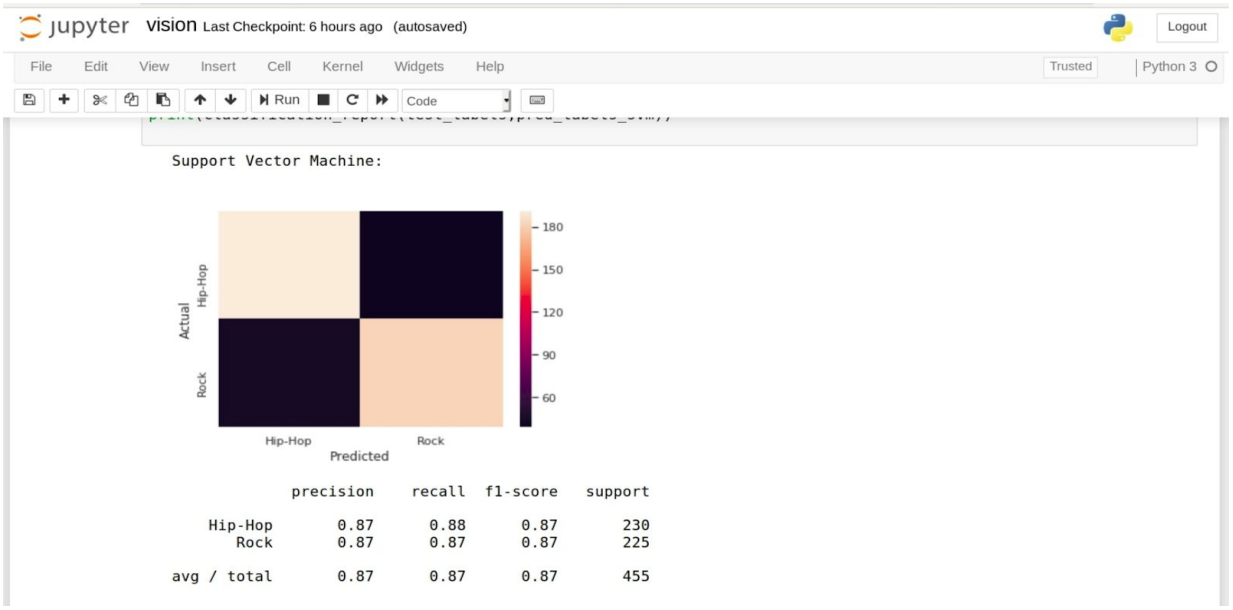
CLASSIFYING SONG GENRES FROM AUDIO DATA

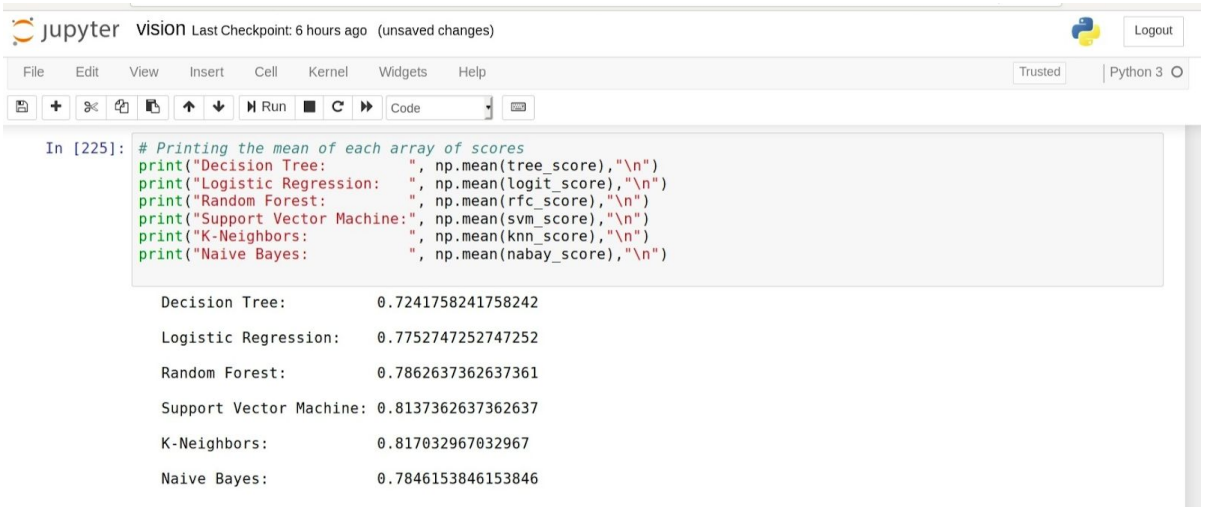
Team Member: Meenal Agrawal (17ucs187)

Project Description: Using a dataset comprised of songs of two genres, classifier is trained to distinguish the two genres based on only track information derived from Echonest .

Screenshots:







Installation Process (Linux):

1. Installation of Python packages:
   pip3 install numpy
   pip3 install matplotlib
   pip3 install sklearn
   pip3 install pandas
   pip3 install seaborn

Technologies:

Python packages : numpy, pandas, matplotlib, seaborn, sklearn.

Detailed Description:

1. Preparing the dataset:

These exist in two different files, which are in different formats - CSV and JSON. We start by creating two pandas dataframe out of the dataset files and then merging them so we have features and labels for the classification later on.

2. Pairwise relationships between continuous variables:

We want to avoid using features having strong correlations with each other (hence avoiding feature redundancy). We analyse such features by creating correlation matrix.

Preprocessing dataset:

3. Normalizing the feature data:

We define the features and labels from the dataset and then normalize the data.

4. Principal Component Analysis on our scaled data

Since we didn't find any particular strong correlations between our features (step -3), we use **principal component analysis (PCA)** to find out by how much we can reduce the dimensionality of our data. We can use scree-plots and cumulative explained ratio plots to find the number of components to use in further analyses.

PCA : PCA rotates the data along the axis of highest variance, thus allowing us to determine the relative contribution of each feature of our data towards the variance between classes.

5. Balance our data for greater performance:

By looking at the number of data points we have for each class, we see that we have far more data points for the rock classification than for hip-hop, potentially skewing our model's ability to distinguish between classes. So we sample the rocks

songs to be the same number as there are hip-hop songs.

___6. Training the classifiers:

1. Decision Tree:

Decision trees are rule-based classifiers that take in features and follow a 'tree structure' of binary decisions to ultimately classify a data point into one of two or more categories.

2. Logistic Regression:

Logistic regression makes use of the logistic function to calculate the odds that a given data point belongs to a given class**.**

3. Random Forest Classifier:

Random forest builds multiple decision trees and merges them together to get a more accurate and stable prediction.

4. Support Vector Machine (SVM):

In this algorithm, we plot each data item as a point in n-dimensional space (where n is number of features we have). Then, we perform classification by finding the optimal hyperplane that differentiate the two classes very well.

5. Naive Bayes classifier:

It calculates the probabilities for every factor. Then it selects the outcome with highest probability.

6. K-Neighbors classifier:

The algorithm makes predictions by calculating similarity between the input sample and each training instance.

7.Using cross-validation to evaluate our models:

To get a good sense of how well our models are actually performing, we can apply cross-validation (CV).
I have used K-fold CV here. K-fold first splits the data into K different, equally sized subsets. Then, it iteratively uses each subset as a test set while using the remainder of the data as train sets. Finally, we can then aggregate the results from each fold for a final model performance score.

The cross-validation scores are:

Decision Tree: 0.7241758241758242
Logistic Regression: 0.7752747252747252
Random Forest: 0.7862637362637361
Support Vector Machine: 0.8137362637362637
K-Neighbors: 0.817032967032967
Naive Bayes: 0.7846153846153846

Hence, we find that K-Neighbors Classifier and Support Vector Machine are the strongest predictors here.