# Musical Genre and Song Popularity Classification

COMP 5411 FB Fall 2019 Final Project Report

Manva Trivedi
*Student Id : 1095816*

Devanshu Mishra
*Student Id : 1104093*

Shreya Bandyopadhyay
*Student Id : 1105134*

***Abstract*—In the course of recent years, streaming services with immense play lists have become essential methods via which a great many people tune in to their preferred music. And yet, the sheer measure of music on offer can mean users may be overwhelmed when attempting to search for more up to date music that suits their preferences. Thus, streaming services have investigated methods for sorting music to take into consideration customized recommendations for which genre classification becomes necessary.**

**In this project, we compare the performance of various genre classification models to classify a specific genre from an initial audio file. The study is based on a freely available collection of audio features and metadata from Million Song Dataset from Echo Nest. For that, we converted audio signals to tabular form. After finding correlation using heat map, which helped in finding correlation between features in the whole dataset. After that we progress into presenting accuracies using different models such as K-Nearest Neighbors (KNN), Support Vector Machines (SVM) and Random Forests (RF). We apply cross validation after dividing data into stratified folds and perform feature selection and Synthetic Minority Over-sampling Technique (SMOTE) sampling to tackle class imbalance problem on training set resulting in classification of genre.**

**For predicting popularity, we have studied correlation among features and divided the class according to whether the song is popular or not using mean of the hotness where, hotness is a feature which is calculated by a combination of track hotness and artist hotness. We then predicted the popularity and applied stratified cross validation using different methods such as KNN, Support Vector Machines and Random Forests and found promising results with Random Forests.**

***Index Terms*—Music Genre Classification, Popularity Prediction, Feature Selection, Synthetic Minority Over-sampling Technique, Stratified Cross Validation, K-Nearest Neighbors, Support Vector Machines, Random Forest, Classifiers**

## I. Introduction

**Problem Statement :** Currently, genre classification is performed manually by humans applying their personal understanding of music. Furthermore, popularity of a song is subjective when it is done manually so it will be a biased representation of the popularity of the song. Proposed classifier can be used for tasks such as automatically tagging music for distributors such as Spotify and Amazon Prime Music and determining appropriate background music for events. The main aim of our project will seek to see if the next hit can be predicted and if audio signals can be categorized into various genres such as jazz, blues, latin, pop, hiphop, rock, classical, country, etc.

**Significance of the problem :** At present, the classification of the musical genre is done manually. Automatic identification of musical genres will aid or substitute the human in this process and would be an useful addition to recovery systems of music data. Moreover, automated popularity prediction can reduce the chances of biasing. Another significance of genre and popularity can be that it can be of great use in trend analysis of songs, for example, regionally categorizing the top billboard songs on the basis of genres to help maximize the widespread of the songs.

**Literature Review :** The problem of forecasting success of songs has been extensively investigated. In the paper ”Musical Genre Classification of Audio Signals” by George Tzanetakis and Perry Cook [1] the automatic categorization of audio signals into a musical genre organizational structure is discussed . In particular, three feature sets are advocated to represent timbral texture, rhythmic content and pitch content.

In the article ”Song Analysis” by University of California’s Thomas Astad Sve , it is concluded that tracking popularity and genres from track metadata is not possible. [10]

Stanford University’s three graduates, Pham, Kyauk, and Park, predicted song success with a view of computer science. [5] They took one percent subset of the million song dataset in their work and then marked the top 25 percent hottest songs as famous. We had a description and extraction of features was more complicated than we did in this venture. We included a feature called artist familiarity, for example, which proved to be the most important feature when predicting the success of the song as not surprisingly. We have not included these features and will rely on track metadata to see if similar results can be obtained.

Salganik, Dodds, and Watts conducted a popularity study and concluded that a quality of songs only partially influences whether a song becomes popular or not [3].

In [4], with cepstral coefficients and a hidden Markov model (HMM), audio signals are segmented and categorized into ”art,” ”voice,” ”laughter,” and non-speech sounds.

**Comparison with previous work :** The automated grouping of audio signals into a musical genre hierarchy is addressed in a paper on [1]. In general, three feature sets were proposed to reflect timbral texture, rhythmic content and pitch content.

In the paper, multiple standard statiftical pattern recognition classifiers such as simple Gaussian classifier (GS), Gaussian mixture classifier (GMM), K - Nearest Neighbors classifier are used for classification.

We have used SMOTE ( Synthetic Minority Over-sampling Technique) for upsampling the data and Random Forest, SVM and KNN classifiers for classification. We have also used K fold and Stratified cross validation for comparing the results obtained through the classifiers.

**What has been achieved in this project :** By performing classification using Random Forest, KNN and SVM classifiers, we could successfully classify the genres and popularity of songs. Comparing the results, we got the best performance using Random Forest.

## II. Data

**Data source :** The dataset used is Million Song dataset provided by the Echo Nest which is now owned by Spotify. We extracted 1% songs and bifurcated into two datasets - one for genre and the other for prediction.

**Data description :** Genre dataset has instances of approximately 60k and 33 fratures, where as, popularity dataset hs 25k instances and 10 features.

## III. Methods and Tools

In our project, we extracted 1% songs from the Million Songs dataset, converted the audio files in tabular format using read_hdf function provided in Pandas library and bifurcated into two datasets - one for genre and the other for prediction.

- **Genre Classification :**
  We checked correlation between features using heatmap.Heat map is a graphical representation of attributes in which the colors represent how much correlated the attributes are with each other. The features with the darkest shade are the most correlated with each other. Figure 1 shows the correlation of features for genre classification.

  We checked class distribution first and saw an imbalance in the class. Therefore we considered performing sampling. However, on performing down sampling, data was being lost. So we performed upsampling using SMOTE (Synthetic Minority Over-sampling Technique). For classification, we use Random Forest classifier, K - Nearest Neighbors (KNN) classifier and Support Vector Machine (SVM) classifier. Standard scaling is used in SVM for genre classification. We first measure the accuracy using Random Forrest classifier. Then apply

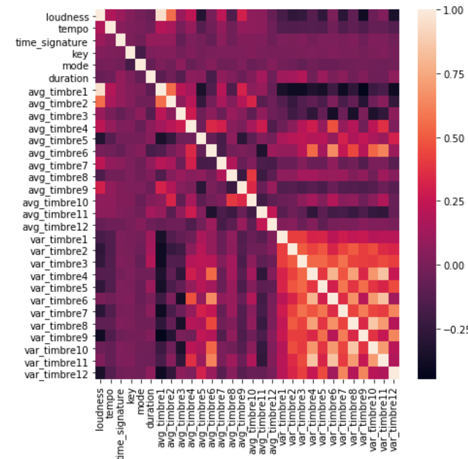


Fig. 1. Heat Map for Genre classification

Cross Validation by dividing the dataset into stratified folds and then performed Feature Selection on the training dataset. Finally, sampling is again performed on the training dataset to remove class imbalance. This process is repeated for KNN and SVM classifiers.

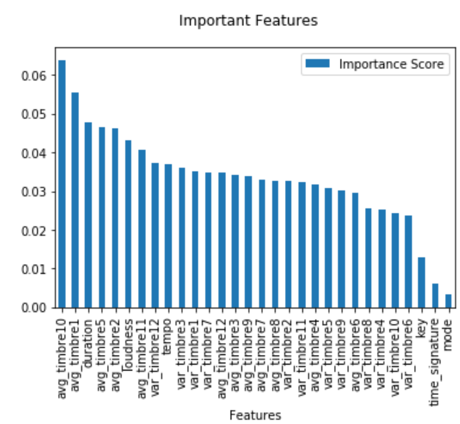


Fig. 2. Gini Indices comparison for Genre classification

We used Random forest to perform feature selection and Figure 3 shows the Gini Index comparisons for features for genre classification. Feature selection helped us understand interpretability.

- **Popularity Classification :**
  We checked correlation between features using heatmap and then removed the missing values. Figure 2 shows the correlation between attributes for popularity prediction.

  We selected top 5 features each for genre and popularity classification.
  Two types of missing values were identified, first being instances having Year value 0 and second being instances having year values but not having song hotness. It was necessary to remove these values since year and song

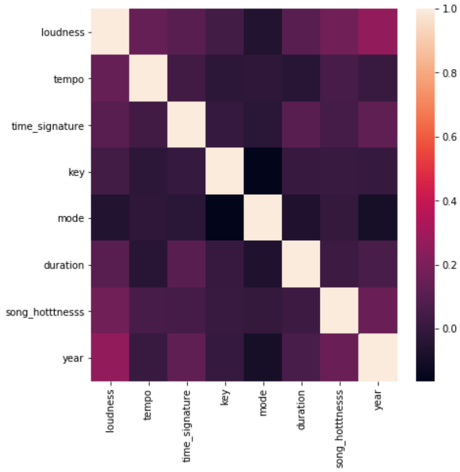


Fig. 3. Heat Map for Popularity Classification

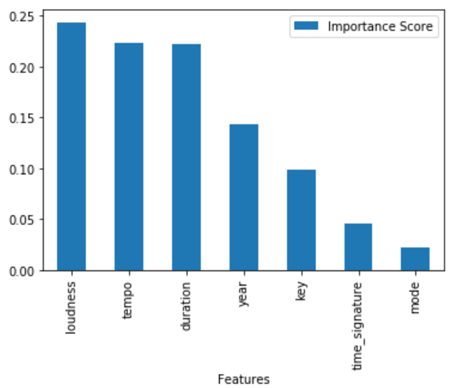


Fig. 4. Gini Indices comparison for Popularity classification

hotness are important features for popularity prediction. We performed class division into classes is popular or not using the average of song hotness and created a new feature "is popular". For classification, we use Random Forrest classifier, K - Nearest Neighbors (KNN) classifier and Support Vector Machine (SVM) classifier. We first measure the accuracy directly using Random Forest classifier. Then apply Cross Validation by dividing the dataset into stratified folds and then performed Feature Selection on the training dataset. This process is repeated for KNN and SVM classifiers.

We also plotted the variations of loudness, tempo and song duration with the year as shown in Figure 5,6,& 7 respectively.

It is seen that the duration and loudness of songs have increase drastically over the years where as there's not much of a change in the tempo.

### A. Anticipated learning methods

*1) Classification:* A variety of classifiers will be used for classification purposes such as K Nearest Neighbors(KNN), Support Vector Machine(SVM), and Random Forest which

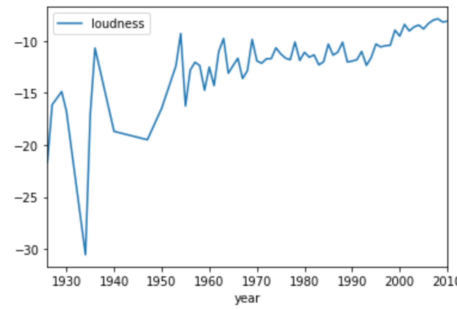


Fig. 5. Loudness vs Year

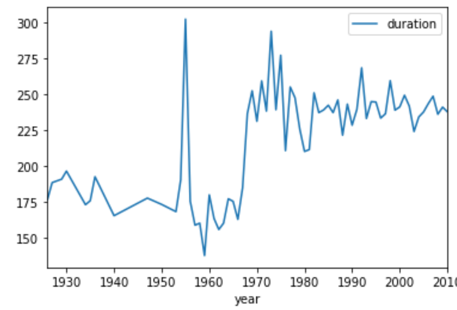


Fig. 6. Duration Vs Year

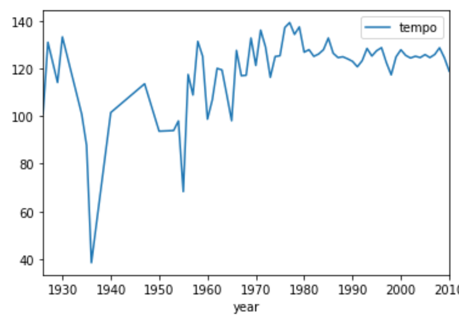


Fig. 7. Tempo vs Year

can be explained as follows:

- **Nearest Neighbors:** K closest neighbors is an algorithm that stores all available cases and classifies new cases based on a measure of similarity (e.g. distance functions). A case is graded by a majority vote of its neighbors, the case being assigned by a distance function to the most common class among its nearest K neighbors. If K = 1, the case will simply be assigned to its closest neighbor's group. [6]

- **Support Vector Machine (SVM):** A Support Vector Machine (SVM) is formally defined by a separate hyperplane as a discriminatory classifier. In other words, given the marked training data (supervised learning), an optimal hyperplane is generated by the algorithm which categorizes new examples. This hyperplane is a line dividing a plane into two sections in two dimentional spaces where it lies on either side in each section. [7]

- **Random Forest:**
  Random forest is a method that builds predictive models together. In this method, the "forest" is a collection of decision trees which function as "strong" classifiers that are poor predictors as individuals but are a robust prediction in aggregate form. Because of their simplicity, lack of assumptions and high overall performance, it is used widely. [8]

### B. Validation methods:

- **K Fold Cross Validation :** In this, dataset is randomly split into k mutually exclusive folds and each fold is trained and tested k times.

- **Stratified Cross Validation :** Stratification is the data rearrangement process to ensure that each fold is a good representative of the whole.

### C. Sampling Methods

- **SMOTE ( Synthetic Minority Over-sampling Technique) :** This is a mathematical method for balancing the number of cases in a dataset. The system operates by creating new instances that is provided as input from current minority events. This SMOTE implementation does not alter the number of cases in the majority. The resulting instances aren't just versions of existing minority cases; rather, the algorithm collects samples of the feature range for each target class and its closest neighbours and produces new examples that combine target case features with neighbouring features.

### D. Tools/libraries :

- **IDE** - Anaconda3 Spyder - 3.3.6
- **Programming Language** - Python - 3.7
- **Libraries** - Scikit-Learn, Numpy, Keras, Pandas, matplotlib

## IV. Results

We applied stratified cross validation for KNN, SVM and Random forest classifiers for result comparisons. Figure 8 shows KNN, SVM and Random Forest accuracy comparisons for each genre class. As seen from Figure 8, we were able to achieve higher accuracies with Random forest for classes classic pop and rock, punk, folk, pop, metal, hiphop, soul and reggae compared to dance and electronica, jazz and blues and classical.

Figure 9 shows the comparison of 2,4 and 5 fold stratified cross validation for KNN,SVM and Random Forest classifiers. Random forest is seen to give promising results for popularity prediction.

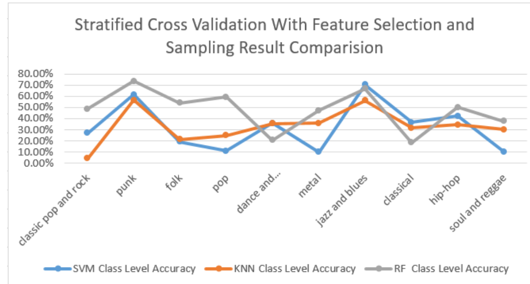


Fig. 8. Graphical Results for Genre classification accuracy

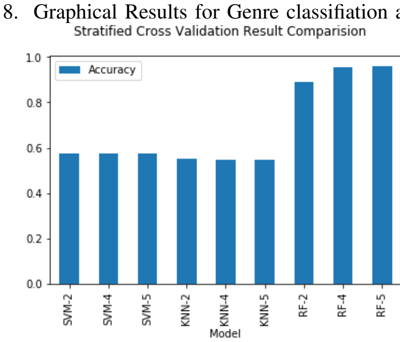


Fig. 9. Stratified Cross Validation result comparison for Popularity

Figure 10 shows the class wise accuracy results with 2 fold Stratified Cross validation, Random Forest Feature selection and SMOTE Sampling for KNN, SVM and Random Forests for Genre classification. We achieved an over all of 45.76% accuracy for Random Forest which is the highest among the three classifiers. In Figure 11, the results are for 5 fold Stratified Cross validation, Random Forest Feature selection and SMOTE Sampling for all the three classifiers for Genre classification . Random forest again performed the best with an overall accuracy of 47.63% which is better than that in 2 fold cross validation. Figure 12 shows results of 2,4,5 fold stratified cross validation, Random forest feature selection on SVM, KNN and Random forest. The accuracy of Random Forest is the best among the three with accuracies of 88.48%, 95.50% and 95.97% respectively for 2,4 & 5 folds, with only slight variations in acccuracies for 4 and 5 fold stratified cross validation.

| Class | SVM Class Level Accuracy | KNN Class Level Accuracy | RF Class Level Accuracy |
|---|---|---|---|
| classic pop and rock | 0.85% | 4.43% | 48.90% |
| punk | 38.53% | 52.40% | 71.72% |
| folk | 53.14% | 20.91% | 52.18% |
| pop | 27.38% | 24.27% | 58.52% |
| dance and electronica | 1.84% | 34.79% | 15.90% |
| metal | 0.00% | 34.49% | 45.50% |
| jazz and blues | 54.02% | 54.02% | 66.62% |
| classical | 11.45% | 29.31% | 15.03% |
| hip-hop | 49.91% | 34.47% | 48.56% |
| soul and reggae | 0.50% | 29.06% | 34.64% |
| | | | |
| Overall | 23.76% | 31.82% | 45.76% |

Fig. 10. Results of 2 fold Stratified Cross Vaalidation for Genre Classification

| Class | SVM Class Level Accuracy | KNN Class Level Accurac | RF Class Level Accuracy |
|---|---|---|---|
| classic pop and rock | 26.87% | 4.23% | 48.72% |
| punk | 61.57% | 56.50% | 73.58% |
| folk | 19.03% | 21.42% | 53.96% |
| pop | 10.97% | 24.74% | 59.39% |
| dance and electronica | 35.82% | 35.49% | 20.49% |
| metal | 9.90% | 35.97% | 47.16% |
| jazz and blues | 70.51% | 56.16% | 66.99% |
| classical | 36.52% | 31.73% | 18.43% |
| hip-hop | 42.41% | 34.56% | 49.97% |
| soul and reggae | 10.06% | 30.08% | 37.60% |
| | | | |
| Overall | 32.37% | 33.09% | 47.63% |

Fig. 11. Results of 5 fold Stratified Cross Vaalidation for Genre Classification

| Model-CV | Accuracy |
|---|---|
| SVM-2 | 57.31% |
| SVM-4 | 57.74% |
| SVM-5 | 57.46% |
| KNN-2 | 55.14% |
| KNN-4 | 54.57% |
| KNN-5 | 54.72% |
| RF-2 | 88.48% |
| RF-4 | 95.50% |
| RF-5 | 95.97% |

Fig. 12. Stratified Cross Validation result comparison for Popularity

## V. Conclusion and Future Work

In our study, we proposed a new approach to classify music genres and song popularity. We performed K fold Stratified Cross validation, Random Forest Feature selection and SMOTE Sampling for KNN, SVM and Random Forests for Genre classification for values of K as 2 and 5. Random Forest gave the best performance among all three classifiers with accuracies being 45.76% and 47.63% respectively. For popularity classification, we got accuracies for Random Forest as 88.48%, 95.50% and 95.97% respectively for 2,4 & 5 fold stratified cross validation.The result signified that Random forest performed the best among all the classifiers. In future, we shall focus on combining both genre classification and popularity classification and apply them to arrange songs in groups of Genre and sort them by their popularity.

## References


[1] George Tzanetakis, Student Member, IEEE, and Perry Cook, Member, IEEE *"Musical Genre Classification of Audio Signals"*, IEEE TRANSACTIONS ON SPEECH AND AUDIO PROCESSING, VOL. 10, NO. 5, JULY 2002

[2] *"URL : http://millionsongdataset.com/pages/getting-dataset/"*

[3] *"URL : https://www.sciencedirect.com/topics/engineering/feature-extraction"*

[4] Salganik, Matthew J., Peter Sheridan Dodds, and Duncan J. Watts, *"Experimental study of inequality and unpredictability in an artificial cultural market"* (2006)

[5] James Pham, Edric Kyauk and Edwin Park, *"Predicting Song Popularity"* (2015) *"URL : http://cs229.stanford.edu/proj2015/140report.pdf "*

[6] *"URL : https://www.saedsayad.com/k_nearest_neighbors.htm "*

[7] *"URL : https://medium.com/machine-learning-101/chapter-2-svm-support-vector-machine-theory-f0812effc72 "*

[8] *"URL : https://blogs.oracle.com/datascience/random-forests%2c-decision-trees%2c-and-ensemble-methods-explained "*

[9] D. Kimber and L. Wilcox, *"Acoustic segmentation for audio browsers"*, in Proc. Interface Conf., Sydney, Australia, July 1996

[10] Thomas Astad Sve, *"Songs Analysis"*, University of California, San Diego & Norwegian University of Science and Technology


## Appendix

1) **Libraries need for the code are as follows:**
    - pip install os
    - pip install glob
    - pip install datetime
    - pip install numpy
    - pip install pylab
    - pip install pandas
    - pip install seaborn
    - pip install pathlib
    - pip install sklearn
    - pip install matplotlib
    - pip install imblearn
2) Datasets (msd_genre_dataset.csv and msd_popularity_dataset.csv) should be in same folder as of the python files (genre_prediction.py and popularity_prediction.py)
3) The program will create a folder Names "Result For Genre Classification {TimeStamp}" in genre_prediction.py and "Result For Popularity Classification {TimeStamp}" in popularity_prediction.py
4) In the generated result folder there will be csv named "Genre_Result_With_CV_FS_Sampling.csv" for genre and "Popularity_Result_With_CV_FS.csv" for popularity containing all the class level accuracies with three different models (KNN, SVM and Random Forest)
5) Execution times are as follows:
    - Genre classification with 2 fold stratified cross validation, random forest feature selection and SMOTE Sampling on all three models (KNN, SVM and RF): 25 mins (Approx)
    - Popularity classification with 2,4 and 5 fold stratified cross validation, random forest feature selection on all three models (KNN, SVM and RF): 7 mins (Approx)