Pune Institute of Computer Technology
Dhankawadi, Pune

A PROJECT REPORT ON

MUSIC GENRE CLASSIFICATION

SUBMITTED BY

Sahil S. Patil	**41250**
Pralhad A. Kulkarni	**41254**
Srushti S. Raybhoge	**41258**

Class : BE 2
Batch : R2

Under the guidance of
Prof. V. V. Bagade

DEPARTMENT OF COMPUTER ENGINEERING
Academic Year 2020-21

DEPARTMENT OF COMPUTER ENGINEERING

Pune Institute of Computer Technology

Dhankawadi, Pune-43

CERTIFICATE

This is to certify that the project report entitled

"Music Genre Classification"

Submitted By

Sahil S. Patil	41250
Pralhad A. Kulkarni	41254
Srushti S. Raybhoge	41258

have satisfactorily completed a project report under the guidance of Prof. V. V. Bagade towards the partial fulfillment of fourth year Computer Engineering Semester I, Academic Year 2020-21 of Savitribai Phule Pune University.

Prof. V. V. Bagade
Project Guide

Prof. M.S.Takalikar
Head
Department of Computer Engineering

Place:
Date:

CONTENT

1. Problem Statement…… 4
2. Abstract…….. 4
3. Introduction…… 4
4. Requirements……...…….. 4
5. Algorithms used…… 4
6. Dataset…… 4
7. Theory……..... 5
   - 7.1. Classification…….. 5
   - 7.2. Types of classification models……... 5
   - 7.3. SVM……... 5
   - 7.4. Decision tree…….. 6
   - 7.5. Random Forest……... 6
8. Architecture…… 8
9. Result…… 8
   - 9.1. SVM……... 8
   - 9.2. Decision…… 9
   - 9.3. Random Forest……9
10. Analysis…….. 10
11. Conclusion…….. 10

1. Problem Statement :

Music Genre Classification. Classify the given GTZAN dataset into their appropriate classes i.e blues, classical, country, disco, hiphop, jazz, metal, pop, reggae, rock.

2. Abstract :

The main motive of this project is to classify the given songs into its song genre .For the same we have to apply various machine learning models . We can further compare the accuracies of these models over the gtzan dataset and select the most appropriate classification model to do the job .

3. Introduction :

Classification problem is a type of problem in which the given data can be identified into different classes. A classification model can identify the class of the object among other classes.A similar classification can be done on songs as to which genre of song is given to the input. This model can be used to classify the song or even arrange the song in a playlist.We will be using different types of ML model to do the same , thus also comparing the accuracies of each applied model.

4. Requirements :

Python libraries, jupyter notebook , GTZAN dataset

5. Algorithms Used :

- SVM (Support Vector Machine)
- Decision Tree
- Random Forest

6. Dataset :

The GTZAN genre collection dataset was collected in 2000-2001. It consists of 1000 audio files each having 30 seconds duration. There are 10 classes (10 music genres) each containing 100 audio tracks. Each track is in .wav format. It contains audio files of the following 10 genres:

- Blues
- Classical
- Country
- Disco
- Hiphop
- Jazz
- Metal

- Pop
- Reggae
- Rock

7. Theory :-

7.1. Classification : -

Classification is the process of predicting the class of given data points. Classes are sometimes called targets/ labels or categories. Classification predictive modeling is the task of approximating a mapping function (f) from input variables (X) to discrete output variables (y).

For example, spam detection in email service providers can be identified as a classification problem. This is s binary classification since there are only 2 classes as spam and not spam. A classifier utilizes some training data to understand how given input variables relate to the class. In this case, known spam and non-spam emails have to be used as the training data. When the classifier is trained accurately, it can be used to detect an unknown email.

These types of problems often have predefined classes present in them.The type of classification which we are dealing with is a multi-class classification problem.

7.2. Types of classification models : -

In machine learning and statistics, classification is a supervised learning approach in which the computer program learns from the input data and then uses this learning to classify new observations. This data set may simply be bi-class (like identifying whether the person is male or female or that the mail is spam or non-spam) or it may be multi-class. Some practical examples of classification problems are: speech recognition, handwriting recognition, biometric identification, document classification etc.

Some types of Classifications are:-

1. Linear Classifiers: Logistic Regression, Naive Bayes Classifier
2. Nearest Neighbor
3. Support Vector Machines
4. Decision Trees
5. Boosted Trees
6. Random Forest
7. Neural Networks

We have used SVM, decision tree, random forest in our mini project.

7.3. SVM (Support Vector Machine) :

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning.

The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.

SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence the algorithm is termed as Support Vector Machine. Consider the below diagram in which there are two different categories that are classified using a decision boundary or hyperplane.

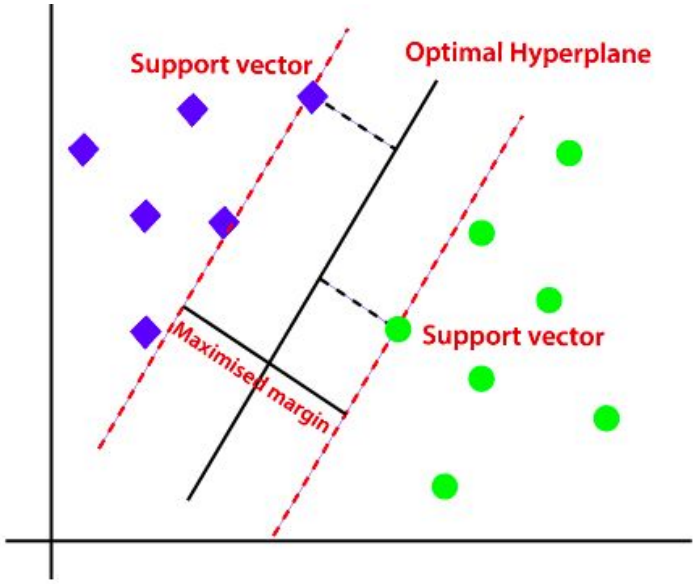

Fig1. *SVM classifies the given input into classes*

7.4. Decision Tree :

Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome.

In a Decision tree, there are two nodes, which are the Decision Node and Leaf Node . Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches.
The decisions or the test are performed on the basis of features of the given dataset.

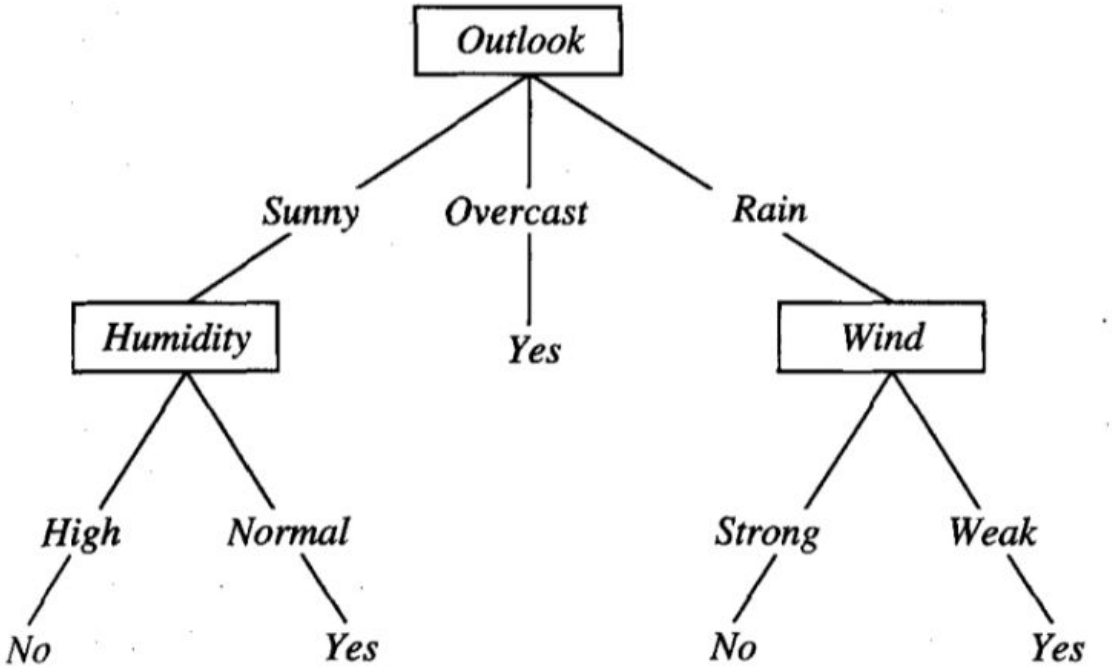

Fig 2. Example of decision Tree over prediction of weather

7.5. Random Forest :

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.

As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

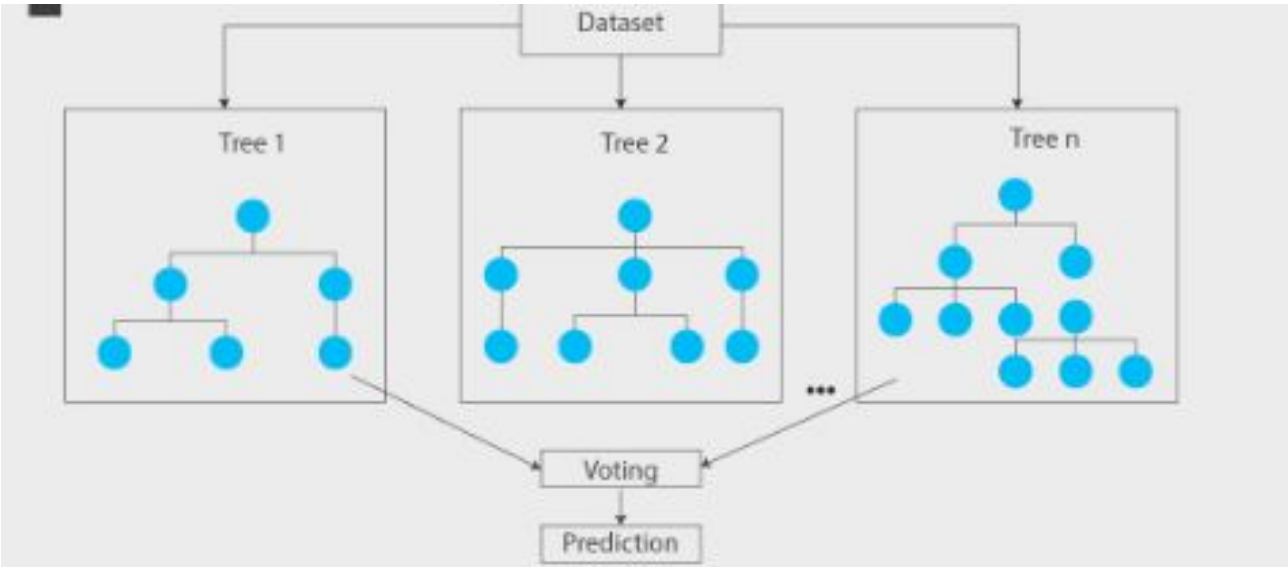

Fig 3. Basic Structure Of Decision Tree

8. Architecture :

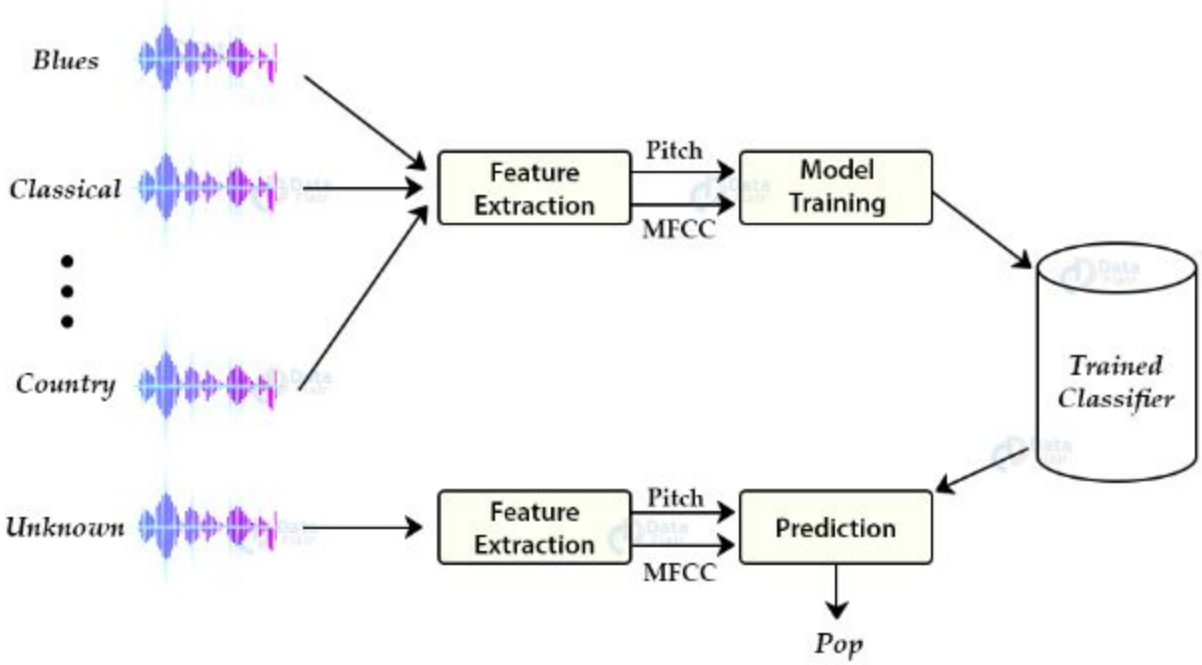

Fig 4. Architecture for music genre classification

9. Results :

9.1 SVM (Support Vector Machine

```
In [32]: preds = grid_svm.predict(X_test)
         print(preds)
```

```
[4 7 0 8 8 6 5 4 4 9 4 2 1 2 9 1 1 2 5 5 7 5 7 0 7 0 8 1 2 9 9 8 8 0 9 6 5
 9 7 3 5 9 1 7 0 6 7 4 3 8 6 4 6 2 3 9 6 9 2 9 1 3 9 1 8 9 2 1 4 6 0 1 2 9
 3 8 5 1 4 6 0 2 7 5 1 0 3 0 7 2 0 8 4 6 7 7 8 5 2 1 3 3 6 2 5 5 9 4 4 0 5
 5 8 7 6 6 8 8 3 0 9 3 1 6 6 0 4 9 4 8 8 1 2 5 9 9 3 7 8 3 2 2 7 1 5 3 2 7
 4 6 2 3 2 4 7 3 4 2 6 7 3 4 9 1 0 2 2 9 1 7 6 3 0 3 9 4 3 4 9 0 7 5 8 5 8
 3 0 2 8 8 0 1 5 2 0 7 1 9 6 7 1 9 7 4 8 4 9 4 4 5 3 5 6 5 8 0 5 3 9 8 9 5
 4 1 6 8 2 3 0 2 2 3 0 0 5 0 7 0 7 1 8 4 0 1 3 1 1 2 8 5]
```

```
In [34]: print(y_test)
         print("Score on validation set (accuracy) = {:.4f}".format(grid_svm.best_score_))
         print("Score on test set (accuracy) = {:.4f}".format(accuracy_score(y_test, preds)))
```

```
[4 7 0 8 8 3 7 4 4 9 4 3 1 9 9 1 5 2 6 5 7 5 7 0 7 0 0 1 2 9 6 8 8 0 5 6 5
 9 7 3 6 9 1 7 9 6 7 4 3 8 6 4 6 2 1 6 6 4 2 9 9 3 9 8 8 7 2 1 4 1 0 1 2 9
 3 8 5 1 4 6 0 2 7 5 1 0 3 0 9 2 3 1 4 6 7 7 3 5 2 1 6 3 6 2 5 5 9 4 4 0 5
 5 8 7 6 6 8 3 3 9 5 1 0 8 6 0 4 9 4 8 8 1 2 9 6 9 3 7 9 3 2 2 8 1 5 3 2 7
 4 8 2 6 2 4 7 3 4 2 6 7 3 4 9 5 0 2 2 9 1 7 6 3 0 3 9 4 3 4 9 0 7 9 8 5 8
 3 0 2 8 7 0 1 5 2 0 7 1 8 6 7 6 9 7 4 1 4 1 4 4 5 3 5 6 5 8 0 5 3 8 8 9 5
 2 1 6 8 2 8 0 2 2 3 0 0 5 0 7 0 7 1 3 4 0 1 6 1 5 1 8 5]
Score on validation set (accuracy) = 0.7827
Score on test set (accuracy) = 0.8040
```

9.2 Decision Tree :-

```
In [28]: preds = grid_cart.predict(X_test)
         print(preds)
```

```
[7 5 0 8 8 5 2 4 4 5 4 3 3 2 9 9 5 2 9 4 7 9 2 3 7 0 0 8 2 4 1 8 1 0 7 8 5
 1 5 3 5 5 1 0 0 6 5 4 0 8 6 4 6 2 9 8 6 9 2 0 6 0 0 9 8 7 2 8 6 8 0 7 2 4
 3 8 5 1 4 6 1 2 7 5 8 3 3 0 1 2 7 8 7 6 7 7 8 5 2 1 6 3 6 2 5 9 7 4 5 0 5
 4 3 7 3 6 8 8 8 3 9 1 0 3 6 0 9 7 7 1 8 8 2 5 5 9 1 0 3 8 2 2 9 1 5 3 2 8
 4 6 5 1 2 4 4 3 4 2 6 7 3 4 5 1 1 2 2 6 6 7 6 9 3 6 9 4 3 4 9 0 8 5 9 5 8
 3 0 2 7 7 0 4 5 2 3 7 5 7 6 7 6 5 1 4 8 2 4 4 2 6 3 1 6 7 8 0 5 8 9 8 6 7
 5 9 6 8 2 8 0 2 2 6 0 0 5 0 7 0 1 6 8 4 0 1 8 1 8 2 8 5]
```

```
In [30]: print(y_test)
         print("Score on validation set (accuracy) = {:.4f}".format(grid_cart.best_score_))
         print("Score on test set (accuracy) = {:.4f}".format(accuracy_score(y_test, preds)))
```

```
[4 7 0 8 8 3 7 4 4 9 4 3 1 9 9 1 5 2 6 5 7 5 7 0 7 0 0 1 2 9 6 8 8 0 5 6 5
 9 7 3 6 9 1 7 9 6 7 4 3 8 6 4 6 2 1 6 6 4 2 9 9 3 9 8 8 7 2 1 4 1 0 1 2 9
 3 8 5 1 4 6 0 2 7 5 1 0 3 0 9 2 3 1 4 6 7 7 3 5 2 1 6 3 6 2 5 5 9 4 4 0 5
 5 8 7 6 6 8 3 3 9 5 1 0 8 6 0 4 9 4 8 8 1 2 9 6 9 3 7 9 3 2 2 8 1 5 3 2 7
 4 8 2 6 2 4 7 3 4 2 6 7 3 4 9 5 0 2 2 9 1 7 6 3 0 3 9 4 3 4 9 0 7 9 8 5 8
 3 0 2 8 7 0 1 5 2 0 7 1 8 6 7 6 9 7 4 1 4 1 4 4 5 3 5 6 5 8 0 5 3 8 8 9 5
 2 1 6 8 2 8 0 2 2 3 0 0 5 0 7 0 7 1 3 4 0 1 6 1 5 1 8 5]
Score on validation set (accuracy) = 0.5080
Score on test set (accuracy) = 0.5440
```

9.3 Random Forest :-

```
In [36]: #Random Forest
         preds = grid_rf.predict(X_test)
         print(preds)
```

```
[4 7 0 8 8 6 7 4 4 9 4 3 1 4 9 1 7 2 9 7 7 1 7 0 7 0 3 8 2 4 9 8 8 0 1 6 5
 7 5 3 5 9 1 9 0 6 7 4 3 8 6 4 6 2 1 8 6 5 2 0 6 3 1 5 8 9 2 1 8 6 0 1 2 9
 3 8 5 1 4 6 0 2 7 5 1 3 3 0 1 2 0 8 4 6 7 7 8 5 2 1 6 8 6 2 5 5 1 4 4 0 5
 5 8 7 6 6 8 8 6 0 9 3 0 6 6 0 4 9 4 8 8 1 2 8 8 9 3 7 8 8 2 2 1 1 5 3 2 8
 4 6 2 1 2 2 7 3 4 2 6 7 3 4 4 1 0 2 2 4 1 7 6 3 0 8 9 4 3 4 7 0 7 5 8 5 8
 3 0 2 8 7 0 9 5 2 0 7 4 4 6 7 6 4 1 4 6 4 9 4 4 5 3 5 6 5 8 0 5 3 1 8 9 5
 2 1 6 8 2 8 0 2 2 3 0 0 7 0 4 0 1 6 8 4 0 3 3 1 6 2 8 8]
```

```
In [37]: print(y_test)
         print("best score on validation set (accuracy) = {:.4f}".format(grid_rf.best_score_))
         print("best score on test set (accuracy) = {:.4f}".format(accuracy_score(y_test, preds)))
```

```
[4 7 0 8 8 3 7 4 4 9 4 3 1 9 9 1 5 2 6 5 7 5 7 0 7 0 0 1 2 9 6 8 8 0 5 6 5
 9 7 3 6 9 1 7 9 6 7 4 3 8 6 4 6 2 1 6 6 4 2 9 9 3 9 8 8 7 2 1 4 1 0 1 2 9
 3 8 5 1 4 6 0 2 7 5 1 0 3 0 9 2 3 1 4 6 7 7 3 5 2 1 6 3 6 2 5 5 9 4 4 0 5
 5 8 7 6 6 8 3 3 9 5 1 0 8 6 0 4 9 4 8 8 1 2 9 6 9 3 7 9 3 2 2 8 1 5 3 2 7
 4 8 2 6 2 4 7 3 4 2 6 7 3 4 9 5 0 2 2 9 1 7 6 3 0 3 9 4 3 4 9 0 7 9 8 5 8
 3 0 2 8 7 0 1 5 2 0 7 1 8 6 7 6 9 7 4 1 4 1 4 4 5 3 5 6 5 8 0 5 3 8 8 9 5
 2 1 6 8 2 8 0 2 2 3 0 0 5 0 7 0 7 1 3 4 0 1 6 1 5 1 8 5]
best score on validation set (accuracy) = 0.7133
best score on test set (accuracy) = 0.7160
```

10. Analysis : -

Methods	**Test Set (**accuracy**)**	**Validation Set (**accuracy**)**
SVM	80.40%	78.27%
Decision Tree	54.40%	50.80%
Random Forest	71.60%	71.33%

Table 1. Comparison of different methods

11. Conclusion : -

Thus we applied different machine learning models to classify the given music into different classes based on its genre . i.e Blue , classical , country , disco , hiphop ,jazz , metal , pop , reggae , rock . By comparing the accuracies of the model we get that in this particular scenario the SVM model gives the best performance.