Project Report On

Music Genre Classification Using Machine Learning & IBM Cloud

**INTRODUCTION**

**1.1 Overview**

Music has become the most favorable area nowadays especially in youth. Most of the people tend to listen music of certain genre such as classical, hip-hop or disco and want a user-friendly way to classify the music as per their preferences. Due to this “Music Genre Classification” came into picture. Music genre classification is a complex task in music information retrieval (MIR) due to selection and extraction of suitable features.

The Machine learning models have been shown to be capable of solving these kinds of real-life problems. Music genre classification can be implemented using various machine learning algorithms. In the proposed system, we are using a deep learning technique. I.e., K-Nearest Neighbours (KNN) for classifying the music in various genres. CNNs are used to solve image pattern recognition tasks. While analyzing music, acoustic feature extraction is the most crucial task. In proposed system, model is trained over GTZAN dataset.

**1.2 Purpose**

A music genre classifier is a software program that predicts the genre of a piece of music in audio format. These devices are used for tasks such as automatically tagging music for distributors such as Spotify and Billboard and determining appropriate background music for events.

Music Genre Classification or classification of music into different categories or genres is a concept that helps the masses to differentiate between 2 genres based on their composition or the beats they withhold. In recent times, music genre classification has become a very popular concept as more and more genres are emerging around the world.

From K-Pop to Jazz, music lovers rely on the technology of music genre classification and are able to listen to songs as per their preferences. While it takes only a click for a listener to switch from Jazz music to Rap, there is certainly much more beneath the surface that fuels our love for music.

**2.LITERATURE**

**2.1 Existing Problem**

* Deep learning models:

1.Convolutional Neural Networks (CNN).

2.Recurrent Neural Networks (RNN).

3.Boltzmann machine.

4.Autoencoders etc.

* Classification:

1.The K-Nearest Neighbours algorithm

2.Decision Tree

3.Support Vector Machines

4.Naive Bayes

* Regression:

1.Linear Regression

2.Lasso Regression

3.Ridge Regression

4.Support Vector Regression (SVR)

5.Ensemble Regression

* Clustering:

1.K means

2.K means++

3.K medoids

4.Agglomerative clustering

5.DBSCAN

Above are some of the existing approaches or methods to solve this problem of Music Genre Classification

The preferable method for solving Music Genre Classification problem is KNN-Algorithm which gives the best accuracy among all available methods or approaches.

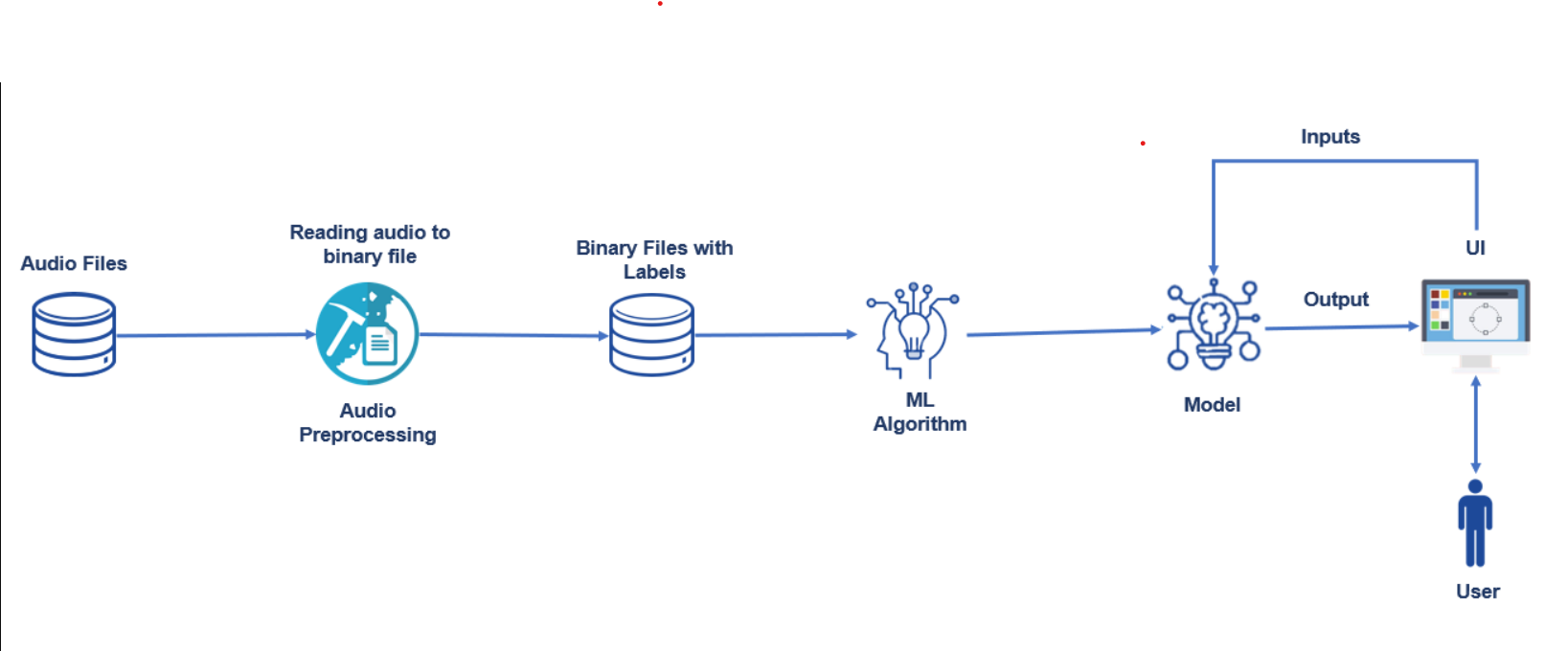
**2.2 Proposed Solution**

We have chosen **K-Nearest Neighbors** machine algorithm for classification of music genre and KNN is a machine learning algorithm used for regression, and classification. It is also known as the lazy learner algorithm. It simply uses a distance-based method find the K number of similar Neighbours to new data and the class in which the majority of Neighbours lies, it results in that class as an output.

From results we found that k NN classifier gave more exact results compared to support vector machine classifier. If the training data is bigger than number of features, k NN gives better outcomes than SVM. SVM can only identify limited set of patterns. KNN classifier is more powerful for the classification of music genre.

**3. THEORITICAL ANALYSIS**

3.1 Block Diagram:



**3.2 Hardware/Software Designing**

1.Software Requirements

1.Downloading of Anaconda Navigator

2.Downloading of python packages like

a. NumPy Package

b. Pandas

c. Keras

d. Tensor Flow

e. Matplotlib

f. scikit-learn

g. Flask

h. pyhton\_speech\_features

I. mfcc

j. from python\_speech\_features import mfcc

k. import sklearn.model\_selection

l. from sklearn.model\_selection import train\_test\_split

m. import scipy.io.wavfile as wav

n. import os

o. import pickle

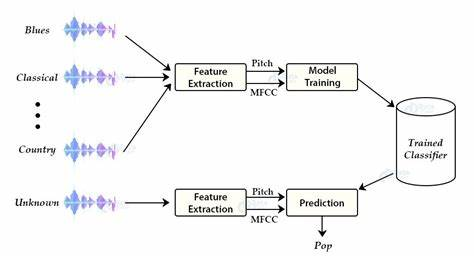
p. import operator

These are some of the software requirements required to implement the music genre classification project using KNN algorithm.

**4.EXPERIMENTAL INVESTIGATION**

**5.FLOWCHART**

* **Flow chart of the music genre classification:**



**6.RESULT**

**Fig:1 output of the GTZAN dataset:**

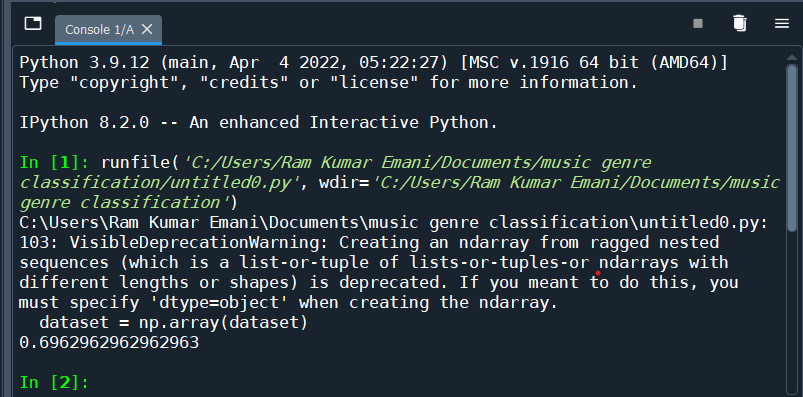


Fig 2: Web Application view :

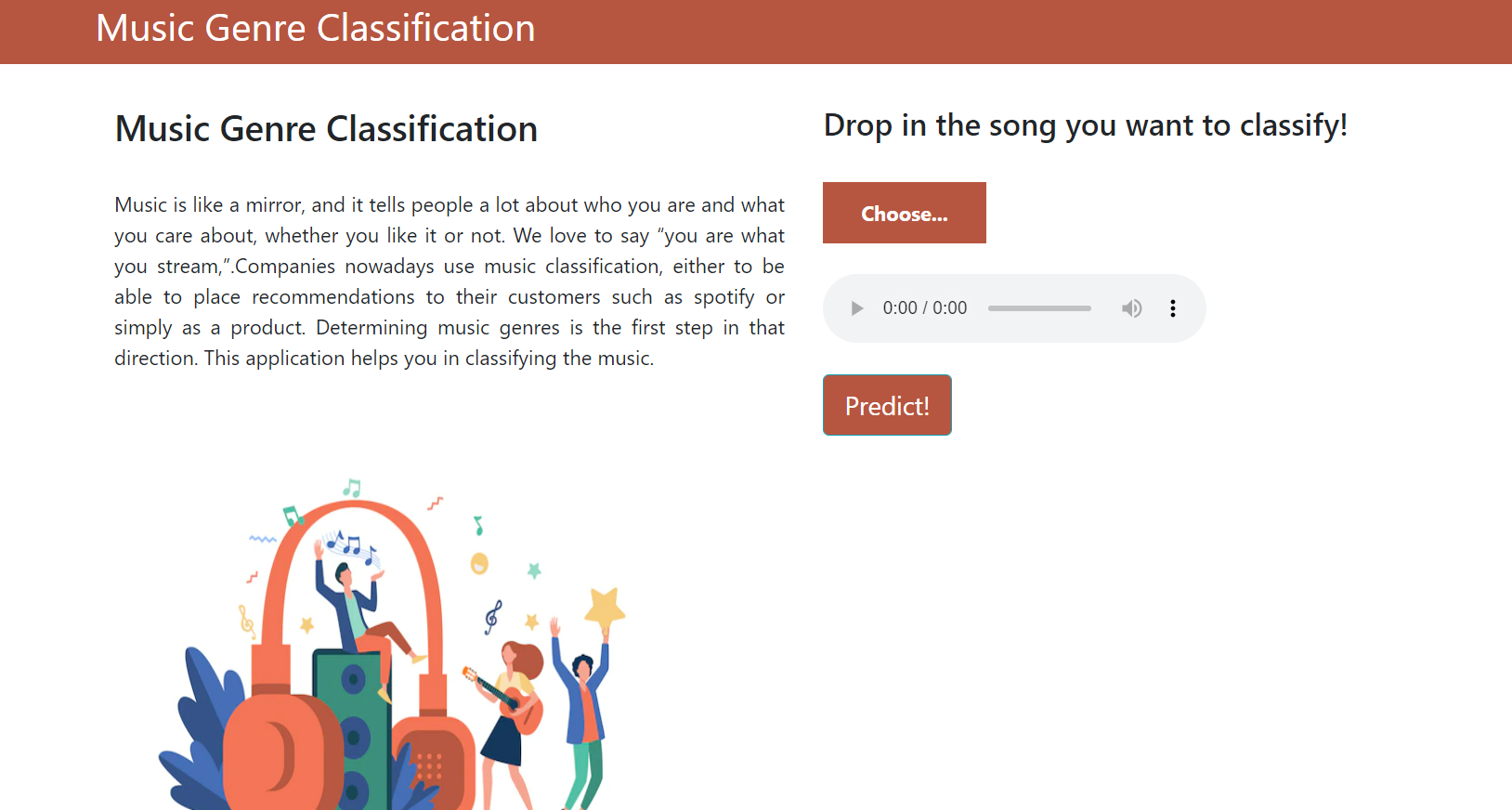


Fig 3: Predicting the music genre type:

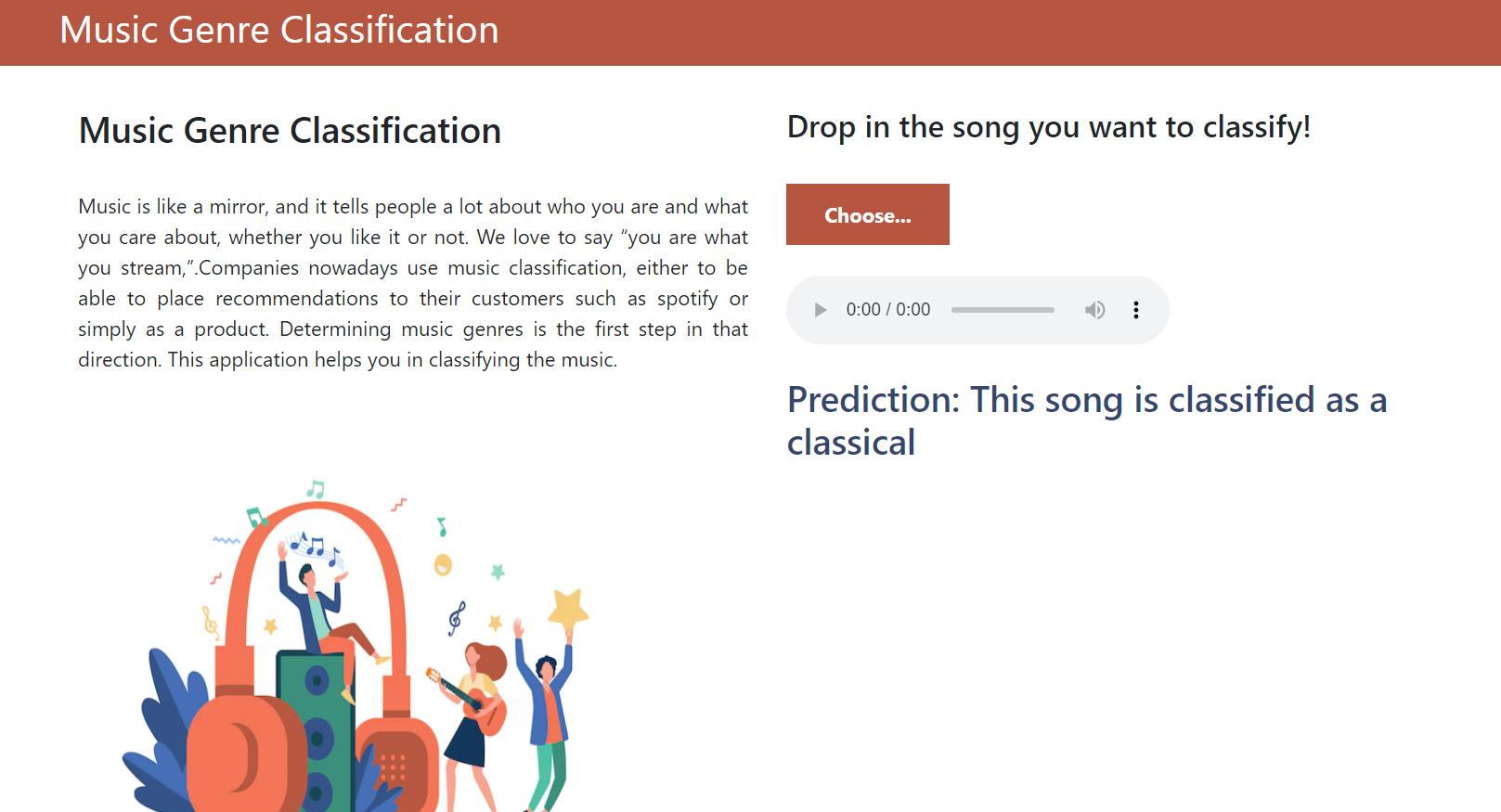
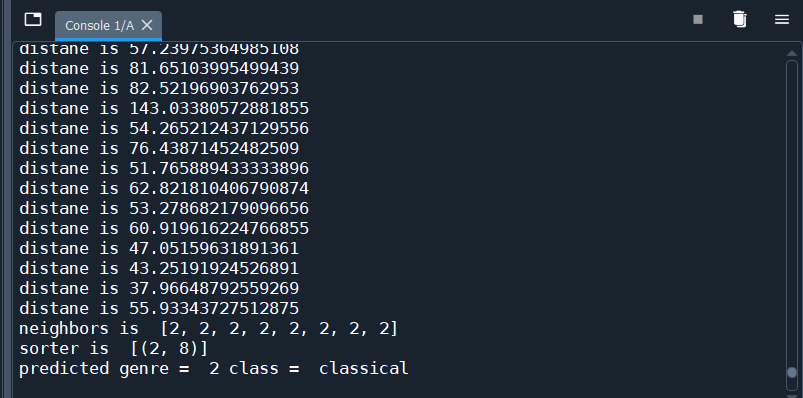


Fig 4: Distances of music beats in classical genre :



**7.ADVANTAGES AND DISADVANTAGES**

7.1 Advantages of KNN approach:

* It is easy to maintain.
* It is user-friendly.
* The system can easily detect music and classify the music genres from the audio file.
* k NN classifier gave more exact results compared to support vector machine classifier.

7.2 Disadvantages:

* It is costly.
* It is lazy.
* Requires full training data plus depends on the value of k and has the issue of dimensionality because of the distance.
* Has the issues of normalization of the data.
* It doesn't handle categorical variables very well.

**8 APPLICATIONS**

**1.Internet platforms like**

* Spotify
* Soundcloud
* Speech detection
* Separating background noise from primary speech.
* Sentimental Analysis.

The need of accurate meta-data requires for database management search/storage purposes.

**9.CONCLUSION**

Hence the Music genre classification using K-Nearest Neighbours was done with the assistance of Spyder and anaconda Jupiter notebook thus results are obtained. As shows below.

**10. FUTURE SCOPE**

Here, we have implemented the k-nearest neighbor (KNN) algorithm for classification. The disadvantages are it is costly and lazy and doesn’t handle the categorical data. We have to work on the classification for music genre classification to be not lazy and less costly and proper handling of the categorical data with best accuracy.

**11.BIBLIOGRAPHY**

**1.** [**https://nevonprojects.com/music-genres-classification-using-knn-system/**](https://nevonprojects.com/music-genres-classification-using-knn-system/)

**2.** [**https://www.kaggle.com/code/rxsraghavagrawal/music-genre-classification-using-knn-begineers/notebook**](https://www.kaggle.com/code/rxsraghavagrawal/music-genre-classification-using-knn-begineers/notebook)

**3.** [**https://github.com/HetGalia/Music-Genre-Classification-using-KNN**](https://github.com/HetGalia/Music-Genre-Classification-using-KNN)

**APPENDIX:**

**Source code:**

* **Model Building:**

**from \_\_future\_\_ import division, print\_function**

**import flask**

**import os**

**import numpy as np**

**from flask import Flask, request, render\_template**

**from werkzeug.utils import secure\_filename**

**from python\_speech\_features import mfcc**

**import scipy.io.wavfile as wav**

**import pickle**

**import operator**

**app = Flask(\_\_name\_\_)**

**dataset = []**

**def loadDataset(filename):**

**with open("my.dat" , 'rb') as f:**

**while True:**

**try:**

**dataset.append(pickle.load(f))**

**except EOFError:**

**f.close()**

**break**

**loadDataset("my-done.dat")**

**def distance(instance1 , instance2 , k ):**

**distance =0**

**mm1 = instance1[0]**

**cm1 = instance1[1]**

**mm2 = instance2[0]**

**cm2 = instance2[1]**

**#Method to calculate distance between two instances.**

**distance = np.trace(np.dot(np.linalg.inv(cm2), cm1))**

**distance+=(np.dot(np.dot((mm2-mm1).transpose() , np.linalg.inv(cm2)) , mm2-mm1 ))**

**distance+= np.log(np.linalg.det(cm2)) - np.log(np.linalg.det(cm1))**

**distance-= k**

**print("distane is",distance)**

**return distance**

**def getNeighbors(trainingSet , instance , k):**

**distances =[]**

**for x in range (len(trainingSet)):**

**dist = distance(trainingSet[x], instance, k )+ distance(instance, trainingSet[x], k)**

**distances.append((trainingSet[x][2], dist))**

**distances.sort(key=operator.itemgetter(1))**

**neighbors = []**

**for x in range(k):**

**neighbors.append(distances[x][0])**

**print("neighbors is ",neighbors)**

**return neighbors**

**def nearestClass(neighbors):**

**classVote ={}**

**for x in range(len(neighbors)):**

**response = neighbors[x]**

**if response in classVote:**

**classVote[response]+=1**

**else:**

**classVote[response]=1**

**sorter = sorted(classVote.items(), key = operator.itemgetter(1), reverse=True)**

**print("sorter is ",sorter)**

**return sorter[0][0]**

**print('Model loaded. Check http://127.0.0.1:5000/')**

**@app.route('/', methods=['GET'])**

**def index():**

**# Main page**

**return render\_template('music.html')**

**@app.route('/predict', methods=['GET', 'POST'])**

**def upload():**

**if request.method == 'POST':**

**# Get the file from post request**

**f = request.files['image']**

**# Save the file to ./uploads**

**basepath = r"C:/Users/Ram Kumar Emani/Documents/music genre classification/Flask"**

**file\_path = os.path.join(basepath, 'uploads', secure\_filename(f.filename))**

**f.save(file\_path)**

**print(file\_path)**

**i=1**

**results = {1: 'blues', 2: 'classical', 3: 'country', 4: 'disco', 5: 'hiphop',**

**6: 'jazz', 7: 'metal', 8: 'pop', 9: 'reggae', 10: 'rock'}**

**(rate,sig)=wav.read(file\_path)**

**print(rate,sig)**

**mfcc\_feat=mfcc(sig,rate,winlen=0.020,appendEnergy=False)**

**covariance = np.cov(np.matrix.transpose(mfcc\_feat))**

**mean\_matrix = mfcc\_feat.mean(0)**

**feature=(mean\_matrix,covariance,0)**

**pred=nearestClass(getNeighbors(dataset ,feature , 8))**

**print("predicted genre = ",pred,"class = ",results[pred])**

**return "This song is classified as a "+str(results[pred])**

**if \_\_name\_\_ == '\_\_main\_\_':**

**app.run(threaded = False)**