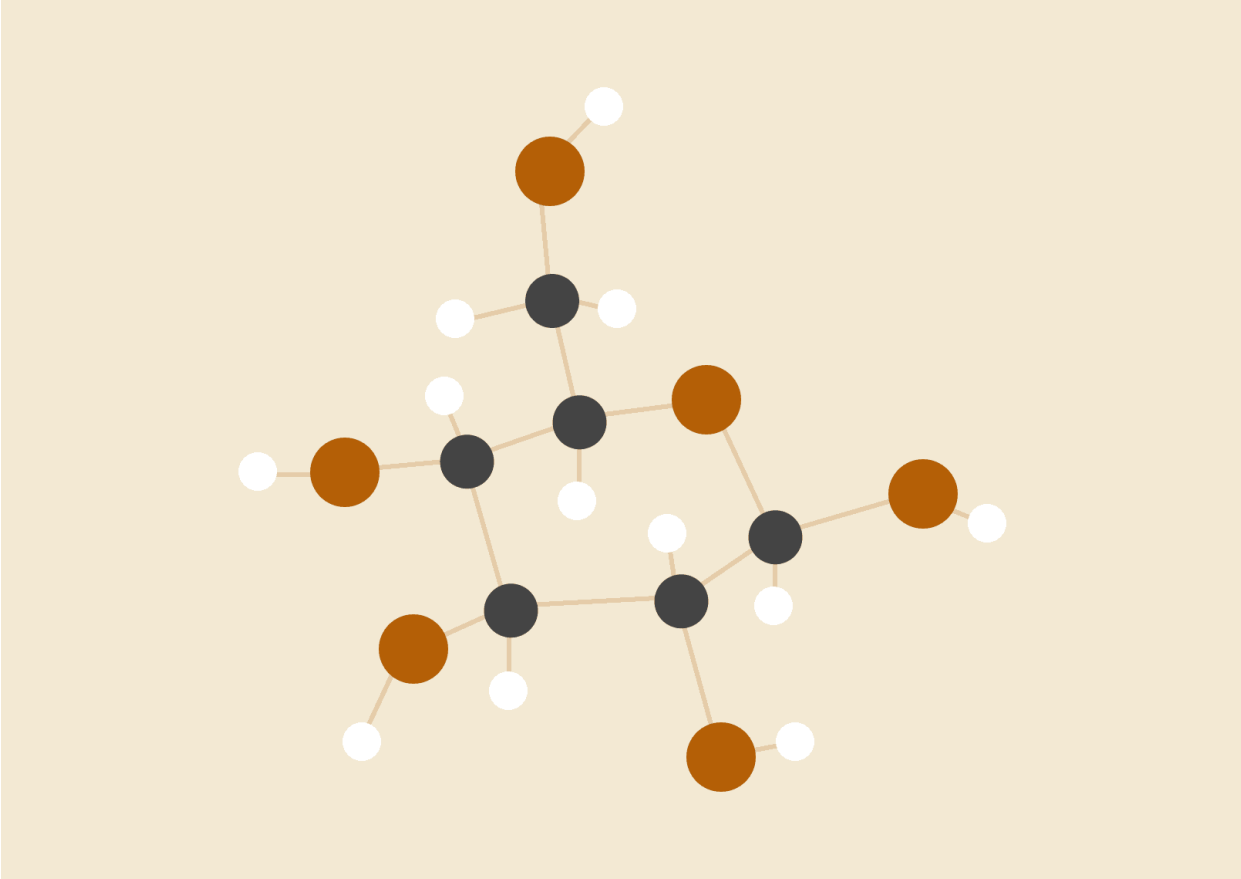
MUSIC GENRE CLASSIFICATION

CS 354 - Report



Bhuvana Sree Peddamallu (190001046)

Pooja Sree Panapakala (190001043)

Likhitha Kyatham (190001032)

PROBLEM DEFINITION

In this problem, we will classify different musical genres from audio files.It is the basic step for recommendation. The idea behind this project is handling sound files in python and run machine learning algorithms. If one has to manually classify songs, one has to listen to all the songs and then select genres which are difficult and time consuming.

ANALYSIS AND DESIGN OF THE PROBLEM

We used two algorithms to solve the problem. They are:

1. KNN - K-nearest neighbor.

   K-nearest neighbors algorithm is a type of supervised ML algorithm which can be used for both classification as well as regression predictive problems. It uses ‘feature similarity’ to predict the values of new data points which further means that the new data point will be assigned a value based on how closely it matches the points in the training set.

2. CNN - Convolutional neural network.

   Convolutional neural network is a multi-layered feed-forward neural network, made by assembling many unseen layers on top of each other in a particular order. Technically, deep learning CNN models to train and test, each input image will pass it through a series of convolution layers with filters (Kernals), Pooling, fully connected layers (FC) and apply Softmax function to classify an object with probabilistic values between 0 and 1.

a. Data Collection/ Preprocessing

In this project, the dataset we used is the GTZAN genre classification dataset which consists of 1,000 audio tracks, each 30 seconds long. It contains 10 genres, each represented by 100 tracks.

Our dataset contains 10 genres :-

1. Blues
2. Classical
3. Country
4. Hiphop
5. Disco
6. Jazz
7. Metal
8. Pop
9. Reggae
10. Rock

We can't take the raw audio signal as input to our model because there will be a lot of noise in the audio signal. Extracting features from the audio signal and using it as input to the base model will produce much better performance than directly considering raw audio signal as input. We used Mel-frequency Cepstral Coefficients (MFCC) which is a widely used technique for extracting the features from the audio signal.

In our data_extract.py file, we used librosa library to use mfcc for extracting features from our audio file. We looped through all the audio files, extracted their mfcc features and stored them in a json file along with it's corresponding genre and label. We extracted 13 features from each audio file.

b. Study and Understanding of Algorithm

We are using a sequential CNN model. We used three convolution layers with activation function relu. The output from these layers is flattened and fed into dense layer with

dropout probability 0.3. We then added an output layer with activation function softmax.

We have used Adam optimizer.

The datasets are divided into three sets train test validation in prepare dataset function.

c. Study of Performance Measurement Criteria

For KNN :-

If we change the hyper parameter - no : of nearest neighbors taken

Accuracy is highest when the number of neighbors is 5.

No : of Nearest Neighbor Taken	Accuracy
2	64.246
5	68.907
10	65.909
20	64.306

For CNN :-

If we change the hyperparameter - no : of convolutional layers

Accuracy is highest when layers=3, dropout = 0.3, learning rate = 0.00001,epochs = 30.

No : of Convolutional Layers	Accuracy

2	70.35
3	71.06

If we change the hyperparameter - drop out

Accuracy is highest when layer=3, dropout = 0.25, learning rate = 0.00001,epochs = 30.

Drop out	Accuracy
0.3	71.06
0.25	72.94
0.35	71.74

If we change the hyperparameter - learning rate

Accuracy is highest when layer=3, dropout = 0.3, learning rate = 0.00005,epochs = 30.

Learning Rate	Accuracy
0.00001	71.06
0.00005	73.79

If we change the hyperparameter - epochs

Accuracy is highest when layer=3, dropout = 0.3, learning rate = 0.00001,epochs = 40.

Epochs	Accuracy

30	71.06
25	69.50
35	71.74
40	74.62

RESULTS

Accuracy for test data is 71.06

The model predicts the given test sets correctly