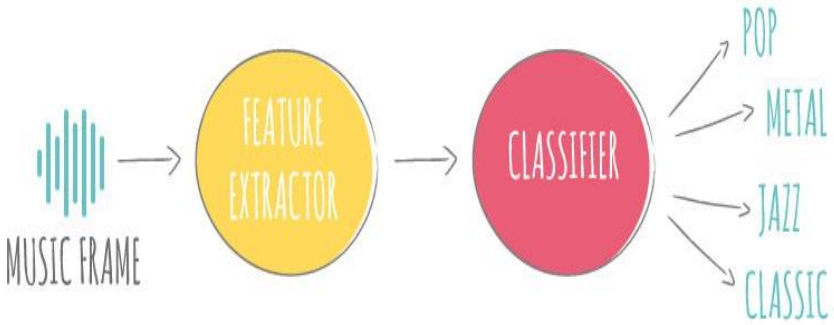
MUSIC GENRE CLASSIFICATION
Final Project Report

Manish Kumar [2018047]

INDRAPRASTHA INSTITUTE *of*
INFORMATION TECHNOLOGY
DELHI

PROBLEM STATEMENT

Music forms a very crucial part of our lives. Genre classification is an essential task with many real-world applications. Since the quantity by which music is getting released daily reaches the mountain peaks, the need for a proper and accurate genre classification rises in proportion. In today's era, internet services have a massive amount of multimedia exchanges and browsing. So the need for an effective way to organise and categorise these data rises in proportion. Searching a query song in a database containing millions of songs can become a cumbersome and time-consuming task. Categorising songs into different genres can reduce the search space for these query songs. The main objective of our project is to come up with an effective and accurate machine learning model that can automatically classify the music based on the genre.



Dataset And Evaluation metrics

- The dataset GTZAN used for building music genre classification has been taken from Kaggle . It consists of 1000 samples of songs.The dataset has 10 genre classes:

  1. Reggae, 2. Jazz, 3. Disco, 4. Rock , 5. Metal , 6. Pop , 7. Country, 8. Blues, 9.Classical, 10. Hiphop.Each of the 10 genre classes have 100 examples of songs. Thus the dataset is balanced.

- We used accuracy as a evaluation metric to analyse the performance of different model as the problem was a classification task and the data set was balanced as shown in the fig1. and our main way of visualizing the performance of our best model is through the confusion matrices.

- We have used Grid Search method along with k- fold cross validation resampling technique to choose the best parameter from given set of parameters. The chosen parameters gives best performance on the dev set and hence these parameters were used to finally estimate the genre of test data.

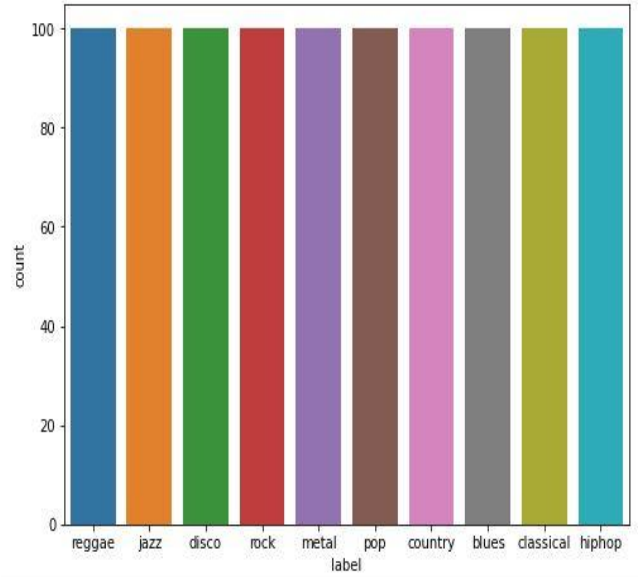


Fig1: Distribution of classes in dataset

APPROACHES USED

Since the problem statement involved classification task to classify music into 10 genre.

So we explored several linear and non linear classifiers.

1. We started with simple Naive bayes and logistic regression and then went through several other classifiers like linear, poly kernel, rbf SVM , LDA , QDA , ANN ,KNN and Random forest, Neural network , CNN and voting classifiers.

2. To start with the classification task a total of 63 features were extracted and then most important features were selected using correlation between the features and variance among the features . If two features had correlation more than 0.8 then one of them which had more correlation with the final label was kept and other was removed and the features having very low variance were removed.

3. Several outliers were detected by observing mean,max and 75th percentile value and using percentile capping method we were able to remove the outliers.

4. For resampling technique we decided to go with cross validation method mainly because it would help us make predictions on all of our data without missing anything.

5. For hyperparameter tuning we decided to go with simple grid search technique.We also tried random search and it also gave the same results.

APPROACHES USED

	spectral_bandwidth_var
count	1000.000000
mean	137110.238662
std	96478.361350
min	10787.185064
25%	67376.554428
50%	111977.548036
75%	182371.576801
max	694784.011540

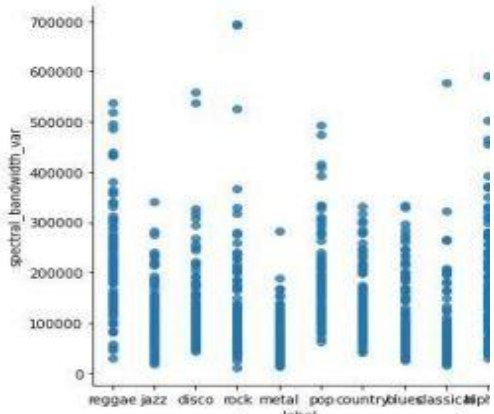
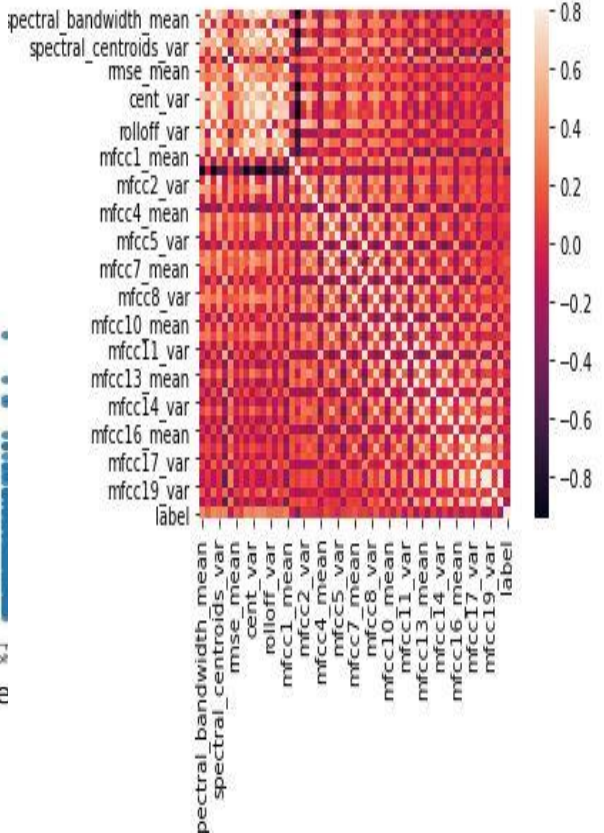


Fig2: Example showing outlie
dataset



spectral_bandwidth_mean	2.770090e+05
spectral_bandwidth_var	9.303696e+09
spectral_centroids_mean	5.125996e+05
spectral_centroids_var	1.607204e+11
tempo	7.988636e+02
...	...
mfcc20_var	2.045618e+03
harm_mean	2.835673e-06
harm_var	1.357998e-04
percp_mean	1.170688e-06
percp_var	4.225423e-05

label	1.000000
spec_bw_mean	0.388800
spectral_bandwidth_mean	0.388800
rolloff_mean	0.385213
spectral_centroids_mean	0.378395
cent_mean	0.378395
chroma_stft_mean	0.364719
mfcc1_mean	0.340101
spectral_centroids_var	0.317572
cent_var	0.317572
rolloff_var	0.273637
zcr_var	0.272210
zcr_mean	0.268278
rmse_mean	0.213968
spectral_bandwidth_var	0.201332
mfcc9_mean	0.198137
mfcc7_mean	0.182958
mfcc12_mean	0.153867
mfcc4_var	0.143586
mfcc11_mean	0.140996
mfcc18_mean	0.119079

Results of State of Art Models

As per states of art model the best accuracy obtained is 91% by Sparse Representation-based Classification. Most of the research works reviewed have made use of models such as KNN, SVM and CNN.

Model	States of Art Model's Test Accuracy
Sparse Representation Classification	91%
SVM	90% for 4 Genres
CNN	82%

Results of our Test Models

Model	Test Accuracy
SVM	73.4%
KNN	72%
CNN	79%

SUMMARY OF MAIN RESULTS:

- The learning model we have explored so far and the corresponding accuracies obtained have been mentioned in Table 1.
  The highest accuracy obtained is by using Poly Kernel SVM model with accuracy as 73.4%.
- Gap between between our best model and state of the art model is about 20% where our poly kernel SVM gives about 73% accuracy the state of the art model is at 91%

Learning Model	Accuracy obtained
Poly Kernel SVM	73.4%
KNN	72%
NN	71%
Linear Kernel SVM	71%
Random Forest	72%
LDA	71%
RBF Kernrel SVM	72%%
Sigmoid Kernrel SVM	70%

Table 1. Results obtained

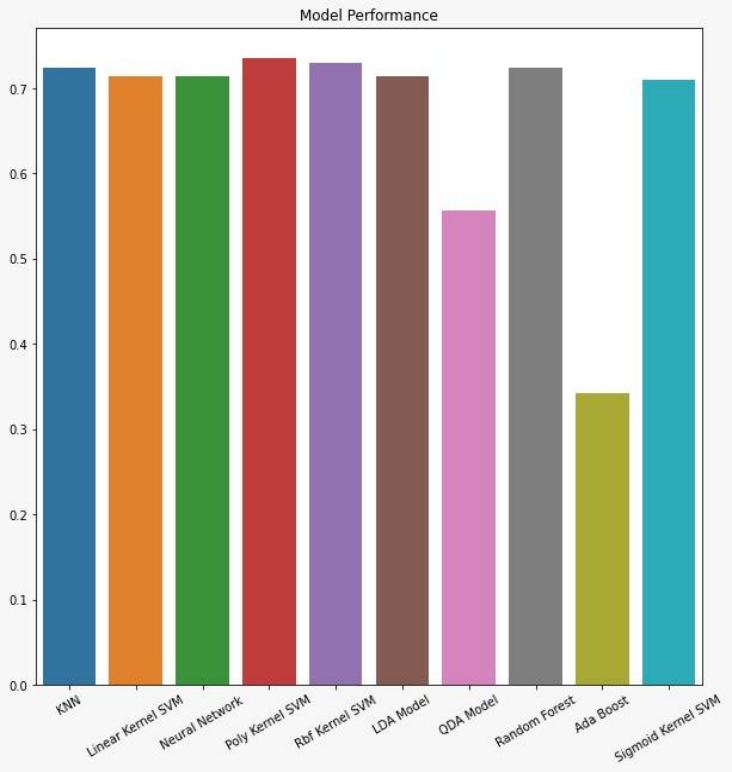
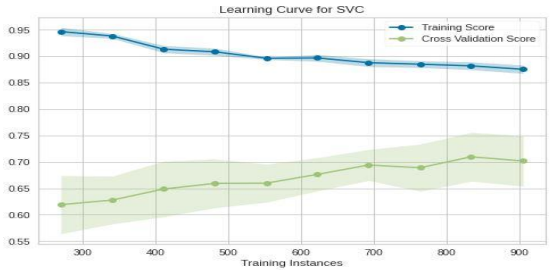
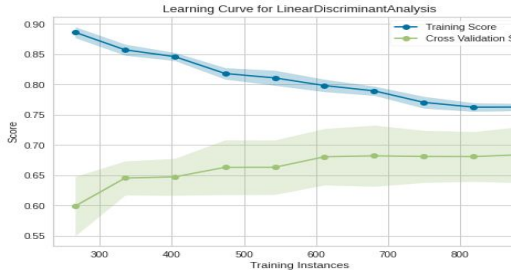


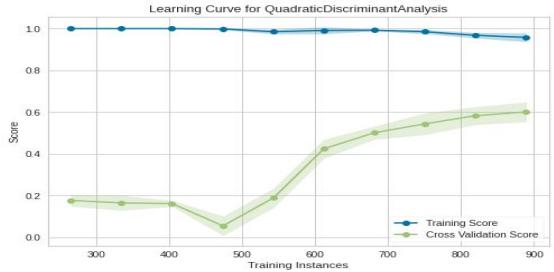
Fig4: Accuracies obtained using different models

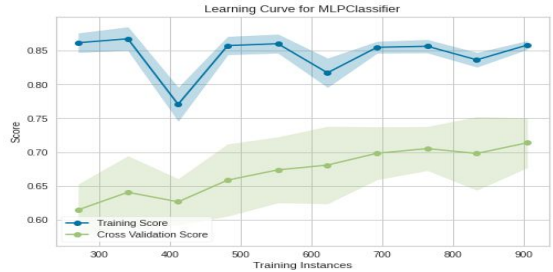
ANALYSIS OF RESULT

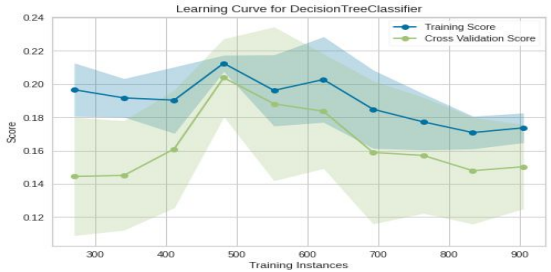
- Evidence of proper training

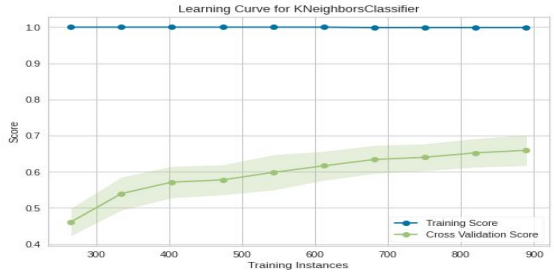












ANALYSIS OF RESULT

Error analysis:

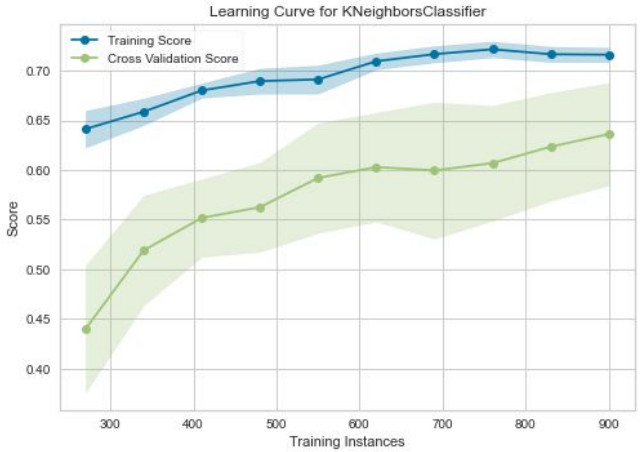
We checked the bias and variance of several models explored to analyse the overfitting and underfitting models.

We tried removing the overfitting by choosing appropriate feature Selection and outliers removal and dimensionality reduction using PCA .However, due to lack of proper data set we could not remove the overfitting completely . We realised that the data set GTZAN has various of the data samples being repeated, and the data set is very noisy and had limited number of samples.

We also observed that this is the dataset that is majorly used for the audio processing and other data sets are very huge.And creating own dataset would have been very time consuming .

So we tried do our best to remove overfitting by several possible techniques.

Also to increase our model accuracy we performed hyperparameter tuning using grid search CV method.



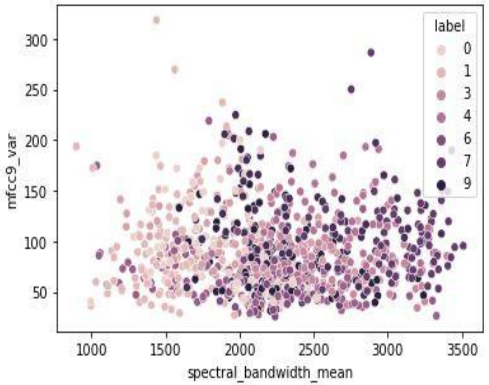
ANALYSIS OF RESULT:

Some conclusions drawn regarding the features:

The dataset is not linearly separable as we observed that the train accuracy obtained on SVM model was not 100% on the best hyperparameter evaluated but was 0.73422. This shows that the data is not linearly separable.The same has been illustrated by taking example of two features in fig.

Naive bayes assumption that features are independent . So the conclusion drawn for model not performing well is that features are dependent.

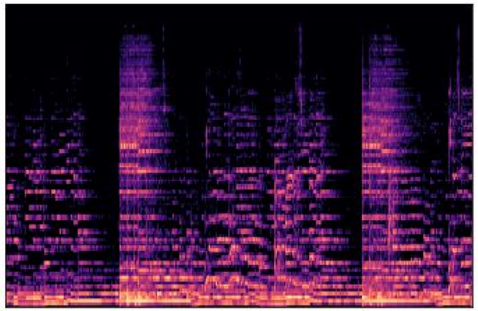
Fig 5: Train and test error analysis



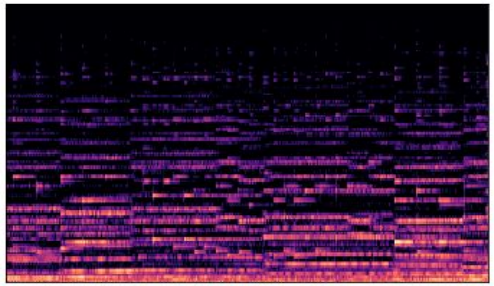
CNN Model

As CNN is best for image classification we tried to convert our audio classification problem to an image classification problem.

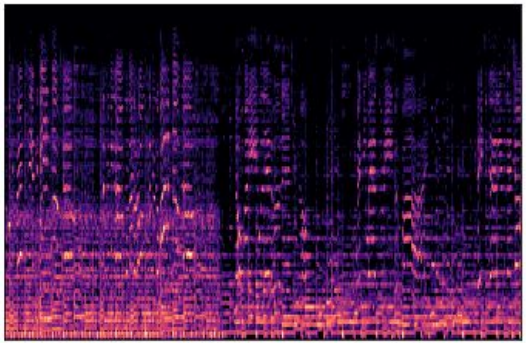
To do this we plotted mel-spectrogram of each audio file and used it as input for our CNN model and tried to classify it.



Rock

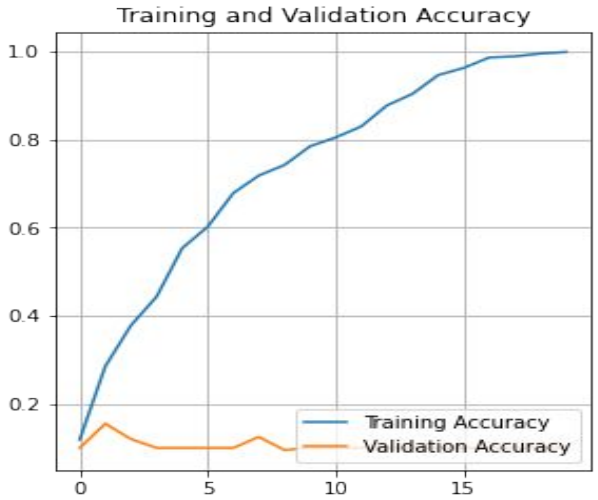


Classical



Metal

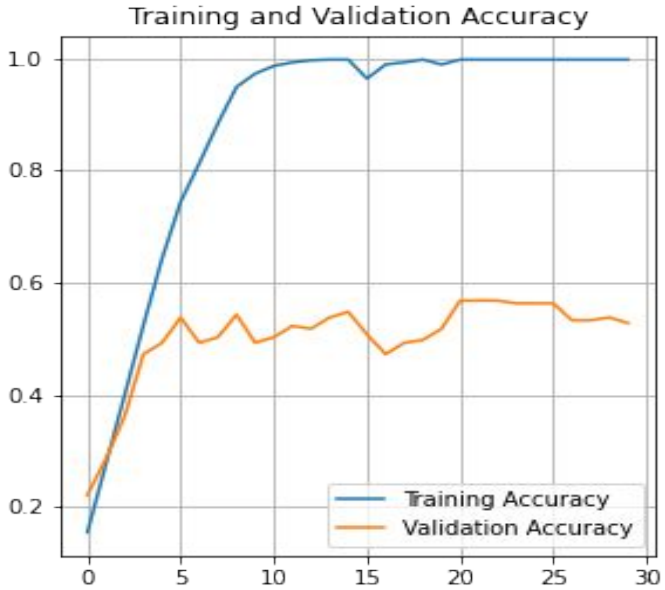
Results obtained on CNN



Accuracy obtained was 11% this was because we can clearly see our model overfitted.
Possible reason for overfitting-
Data was less there were only 100 images for each genre .
Model was too complex for that data and model easily covered the whole data spectrum which resulted in an overfit.

Trying a Less complex model

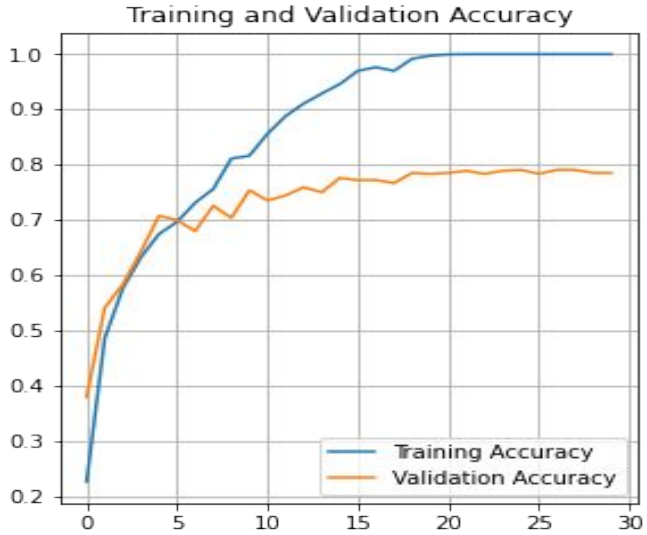
The previous CNN had 5 convolutional layers to reduce the model complexity we made a model of 3 convolutional layers

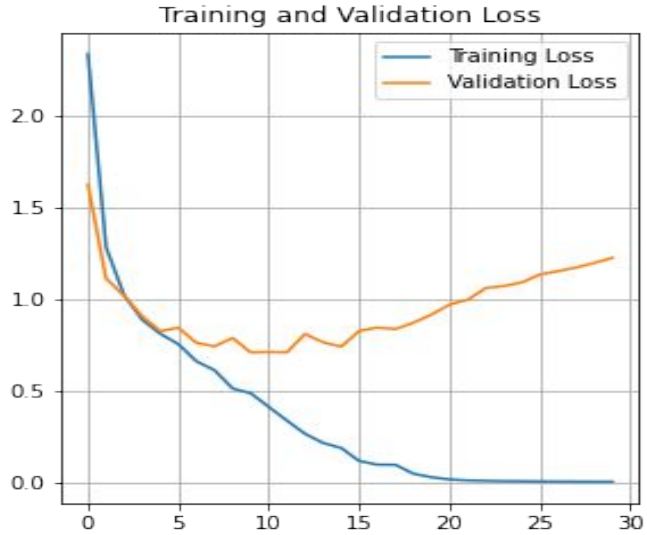


Accuracy we obtained was 56% still the model was overfit as CNN are in general complex models simpler models like SVM performed better as data was less.

Getting More Data

After doing a bit of research we found out that for genre classification only 5 sec audio is enough to classify it correctly . So we divided each audio in 3 audios of 10 sec and saved it mel-spectogram. So our dataset was increased from 1000 to 3000 images.





After training the CNN on this increased dataset the accuracy increased to 79%.

CONCLUSION AND LEARNING

- We learnt that a good prediction algorithm consists of three components : processing of data, choosing the right ML technique and hyperparameter tuning. All three steps are equally important to properly execute any of these step it is first important to do a thorough EDA of the dataset so we can choose the right learning technique.

- Dividing the data into training, testing and validation is important so that testing data act as completely unseen data and the hyperparameter tuning can be performed using the dev set.

- It is important to analyse the results and the error obtained so that further steps can be taken in right direction.

- Overfitting and underfitting check using three different data sets which are training,testing and validation(by analysing the bias and variance) is important to generalise the data.