Aphelion Music: Model Training & Testing Report

Team 5:

Aditya Panwar
Prasanna
Aneesh
Naman

INTRODUCTION

For the project implementation, many libraries are employed, models are tested and finally Music Genre is classified.

This report focuses solely on Models' outputs (training & testing), for the full report kindly refer to the file named ***Aphelion Music Report (Team 5).pdf***.

You may also refer the GitHub repo: **https://github.com/adityapanwar94/ME781/**

Models:

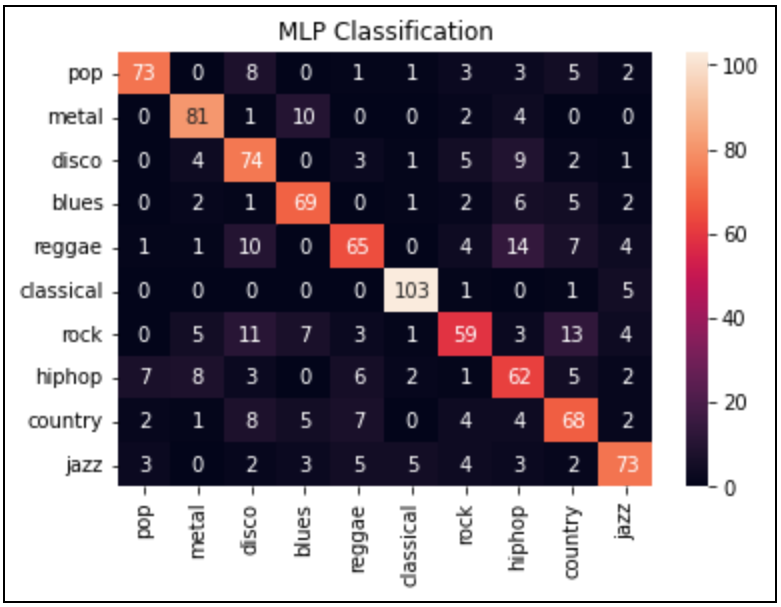
Various models as described in the full report were tested against the extracted features (MFCC) of the GTZAN dataset.

1. ANN (artificial neural network):
    a. Code file: ANN.ipynb
    b. Model Summary:

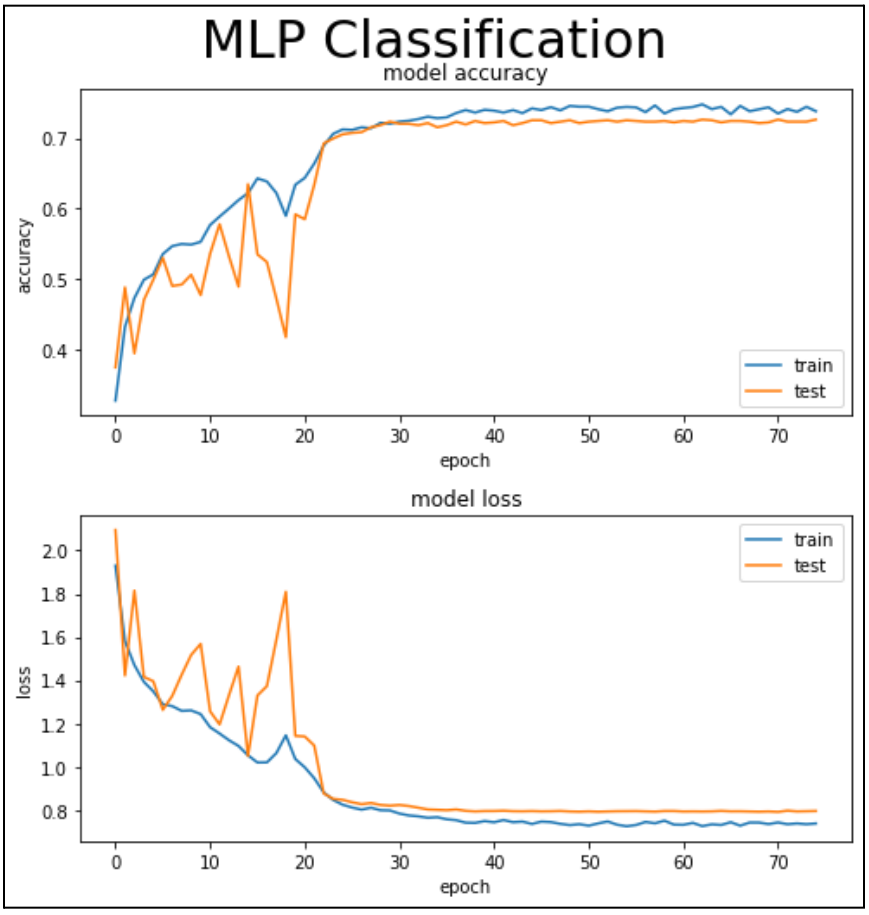
```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 flatten (Flatten)           (None, 1690)              0

 dense (Dense)               (None, 512)               865792

 dense_1 (Dense)             (None, 512)               262656

 batch_normalization (BatchN  (None, 512)              2048
 ormalization)

 dropout (Dropout)           (None, 512)               0

 dense_2 (Dense)             (None, 256)               131328

 dense_3 (Dense)             (None, 256)               65792

 batch_normalization_1 (Batc  (None, 256)              1024
 hNormalization)

 dropout_1 (Dropout)         (None, 256)               0

 dense_4 (Dense)             (None, 128)               32896

 dense_5 (Dense)             (None, 128)               16512

 batch_normalization_2 (Batc  (None, 128)              512
 hNormalization)

 dropout_2 (Dropout)         (None, 128)               0

 dense_6 (Dense)             (None, 64)                8256

 dense_7 (Dense)             (None, 64)                4160

 batch_normalization_3 (Batc  (None, 64)               256
 hNormalization)

 dropout_3 (Dropout)         (None, 64)                0

 dense_8 (Dense)             (None, 10)                650

=================================================================
Total params: 1,391,882
Trainable params: 1,389,962
Non-trainable params: 1,920
_________________________________________________________________
```

Fig: ANN model summary

c. Confusion Matrix:



d. Plots:



2. CNN (convolutional neural network):

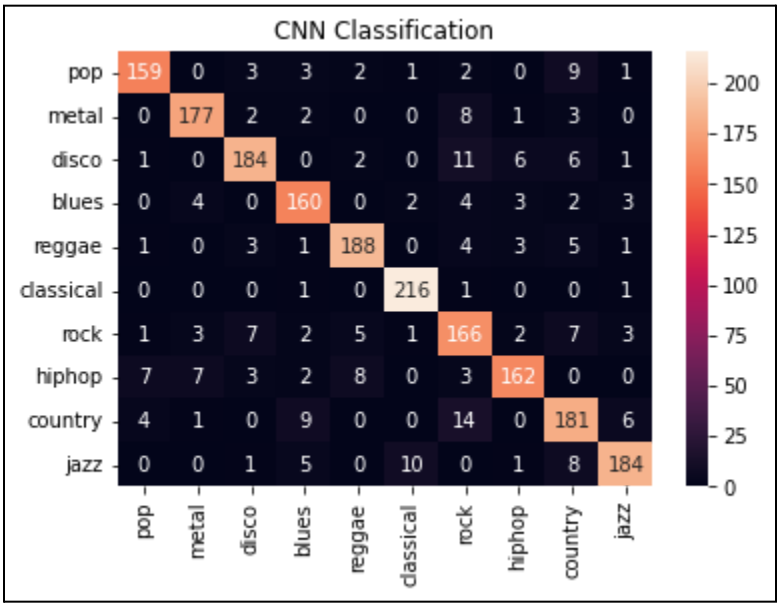
a. Code file: CNN.ipynb

b. Model Summary:

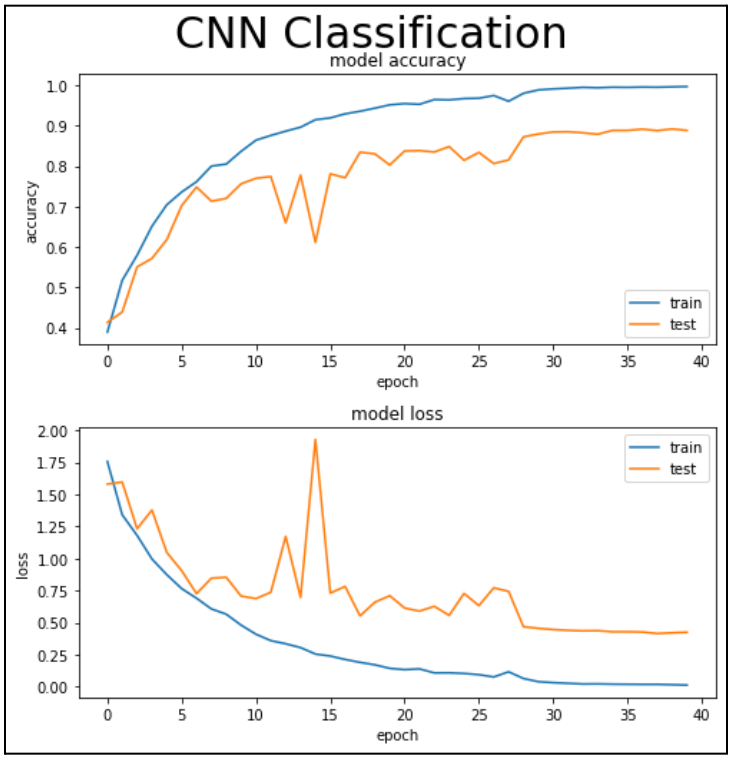
```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 130, 13, 256)      6656

 batch_normalization (BatchN  (None, 130, 13, 256)     1024
 ormalization)

 max_pooling2d (MaxPooling2D  (None, 65, 6, 256)       0
 )

 conv2d_1 (Conv2D)           (None, 65, 6, 128)        819328

 batch_normalization_1 (Batc  (None, 65, 6, 128)       512
 hNormalization)

 max_pooling2d_1 (MaxPooling  (None, 32, 3, 128)       0
 2D)

 conv2d_2 (Conv2D)           (None, 32, 3, 64)         204864

 batch_normalization_2 (Batc  (None, 32, 3, 64)        256
 hNormalization)

 max_pooling2d_2 (MaxPooling  (None, 16, 1, 64)        0
 2D)

 flatten (Flatten)           (None, 1024)              0

 dense (Dense)               (None, 128)               131200

 batch_normalization_3 (Batc  (None, 128)              512
 hNormalization)

 dropout (Dropout)           (None, 128)               0

 dense_1 (Dense)             (None, 64)                8256

 batch_normalization_4 (Batc  (None, 64)               256
 hNormalization)

 dropout_1 (Dropout)         (None, 64)                0

 dense_2 (Dense)             (None, 10)                650

=================================================================
Total params: 1,173,514
Trainable params: 1,172,234
Non-trainable params: 1,280
_________________________________________________________________
```

Fig: CNN model summary

c. Confusion Matrix:



d. Plots:



3. LSTM (Long Short term Memory):

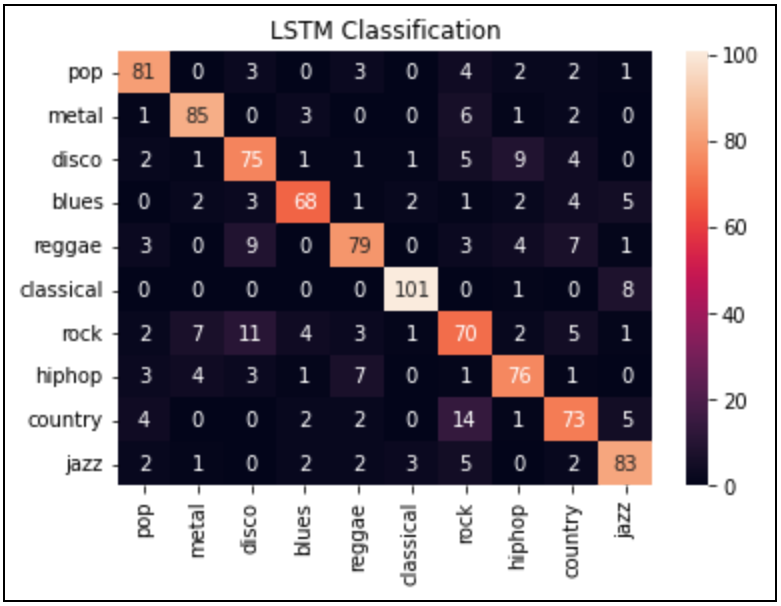
a. Code file: LSTM.ipynb

b. Model Sumary:

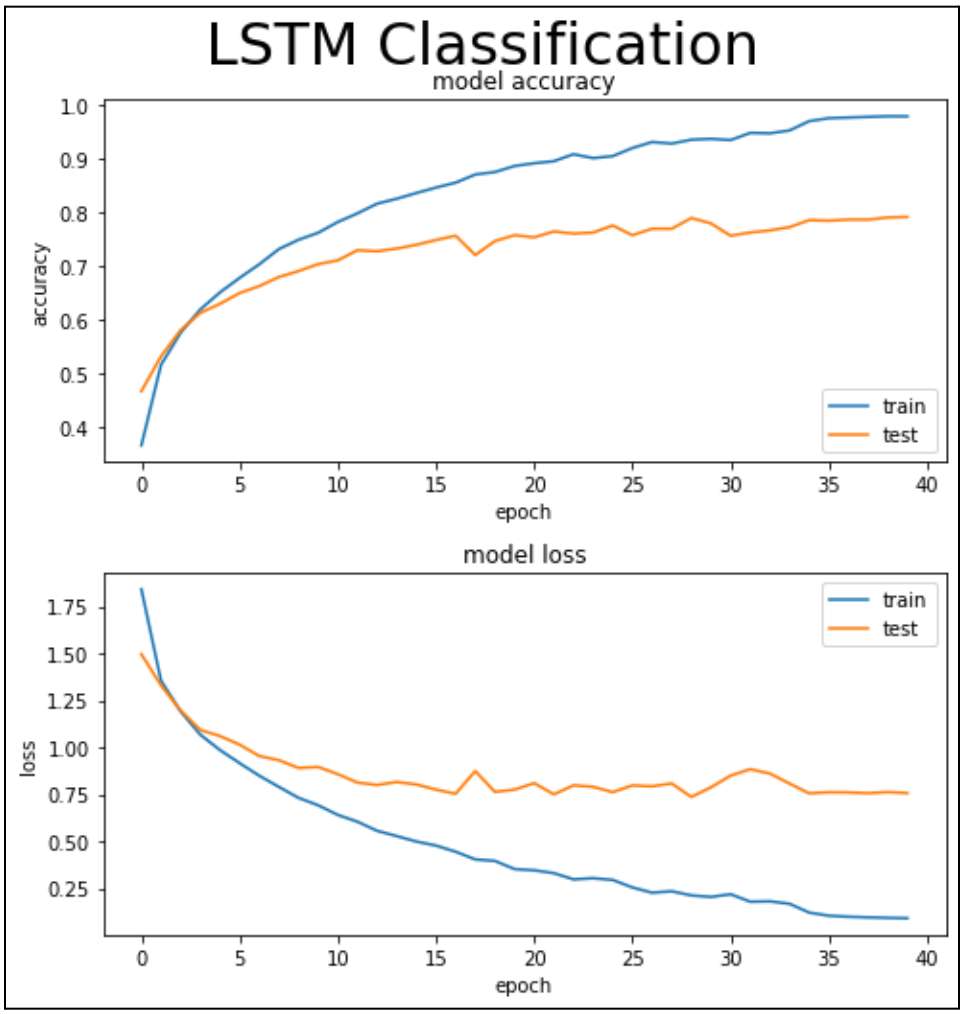
```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm (LSTM)                 (None, 130, 128)          72704

 lstm_1 (LSTM)               (None, 128)               131584

 dense (Dense)               (None, 64)                8256

 dense_1 (Dense)             (None, 64)                4160

 dense_2 (Dense)             (None, 10)                650

=================================================================
Total params: 217,354
Trainable params: 217,354
Non-trainable params: 0
_________________________________________________________________
```

Fig: LSTM model summary

c. Confusion Matrix:



d. Plots:


LSTM Classification
model accuracy
accuracy
epoch
train
test
model loss
loss
epoch
train
test

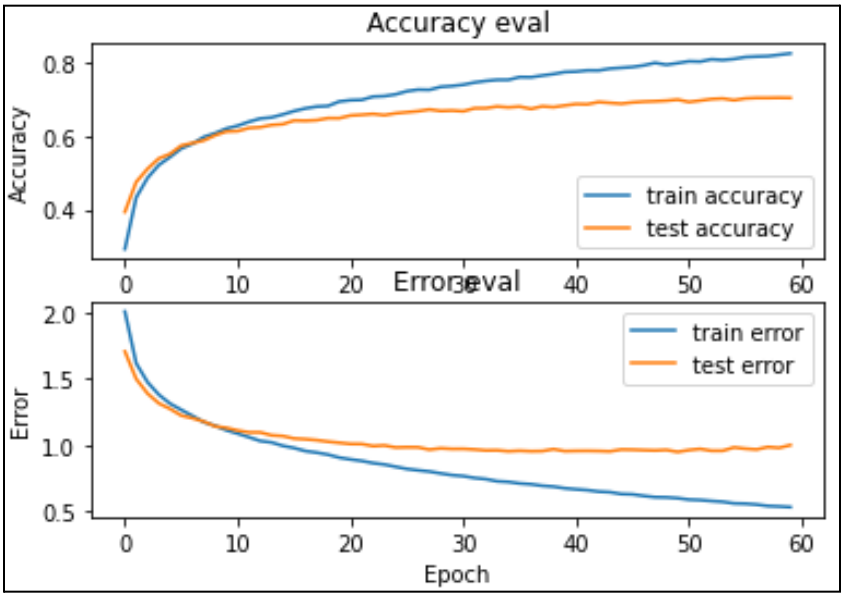
4. RNN (Recurrent Neural Networks):

a. Code file: RNN.ipynb

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm (LSTM)                 (None, 44, 64)            19968

 lstm_1 (LSTM)               (None, 64)                33024

 dense (Dense)               (None, 64)                4160

 dropout (Dropout)           (None, 64)                0

 dense_1 (Dense)             (None, 10)                650

=================================================================
Total params: 57,802
Trainable params: 57,802
Non-trainable params: 0
```

Fig: RNN model summary

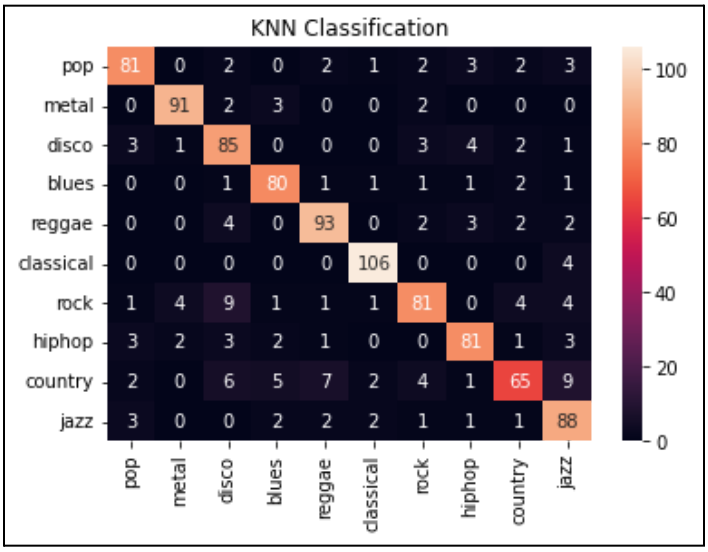
b. Plots:



5. ML models (Long Short term Memory):

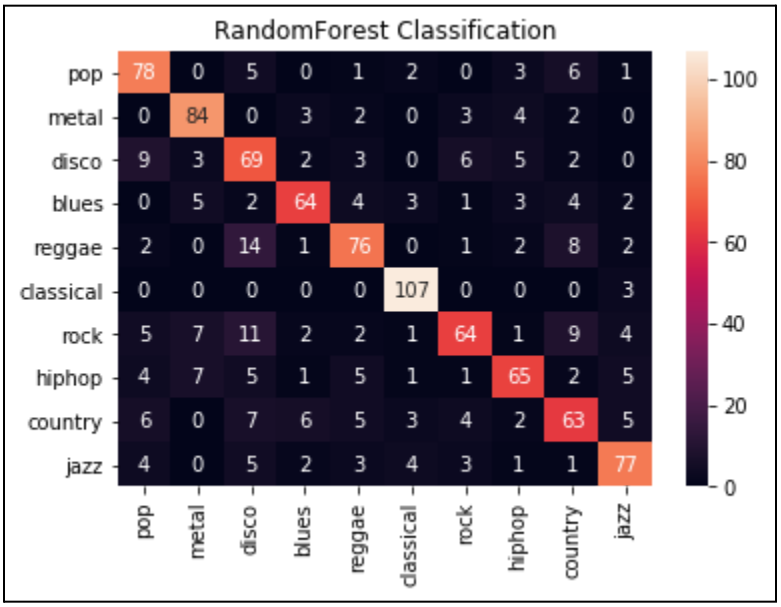
a. Code file: ML.ipynb

b. kNN Confusion Matrix:



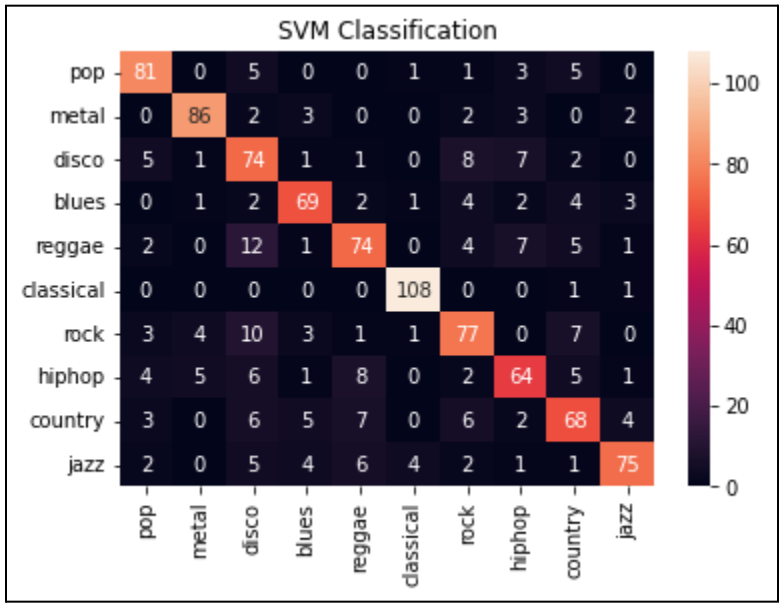

```
Train Accuracy:  0.9988883948421521
Test Accuracy:  0.851
```

c. Random Forest Confusion Matrix:



```
Train Accuracy:  0.9851044908848378
Test Accuracy:  0.747
```

d. SVM Confusion Matrix:




```
Train Accuracy:  0.8724988883948421
Test Accuracy:  0.778
```

Log Files:

Navigate through the **base submission folder** > **Log Files**

Find the relevant log files for various models:

- *CNN_logs*
- *ANN_logs*
- *LSTM_logs*
- *RNN_logs*