CS 464 – Introduction to Machine Learning

FINAL REPORT

A Machine Learning Approach to Music Genre Classification

Group - 7

Atilla Emre Söylemez - 22103189

Berk Sara - 22102354

Dila Tosun - 22102100

Ege Özkan - 22001921

Sezer İlik - 21902929

1. Introduction

Music is an important fabric of human culture which serves as a way to entertain, express emotions. Since digital platforms allowed ways to access and interact with music, the volume of available music tracks has increased which created new opportunities. Thus, in our project, 'A Machine Learning Approach to Music Genre Classification' aims to address the challenge of accurate classification of music into different genres which is a task that is becoming increasingly important for online music platforms. Our aim is to develop models to categorize music into specific genres; jazz, rap, arabesque, Turkish classical music, rock and pop, based on 30 second tracks of Spotify audios.

As music genre classification is a complex process, multiple models are used to validate the genres and different methods are assessed in the upcoming parts of the report. Our approach includes the use of Support Vector Machines (SVM) due to their efficiency in high-dimensional spaces, Random Forests in music genre classification provide robustness to noise, handle high-dimensional data, and offer accurate predictions ,Multilayer Perceptrons (MLP) as they are useful for learning non-linear relationships and Convolutional Neural Networks (CNN) as they can extract hierarchical features from spectrograms of music tracks. The feature extraction techniques that we used include Mel-Frequency Cepstral Coefficients (MFCCs), Chroma features and Spectrograms. They are especially useful for capturing rhythm, harmony and timbre of different music tracks as they are critical for distinguishing between genres. Other techniques and methods are expressed in detail in the upcoming sections.

We have tested our models with different hyperparameters and the results are expressed with confusion matrices, accuracy results and graphs. Our test results indicate promising directions in genre classification tasks, laying the groundwork for further refinement. It was clear that music genres that are close to each other in technical characteristics were hard to identify and the models that we tried did not yield high accuracies for such genres such as Turkish classical music and arabesque. At the conclusion section, the challenges encountered, solutions developed and further directions of interest are expressed.

2. Problem Description

In the highly increasing area of digital music with new digital music platforms which allow millions of tracks with a click away, the ability to accurately classify music into genres is an important task to accomplish. Thus, in our project

we aim to overcome the problem of classifying broad data into correct genres in order to increase user engagement and operational efficiency in platforms that will use our model. Currently, our project focuses on the classification of the following genres: jazz, rap, arabesque, Turkish classical music, rock and pop.

One of the problems that we also want to address is the high variability in music content. Even within a single genre, the range of musical expression is vast. For example, a rock song can range from soft rock ballads to heavy metal, all of which have different expressions and signatures. In music information retrieval (MIR), accurate genre classification of highly varied content is important. Audio data is generally complex as music is a rich, multi-dimensional form of data involving frequency, time, and amplitude. Thus, selecting or extracting features which capture the crucial notes of audio tracks is a problem that we also want to address.

In our project, we also want to address problems that are related to model efficiency. We developed our models to see how we can effectively extract and utilize audio features to distinguish between music genres. Different feature extraction techniques were used. Another question was which machine learning models are most effective for this classification task. Thus, we tried several models, such as but not limited to SVM, MLP, and CNN, to determine their efficiency. Another important issue was the subjectivity and cultural variability in music genre labels. Overall, our main objective is to develop a machine learning solution that classifies music tracks into genres by addressing the above questions.

3. Methods and Dataset

Dataset Accumulation and Preprocessing

The genre classification requires a diverse dataset of music clips. For the implementation of the project, songs from different genres were collected which were all 30 seconds in length, by using the Spotify API that allows access to 30-second previews across different genres. The data needs to be preprocessed in order to standardize the sampling rates. Meaningful features should be extracted to have a better analysis of the audio data and all of the different features provide a different viewpoint for the data.

Feature Extraction

Audio signals are representations of sound data. They have 3 dimensions: amplitude, frequency, and time. They have different parameters that shape their characteristics, which are also important for the feature extractions [1]. Feature extraction for the data processing is an important step since it transforms the raw

audio signals into a format that is more manageable and usable by machine learning models. Different feature extraction methods are being used, such as Mel-frequency cepstral coefficients (MFCCs), spectrograms, and Chroma features.

- *Mel-Frequency Cepstral Coefficients (MFCCs)* [1]: MFCCs are coefficients that are derived from a cepstral representation of an audio clip. It begins by dividing the audio signal into short frames as the sounds change rapidly. Then, those frames are passed through a Fourier Transform for time-to-frequency domain transition. They are the results of a cosine transform of the real logarithm of the short-term energy spectrum expressed on a mel-frequency scale, which is the scale that mimics the human ear's response to different frequencies. Lastly, the finer details of sound, the higher coefficients, are usually discarded as they are not that useful for distinguishing the music genre. MFCCs are mostly used for distinguishing timbre, which is the tone quality of sound. Here is an example representation of MFCC:

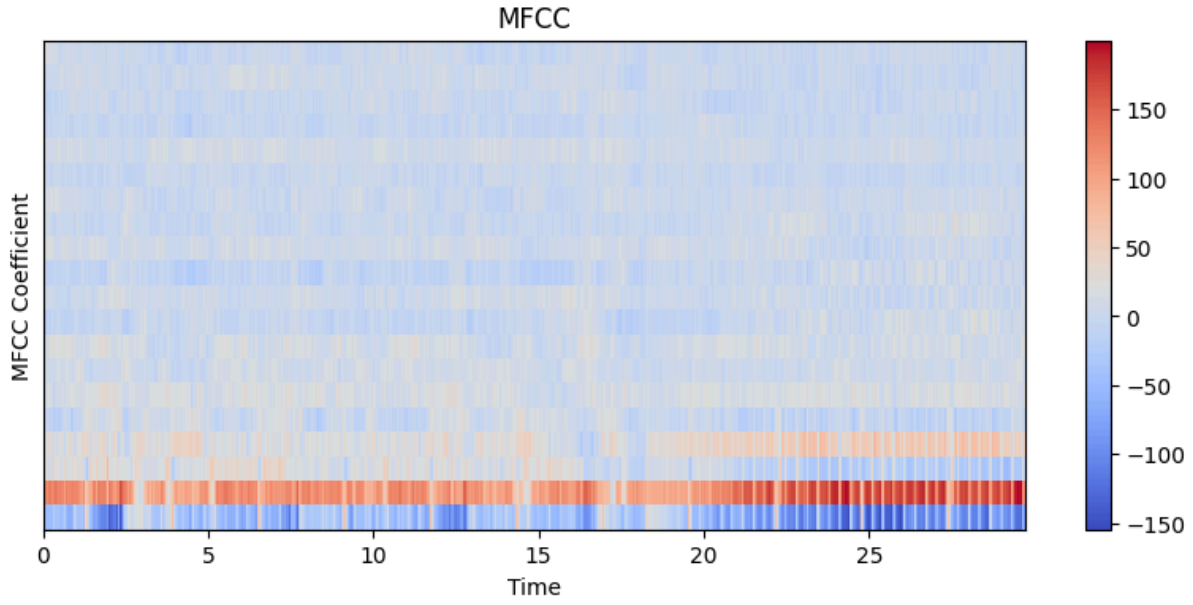

Fig 1 : Example MFCC representation

- *Spectrogram* [2]*:* A spectrogram is a graph that shows the spectrum of frequencies in a sound signal as it varies over time. The graph indicates the frequencies where the signal's energy is changing. The power of frequencies is represented with different colors. Spectrograms are used for identifying different characteristics of the audio signals, such as rhythmic patterns, pitch variations, and similar dynamics. They are useful for genre classification as they capture unique elements such as tone, pitch, and rhythm of sounds across frequencies, which is helpful for highlighting different genres. Here is an example representation of a Spectrogram:

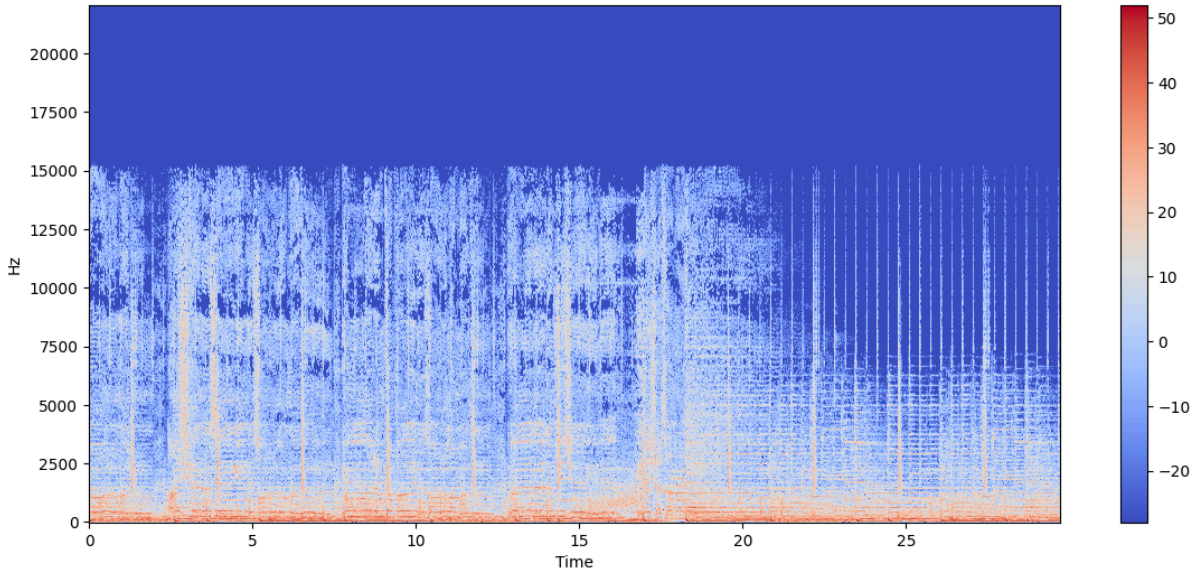

Fig 2 : Spectrogram representation

- *Chroma Feature* [3,4]: The Chroma features relate to 12 different pitch classes. A pitch class represents all pitches that are related to each other by octave or enharmonic equivalence. Pitches of the audio sounds are categorized into those pitch classes, which allows us to capture the harmonic and melodic characteristics of the sound. This approach is especially useful for differentiating between genres that have distinct harmonic structures, such as classical, which usually has harmonic progressions, versus techno, which usually doesn't have such harmonic content. Here is a representation of the Chroma feature:

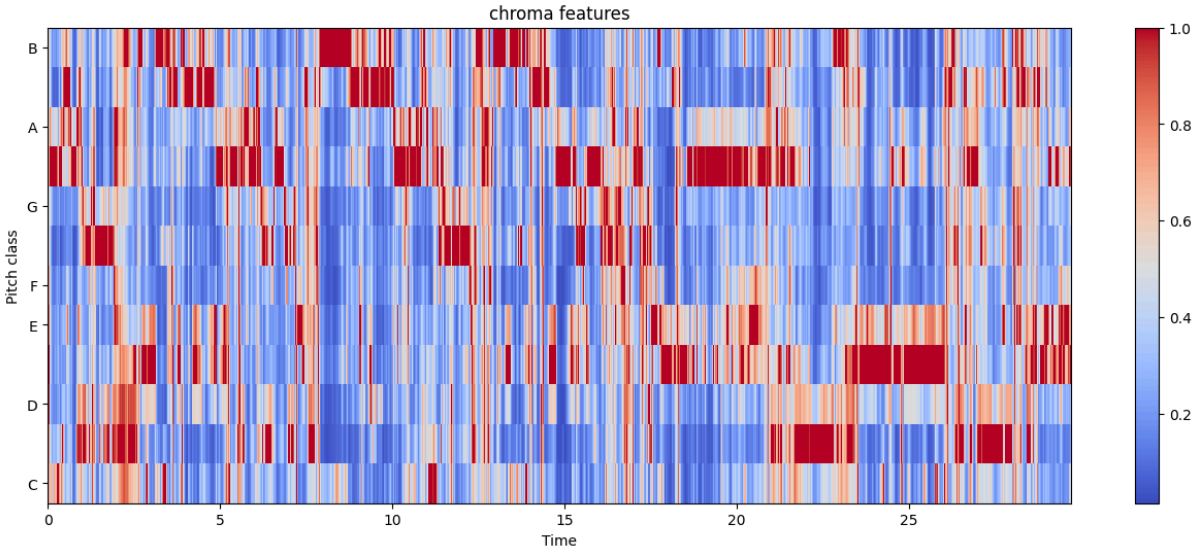

Fig 3 : Chroma feature representation

- *Spectral Rolloff* [5]*:* The Spectral Rolloff gives information about where most

of the energy is stored in the frequency graph at a given time. If the value is lower, this means lower frequency prominence while a higher one indicates otherwise. This is useful in determining where in the spectrogram the music lives generally. For example, a song from Turkish classical music tends to be lower compared to a pop or rock due to the more prominent usage of high-frequency elements in the said genres. A sample spectral rolloff is given as follows:

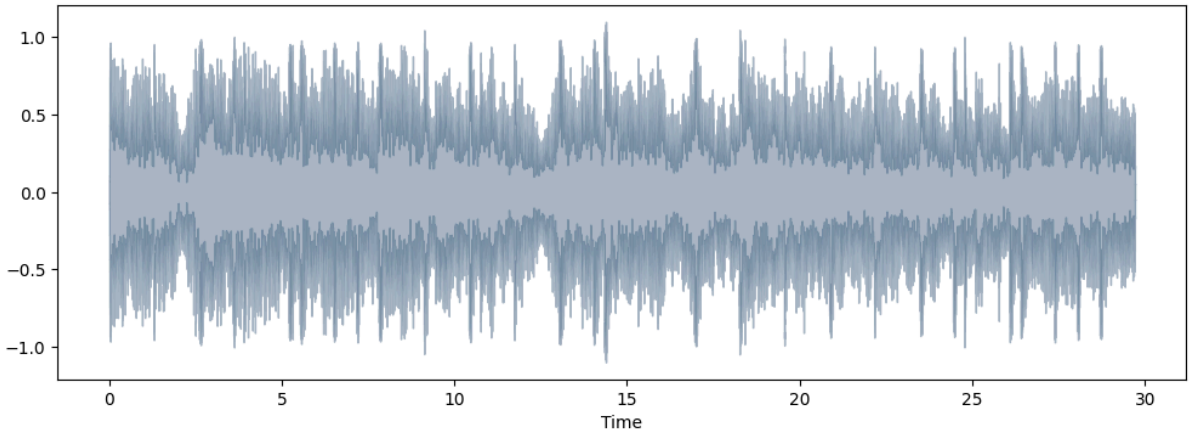

Fig 4: Sample Spectral Rolloff

Zero-Crossing Rate [6]*:* Zero-Crossing Rate is the rate at which a signal switches its sign, i.e., from negative to positive. It can provide context about the "smoothness" of the music as well as its tempo and frequency. For example, a high level of electronic noise is reflected in this graph. A rap song tends to have larger values than a jazz song for example. A sample zero-crossing rate graph is given as follows:

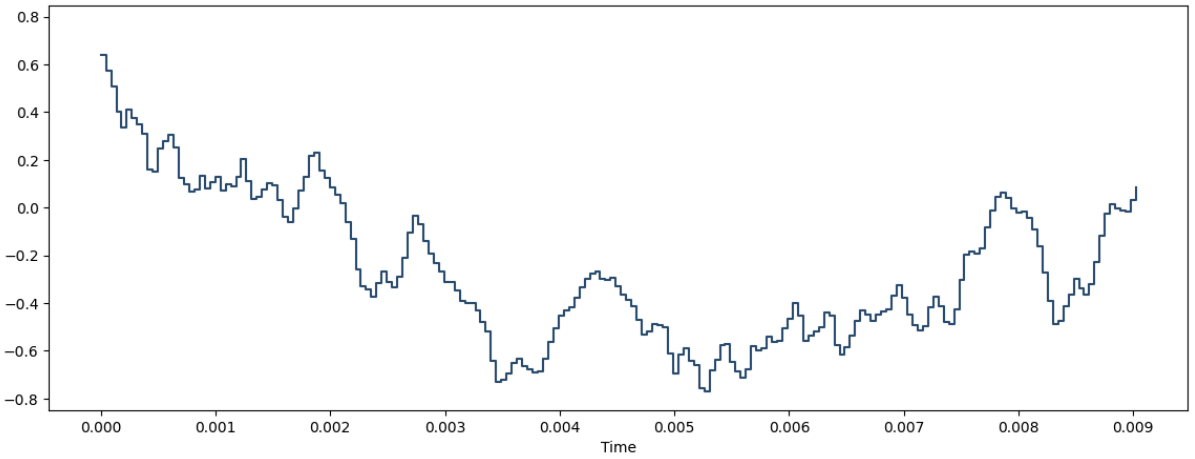

Fig 5: Sample Zero-Crossing Rate Graph

- *Spectral Bandwidth* [6,7]*:* Spectral Bandwidth displays the largest difference between the highest and lowest frequency present at a given time in the signal. Combined with Spectral Centroid, this could provide context about the frequency accumulation of the song. Here is a sample spectral bandwidth:

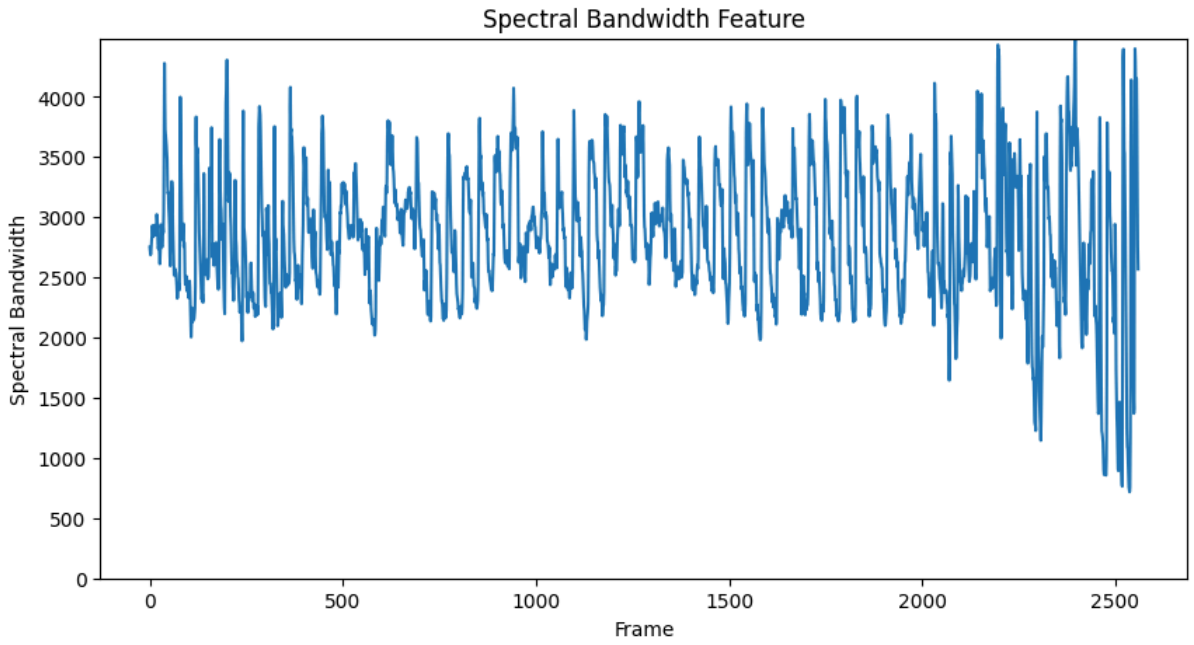

Fig 6: Spectral Bandwidth

- *Spectral Centroid*[5,7]*:* Spectral Centroid displays the center of all frequencies by taking their amplitude into account and averaging them. A sample spectral centroid is as follows:

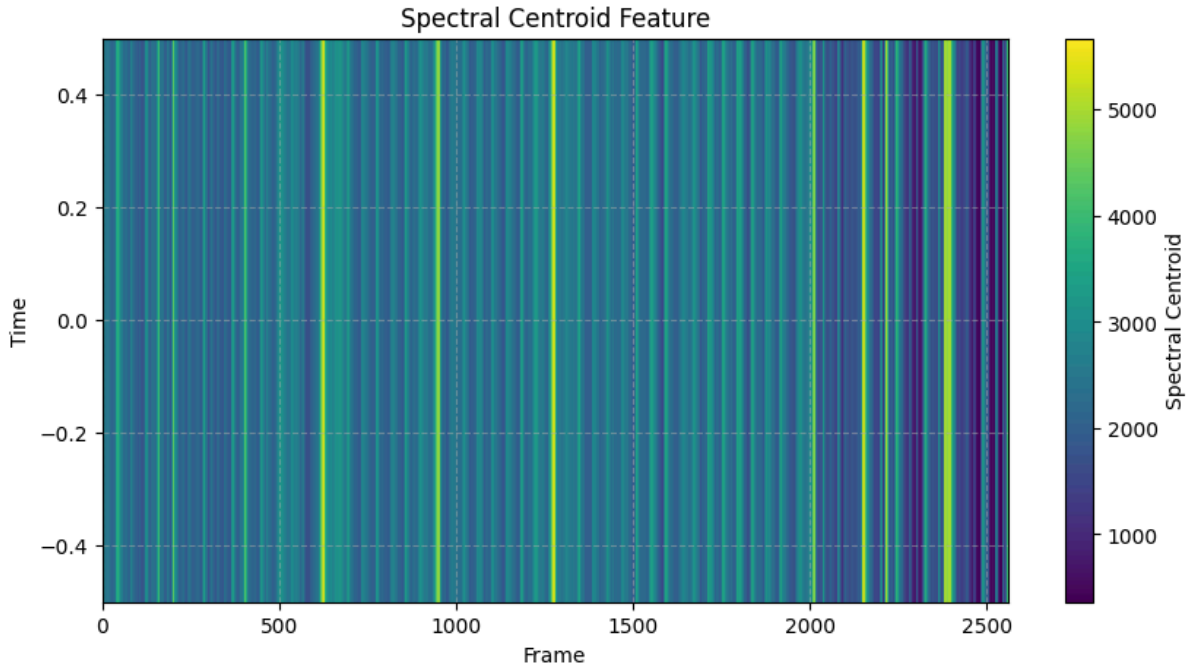

Fig 7: Spectral Centroid

Models Used

For the models we used machine learning algorithms to complete the classification of the songs. The models were Support Vector Machine (SVM)[8], Multi-Layer Perceptron (MLP)[9], Convolutional Neural Network (CNN)[10].

Convolutional Neural Networks (CNNs) are a class of deep neural networks which are particularly used for images[10,11]. They operate through multiple layers that each perform distinct jobs. The first layer is the convolutional layer that applies filters to the input to create feature maps. Specific features are detected by those filters applied. After the convolution, ReLU is applied for non-linearity which is very useful for our music tracks as they have more complex patterns. After the convolutional layers, pooling layers reduce the spatial dimensions and lastly there are fully connected layers to combine the features into a final output, in our case genre labeling. For CNN we built our own model from scratch. We tried several different models and assessed their performance. These models are Sequential only, VGG16 [12], LeNet-5 and ResNet models.

Multilayer Perceptron (MLP) consists of multiple layers of nodes such as an input layer, one or more hidden layers, and an output layer. Each node in a layer connects with a certain weight to every node in the following layer. The input layer has neurons which receive different forms of input features such as raw data values. Hidden layers consist of weighted computations that transform values from the previous layer. The output layer gets the values from the last hidden layer and transforms them into output values. MLP was done without any convolution layers and it has a basic sequential model with dense and dropout layers. The data was shrunk using mean and variance. Also we calculated the mutual information for each feature in this model and eliminated the worst 5. Furthermore, we added the confusion matrix to analyze the data.

Transfer Learning is a technique where knowledge gained while solving one problem is applied to some other similar problem. There are source and target domains. The source domain is the dataset and the corresponding task where the model is initially trained. The target domain is the new dataset that the knowledge learned is stored. We initially tried working with transfer learning models, however there are fewer pre-trained models available that are directly applicable to music data compared and the ones that are available are not suitable much for the initial data that we had. The variability within genres also created problems as the model trained on one subset of a genre might struggle to generalize across the entire genre.

Support Vector Machines (SVMs) are a class of supervised machine learning algorithms that are used for classification tasks. The best separating

hyperplane that divides the data into classes with the minimum margin is found. They are efficient in high-dimensional spaces and they can model complex non-linear relationships. Thus, it was especially useful for our task because of the complexity of audio tracks.

4. Results

Multi-Layer Perceptron (MLP)

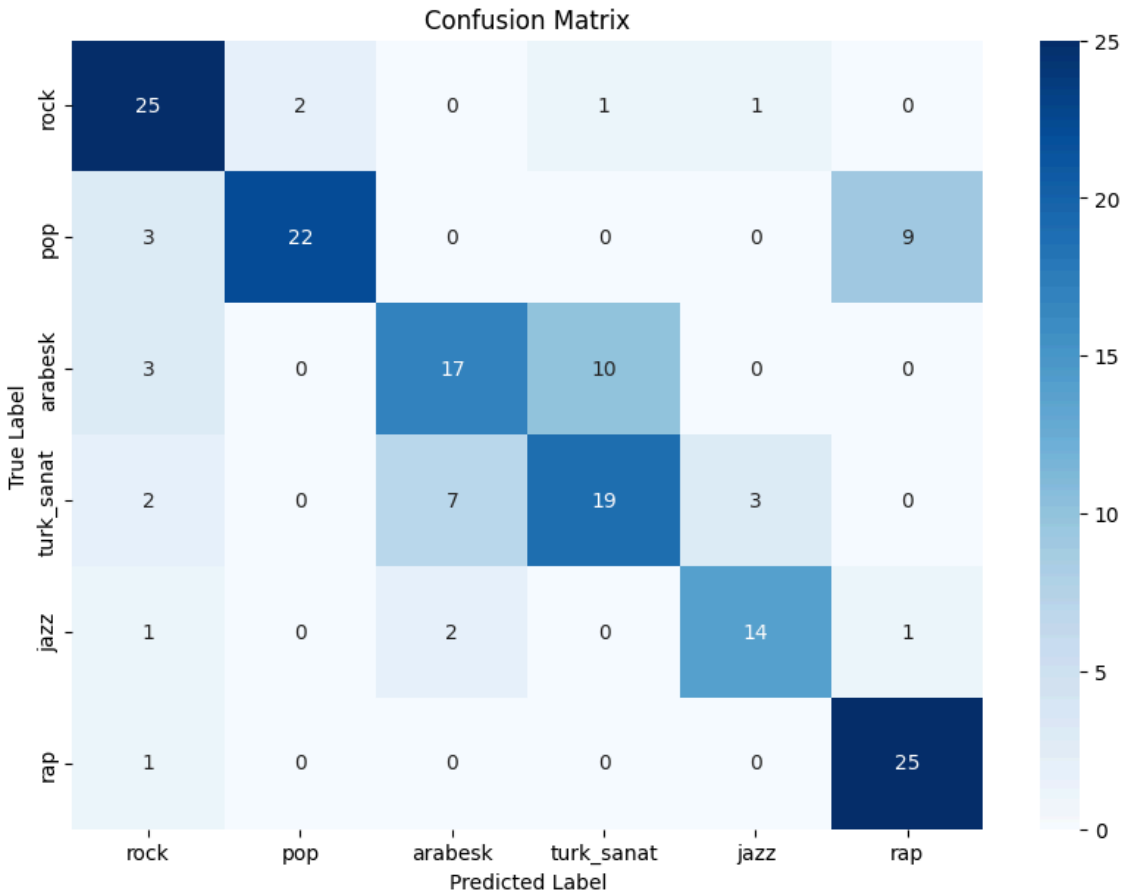

Fig 8: Confusion Matrix for Multi-Layer Perception

Genre	Precision	Recall	F1-Score	Support
Arabesk	0.71	0.86	0.78	29

Jazz	0.92	0.65	0.76	34
Pop	0.65	0.57	0.61	30
Rap	0.63	0.61	0.62	31
Rock	0.78	0.78	0.78	18
Turk Sanat	0.71	0.96	0.82	26
Macro Avg	0.74	0.74	0.73	168
Weighted Avg	0.74	0.73	0.72	168

Table 1: Results for MLP

- Overall Accuracy for sequential only: 0.72

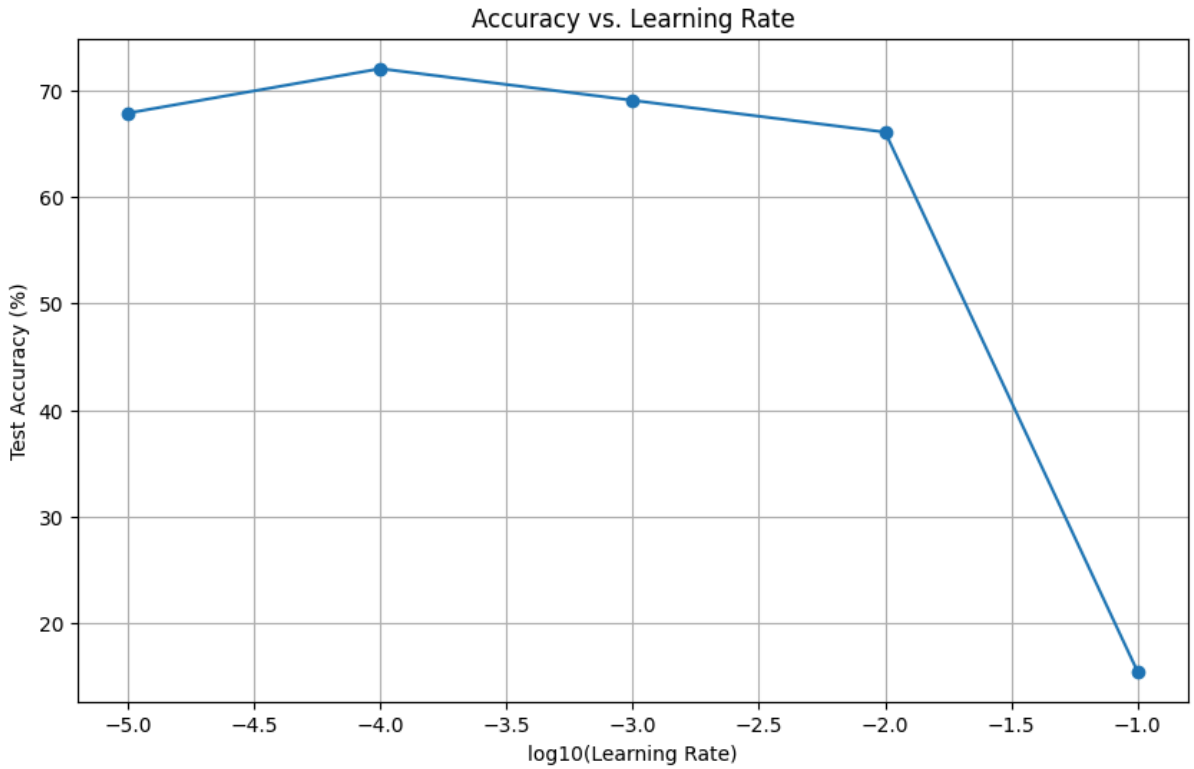

Fig 9: Accuracy-Learning Rate Graph

- Learning Rate: 0.01, Test Accuracy: 9.03%
- Learning Rate: 0.01, Test Accuracy: 66.07%
- Learning Rate: 0.001, Test Accuracy: 69.05%

- Learning Rate: 0.0001, Test Accuracy: 72.02%
- Learning Rate: 1e-05, Test Accuracy: 67.86%

Convolutional Neural Network (CNN)

Sequential

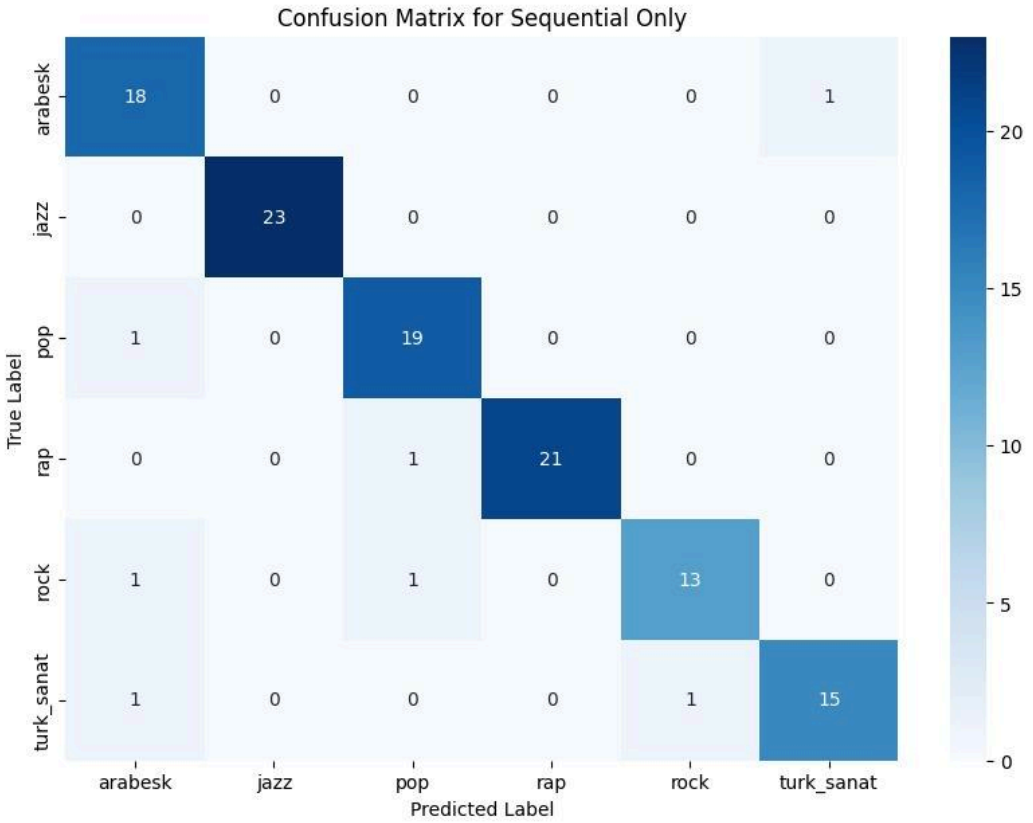

Fig 10: Confusion Matrix for Sequential Model (CNN)

Genre	Precision	Recall	F1-Score	Support
Arabesk	0.86	0.95	0.90	19
Jazz	1.00	1.00	1.00	23
Pop	0.90	0.95	0.93	20
Rap	1.00	0.95	0.98	22
Rock	0.93	0.87	0.90	15
Turk Sanat	0.94	0.88	0.91	17

Macro Avg	0.94	0.93	0.93	116
Weighted Avg	0.94	0.94	0.94	116

Table 2: Results for Sequential

- Overall Accuracy for sequential only: 0.72

VGG16

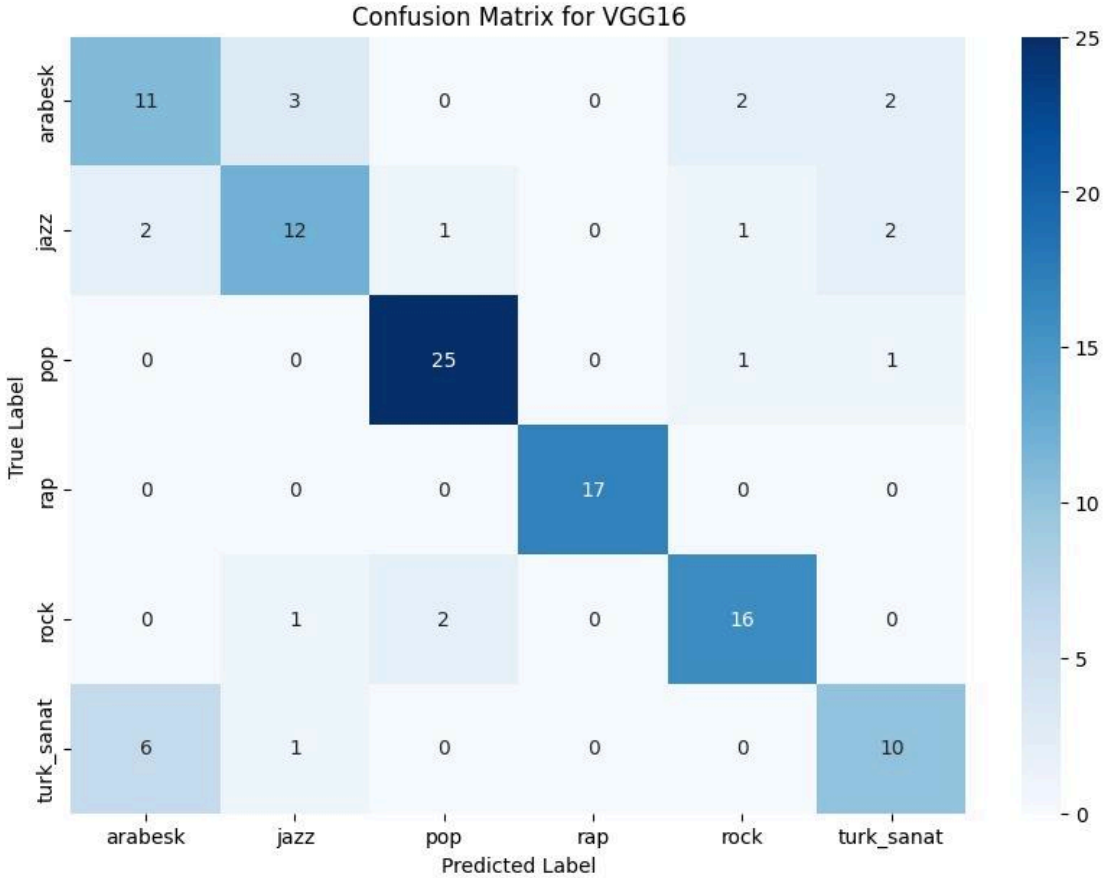

Fig 11: Confusion Matrix for VGG16 (CNN)

Genre	Precision	Recall	F1-Score	Support
Arabesk	0.90	0.95	0.92	19
Jazz	0.96	0.96	0.96	23

Pop	0.95	0.95	0.95	20
Rap	1.00	1.00	1.00	22
Rock	1.00	0.93	0.97	15
Turk Sanat	0.88	0.88	0.88	17
Macro Avg	0.95	0.94	0.95	116
Weighted Avg	0.95	0.95	0.95	116

Table 3: Results Table for VGG16 (CNN)

- Overall Accuracy: 0.95

LeNet-5

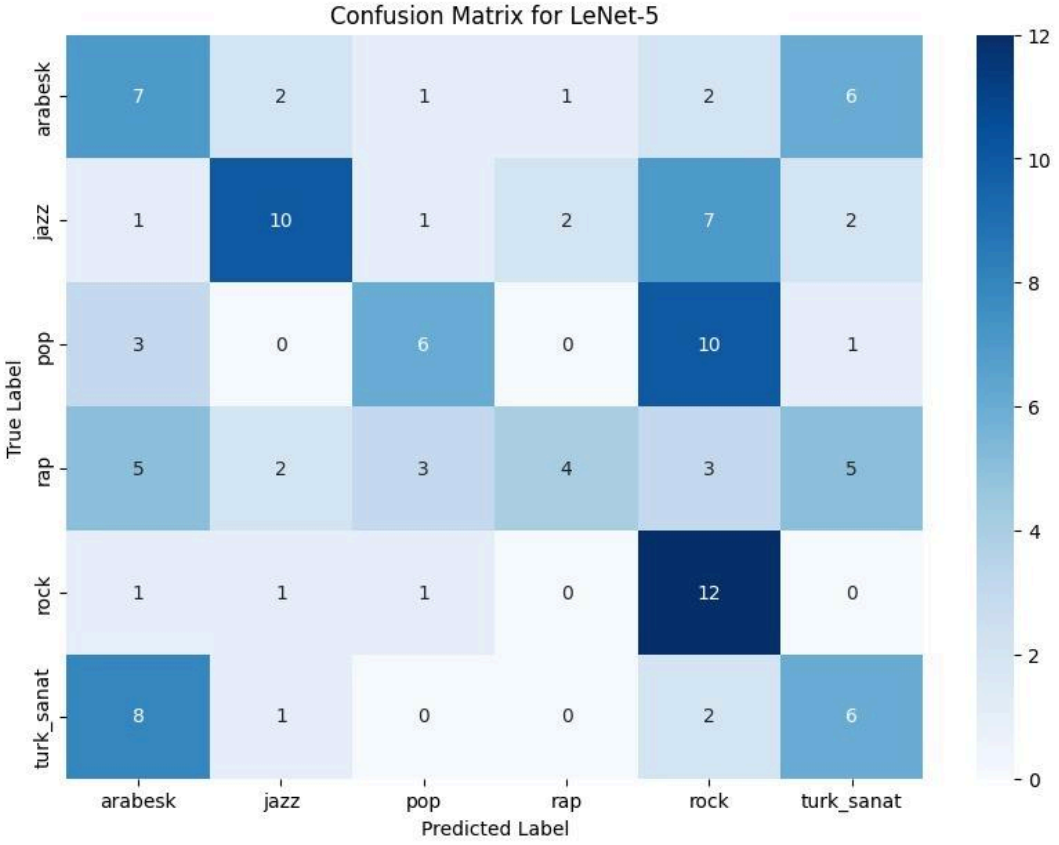

Fig 12: Confusion Matrix for LeNet-5 (CNN)

Genre	Precision	Recall	F1-Score	Support
Arabesk	0.28	0.37	0.32	19
Jazz	0.62	0.43	0.51	23
Pop	0.50	0.30	0.38	20
Rap	0.57	0.18	0.28	22
Rock	0.33	0.80	0.47	15
Turk Sanat	0.30	0.35	0.32	17
Macro Avg	0.43	0.41	0.38	116
Weighted Avg	0.45	0.39	0.38	116

Table 4: Results Table for LeNet-5 (CNN)

- Overall Accuracy: 0.39

ResNet

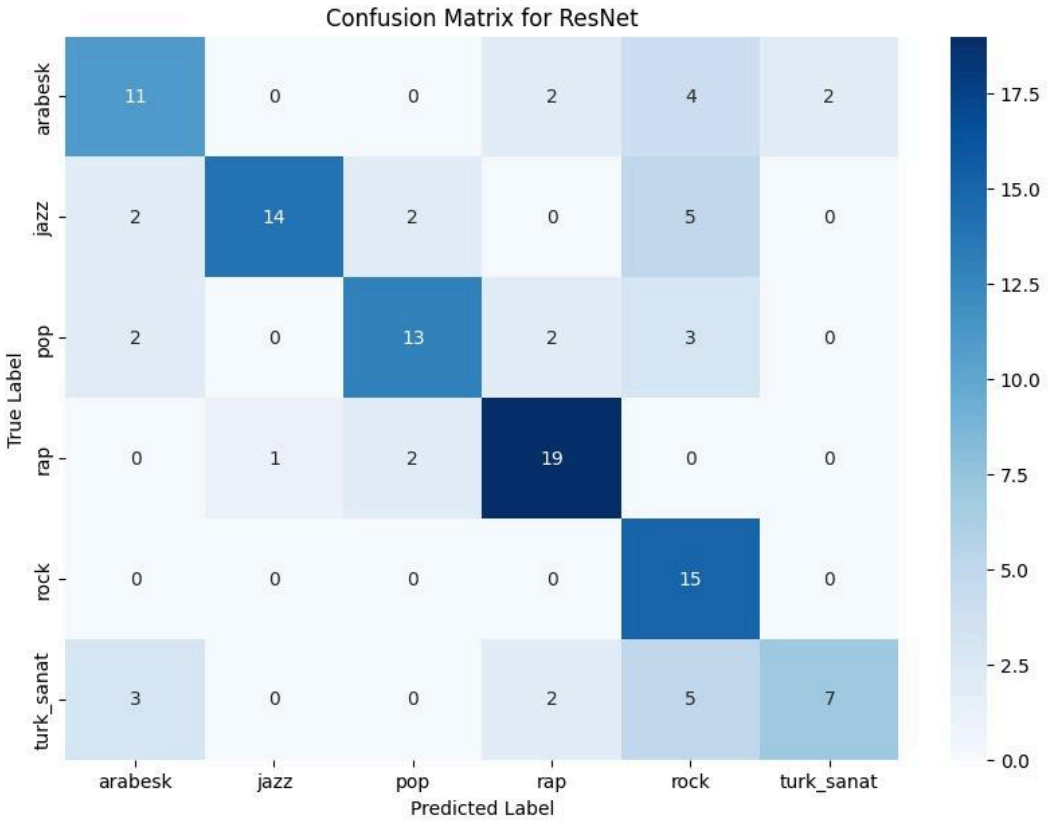

Fig 13: Confusion Matrix for ResNet (CNN)

Genre	Precision	Recall	F1-Score	Support
Arabesk	0.61	0.58	0.59	19
Jazz	0.93	0.61	0.74	23
Pop	0.76	0.65	0.70	20
Rap	0.76	0.86	0.81	22
Rock	0.47	1.00	0.64	15
Turk Sanat	0.78	0.41	0.54	17
Macro Avg	0.72	0.69	0.67	116
Weighted Avg	0.74	0.68	0.68	116

Table 5: Results Table for ResNet (CNN)

- Overall Accuracy: 0.68

Support Vector Machine(SVM)

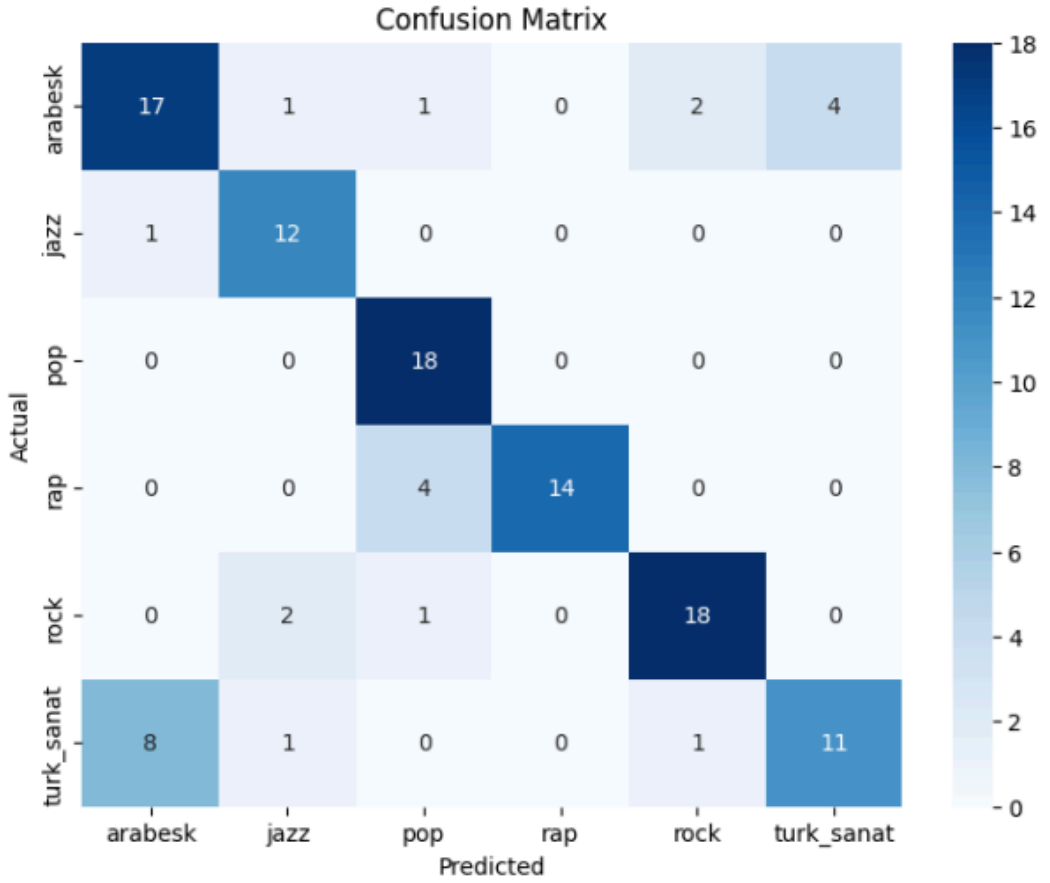

Fig 14: Confusion Matrix for Support Vector Machine

Genre	Precision	Recall	F1-Score	Support
Arabesk	0.65	0.68	0.67	25
Jazz	0.75	0.92	0.83	13
Pop	0.75	1.0	0.86	18
Rap	1.0	0.78	0.88	18
Rock	0.86	0.86	0.86	21
Turk Sanat	0.73	0.52	0.61	21
Macro Avg	0.79	0.79	0.78	116
Weighted Avg	0.78	0.78	0.77	116

Table 6: Results Table for Support Vector Machine

- Overall Accuracy: 0.77

Random Forest Classifier:

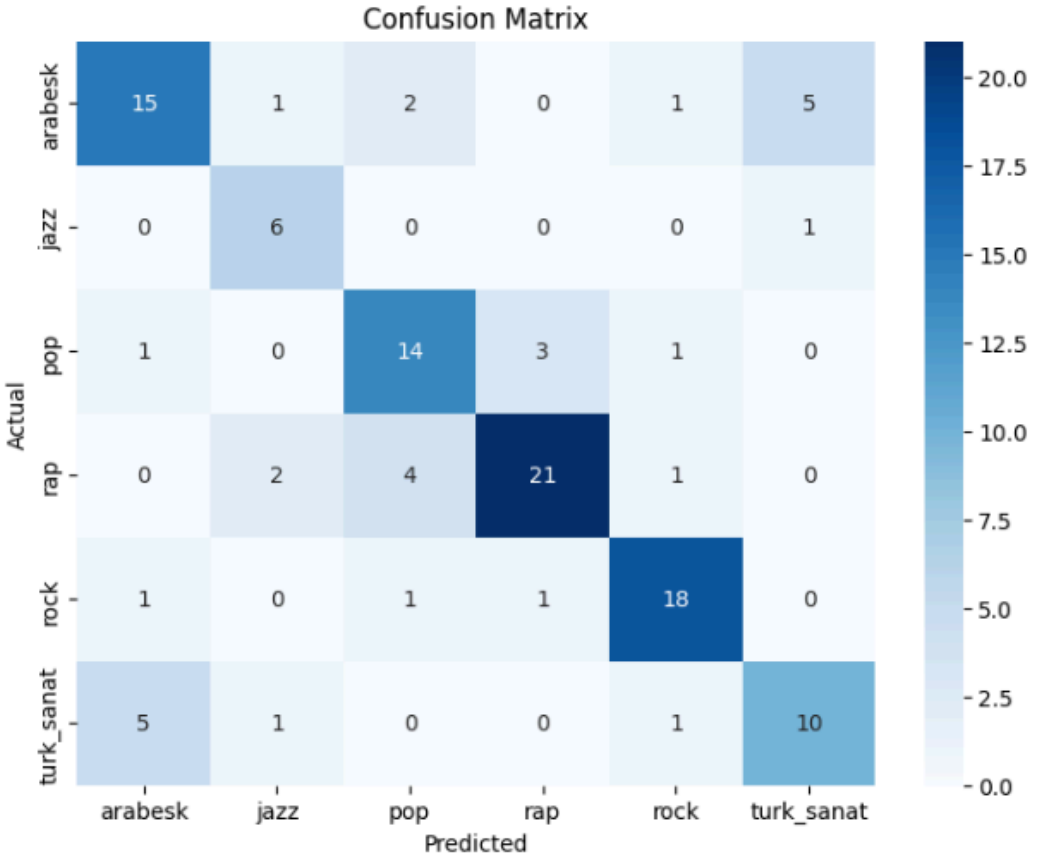

Fig 15: Confusion Matrix for Random Forest Classifier

Genre	Precision	Recall	F1-Score	Support
Arabesk	0.68	0.62	0.65	24
Jazz	0.60	0.86	0.71	7
Pop	0.67	0.74	0.70	19
Rap	0.84	0.75	0.79	28
Rock	0.82	0.84	0.86	21
Turk Sanat	0.62	0.59	0.61	17

Macro Avg	0.71	0.74	0.72	116
Weighted Avg	0.73	0.72	0.72	116

Table7: Random Forest Classifier

- Overall Accuracy: 0.72

5. Discussion

Random Forest Classifier:

In our study, we employed the Random Forest classifier, a powerful ensemble learning technique. The Random Forest classifier exhibited robust performance across various hyperparameters, particularly in terms of the number of estimators (trees) in the forest. We experimented with different numbers of estimators, ranging from 50 to 150, and observed notable variations in performance metrics such as accuracy and F1 score. Interestingly, the Random Forest classifier demonstrated competitive accuracy rates, with peak performance observed around 100 estimators, achieving an accuracy of approximately 72% on the test dataset. Furthermore, unlike some CNN architectures, Random Forest's training time remained relatively consistent across different hyperparameter settings, making it computationally efficient for our music genre classification.

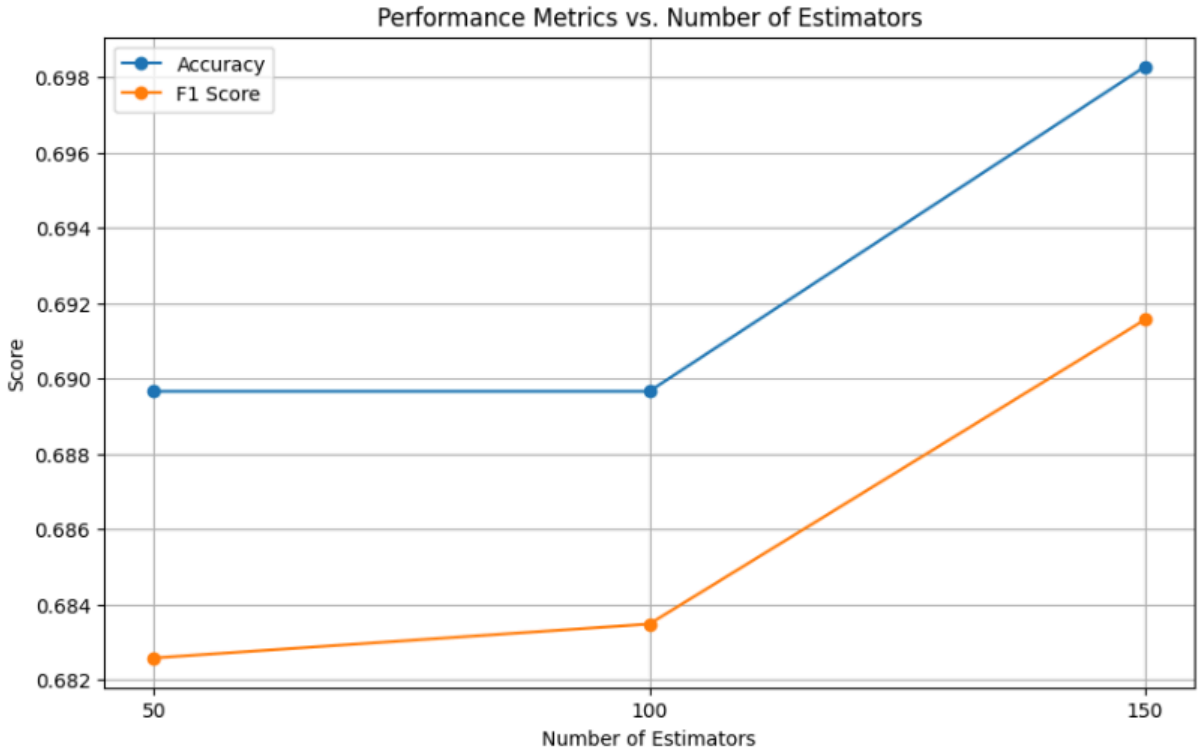

Fig 16: Accuracy/F1 Score- Number of Estimators Graph

SVM

In our study, we incorporated the Support Vector Machine (SVM) classifier, a widely-used supervised learning algorithm. We investigated SVM's performance across different learning rates, aiming to optimize its classification accuracy for music genre recognition. Despite SVM's sensitivity to hyperparameters, particularly the learning rate, we observed varied performance across different settings. With careful experimentation, we found that SVM achieved peak accuracy around a learning rate of 1, reaching approximately 72% accuracy on the test dataset. However, our results also revealed challenges in distinguishing certain genres, such as Arabesque and Turkish Classical, which may have contributed to fluctuations in accuracy across different learning rates. Despite these challenges, SVM showcased stability in performance and computational efficiency, making it a viable option for music genre classification tasks, especially when combined with appropriate hyperparameter tuning strategies. Overall, our findings underscore the importance of exploring SVM's sensitivity to learning rates and its potential for accurate and efficient classification in music genre recognition tasks.

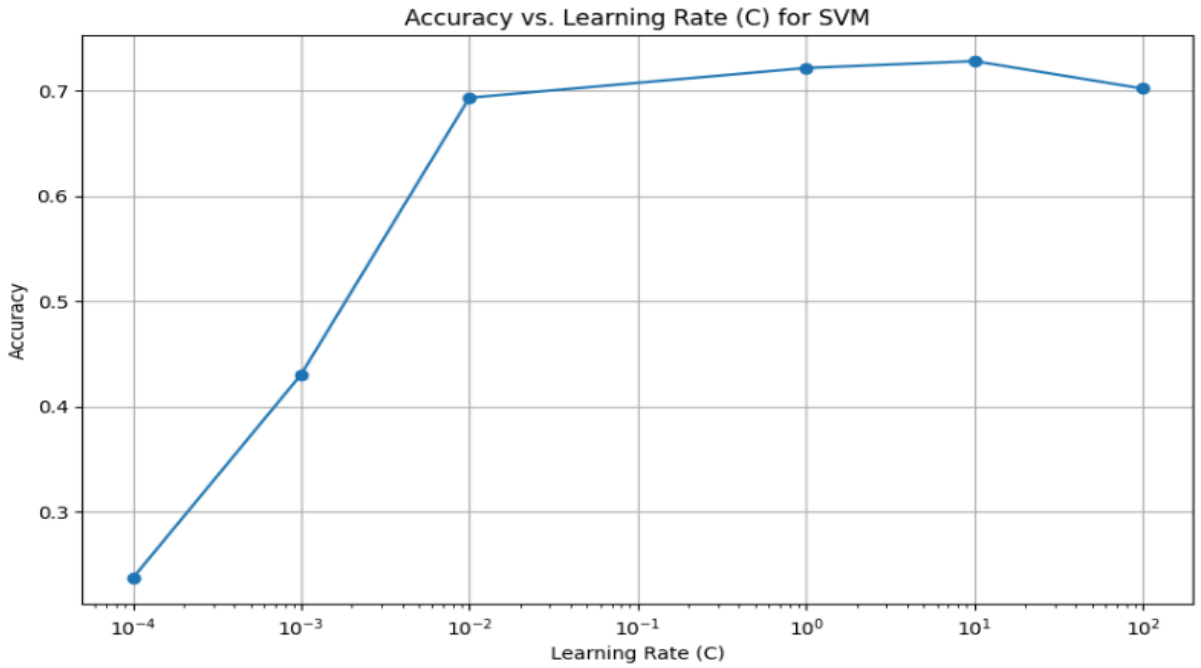

Fig 17:

MLP

For the MLP model we tried the training for the 6 music genre classification. Rock and Rap were easily distinguished in this model however turk sanat music and arabesk music were not easily distinguishable from each other using this model. With different learning rates that goes from 10**-1to 10**-5 we got a model that reached a maximum %72 accuracy and worst %9 accuracy with Adam optimizer with 30 epoch and 16 batch size.

```
model = k.models.Sequential([
    k.layers.Dense(1024, activation="relu", input_shape=(X_train.shape[1],)),
    k.layers.Dropout(0,2),
    k.layers.Dense(512, activation="relu"),
    k.layers.Dropout(0,2),
    k.layers.Dense(256, activation="relu"),
    k.layers.Dropout(0,2),
    k.layers.Dense(128, activation="relu"),
    k.layers.Dropout(0,2),
    k.layers.Dense(64, activation="relu"),
    k.layers.Dropout(0,2),
    k.layers.Dense(6, activation="softmax"),
])
```

Fig 18: MLP Structure

This neural network model, implemented using the Keras Sequential API, comprises several densely connected layers with varying numbers of neurons and ReLU activation functions. Dropout layers with a dropout rate of 0.2 are

inserted after each dense layer to mitigate overfitting. The model culminates in an output layer with a softmax activation function, facilitating multi-class classification by producing a probability distribution over the output classes.

The architecture's depth, ReLU activation, and dropout regularization contribute to its effectiveness. The deep structure enables the model to learn complex features, while ReLU activation fosters rapid convergence and combats the vanishing gradient issue. Additionally, dropout layers enhance generalization by stochastically dropping units during training, preventing overfitting. Finally, the softmax output layer ensures probabilistic class predictions, rendering the model suitable for multi-class classification tasks.

CNN

Different CNN models were used for this project were Sequential, VGG16, LeNet-5, ResNet.

For Sequential and VGG16, we managed to get quite accurate models with ~95% accuracy for both in the best cases. Sequential model was trained with 30 epochs and VGG16 10 epochs; they both use Adam optimizer. However, when we tried the Sequential model for different learning rates, the maximum accuracy we reached was 72% with 0.0001 learning rate. We believe that this is due to the randomized split of the testing data because the genres Arabesque and Turkish Classical proved to be the hardest ones to distinguish, so their distribution along this train/test splits affect the final result dramatically. Nonetheless, the accuracy seems to drop before and after the 0.0001 learning rate mark.

For LeNet-5 the accuracy is quite low because the execution took too long and kept failing when we tried to increase the 2D convolution filter number, so we had to keep it at 16 after 12 instead of 32 as we initially tried. The structure is below:

```
checkpoint_dir = "lenet5_models/"
os.makedirs(checkpoint_dir, exist_ok=True)

model = Sequential()
model.add(Conv2D(12, kernel_size=(5, 5), strides=(1, 1), activation='relu', input_shape=train_spec.shape[1:], padding='same'))
model.add(MaxPooling2D(pool_size=(2, 2), strides=(2, 2)))
model.add(Conv2D(16, kernel_size=(5, 5), strides=(1, 1), activation='relu', padding='valid'))
model.add(MaxPooling2D(pool_size=(2, 2), strides=(2, 2)))
model.add(Flatten())
model.add(Dense(120, activation='relu'))
model.add(Dense(84, activation='relu'))
model.add(Dense(num_classes, activation='softmax'))

model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
```

Fig 19: LeNet-5 Structure

Similar problem was prevalent for ResNet too. The accuracy at the end was 68%. The computational resources required for these two models were more than

we had at the time. Both were compiled with 10 epochs. However, they still yield some interesting results as the trend of confusing the Arabesque and Turkish Classical music continued for less accurate models, especially for LeNet-5, while ResNet struggled with Jazz the most. Pop and Rap genres seem to be more distinct and easily recognizable overall.

6. Conclusions

Our project, A Machine Learning Approach to Music Genre Classification, aimed to develop an effective model to categorize 30 second Spotify audio tracks into jazz, rap, arabesque, Turkish classical music, rock, and pop genres. We created multiple models including Support Vector Machines (SVMs), Random Forest Classifier, Multilayer Perceptrons (MLP), Convolutional Neural Networks (CNNs) which all have different strengths in handling the complexity and high dimensionality of audio data. Different feature extraction techniques were used to help in capturing the essential elements of rhythm, harmony and timbre. The MLP model showed moderate success, an accuracy rate of 72%. It distinguished genres like rock and rap were notable however with genres with closer musical signatures such as Turkish Classical music and arabesque weren't as successful. Our CNN models varied in performance with the Sequential and VGG16 models, which achieved high accuracies, like 95%. But models like LeNet-5 and ResNet had lower accuracy that showed the challenges in their application due to computational demands. The SVM model also provided a moderate performance of 77% accuracy. There were trade-offs between model complexity and computational efficiency and the diversity of models showed the importance of model selection based on specific characteristics of data.

One of the main challenges that we came across was the subjectivity and cultural variability in the music genre labels, thus it reflected in the confusion between similar genres. For future direction, further refinement for labeling techniques could be used. Also other feature extraction techniques could enhance model accuracy. Transfer learning accuracy rates could be tried to be improved by using distinct genres. Overall, this project lays a solid foundation for future explorations in music genre classification, providing information about the challenges and limits of the current techniques in handling complex and culturally nuanced data.

7. Appendix

Division of Work

Atilla - Data Accumulation, Background Research, Report Writing

Berk - Data Accumulation, Report Writing, MLP model building
Dila - Background Research, Report and Presentation writing, Transfer Learning
Ege - Background Research, CNN model building
Sezer - Background Research, Report Writing, SVM model building

References

[1] Mallat Stéphane, Compression, *A Wavelet Tour of Signal Processing (Third Edition),* Academic Press, 2009, 481-533, ISBN 9780123743701, https://doi.org/10.1016/B978-0-12-374370-1.00014-8.

[2] "What Is A Spectrogram?" Pacific Northwest Seismic Network. pnsn.org/spectrograms/what-is-a-spectrogram.

[3] F. Zalkow and M. Müller, "CTC-Based Learning of Chroma Features for Score–Audio Music Retrieval," in IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 29, pp. 2957-2971, 2021, doi: 10.1109/TASLP.2021.3110137

[4] "Chroma feature," Wikiwand, https://www.wikiwand.com/en/Chroma_feature (accessed Apr. 21, 2024).

[5] Goksselgunduz, "Fundamental terms of signal processing," Medium, https://medium.com/@goksselgunduz/fundamental-terms-of-signal-processing-2826a1b5543d (accessed Apr. 21, 2024).

[6]M. Chaudhury, A. Karami, and M. A. Ghazanfar, "Large-scale music genre analysis and classification using Machine Learning with apache spark," MDPI, http://dx.doi.org/10.3390/electronics11162567 (accessed Apr. 21, 2024).

[7] Rajeeva Shreedhara Bhat,Rohit B. R., Mamatha K. R, "Music Genre Classification," *SSRG International Journal of Communication and Media Science*, vol. 7, no. 1, pp. 8-13, 2020. *Crossref,* https://doi.org/10.14445/2349641X/IJCMS-V7I1P102

[8] M. F. Türkoğlu, "Support Vector Machine ," Medium, https://mfatihto.medium.com/support-vector-machine-algoritmas%C4%B1-makine-%C3%B6%C4%9Frenmesi-8020176898d8 (accessed Apr. 21, 2024).

[9] E. Işıkhan, "Multi Layer Perceptron (MLP)," Medium, https://isikhanelif.medium.com/multi-layer-perceptron-mlp-nedir-4758285a7f15 (accessed Apr. 21, 2024).

[10] M. Nabil, "Unveiling the diversity: A comprehensive guide to types of CNN Architectures," Medium, https://medium.com/@navarai/unveiling-the-diversity-a-comprehensive-guide-to-types-of-cnn-architectures-9d70da0b4521 (accessed Apr. 21, 2024).

[11] “Music genre classification using CNN,” Clairvoyant, https://www.clairvoyant.ai/blog/music-genre-classification-using-cnn (accessed Apr. 21, 2024).

[12] G. Learning, “Everything you need to know about VGG16,” Medium, https://medium.com/@mygreatlearning/everything-you-need-to-know-about-vgg16-7315defb5918 (accessed May 10, 2024).