Music Genre Classification

BT5153 Applied Machine Learning in Business Analytics – Group 04 Project Report
Anusha Mediboina (A0262847U), Arian Madadi (A0231939X), Asok Kaushik (A0262739U), Hongyu Ren(A0262688M), Naveen Mathew Verghese (A0262734A)

Abstract

Music has become an inseparable aspect of people's daily lives today. With numerous musical genres, it is no surprise that people have varying musical tastes. Genres are extremely helpful for music discovery. They bond fans and listeners and facilitate shared experiences. Hence, the classification and the recommendation of contemporary music in music streaming platforms is up-to-date issue.

Hence, in this project we explore machine learning model for music genre classification to improve business objectives like improving search quality for users, reducing distributor / artists upload time, reduce search time of users and broader exposure to other language artists

1. Objective

In the real world a company can adopt the model and deploy it within an operational pipeline where continuous data collection, model updating and testing should be performed for optimal performance.

- The business success metrics that we are considering are:
- Search time: By reducing the search time of a user, they would be able to increase their time with the platform for their intended purpose of listening to music, rather than searching
- CTR (Click Through Rate): is a good measure to understand how well the genre has been classified. Better accuracy of music genre classification increases the CTR of the keywords of different genres, which in turn increases recommendation quality and customer satisfaction.
- Number of users: By providing accurate genre classification, both user and artist engagement will increase. Platforms can help artists better understand their target audience, which would attract more artists/distributors to the platform to upload their content. It also helps users to discover new artists and songs that fit their preferences.

2. Dataset and Features

Data Collection: We made use of the 10,000 song dataset available on the MillionSongDataset website. This is a freely available collection of audio features and metadata for western popular music pieces. The dataset comes with a set of features such as tempo, rhythm, pitch, timbre etc. We have extracted xx features from hdf5 and arff files that were available on the platform. The website provided HD5 wrappers in python as pytables to extract the features from the hdf5 files. We built a python file to read .arff files. The features extracted from these files are detailed below.

Features:

Features	Desciption
Spectral Centroid	It indicates the "center of mass" of the spectrum. It can be thought of as a measure of the brightness or tonal quality of a sound.
Roll-off Point	The frequency at which the magnitude of the spectral content of a signal fall below a certain threshold
Flux	The rate of change of the spectral content of a signal over time
Variability	Spectral Variability measures the degree to which the power spectrum of an audio signal changes over time.
Compactness	A measure of how closely the energy of a signal is distributed across the frequency spectrum. E.g. a more compact signal has energy distributed more evenly across the spectrum,
Root-Mean-Square	the RMS value is a measure of the average energy of the signal over time
Fraction of Low Energy Windows	measures the proportion of time windows in which the energy of an audio signal is below a certain threshold

Zero Crossing Rate	Zero Crossing Rate is the number of times the audio waveform crosses the horizontal zero axis per second.
Rhythm histogram	the magnitudes of each modulation frequency bin of all critical bands are summed up, to form a histogram of "rhythmic energy" per modulation frequency
Loudness	measures the overall volume of a song, and is expressed in decibels (dB).
Pitch	refers to the perceived highness or lowness of a sound. It is determined by the frequency of the sound wave, where a high frequency corresponds to a high pitch and a low frequency corresponds to a low pitch
Timbre	refers to the tone color or quality of a sound, which distinguishes it from other sounds of the same pitch and loudness.

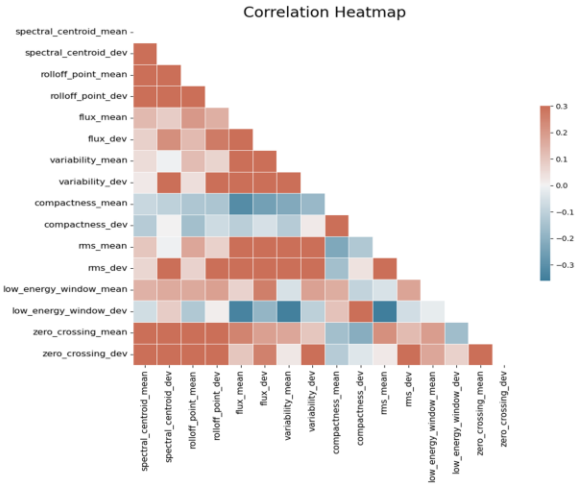
3. Exploratory Data Analysis:

Exploratory data analysis, helped in determining if the dataset is balanced or imbalanced, and whether additional steps were required to be taken to balance the dataset. Examining the audio features of different genres: EDA can help in examining the audio features like loudness of different genres, This helped in identifying whether audio features like loudness are important for distinguishing between different genres. Following are some of the EDA findings:

a) Correlation Heatmap

We can observe from the correlation heatmap that features like spectral_centroid, roll-off point, zero crossing rate are positively correlated with each other. Compactness and RMS are also positively correlated. Also flux and compactness are negatively correlated. Overall, the correlation heatmap suggests that there are relationships between the different audio features considered in this analysis, and these features can be used in combination to accurately classify different music genres.

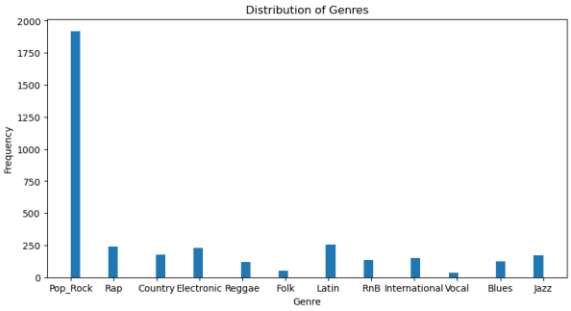
Fig1: Correlation Heatmap



b) Genre Distribution

By plotting a histogram of the genre labels for all the data points, we observe that our current dataset is an imbalanced one with majority of the data points belonging to the 'Pop_Rock' genre. Due to this, it would be required that the samples be balanced across genres during model training using various techniques. To correct this, SMOTE was used to balance the dataset and also 'Vocal' and 'Folk' genres were removed, as they had less than 50 datapoints.

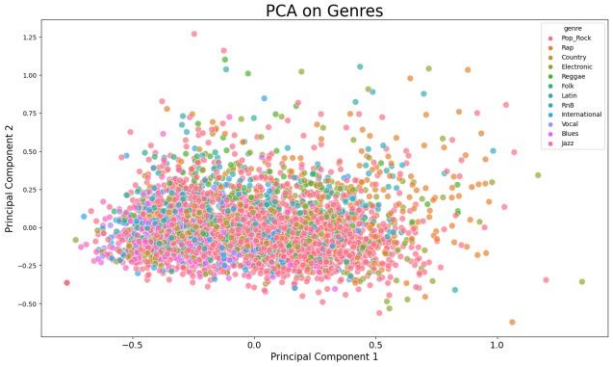
Fig2: Genre Distribution



c) Principal component analysis

Principal Component Analysis is applied to gain insights into the structure of the data. The audio data is first pre-processed using MinMaxScaler to normalize the data. Then, PCA is applied to reduce the dimensions of the data to two principal components (PCs), which would be easy to visualize. The resulting scatterplot shows the distribution of the data points in the two-dimensional space defined by the two principal components, which together account for 80% variation of the entire data. By observing the scatterplot, we can say that the genres do not exhibit any clustering wrt PCs and that the data does not exhibit any clear patterns or structures that can be easily identified through the two principal components.

Fig3: Principal Component Analysis



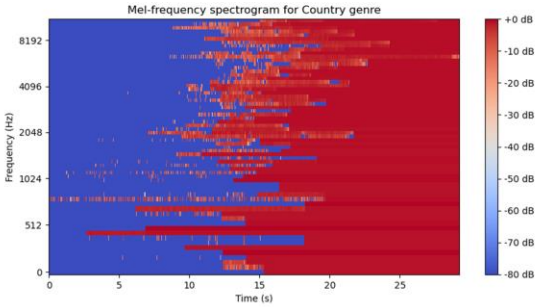
d) Mel Frequency Spectrogram

A Spectrogram is a way to visually represent a signal's loudness, or amplitude, as it varies over time at different frequencies. The y-axis is converted to a log scale, and the color dimension is converted to decibels. In the context of a spectrogram, the darkness of the color represents the magnitude of the spectrogram in decibels (dB), which is a measure of the power or intensity of the sound. Therefore, darker colors in the spectrogram indicate higher power or intensity in that frequency range at that time, which can be associated with louder sound. A mel spectrogram is one where the frequencies are converted to the mel scale. As perception of frequency is not linear. Mel spectrograms take this into account by scaling the frequency axis according to the mel scale, which is a non-linear scale that approximates the human perception of frequency. Mel spectrograms for loudness would differ for different genres due to differences in the characteristics of the music. For some genres, there may be more emphasis on loud, distorted guitar tones, while in others, there may be more emphasis on dynamics and the interplay between different instruments. This could result in differences in the overall loudness profile of the music, as well as differences in the distribution of loudness across the frequency spectrum.

For plotting the spectrograms, code was iterated through all the .h5 files in the 10,000 songs dataset directory and collecting the segment information for those songs that match the track IDs in the dataset having 3953 songs.For each song that matches a track ID in the 3953 dataset, the code extracts the loudness, for each segment using the hd.get_segments_loudness_max() function.The loudness, information for each segment is then stored in separate list The song_id list is used to store the track ID for each song. The mel frequency spectrogram was plotted using the melspectrogram function of librosa library for all genres. The function takes the loudness values of all segments oa a genre and converts them to a power spectrogram using a Fast Fourier Transform (FFT) with a window size of 2048 and a hop length of 512. It also applies a mel filterbank with 128 filters to convert the power spectrogram to a mel spectrogram. Then, it converts the mel spectrogram to a dB scale using the power_to_db function with the maximum value of the spectrogram as the reference. Finally, it plots the spectrogram using specshow and displays the plot with a colorbar. The x-axis represents time in seconds, the y-axis represents frequency in Hz, and the color represents the magnitude of the spectrogram in dB.

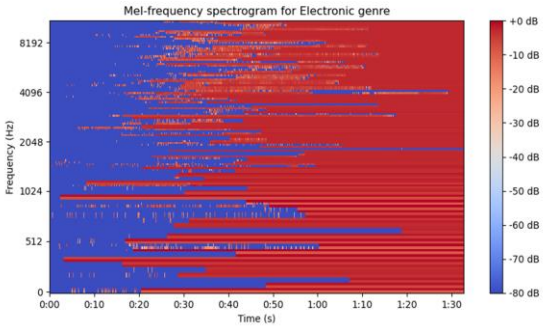
Below is the mel frequency spectrogram for the songs of 'Country'genre from our dataset. Country music often features acoustic instruments such as guitars, fiddles, and banjos, which tend to produce sounds with a relatively narrow frequency range and a limited range of dynamics. As a result, the mel spectrogram of country music shows relatively consistent energy with occasional spikes in energy corresponding to the attacks of individual notes or chord

Fig4: Mel Frequency Spectrogram for Country Genre



In contrast, electronic music often features synthesized sounds that can have a much broader frequency range and greater dynamic range. The mel spectrogram of electronic music shows energy with more complex patterns of spikes and dips corresponding to different sounds and rhythms.

Fig5: Mel Frequency Spectrogram for Electronic Genre



4. Data Preprocessing

The dataset present is more structured. The following pre-processing steps have been applied:

1) Standard scalar feature was used by removing the mean and scaling to unit variance
2) Label Encoder was used to encode the target label ***Genre*** with value between 0 and 9.

The dataset is imbalanced, and the Folk and Vocal genres were removed because they had less than 50 data. The pop_rock genre is the majority class containing around 1900 records, so several techniques need to be applied to address the imbalance of the dataset.

4.1 Train Test Split

To compare the performances between different classification models, train test split is applied to the genre dataset. 80% of the song tracks are kept as a train set while the other 20% are used as a test set. Since the dataset is imbalanced and for ensuring the same class proportion is maintained in test set as well, the stratify parameter available in train test split is used.

Two techniques on improving the performance of imbalanced dataset were used individually.

1) SMOTE is an over-sampling method. It creates synthetic samples of the minority class. The following imblearn python package to over-sample the minority classes was used.

2) Another method is to estimate class weights in scikit_learn by using compute_class_weight and use the parameter 'class_weight', while training the model. This can help to provide some bias towards the minority classes while training the model and thus help in improving performance of the model while classifying various classes.

5. Models Building

5.1 Classification Models Comparison

Table 1: Accuracy and F1score of Trained Models with improving the performance of imbalanced dataset by applying SMOTE in train dataset.

	Accuracy_smote	f1score_smote	Algos
3	0.45	0.47	XGBClassifier
2	0.43	0.46	RandomForestClassifier
1	0.33	0.37	DecisionTreeClassifier
0	0.29	0.34	KNeighborsClassifier

Table 2: Accuracy and F1score of Trained Models with improving the performance of imbalanced dataset by applying Class weight balancing in the models trained.

	Accuracy_classweights	f1score_classweights	Algos
0	0.56	0.46	RandomForestClassifier
1	0.55	0.49	XGBClassifier
2	0.40	0.41	DecisionTreeClassifier
3	0.32	0.33	KNeighborsClassifier

From the observations the XGboost provide the highest Test Accuracy when the model is trained with oversampled dataset and Random Forest provide the highest Test Accuracy when the model is trained with class balance weights. The different model accuracy is also improved overall after adding class weight balance in model training.

The F1 test scores of XGboost is highest for both the imbalance methods applied to the trained model.

F1 score, is a harmonic mean of precision and recall, and it provides a better evaluation metric than accuracy in case of imbalanced datasets. F1 score gives equal weightage to precision and recall and is a good measure when the target variable is imbalanced. So XGboost would be better models compared to rest of the models.

Overall, the accuracy and F1 scores are lower, and the model has not been trained well to predict the unseen data due to imbalance in dataset.

5.2. Deep Learning Models

In our study, we investigated three deep learning models to address the challenge of music genre classification: a feedforward neural network (NN), a convolutional neural network (CNN), and a recurrent neural network (RNN). Our goal was to create models capable of processing the input features we derived

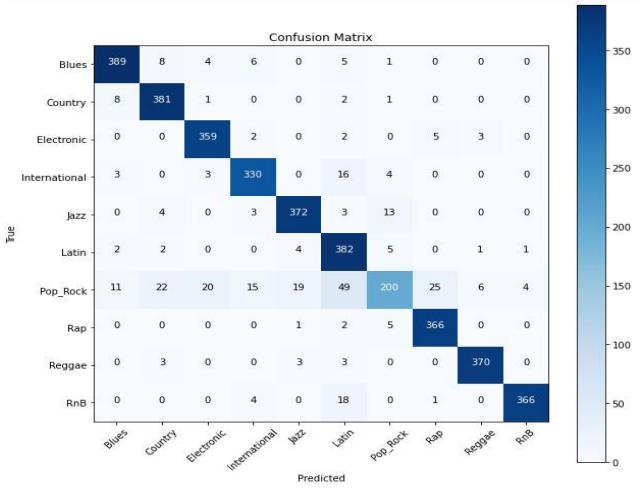
from the audio data in order to accurately predict the song's genre.

The feedforward neural network is composed of three layers of nodes. Each layer processes the input data by applying a weighted sum of the inputs, adding a bias term, and passing the result through an activation function. In our design, the input layer consists of 16 nodes representing the 16 features, followed by hidden layers with 128 and 64 nodes, respectively. We chose these sizes to gradually reduce the number of nodes in each layer, helping the model learn a hierarchy of features and compress information for improved generalization. Both hidden layers employ the ReLU (Rectified Linear Unit) activation function, which introduces non-linearity, allowing the model to capture more complex relationships between features.

The output layer has a number of nodes equal to the genre classes and uses a softmax activation function. This function converts the weighted sums into probabilities that sum to 1, estimating the likelihood of an input belonging to each genre class. We trained the model using backpropagation and gradient descent optimization to minimize the cross-entropy loss. We assessed the feedforward neural network's performance using a validation set and the F1 score metric, achieving a training loss of 0.0178 and a validation F1 score of 0.9084.

Below is the Confusion Matrix for the feed forward Neural Network

Fig6: Confusion Matrix for Feed forward NN



For our alternative approach, we utilized a convolutional neural network. Our CNN design includes three convolutional layers and two fully connected layers. Each convolutional layer applies a set of filters, followed by a ReLU activation function and a max-pooling layer. The first convolutional layer contains 32 filters, while the second and third layers have 64 and 128 filters, respectively. The max-pooling layers reduce the spatial dimensions of the feature maps, aiding in managing the computational complexity of the model.

After processing the data through the convolutional layers, the feature maps are flattened and passed through two fully connected layers. The first layer consists of 256 nodes and a ReLU activation function, while the second layer, the output layer, has a number of nodes equal to the genre classes. Like the feedforward neural network, a softmax activation function is used in the output layer. We trained the CNN using the same loss function, optimization method, and evaluation metric as the feedforward neural network. The CNN yielded a training loss of 0.0008 and a validation F1 score of 0.9560, showing improved performance compared to the feedforward neural network.

As well as the previously mentioned models, we also explored the use of a recurrent neural network (RNN) for music genre classification. Specifically, we employed a Long Short-Term Memory (LSTM) architecture, which is a type of RNN designed to capture temporal dependencies in sequential data effectively.

The dataset we used was different to the other two. For this dataset, we created 60 ticks of rhythm data evenly spaced out within the song. This then creates a rhythm histogram, which shows how the rhythm of the song changes over time. We believe that this temporal dataset contains more information than simply one that utilizes averages, and so an RNN was selected. We started by preprocessing the dataset, removing specific genres, encoding the genre labels, and balancing the class distribution using random oversampling. We then divided the data into training and validation sets and reshaped the input features to fit the LSTM model.

Our LSTM-based model is composed of an input layer, two LSTM layers with 64 hidden units each, and a fully connected output layer with nodes corresponding to the genre classes. The model processes a single feature at each time step.

We trained the model using cross-entropy loss and the Adam optimizer with a learning rate of 0.01. To compute the running loss and weighted F1 score for

each epoch, we defined separate functions for training and validating the model. We trained the model for a total of 20 epochs, monitoring its performance by printing the training and validation loss and F1 scores after each epoch.

With a training loss of 2.4140 and a validation F1 of 0.2532, it didn't perform nearly as well as expected" after this "By employing this LSTM-based RNN architecture, we aimed to capture the temporal relationships in our audio feature data, potentially leading to improved music genre classification performance, however, this wasn't the case."

In conclusion, we developed and compared three deep learning models for music genre classification: a feedforward neural network, a convolutional neural network, and a recurrent neural network. Both the first models exhibited promising results, with the CNN outperforming the feedforward neural network in terms of the validation F1 score, however, the RNN severely underperformed. This is likely due to the data not containing as much information as we had initially anticipated.

5.3. Explainability

XAI, or Explainable Artificial Intelligence, is a set of techniques and tools that aims to make machine learning models more transparent and interpretable to human users. This is crucial in domains such as music genre classification, where machine learning is used to make decisions that can have a significant impact on user experience.

However, complex machine learning models like random forests and XGBoost we used in previous part can be challenging for humans to comprehend. This lack of interpretability can limit their usefulness in the music genre classification domain since users may not trust the predictions of a model perceived as a "black box."

To address this issue, PFI and LIME are two popular techniques for model interpretation. PFI allows us to identify the most critical features in a model, providing insight into the factors driving the model's predictions. LIME, on the other hand, provides local explanations of individual predictions, enabling us to understand how the model is making decisions on specific music tracks.

By using these techniques, we can gain a better understanding of how our random forest and XGBoost models make decisions, which can help us build trust in the models and identify areas for improvement. In music genre classification, this can result in better predictions and ultimately improve the user experience.

1. Global Explanation

We used PFI to further understand the feature importance of Random Forest model and XGBoost model. Both the Random Forest and XGBoost models underwent PFI analysis on their respective training sets. The findings of both models revealed that the top features contributing to genre classification are similar. Specifically, "compactness_mean", "compactness_dev", "rms_mean", "variability_dev", and "rolloff_point_mean" were identified as the most influential features.

Fig7: PFI for Random Forest

Weight	Feature
0.1061 ± 0.0064	compactness_mean
0.0822 ± 0.0047	compactness_dev
0.0404 ± 0.0034	rms_mean
0.0342 ± 0.0051	variability_dev
0.0294 ± 0.0052	rolloff_point_mean
0.0218 ± 0.0038	rms_dev
0.0207 ± 0.0052	flux_mean
0.0183 ± 0.0028	zero_crossing_dev
0.0164 ± 0.0015	rolloff_point_dev
0.0149 ± 0.0027	variability_mean
0.0118 ± 0.0008	low_energy_window_mean
0.0118 ± 0.0029	zero_crossing_mean
0.0086 ± 0.0026	spectral_centroid_dev
0.0078 ± 0.0020	low_energy_window_dev
0.0040 ± 0.0013	spectral_centroid_mean
0.0035 ± 0.0011	flux_dev

Fig8: PFI for XgBoost

Weight	Feature
0.1061 ± 0.0064	compactness_mean
0.0822 ± 0.0047	compactness_dev
0.0404 ± 0.0034	rms_mean
0.0342 ± 0.0051	variability_dev
0.0294 ± 0.0052	rolloff_point_mean
0.0218 ± 0.0038	rms_dev
0.0207 ± 0.0052	flux_mean
0.0183 ± 0.0028	zero_crossing_dev
0.0164 ± 0.0015	rolloff_point_dev
0.0149 ± 0.0027	variability_mean
0.0118 ± 0.0008	low_energy_window_mean
0.0118 ± 0.0029	zero_crossing_mean
0.0086 ± 0.0026	spectral_centroid_dev
0.0078 ± 0.0020	low_energy_window_dev
0.0040 ± 0.0013	spectral_centroid_mean
0.0035 ± 0.0011	flux_dev

2. Local Explanation

We used LIME(Local Interpretable Model-agnostic Explanations) to perform local explanation on the first row of testing data, which is used to provide local explanations for the predictions made by both the Random Forest and XGBoost models. This was performed on the test set, and the results showed that the first testing data was classified as Jazz by both models. Interestingly, LIME revealed that the feature "low_energy_window" had the greatest contribution to the positive score for this classification. These insights provided by LIME can be crucial for understanding how the models are making predictions and can help in improving the interpretability and trustworthiness of the models.

Fig9: LIME for Random Forest

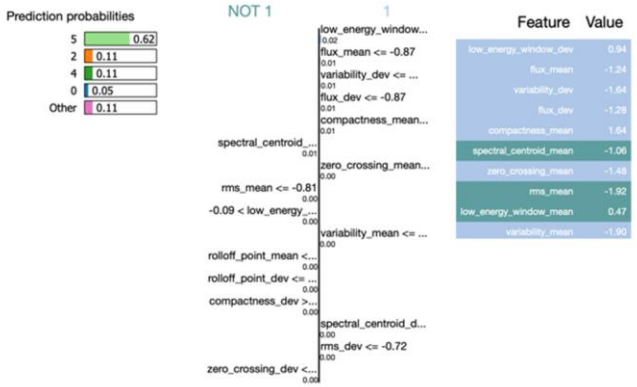
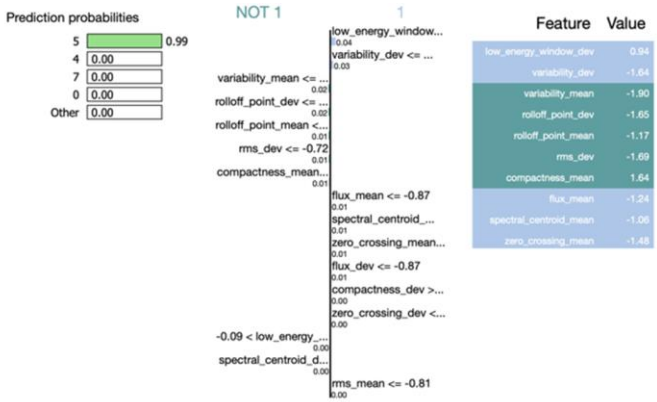


Fig10: LIME for XgBoost



6. Conclusion

Based on our analysis of various classification model of Machine Learning and Deep Learning, we can say that Deep Learning models are able to perform better than other models.

7. References

. References:

Github Link for all the code of this Project: https://github.com/naveenmv24/BT5153_Group4_Final_Project.git

1. http://www.ifs.tuwien.ac.at/mir/msd/

2. http://millionsongdataset.com/

3. http://www.ifs.tuwien.ac.at/mir/msd/download.html#groundtruth

4. http://millionsongdataset.com/pages/code/