**\*Unable to upload the whole project folder to blackboard. Please download it from** <https://exchangelabsgmu-my.sharepoint.com/:u:/g/personal/ckasula_masonlive_gmu_edu/EYxn8zEHMWdLsghuuhIUu0QBrX950FzOCSHE-xobS31OAQ?e=Pauog2>

-------------------------------------------------------------------------------------------------------------------------------------------

**A clear introduction to the problem:**

Music is categorized into different classes called genres. Each of these genres is defined by a style, contains a form, and is influenced by culture. The boundaries between different genres are also very intricate. For example, rap and metal are branches of the rock genre. But they have their own core variants that can generate a hybrid genre, metal-rap. With the increase in the number of genres, its ability to contain a behavior of music will be lost, thereby making the classification task meaningless. Given the complexities, modeling the data to identify, define, and understand the patterns that distinguish the genres is a challenging task.

**Task: Given the lyrics of a song, the task is to categorize it into one of the 11 genres present in the dataset by using Natural Language Processing, Machine Learning and Deep Learning.**

The project focuses on the application of Machine Learning and Deep Learning techniques in the classification of a song to its genre by only using its lyrics. Specifically, Machine Learning models such as Naïve Bayes and Logistic Regression were tried followed by the application of Convolution Neural Networks in a seq2seq paradigm. Various ways to develop models in a large data set with a huge variation in the class distribution is included. The trained models are evaluated, and their performance is compared.

The proposed CNN model was identified as the best performer. The CNN model, Logistic Regression model and the Naïve Bayes model (trained on oversampled dataset) have performed better than the baseline models.

To perform the classification task, the dataset is extracted from Kaggle. It consists of 380,000+ lyrics associated with 20,000 artists and various genres arranged by year. The folder structure is artist/year/song. Every artist folder has a genre.txt that specifies the genre of the musician.

The features of the dataset are as follows:

• Song: Name of the song

• Artist: The singer of the song.

• Year: The year of song release.

• Genre: The type/class of the song.

• Lyrics: The lyrics of the song in text.

The total number of genres in the dataset are 11 in number, which are, R&B, Electronic, Other, Pop, Indie, Metal, Jazz, Hip - Hop, Country, Rock, and Folk.

**Point by point outline of solution**

Four approaches have been proposed to tackle the current task.

Before understanding the approaches, the features are to be noted.

Unigrams and Bigrams are considered as features for the Machine Learning algorithms (Naïve Bayes and Logistic Regression). Word vectors extracted from the Glove embeddings are used for the Deep Learning models. For the raw dataset, the number of uni-grams and bi-grams (features) that are generated by the TfidfVectorizer are 4,048,833 in number. For the sampled data distribution, the number of uni-grams and bi-grams (features) generated by the same vectorizer are 3,731,150 in number.

1. **Approach I:**

The raw data set has been used in this approach. The train and test sets are split in the ratio of 80:20 respectively. K-fold cross-validation has been performed for every model to determine the best set of hyperparameters and to analyze the model’s performance in different parts of data. For training the model, uni-grams, and bi-grams are generated as features of the data set. For each n-gram, term frequency and inverse document frequency is computed. This way, each lyric is represented by a sparse vector.

The training and testing performed in the K-fold cross validation results in the identification of the optimal set of hyperparameters for the Naïve Bayes and Logistic Regression classifiers. The saga algorithm is used as the optimization technique for the Logistic Regression model.

The files related to this approach are:

**Baseline Logistic Regression Model:** Nonsampled\_baseline/LogisticRegression\_NonSampling.ipynb

**Baseline Naïve Bayes Model:** Nonsampled\_baseline/Naive\_undersampling.ipynb

1. **Approach II:**

In the second approach, the sampled dataset was used. The same process in Approach I has been repeated. The same classifiers were trained on the current data set and its performance is evaluated on the test set.

The files related to this approach are:

**Logistic Regression Model**: OverSampling/Logistic\_oversamp.ipynb

**Naïve Bayes Model:** OverSampling/NaiveoverSamp.ipynb

1. **Approach III:**

In this approach, the sampled data set was used. An experiment was performed to check whether Parts of Speech (POS) data can help in increasing the accuracy of the models. Hence, each lyric is processed through the pos\_tag function of NLTK to identify the Parts of Speech tag for each word in the lyric. The number of nouns, verbs, adjectives, adverbs, and other Parts of Speech in each lyric is calculated and appended as a feature to the tf-idf matrix generated during training. A total of 39 POS related columns are generated making the total number of features 3,731,189. The resultant POS features are normalized in the range of [0,1] by column. Normalization by column is necessary because the magnitude of numbers in the POS features will be very large when compared to the tf-idf values of the generated n-grams. In such a case, during the training process, the tf-idf values of the n-grams will have a negligible effect over the model in predicting the target class. Hence, normalizing the newly formed POS columns is mandatory.

The Naïve Bayes and the Logistic Regression classifier were trained on the new data with the hyperparameters obtained through cross-validation. The trained models are later used to predict the genres for the test records.

These models did not show any improvement but instead underperformed when compared to the other models in other approaches.

The files related to this approach are:

**Logistic Regression Model**: POS/Datapreprocessing\_rockreduced.ipynb

**Naïve Bayes Model:** POS/Naive\_Bayesandlogi.ipynb

1. **Approach IV:**

In this approach, the sampled data set was used. Preprocessing the data and converting it to sequences from text is necessary for a seq2seq paradigm. A word-index dictionary is constructed such that for each unique word in a vocabulary, a number is assigned depending on the frequency of its occurrence. The smaller the number, the higher the frequency of occurrence. The text to sequences function from the Keras library is used to convert each lyric in the train set into an array of numbers that represent the sequence of words in that lyric.

Word embeddings are used as features for the deep learning model. Global Vectors for Word Representation famously called as GloVe embeddings contain a global array of 300-dimensional vectors for many words in the literature. Unlike the Word2Vec models, GloVe provides a vector for a large variety of words. Hence, GloVe was preferred over Word2Vec embeddings.

CNNs were selected to be applied for this task. The longest lyric in the data set consists of 3432 words/tokens. After converting the lyrics to sequences, padding is performed for all the lyrics to match the length of the largest lyric which is 3432. The first layer is the input layer which consists of 3432 neurons. The embedding layer then replaces each word/token with 300-dimensional vectors extracted from the GloVe dictionary. A series of convolution operations followed by max-pooling operations are performed through different layers. The number of neurons in the first, second, third, fourth, and fifth layers is 500, 400, 300, 200, and 100 respectively. MaxPooling1D(5) was performed after every convolutional layer. SpatialDropout1D(0.2) was introduced in the architecture to prevent overfitting. ‘ReLU’ was used as the activation function in the convolutional layers. ‘Adam’ is selected as the optimizer and ‘categorical cross-entropy’ was used as the loss function. The last layer is the dense layer which consists of 11 neurons as there as 11 genres and they use softmax as their activation function. The model was trained for 30 epochs over the train set and its performance was tested over the test set.

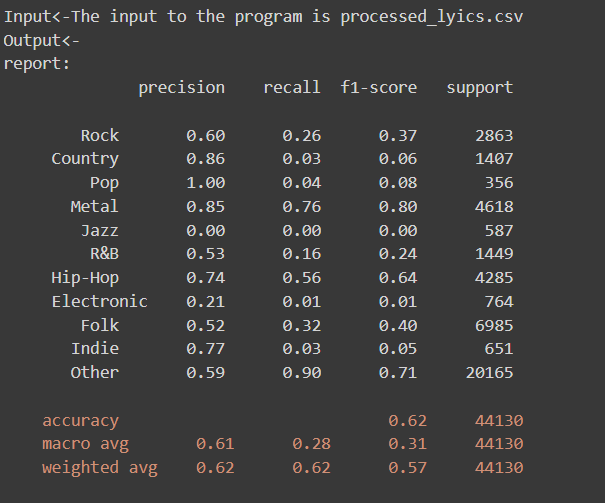
The files related to this approach are:

**Deep Neural Network Model:** NeuralNetwork/oversampled/NeuralCNN\_oversam.ipynb

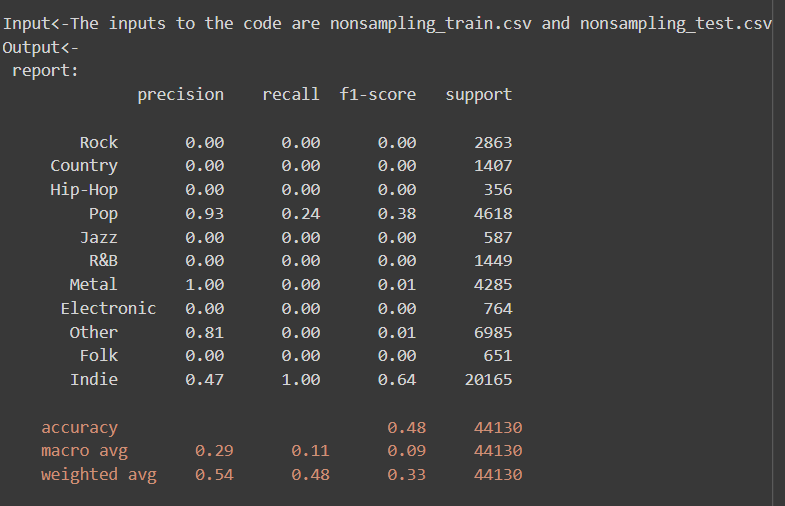
**Examples of actual program input and output**

**Input and Output for Approach I:**

**Logistic Regression**

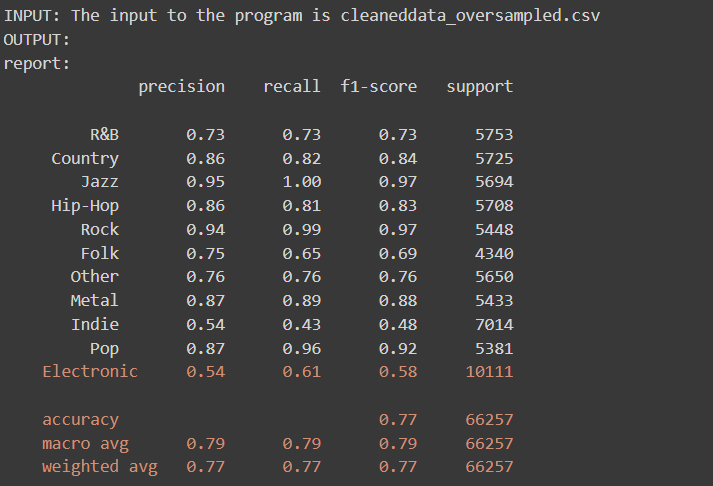


**Naïve Bayes:**

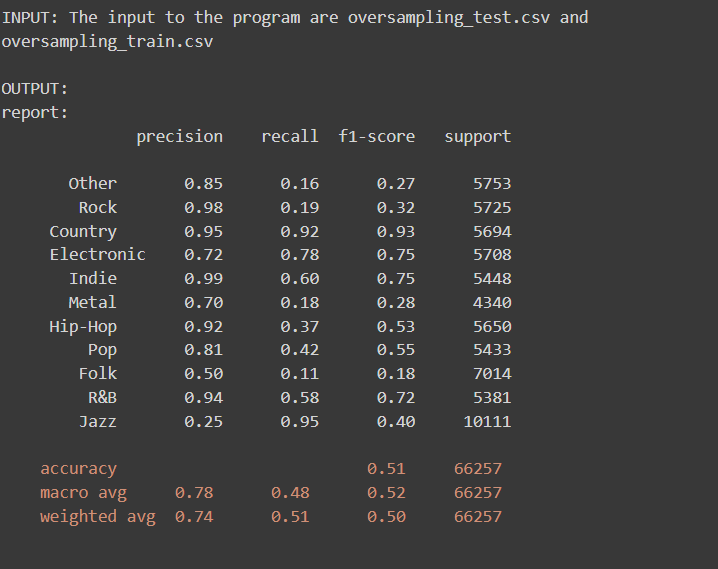


**Input and Output for Approach II:**

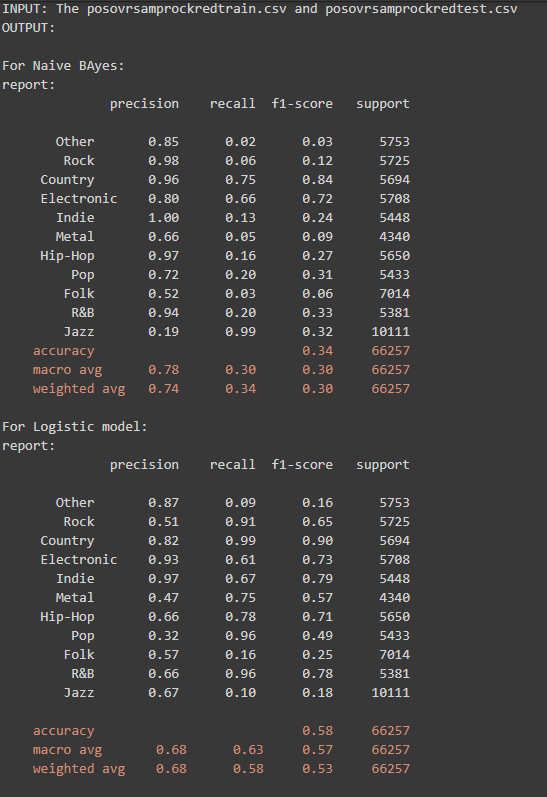
**Logistic Regression:**



**Naïve Bayes:**



**Input and Output for Approach III:**



**Input and Output for Approach IV:**

