Music Genre Classification using various Machine Learning methods

Arnova Abdullah
Communication Systems and Networks (Master)
Cologne University of Applied Sciences
Cologne, Germany
arnova.abdullah@smail.th-koeln.de

Serkan Kutludag
Communication Systems and Networks (Master)
Cologne University of Applied Sciences
Cologne, Germany
serkan.kutludag@smail.th-koeln.de

Soumya Sambeet Mohapatra
Communication Systems and Networks (Master)
Cologne University of Applied Sciences
Cologne, Germany
soumya_sambeet.mohapatra@smail.th-koeln.de

Adnan Abbas
Communication Systems and Networks (Master)
Cologne University of Applied Sciences
Cologne, Germany
adnan.abbas@smail.th-koeln.de

***Abstract*—Music has been always a part of the human history from the beginning. The music songs differ from each other in terms of style, in other words, genre. Machine Learning can be used for both classification of the genre and to understand the reason behind it. Different machine learning techniques with different preprocessing and some additional implementations are done in this project to see the performance of these models for recognizing music genre. In this paper, the classification models are trained on the Spotify's audio feature dataset.**

***Index Terms*—machine learning, classification, training, encoding, clustering**

I. Introduction

Music is a way of expressing yourself in an abstract way and it has always played a major part in shaping people's emotions. Throughout the ages, different kinds of music have been created, some being very similar and some being distinct. This similarity/distinction is defined as the genre of the music. However, what are the border lines between different genres? Is there any mathematical formulation that gives the correct genre for a given music? Even though the answer is not too obvious, machine learning can be used to learn the relationship between genres. According to the Feng [1], Deep Learning with parallell CNN and Bi-RNN works highly effective for music genre classification. In addition, there are lots of different ML models which also give as promising result as DL. In this project, different preprocessing techniques are used to observe their benefits. Different ML models are used to compare and select the best one out of all. Additionally, some extra techniques like hard voting system, semi-supervised learning technique are also used to increase score at the end. All these analysis are implemented with the help of the Scikit-learn [2] library.

II. Setup

A. Data collection

The data required for creating the machine learning models was obtained from Spotify. In order to send requests to the Spotify API [3] , a spotify developer account was created to obtain a client ID and a client secret. Using these credentials, an authorization token was created. This authorization token was then used to send requests to the "Get Tracks' Audio Features" API end point. A total of 129179 data points consisting of 115 different genres and 19 different features was obtained.

B. Data cleaning

Upon initial inspection of the dataset, the features `type`, `id`, `uri`, `track_href`, `analysis_url`, `year` and `duration_ms` were identified to be of no importance as they contain only the meta information about the tracks. Hence these features were removed. Secondly, the dataset consisted of 115 genres. For the purpose of this paper, we reduced the number of target genres to 10. The selected genres were `acoustic`, `blues`, `classical`, `country`, `dance`, `jazz`, `metal`, `pop`, `rock` and `techno`. The data cleaning process was carried out with the help of the Pandas [4] library in Python.

C. Feature selection and preprocessing

The heatmap of correlation matrix was examined to see the correlation between different features. Based on this, we divided the feature selection and preprocessing into 3 different cases:

- **Case 1:** The feature `key` was removed because it was the least correlated amongst the others. The features `energy` and `loudness` were also dropped as they were highly correlated with `acousticness` as seen from the heatmap of the correlation matrix.
- **Case 2:** Since the genre values are categorical, they were label encoded. Additionally, one-hot encoder was applied for the columns `mode`, `key` and `time_signature` since these features are categorical in nature. For the re-

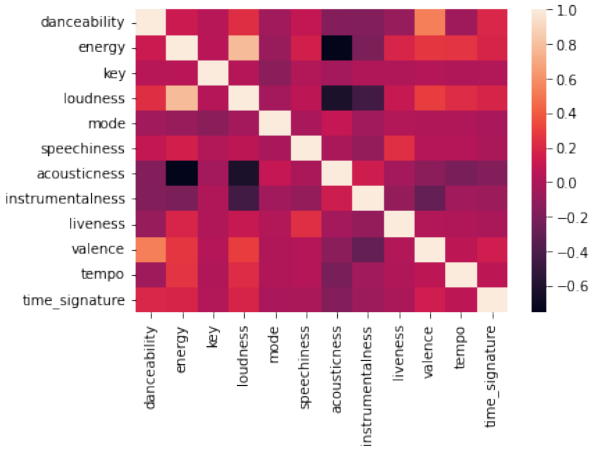

Fig. 1. Heatmap of correlation matrix

maining numerical columns, MinMax Scaler was applied. This resulted in the dataset having 27 columns.

- **Case 3:** In this case, it was decided to preserve all the features because removing feature decreased accuracy of the machine learning model. Moreover, since our target feature was categorical, the correlation matrix might not have been the correct metric based on which feature selection can be made.

III. Model Training

A. Decision Tree Classifier

A Decision tree [5] is a very versatile machine learning model that make predictions based in a tree like layout. It begins at a root node where it tests whether a condition is satisfied. If the result is true, then it moves to the left child node and if it is false, it moves to the right child node. At the new node, it again tests the condition at that node. This continues until a leaf node is reached and this node gives the final result.

A Decision Tree Classifier was trained preserving all the 12 audio features and the target feature was set to 'genre'. An initial training resulted in a training set accuracy score of 92.3% and test set accuracy score of 35.8%. The cross validation accuracy score for 3 folds were 36.6%, 36.4% and 35.8%.

To improve the model, a grid search was used to find the set of hyperparameters that produced the most optimal performance. From the grid search, the best parameter set obtained is as follows:

- `max_depth: 8`
- `max_features: None`
- `min_leaf_samples: 9`
- `min_samples_split: 3`

Using these hyperparameters, a new Decision Tree classifier was trained. The accuracy scores for the tuned classifier were 47.8% and 42% on the training and test set respectively. The cross validation accuracy score improved to 42.5%.

From these performance metrics, it can be concluded that the Decision Tree Classifier does not perform so well in classifying genres when it is trained with the Spotify dataset.

B. Random Forest Classifier

Random forest [5] is an ensemble supervised machine learning method that works by creating a set of decision trees from a randomly selected subset of the training set. It gives the final prediction by taking majority voting from different decision trees. Random forest is used widely in various applications because it is extremely easy and efficient to train it. Random Forest can reduce the overfitting of data by introducing extra randomness when growing trees.

For traiing the Random Forest Classifier, in addition to the dataset consisting of 10 genres, an additional dataset consisting of 6 genres was also created. The 6 genres were `acoustic`, `classical`, `jazz`, metal, `rock`, `techno`. This was done to compare Random Forest's classification accuracy with reduced number of gernre.

While training this model, the features were encoded as described in **Case 2**, however none of the features were removed. Stratified train-test splitting was done with 10000 instances in the training set and 2000 instances in the test set. For the additional dataset with 7200 instances and 6 genres, the training set contained 6000 instances, and the test set contained 2000 instances. Applying Random Forest in the dataset with 12000 data points gave an accuracy score of 91.87% on the training set and 49% accuracy score on the test set. Similarly, the Random Forest was created with the dataset consisting of 6 genres and it gave 97.87% accuracy score on the training set and 65.58% accuracy on the test set.

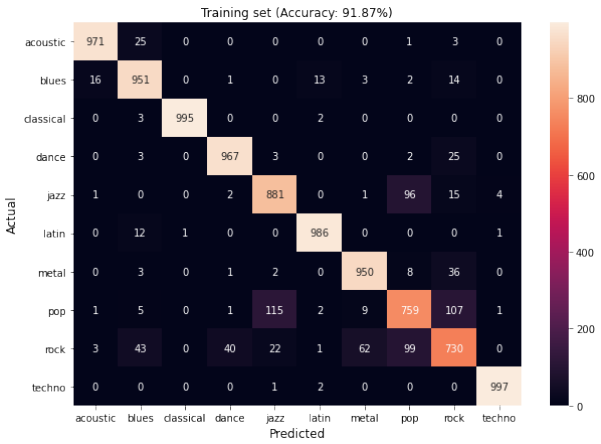

Fig. 2. Confusion matrix for RF on training data

It was observed clearly that RF was overfitting the data. To avoid this, cross-validation and hyperparameter tuning were implemented. The parameters for cross-validation are adjusted, the number of folds is set to 10, allowing shuffling of each class's sample before splitting and `random_state` is set to 42. The adjustable parameters for The RF model are `n_estimators`, `max_features`, `max_depth`,

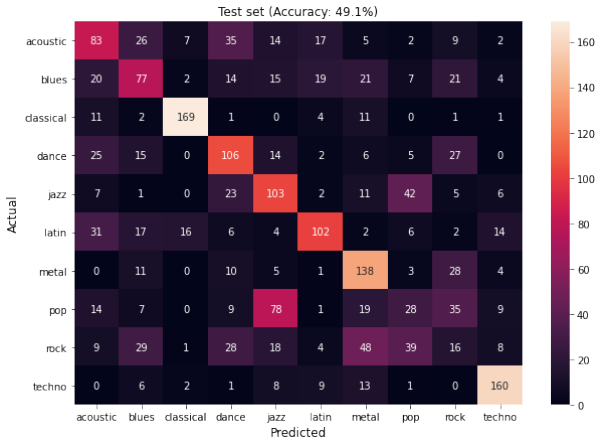

Fig. 3. Confusion matrix for RF on test data

`min_samples_split`, and `min_sample_leaf`. The number of trees in the forest (`n_estimator`) is varied between 150, 200, 250, and 300. The maximum depth of the tree (`max_depth`) is varied between 20, 26, 32, and 38. The maximum feature (`max_feature`) is set to default (`sqrt`).

From the grid search, the best parameter set obtained for the model trained on dataset with 10 genres is as follows:

- `max_depth: 32`
- `max_features: 'sqrt'`
- `min_leaf_samples: 7`
- `min_samples_split: 16`
- `n_estimator: 300`

Using these hyperparameters, a new Random Forest classifier was trained. The accuracy scores for the tuned classifier were 49.7% on the test set. Similar hyperparameter tuning for the model trained on dataset with 6 genres resulted in a test set accuracy score of 65.75%. Keeping all the features gives approximately 1% increase in accuracy but it increases computing time in Grid Search. In our case, the computation time is 1556.052 seconds.

Some genres are very similar and so, while predicting those genres, accuracy score gets decreased. In the next section a semi-supervised technique is implemented to identify the best genres for classification.

C. Other models

In addition to Decision Tree and Random Forest classifier, several other models were also trained. The models were trained on preprocessed dataset as descirbed in **Case 1**.

- The accuracy of LR on training set was 0.467 whereas on test set is 0.458.
- The accuracies of KNeighborsClassifier on the training and testing set were 0.429 and 0.414 respectively.
- For the case of SVM, the accuracy on training and testing were 0.462 and 0.456 respectively.
- The accuracies of AdaBoostClassifier on training and testing were 0.444 and 0.436 respectively.
- The accuracies of GradientBoostingClassifier on training and testing were 0.598 and 0.473 respectively.
  The best score in this dataset is derived with the hard voting scheme combination of Logistic Regression, Random Forest, GradientBoostClassifier and SVM. This gave an accuracy of 0.552 on the training set and 0.480 on the test set.

IV. Semi-supervised approach

After training and evaluating several models, we found that the accuracy was still low. One of the main reasons for this was that the genres that was selected initially were having very strong similarities amongst each other. The genres were selected by intuition based on their current popularity. However, technically some genre were very difficult to distinguish by the machine learning models. For example, the audio features of the genre rock have strong similarities with that of pop, techno has strong similarity to metal and classical is very close to country. In order to improve the classification accuracy, we therefore needed to select a set of genres that are very distinct.

In the final attempt, we took a semi-supervised learning approach to identify genres that are distinct from each other. Our initial idea was to identify clusters of similar genres using the KMeans [5] method. But this method failed to recognize any clusters due to the nature of the Spotify dataset. Therefore, we we manually defined grid of clusters and selected the genre that has highest count in that grid.

For our purpose of creating distinct cluster regions, we required comparison of two features that have the data points spread through out the X-Y plane so that every grid would contain uniform number of data points. From the scatter matrix of the dataset, the comparison between the features `acousticness` and `valence` were chosen to be the best candidate for creating grids. The two features were segmented into 9 different grids and the genre having the highest count in each grid was identified.

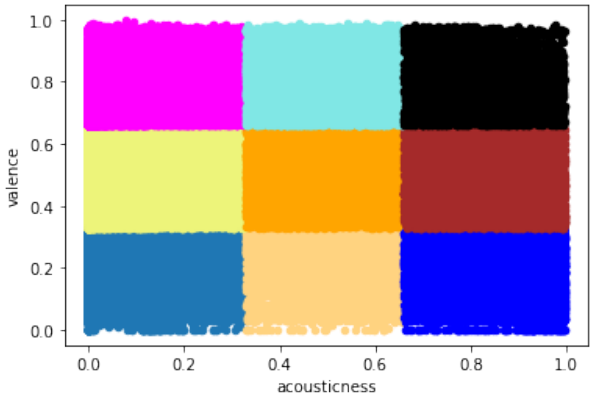

Fig. 4. Manually defined Grid cluster

From the grids, the following genres were identified to have maximum count: `black-metal`, `mandopop`, `classical`, `grunge`, `sertanejo`, `tango`, `reggae`, `salsa`, `tango`. The rows containing these genres were extracted from the

original dataset. The new dataset contained 9200 instances. A train-test split was created with 20% ratio and the training set was used to retrain a Random Forest classifier.

With the new model, an accuracy score of 99.99% was obtained on the training set and 80.70% on the test set. The 3 fold cross validation scores obtained were 81.37%, 79.9, 81.08%.

From the performance metrics obtained with the new list of genres, it can be concluded that if the genres are distinct enough, the model is capable to classify new data with much higher accuracy.

V. Results

TABLE I
ACCURACIES OF VARIOUS MODELS

Training Models	Training set accuracy	Test set accuracy
Decision Tree	47.84%	42.08%
Random Forest	64.9%	49.7%
Logistic Regression	46.7%	45.8%
KNC	42.9%	41.4%
SVM	46.2%	45.2%
ADA	44.4%	43.6%
GBC	59.8%	47.3%

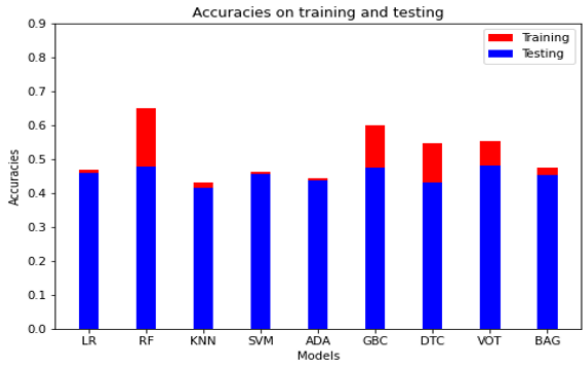

Fig. 5. Accuracies of various models

VI. Conclusions

In this paper, various machine learning methods to classify genre of music were discussed. From the results, it can be concluded that the Random Forest model is the most suitable model for this purpose. It was also seen that the selection of target genres affected the accuracy of the model. Therefore, for higher accuracy, it is necessary to select the genres which are very distinct and different from each other.

References

[1] Lin Feng, Shenlan Liu, and Jianing Yao. Music genre classification with paralleling recurrent convolutional neural network. *arXiv preprint arXiv:1712.08370*, 2017.

[2] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[3] Spotify web api. https://developer.spotify.com/documentation/web-api/reference/#/.

[4] The pandas development team. pandas-dev/pandas: Pandas, February 2020.

[5] Aurélien Géron. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems.* " O'Reilly Media, Inc.", 2019.