International Journal of Advanced Computational Engineering and Networking, ISSN(p): 2320-2106, ISSN(e): 2321-2063
Volume-7, Issue-8, Aug.-2019, http://iraj.in

MUSIC GENRE CLASSIFICATION ON AWS CLOUD PLATFORM

[1]ASTHA BARANWAL, [2]PRIYA.V

[1,2]School of Information Technology and Engineering, VIT, Vellore, India
E-mail: [1]astha.baranwal2016@vitstudent.ac.in, [2] vpriyacse@vit.ac.in

Abstract - Music genre classification is a trending topic in the music industry. Genre classification is crucial to the music industry as it plays a vital role in song recommendation systems. This paper uses the AWS Cloud platform's machine learning model to classify songs into genres (viz. rock, pop, blues, country, classical and hip-hop) based on the predicted scores. AWS Machine Learning models are fast and easy to build. Stochastic Gradient Descent is used for classification for the AWS Machine Learning multi-class model. The predicted scores for each genre for a song can help determine multi-labeled classification for a song as well. Real-time classification of songs is possible through the AWS Machine learning model created. Random forest model is built for the dataset using Python machine learning libraries. This model is then hyper parameter tuned and the performance metrics are calculated. F1-scores for the two models are then compared.

Keywords – Machine learning, Multi-classification, Cloud Computing, Multi-label classification, Stochastic Gradient Descent, Random Forest, AWS Machine Learning

I. INTRODUCTION

Genre classification is a very subjective topic. Based on the approach, songs might get classified into genres differently by different people. It is fairly simple for a human being to identify the genre of a song. One thinks about how fast the beat of the song is, the mood of the song and the ambiance it creates. All these help create a mental picture of the song and thus the genres associated with it are determined.

The motivation behind this paper is to find songs that belong to a particular genre based on attributes like, beats, keys, bars, tatums, loudness etc. The goal here is to build a model that determines the genre of the song based on these attributes. Genre classification is crucial to the music industry as it plays a vital role in song recommendation systems. Song recommendations systems work in a way that help listeners find similar music. Automated systems for genre classification are gaining momentum in this age. [1]

A. Multi-class Classification

The dataset used in this paper has six mainstream music genres as class labels. They are rock, pop, blues, country, classical and hip-hop. The classification models built classify the songs into one of these genres. Since there is more than one class to be classified, the models built are multi-class classification models. [2]

B. Multi-label Classification

The AWS ML provides real-time prediction of genre through its prediction API. This API gives prediction results which contains predicted Scores map, having the predicted scores for all the genres. The higher the score, the greater is the vote for a particular genre. Hence, this can be used to determine if a song can be classified into two or more genres based on predicted score. This is calledmulti-label classification. Since genre classification of songs is often ambiguous, multi-label classification of songs is important. [3][4]

C. Stochastic Gradient Descent

Stochastic Gradient Descent (SGD), also known as Incremental Gradient Descent, uses an iterative method in order to optimize a differentiable objective function. Since the samples are selected in a random order from the training dataset, it is called 'stochastic'. [5]

D. Random Forest

Random forest is an ensemble learning model of multiple decision trees for classification and regression that operates by creating several decision trees at the time of training the model. It outputs the class that is the mode of the classes for classification or gives the mean prediction (in case of regression) of the individual trees. [6]

E. AWS Cloud ML

Amazon Machine Learning is an AWS service for building ML models within a few minutes and generating predictions. Amazon ML enables the usage of powerful machine learning technologiesand there is no need of having an extensive knowledge of machine learning algorithms and techniques. It has provisions for robust system and services for hosting Jupyter Notebook on Amazon EC2 instance.

II. LITERATURE SURVEY

Music genre classification has been a trending topic for many years. Costa et al. [7] proposed a convolution neural network model for the classification of songs on the Latin American music (LMD database). Goulart et al. [8] presented various approaches for music genre classification. They did feature selection followed by classification into three main styles of music: blues, classical, and lounge.

Vlegels et al. [9] measured music taste based on artist preferences using a ground-up technique by two-mode network. Zhao et al. [10] used bi-partite graph for computing the users' similarity by genres weight relations. Venrooij et al. [11] compared the effects of genre ambiguity present in albums for the two subfields of pop music genre taking genre fuzziness into account. Huang et al. [12] proposed a self-adaptive harmony search algorithm for genre classification. They also used the Support Vector Machine (SVM) for classification by intensity, pitch, timbre, tonality, and rhythm of a song. Fu et al. [13] proposed the bag-of-features approach in which a song is split into numerous frames and then a feature vector is extracted from each of the local frames.

The proposed system classifies the songs into six genres, namely, rock, pop, blues, country, classical and hip-hop using SGD and Random forest models. A predicted Score for each genre is given which can help in multi-label classification of songs.

III. METHODOLOGY

A. The Dataset

The dataset taken is a subset of the Million Song dataset from the CORGIS dataset project. It has around 10 k instances and 32 attributes. It contains instances from the Million Song Dataset, which is a collaboration between the EchoNest and LabROSA about one million popular contemporary songs.

Attribute	Description	Datatype
Song_id	Unique ID of the song	string
Song_name	Name of the song	string
Album_id	Album ID of the song	int
Album_name	Name of the Album the song was released in	string
Artist_id	Unique ID of the artist	string
Artist_name	Name of the artist	string
Song_popularity	Popularity of song	float
Artist_popularity	This corresponds to how much buzz the artist is getting right now. This is derived from many sources, including mentions on the web, mentions in music blogs, music reviews, play counts, etc.	float
Familiarity	This corresponds to how well known the artist is. You can look at familiarity as the likelihood that any person selected at random will have heard of the artist.	float
Duration	Duration of the song in seconds	float
Bars_confidence	Confidence value (between 0 and 1) associated with each bar by The Echo Nest	array float
Bars_start	Start time of each bar according to The Echo Nest. It denotes the number of bars in the song	array float
Beats_confidence	Confidence value (between 0 and 1) associated with each beat by The Echo Nest	array float
Beats_start	Start time of each beat according to The Echo Nest. It denotes the number of beats in the song	array float
End_of_fade_in	Time of the end of the fade in, in seconds , at the beginning of the song	float
Key	Estimation of the key the song is in	int
Key_confidence	Confidence of the key estimation	float
Location	Location of the artist	string
Latitude	Location latitude of artist	float
Longitude	Location longitude of artist	float
Loudness	General loudness of the track in dB	float
Mode	Major or minor mode	int
Mode_confidence	Confidence measure of mode	float
Start_of_fade_out	Start time of the fade out, in seconds, at the end of the song	float
Tatums_confidence	Confidence value (between 0 and 1) associated with each tatum	array float
Tatums_start	start time of each tatum (smallest rhythmic element)	array float
Tempo	Estimated tempo in BPM (Beats Per Minute)	float
Terms_freq	Frequency of tags of artists in the Echo Nest API	array float

Time_signature	Estimate of number of beats per bar	int
Time_signature_confidence	Confidence of the time signature estimation	float
Year	Year of release	int
Genre	The genre of the song	string

Table 1: Dataset Description.

Genre is the target variable while all others is the independent variables on which Genre depends on.

B. Platforms, Softwares and Languages

1) AWS Cloud

An Amazon S3 bucket is used to store the dataset and multi-class classification model is built using AWS ML service. The AWS ML model is fast and is built within 3 minutes. It provides a real-time prediction API for predicting genres based on input records. An EC2 instance is used to deploy the Jupyter Notebook on which the Random Forest model is coded.

2) Python and Anaconda

Python machine learning libraries like NumPy, pandas, matplotlib, scikit-learn, etc. are used to build the random forest model and hyperparameter tune it.

Anaconda is a professional platform which can be used as a GUI as well as a console. Anaconda is a Python distribution which brings a lot of useful libraries with it, which are not included in Python standard library, like Jupyter Notebook.

C. The System Architecture

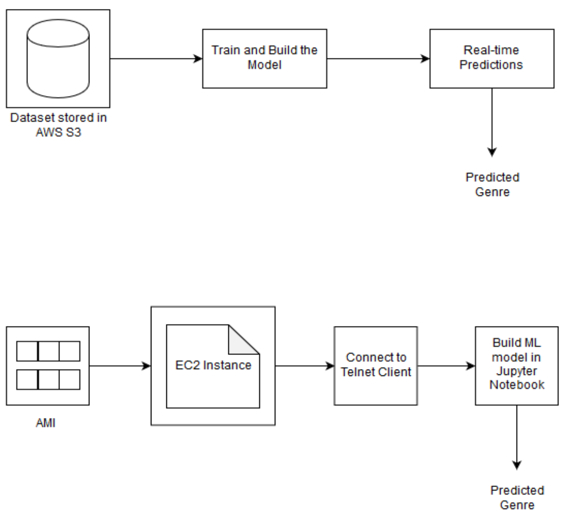

Fig. 1: The cloud architecture for the system.

The dataset had around 400+ genres which have been distributed into six mainstream genres i.e., rock, pop, blues, country, classical and hip-hop in order to create machine learning models for this paper. The missing/ null valued instances are either removed or the values have been filled by the mean.

There was no clear correlation found between the attributes after data visualization and analysis.

Feature engineering is done and irrevant attributes for classification are dropped. AWS ML is used to build the stochastic gradient descent model and python is used to build random forest model which is then hosted on the cloud using EC2. Performance metrics are used to analyze the model performance.

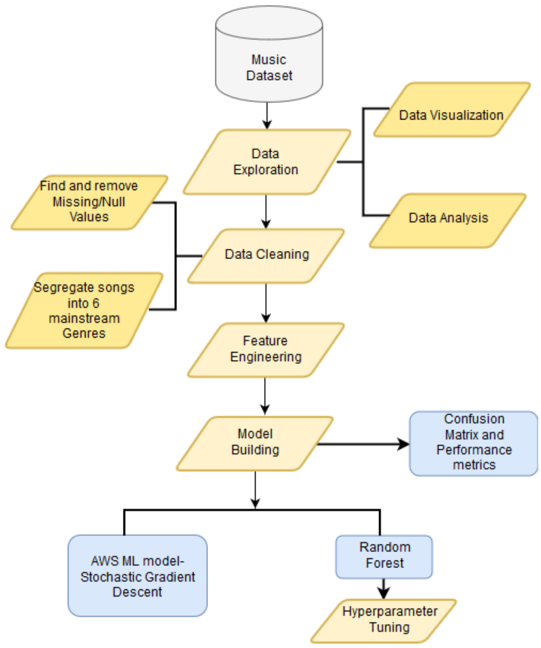

Fig. 2: The proposed system for song classification.

D. Stochastic Gradient Descent Model

AWS ML is used to build this model.

1) The Model

Pseudocode of Stochastic Gradient Descent model:

1. Divide the music dataset into train and test dataframes
2. Randomly shuffle instances in the training dataset
3. For i=1, 2, …, n, do:
   $x := x - \eta \nabla f B(x)$
4. Repeat steps 2 to 4 until an approximate minimum is obtained.
5. Obtain the predicted results (predicted genres) for each instance present in the test data

2) Real-time Prediction

Any ML model created with Amazon Machine Learning can be queried for predictions in real-time with a Predict API. The Amazon ML gets the request with values of the independent variables and the prediction is made for the dependent (target) variable.

The Predict APItakes a single input observation in request payload, and the prediction is synchronously returned in response. Any new song's genre can be predicted using this API.

3) Multi-label Classification

The predicted scores can be used for multi-labeled classification of songs. The strength of prediction related to each class is determined by examining the predicted Scores map. The higher the score of a class is in this map, the stronger is the prediction related to the class. The class with the highest value of predicted Scores is ultimately selected as the predicted Label for the song.

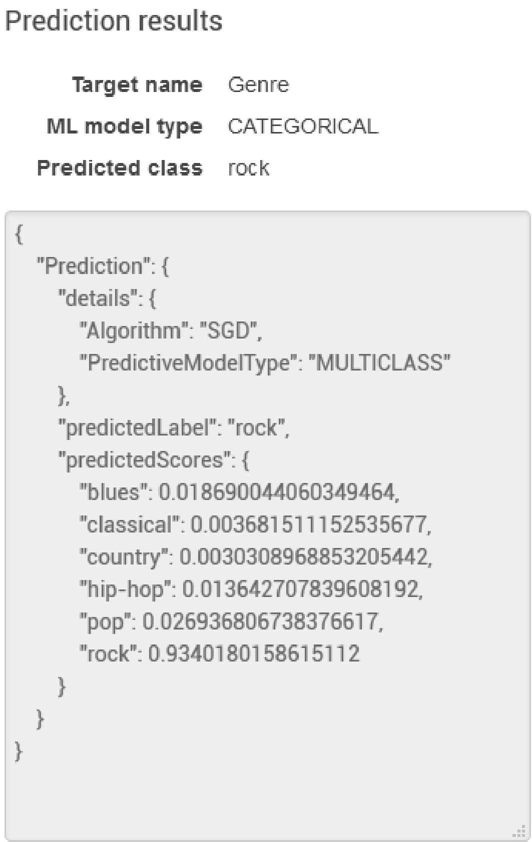

Fig. 2: Prediction results for real-time prediction.

Here, the model classified the song into the rock genre because of the highest predicted score for rock label at 0.9340. The second highest predicted score is for the pop genre at 0.0269 and hence, the song can be classified as 'rock pop' and so on. This is multi-labeled classification.

E. Random Forest Model

The model is built and then hyperparameter tuned to gain maximum accuracy. An EC2 instance is used to run the Jupyter Notebook on cloud and the model is built.

1) The Model

Pseudocode of Random Forest model:

1. Divide the music dataset into train and test dataframes
2. From the total 'm' features, randomly select 'k' features, where k<m
3. Among the k features, calculate the node 'd' using the best split
4. Split the node into daughter nodes through the best split
5. Repeat steps 2 to 4 for all the other instances of the dataset to construct single decision tree
6. Repeat steps 2 to 5 to build a forest with 'n' number of trees
7. Obtain the predicted results (predicted genres) for each instance present in the test data
8. Cross validate predicted genres with actual genres
9. Calculate the accuracy of the model through confusion matrix

2) Hyperparameter Tuning

Models are hyperparameter tuned to obtained the maximum accuracy possible. After hyperparameter tuning, the optimal accuracy is found at: n_estimators=1800, max_depth=140, max_features='auto'. The accuracy rate increased from 0.39 to 0.49.

IV. RESULTS AND DISCUSSION

A. Performance Metrics

True Positive (TP) – The instances which are classified as positive and are actually positive.
True Negative (TN)– The instances which are classified as negative and are actually negative.
False Positive (FP)– The instances which are classified as positive but are actually negative.
False Negative (FN)– The instances which are classified as negative but are actually positive.

1) Accuracy

It is a metric used to predict the correctness of a machine learning model.
Accuracy = (TP + TN) / (TP +FN + TN +FP)

2) Precision

It is the ratio of true positives to the sum of the true positives and false positives.
Precision = TP / (TP + FP)

3) Recall/ Sensitivity

It is the ratio of true positives to the sum of the true positives and false negatives.
Sensitivity = TP / (TP + FN)

4) F1-Score

F1 score is a combination function of precision and recall. It is used when we need to seek a balance between precision and recall.
F1-Score = 2 * (Precision * Recall) / (Precision + Recall)

B. AWS Machine Learning Model Results

1) The Confusion Matrix

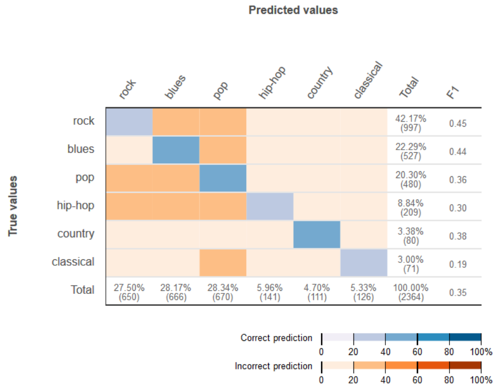

Fig. 3: The confusion matrix and classification report for the SGD model created using AWS ML.

The model gives a F1- Score of 0.353.

C. Random Forest Model Results

1) The Confusion Matrix

```
Model Accuracy: 0.39
Confusion Matrix
[[333  52  14  24 113  87]
 [ 73  81   8   9  47  24]
 [ 80  24  27   5  28  26]
 [ 51  23   2  38  53  60]
 [176  46  19  35 243 147]
 [165  21  13  21 154 298]]


Classification Report
             precision    recall  f1-score   support

      blues       0.38      0.53      0.44       623
  classical       0.33      0.33      0.33       242
    country       0.33      0.14      0.20       190
    hip-hop       0.29      0.17      0.21       227
        pop       0.38      0.36      0.37       666
       rock       0.46      0.44      0.45       672

avg / total       0.38      0.39      0.38      2620
```

Fig. 4: The confusion matrix and classification report for the untuned Random Forest model.

```
Model Accuracy: 0.49
Confusion Matrix
[[348  34   5   3 128 105]
 [ 56 103   3   1  54  25]
 [ 57  21  28   0  44  40]
 [ 31   7   0  35  78  76]
 [101  40   5   6 343 171]
 [ 93   5   4   3 141 426]]


Classification Report
             precision    recall  f1-score   support

      blues       0.51      0.56      0.53       623
  classical       0.49      0.43      0.46       242
    country       0.62      0.15      0.24       190
    hip-hop       0.73      0.15      0.25       227
        pop       0.44      0.52      0.47       666
       rock       0.51      0.63      0.56       672

avg / total       0.51      0.49      0.47      2620
```

Fig. 5: The confusion matrix and classification report for the hyperparameter tuned Random Forest model.

The model gives a F1- Score of 0.47, which is higher than that of the AWS ML model created using Stochastic Gradient Descent, which had F1-Score of 0.353 only.

Random Forest model	Accuracy	Recall	Precision	F1-Score
Not Tuned	0.39	0.39	0.38	0.38
Hyperparameter Tuned	0.49	0.49	0.51	0.47

Table 2: Performance metrics for unturned and hyperparameter tuned Random Forest models.

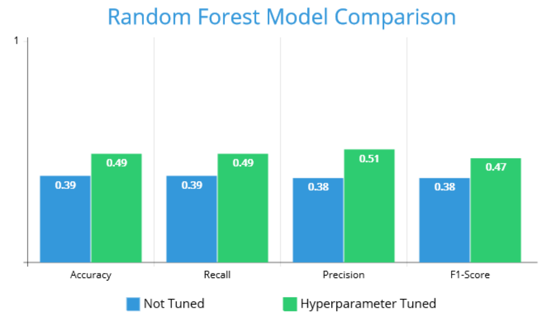

Fig. 3: Comparison between the two random forest models.

V. CONCLUSION

Music genre classification always has been a challenging topic. This paper proposed two models for classifying songs using AWS Stochastic Gradient Descent machine learning model and Random forest model with hyperparameter tuning. The Random forest model out-performed the AWS ML SGD model with an F1-score of 0.47. However, it took only 3 minutes for the creation of the machine learning model on AWS Platform whereas, the random forest model required hours of coding. This is one reason why cloud applications are really fast and easily deployable without any prior knowledge about algorithms and techniques. The model even created a prediction API for classifying new songs based on genre. Multi-label classification is also possible through the AWS model.

REFERENCES

[1] Cory McKay, Ichiro Fujinaga,"Musical genre classification: Is it worth pursuing and how can it be improved?," Published in ISMIR 2006.

[2] Muhammad Asim Ali, Zain Ahmed Siddiqui, "Automatic Music Genres Classification using Machine Learning," (IJACSA) International Journal of Advanced Computer Science and Applications,Vol. 8, No. 8, 2017.

[3] Sanden, Chris; Zhang, John Z., "Algorithmic multi-genre classification of music: an empirical study," Proceedings of the International Computer Music Conference 2011, University of Huddersfield, UK, 31 July - 5 August 2011.

[4] Sanden Sergio Oramas, Oriol Nieto, Francesco Barbieri, Xavier Serra, "Multi-Label Music Genre Classification From Audio, Text, and Images Using Deep Features," Proceedings of the 18th International Society of Music Information Retrieval Conference (ISMIR 2017).

[5] Léon Bottou, "Large-Scale Machine Learning with Stochastic Gradient Descent," Proceedings of COMPSTAT'2010, pp 177-186.
[6] Sanden Sergio Oramas, Oriol Nieto, Francesco Barbieri, Xavier Serra, "Multi-Label Music Genre Classification From Audio, Text, and Images Using Deep Features," Proceedings of the 18th International Society of Music Information Retrieval Conference (ISMIR 2017).
[7] Yandre M. G. Costa, Luiz S.Oliveira, Carlos N.SillaJr., "An evaluation of Convolutional Neural Networks for music classification using spectrograms," Applied Soft Computing, Volume 52, March 2017, Pages 28-38.
[8] Antonio Jose HomsiGoulart, Rodrigo CapobiancoGuido, Carlos DiasMaciel,"Exploring different approaches for music genre classification," Egyptian Informatics Journal, Volume 13, Issue 2, July 2012, Pages 59-63.
[9] Jef Vlegels, JohnLievens, "Music classification, genres, and taste patterns: A ground-up network analysis on the clustering of artist preferences,"Poetics, Volume 60, February 2017, Pages 76-89.
[10] Daozhen Zhao, Lingling Zhang, WeiqiZhao, "Genre-based Link Prediction in Bipartite Graph for Music Recommendation," Procedia Computer Science, Volume 91, 2016, Pages 959-965.
[11] Alex van Venrooij, VaughnSchmutz, "Categorical ambiguity in cultural fields: The effects of genre fuzziness in popular music," Poetics, Volume 66, February 2018, Pages 1-18.
[12] Yin-FuHuang, Sheng-MinLin, Huan-YuWu, Yu-SiouLi, "Music genre classification based on local feature selection using a self-adaptive harmony search algorithm," Data & Knowledge Engineering, Volume 92, July 2014, Pages 60-76.
[13] ZhouyuFu, GuojunLu, Kai MingTing, DengshengZhang, "Music classification via the bag-of-features approach", Pattern Recognition Letters, Volume 32, Issue 14, 15 October 2011, Pages 1768-1777.

★★★