Turkish Online Journal of Qualitative Inquiry (TOJQI)

Volume 12, Issue 3, July 2021:2494- 2510

Research Article

Machine Learning in Music Genre Classification

Akshat Ramanathan[1], Priyanshu Srivastava[2], R. Jeya[3]

Abstract

This paper compares machine learning algorithms in how they classify audio signals into musical genres. A literature review of previously present techniques and approaches is also done, based on feature engineering and algorithms. Unique dataset is formed using an online GTZAN music database, with sampling and other techniques to balance classes. The audio analysis library is used for extracting key features from the samples.

Analyses are then performed, for each feature and genre. Classifiers are created based on machine learning concepts (SVM and NN) and training and testing are performed on algorithms, with the dataset made. In the end, a wholistic process evaluation is conducted (an assessment of the features and dataset chosen and other parameters of the project) to infer an outcome.

Keywords: *SVM - Support vector machine, NN - neural network.*

[1] UG Scholar, Department of Computer Science and Engineering, SRM Institute of Science and Technology, Chennai, India, ar7782@srmist.edu.in

[2] UG Scholar, Department of Computer Science and Engineering, SRM Institute of Science and Technology, Chennai, India, pr4903@srmist.edu.in

[3] Assistant Professor, Department of Computer Science and Engineering, SRM Institute of Science and Technology, Chennai, India, jeyar@srmist.edu.in

Received: 23.06.2016, Accepted: 03.10.2016

Introduction

A genre of music is a category that identifies pieces of music to a set of preexisting conventions. It is separated based on musical form and musical style.

Nowadays, Music Classification is used by companies to suggest recommendations for their customers (like Shazam, Spotify) or to simply be available as an endproduct (for example SoundCloud). Music Genres Recognition is the starting point for such scenarios. Machine Learning techniques are useful in extracting data trends and patterns from the large pool of information. Similar ideologies are used for Music Analysis.

Machine Learning is the core of this research. Songs are analyzed based on their digital signatures including tempo, acoustics, etc. to answer that old question: What type of music are you into?

Classification in Machine Learning is a Supervised learning approach in where the program learns from the data given to it and make new observations or classifications. This process involves categorization of structured and unstructured data into specific classes.

These are often referred to as labels, target, or categories. It can be either a binary classification or a multi-class problem. Classification modeling is the task of approximating the mapping function from input to output variables. The purpose is to classify the new data into specific categories.

The focus of this paper is to compare Machine Learning algorithms ability to classify songs into the correct musical genre. For this process a short audio sample is required. The higher the “expertise”, the classification is more accurate.

To summarize, when datasets are easily “separable” from one another, machine learning classifiers are able to categorize genres accurately.

State of the Art (Literature Survey)

Machine Learning for Music Genre Classification

Bryan Lansdown [1] proposed a model in 2019 which compares machine learning algorithms ability to classify songs. Various candidate datasets are evaluated and an explanation of pre-processing the data is and size reduction is explained. An explanation of the extracted features and their analysis is presented. After this, a presentation of the results of each classifier. It was based on MFCC, ZCR and other Spectral spread timbral texture feature.

Music Genre Classification using Machine Learning Algorithms: A Comparison

Snigdha Chillara, Kavitha A S, Shwetha A Neginhal, Shreya Haldia, Vidyullatha K S [2] proposed a model in 2019 that classifies music into genres and prove that there exists a solution in which automatic genre classification is based on different features, instead of the need for manual entry of the genre. A good accuracy was also reached and hence the model classifies new music into its genre correctly.

Music Genre Classification using Machine Learning Techniques

Hareesh Bahuleyan experimented on a paper in 2010 [3]. In this study the classification approach is based on providing tags to the songs present in the database. The tags present help form classification labels and improve accuracy. The first approach uses Convolutional Neural Network. The second approach uses algorithms like Logistic Regression, Random forest etc., where features from time and frequency domain of the signal are used. The extracted features like Mel-Frequency Cepstral Coefficient and other Spectral features are used to classify the music into its genres.

The common Inference of the above studied surveys are listed below –

1. The survey papers consist high use of computation power for calculation and analysis of features.
2. Requirement of high bandwidth for the searching and selection of data is more in these papers.

3. The papers also used methods which are expensive to implement and test.
4. Complex data integrity and repeated processing of data can be found in many of these survey papers.
5. Lack of organization in resource management also weighs in on making the above-mentioned survey papers lack the accuracy.

Proposed Work

Our proposed system consists of 4 phases –

1. In the first phase, the required songs are downloaded and an audio analysis tool is used to extract the features and store them into the system locally.
2. The second phase consists of splitting the dataset into training and testing samples for multiple data. The Machine learning algorithms and the models are then verified using the test data with the help of appropriate graphs and convolution matrices.
3. In the third phase, a comparison of all the algorithms and models is carried out and the best suitable model is declared.
4. The last phase consists of using this best suitable model to classify and recognize the genre of an unknown input audio signal. The features extracted from this audio signal are fed into the model as parameters for genre recognition.

Data-set Creation Using Extraction of Features

After downloading the GTZAN dataset audio files for 6 genres, a python script is written to extract the required features, as depicted in Fig.1. The genres to be classified for this paper include, Classical, Blues, Hip-hop, Metal, Pop and Reggae. The features to be extracted are Spectral Centroid, Zero Crossing Rate, Spectral Bandwidth, Spectral Contrast and Spectral Rolloff. The average and standard deviation values of each of these features helps us find the onset points in the audio signal. 13 types of Mel Frequency Cepstral Coefficient values are also extracted and stored in the system.

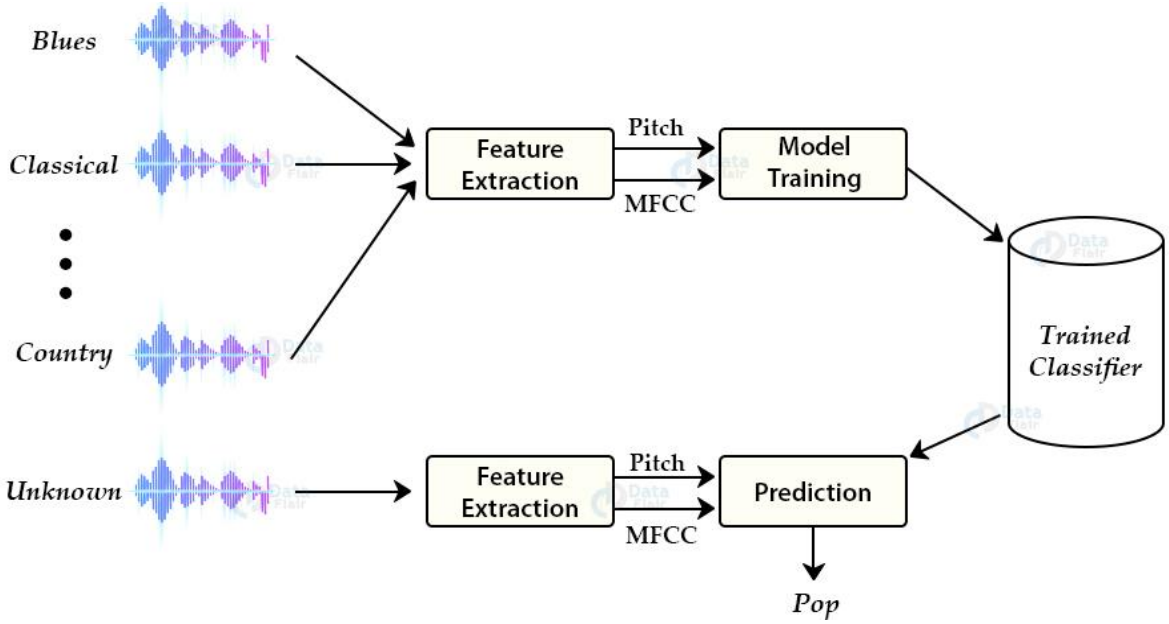

Fig. 1. Project System Architecture.

Model Training and Normalization of Data

The extracted data are then normalized using Min-Max Scalar into required range (-1,1) and are transformed to scale the data and learn its parameters. Here, the models learn the mean and standard deviations of the audio signals. These features are normalized and added to the dataset. This is done so that our model is not biased towards a particular feature of the dataset and also prevent it from learning features of our test data. The Machine learning models used to train for this paper include, K-Nearest Neighbor classifier, Support Vector Machine classifier, Random Forest classifier and Multi-layer Perceptron neural network. These algorithms consist of 36 extracted features as parameters to train and test the models.

Comparison and Testing

The output from each of these models are compared with test data genre present in the data-set prior to splitting. A confusion matrix is made for each of the algorithms to determine their accuracy and precision. A graph is also plotted for K-NN classifier for the best neighbour value of k as well as for the best estimator value for the Random Forest classifier. Post-comparison, the best model suitable for Music Information Retrieval is declared. Primitive testing and results indicate that SVM is the algorithm most suitable for this task, with music accurately classified genre.

Genre Recognition

An unknown audio signal sample is taken as input and the required features are extracted in a similar manner. The scaling is applied to this test data as well to remove bias from our model. These features are taken as input for the above-mentioned suitable model and genre of the audio signal is classified.

The benefits of the proposed system include –

- Moderate use of computational power but provides better accuracy.
- Cost effective and Cheaper to implement.
- Less complex resource management with simple modules.
- Reduced Redundancy and computational complexity.

Implementation

This project is based on 6 major music genres namely, Metal, Pop, Reggae, Hip-hop, Blues and Classical.

The database created consists of music in the waveform format (.wav) and their features (easy to categorise due to lack of digitally produced sound for better results). These songs are processed using audio analysis library - Librosa for pre-extraction of features and stored into the database itself. The database is then scaled and transformed to fit as parameters for the Machine learning algorithms.

The confusion matrix is generated to compare accuracy and precision. With graphs for appropriate algorithms The features to be extracted vary based on model to be trained and hence include, Time Domain Features - Zero Crossing Rate, Central Moments and Frequency Domain Features – Spectral Centroid, Spectral Roll-off, Spectral Contrast and Spectral Bandwidth, MFCC (Mel Frequency Cepstral Coefficients).

Analog signals are sine waves (continuous) that vary in strength (amplitude) or frequency (time). Alternatively, binary signals (0’s and 1’s) used in computer processing, cannot take

any fractional values. Digital waves are quantified from analog signals (from computers), to capture data at discrete moments in time. Sampling rate is defined as the speed at which the audio data is captured. In this study, we use 22.05 KHz sampling rate, it means that we use 22050 samples per second as described by the GTZAN dataset as well as the Librosa tool default documentation.

Using the Librosa library tool, we load the audio signal into our project. In audio processing, we operate on one frame at a time using a constant frame size and hop size. Frames are typically chosen to be 10 to 100 ms in duration and the hop size determines the rate at which the frames change. In our project we use the frame length of 2048 frames (number of samples in a frame) with hop length at 512 samples (length between samples). We compute features from frames using these constants present in a configuration file.

Segmentation of information in a signal can be done with less frames if we can find the most valuable points. These valuable points in a signal are called **ONSETS**. **Onset Detection** is a fundamental task in Music Information Retrieval (MIR) that consists of detection of important events in the signal. These events are our Onsets. We detect an onset using the Librosa library, that gives us the set of inception points of a sound, or the earliest moment at where it is detected.

The digitized signal only gives us Amplitude and time information. Thus to extract the required features from the audio signal, we use Fast Fourier Transform on the audio signal. The Fourier Transform operation is used to change time-domain signal into frequency domain. The audio signal is represented as a sequence of samples and hence requires conversion. The frequency domain is the position of sinusoidal waves with different frequencies, magnitudes, and phase.

As seen in Fig.2, the data is used with a MinMax Scalar and is transformed to fit the model. After scaling of the ranges, the values are fit to learn the mean and variance of the data.

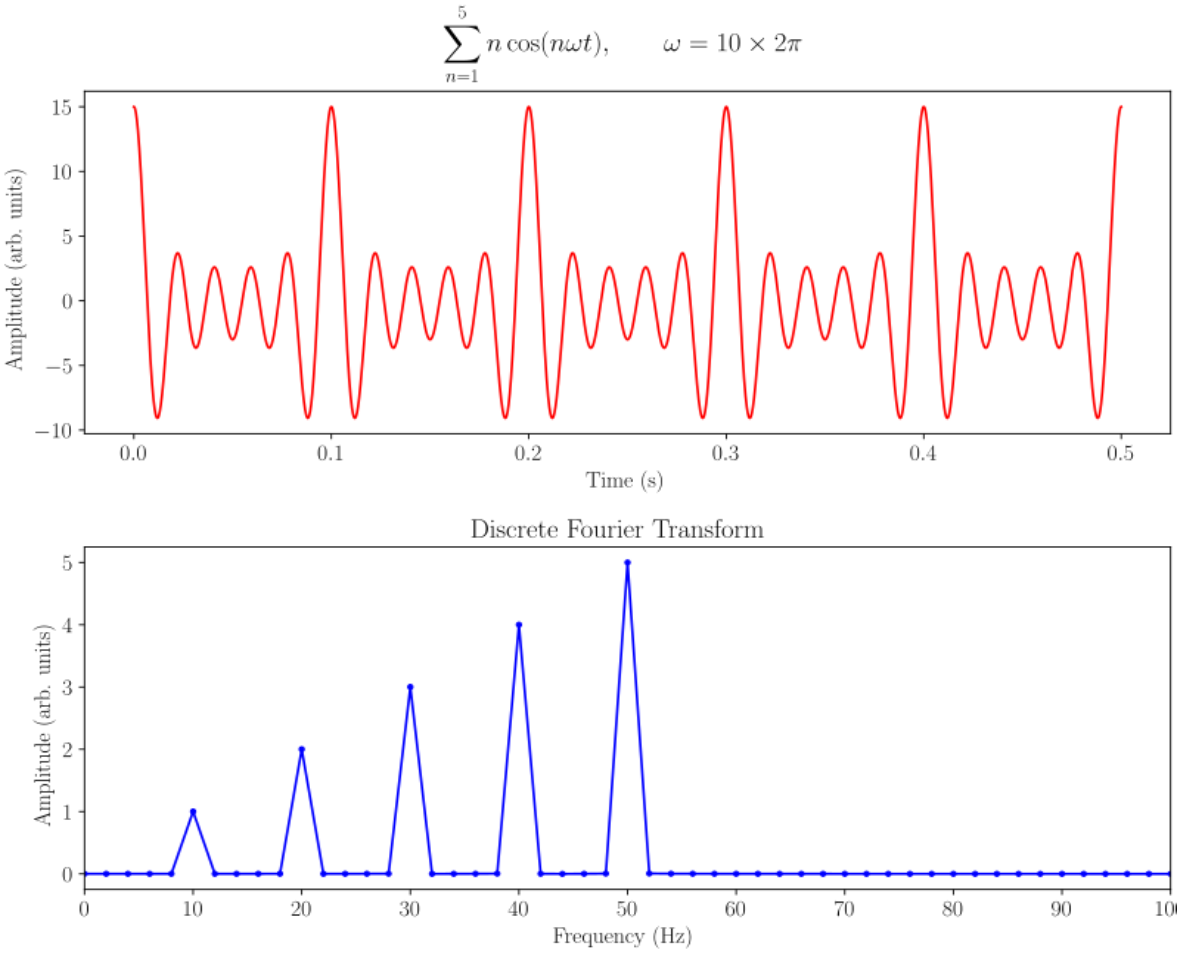

Fig. 2. Discrete Fourier Transform

The fit method is used to calculate these for each features of the data using its mean and variance. The discrete cosine transforms of the list of mel log powers are taken as signal input for our Machine learning Models. Our Mel-frequency Cepstral Coefficient data is the amplitudes of the resulting spectrum.

The trained model uses this information to make certain assumptions and calculations which is required to make a classification class (genre). During testing, similar process is done to the input audio signal and the feature data extracted from test signal is taken as input for the model. The model outputs the most similar pattern and gives the output class (genre) of the test file with appropriate graphs for estimators and neighbours as well as with confusion matrices. Testing of multiple files and comparing them with their original value helps us calculate the accuracy score of the trained model.

Classifications are made by using several machine learning algorithms. The algorithms present in this paper are mentioned below –

K-Nearest Neighbors

K-nearest neighbors relies on labeled input data for learning a function that will generate an appropriate output when we provide it with a new unlabeled data.

It takes a data point and then calculates the distance between the k numbers of data points which are labelled, and classifies the point based on the number of votes obtained from the nearest k-data points. The ‘supervised’ part is the training data as we already know what genre the song is labelled before testing.

KNN is a simple machine learning algorithm which is based on supervised learning. It stores the available data and then uses it to classify new data points based on similarities. K-NN is mostly used in classification problems although it can be used for regression problems too. K-NN is a non-parametric classifier which compares the 'k' nearest training points to the test points and classify based on the majority of these 'k' that are the nearest neighbors. High dimensionality results to decrease in effectiveness, therefore it is suggested to avoid high dimensional inputs.

Module Description –

- The first major step to classify genre using KNN involves feature extraction. Various different features are extracted that will be used to train the model. Approximately 36 features are extracted.
- The second step is training the feature vectors obtained using the K Nearest Neighbor classifier.
- The KNN makes predictions and get the test data accuracy.
- The output value derived is the genre-classified value.

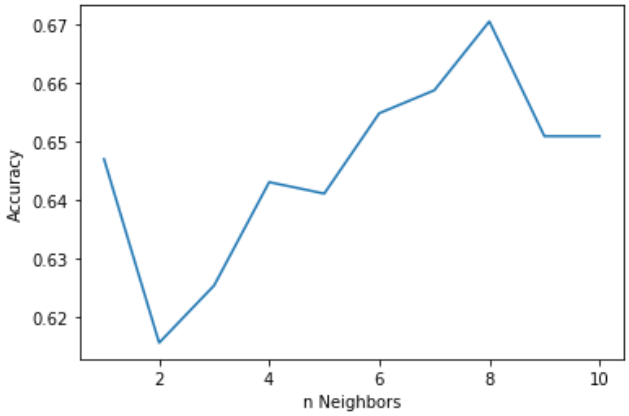

Fig. 3. K-Nearest neighbor Model Accuracy Comparison Graph.

K-NN classifier compares the 'k' nearest training points to the test points and classify based on the majority of these 'k' that are the nearest neighbors. Various Studies suggest that an increasing 'k' tends to tip the accuracy to the expected value of the sample. Studies also shows that an even 'k' has less accuracy than when 'k' is odd around it, K+1, K-1. There is no right way to selecting the optimal 'k' value. A small k value might lead to unstable decision boundaries. On the other hand, a bigger k value will smoothen out these decision boundaries. So, we select a random k value and plot error rates as you increase the k till you find the value with the least error rate.

Random Forest

Random forest is a classification model that works by forming a set of decision trees during training and giving the class output that is the statistical mode of all the individual tree classes. Random decision forests, as compared to decision trees, correct the overfitting of data in the training dataset. Hence Random forests perform better than decision trees in terms of accuracy. However, data characteristics can affect their performance.

Module Description –

- Each sub-tree consists of entropy calculation based on certain decisions of the individual Decision Tree.
- The predictions are the genre output values of all the sub trees.

- The output value is the final genre-classified value which is the most recurring value of the individual decision trees.

Estimators are nothing but an equation which helps pick the best or nearly accurate data model based on real scenario observations. They are not general estimations but are the equations that evaluates a given quantity i.e., the “estimand” and generates an estimate. This generated estimate is then fed to the machine learning classifier to determine what action to take, in this case, which genre to be presented as output.

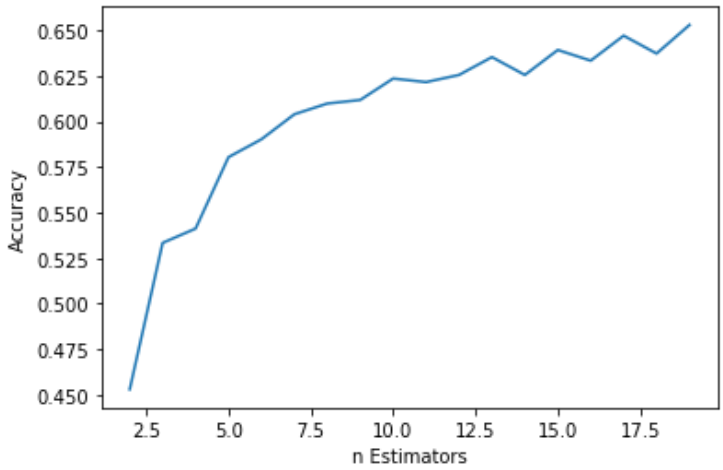

Fig. 4. Random Forest Model Accuracy Comparison Graph

Estimators are really useful in reducing errors and hence also in increasing the accuracy of the model. Various studies show that by increasing the number of estimators we can see a gradual decrease in error values. While true, after a certain point increasing estimators might not decrease the error value as much, so we can say that there is a threshold for the number of estimators that can be used to increase accuracy. We can use trial and error to find the best number of estimators, and in this case, we calculate the value of 19 as the highest estimator count that offset improving the accuracy of out model.

Support Vector Machine

The main objective of the support vector machine algorithm is to create a hyperplane that divides the data points into N-dimensional classes (N — the number of features). To separate

the data points, multiple planes are possible and hence the one that fits all the classes accurately must be chosen.

The required hyperplane maximizes the margin (the distance between both class data points). Maximizing the margin distance helps increase the accuracy of the test data.

Module Description -

- Input Vector is the extracted and calculated metadata.
- The Kernel function consists of space/time decomposition, and Principal Component Analysis of extracted features.
- The output value is the genre-classified value.

Multi-Layer Perceptron Neural Network

A multi-layered perceptron or MLP is a common neural network models that is used in the field of deep learning. MLP is a simpler neural network model than most models use.

However, the techniques that has been introduced by MLP has further paved a path for more advance forms of neural networks. An MLP model, like a human brain, is made up of neurons that carry data amongst themselves.

Each neuron consists of a set value. The network has three main layers - Input layer, Hidden layer(s), Output layer.

Module Description –

- Input Vector is the extracted and calculated metadata.
- In a multilayer perceptron that consists of a linear activation function (maps weighted inputs to each neuron output), all layers can virtually be reduced into a two-layer (input-output) model.
- The Default Activation function consists of the hyperbolic tan function that maps inputs to outputs.
- The output value is the genre-classified value.

Artificial Neural Networks are generally used for the classification of remotely sensed data. The Neural Network algorithm mimics the operation of that of a human brain. It can adapt to changing inputs for the best possible result. The Neural Network accuracy can be predicted by looking at the weights and it does a good job at that. Accuracy of the neural network can be further improved by eliminating overfitting and underfitting and using the best fit or by generating the best set of hyperparameters or by providing enough training data so that the model learns perfectly and performs according to the standards. Neural Networks generally produces highly accurate models when these conditions are met.

Table 1.

Algorithms Accuracy Comparison

Sr. No	**Algorithm Accuracy of Trained and Tested Data**		
	Algorithm used	***Train Accuracy (%)***	***Test Accuracy (%)***
1	K- Nearest Neighbor	78.8	67.1
2	Random Forest	100.0	65.3
3	Support Vector Machine	100.0	70.0*
4	Multi-Layer Perceptron	100.0	67.6

a. SVM has the highest accuracy.

Results Discussion

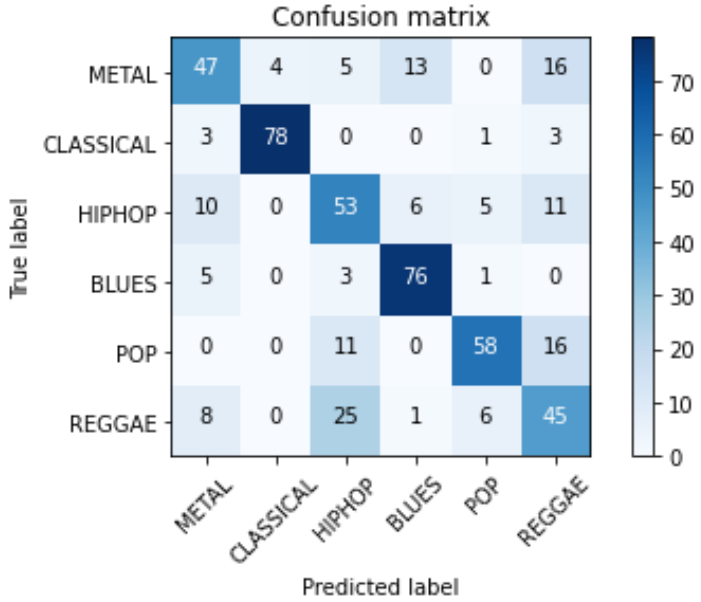

Fig. 5. Support Vector Machine Model Confusion Matrix

Support Vector Machines (SVM) results in the formation of a hyper-plane separating the points. It is used with a kernel for linear and non-linear ways for multiple dimensions. When there are limited set of points in many dimensions SVM tends to have improved accuracy as it is required to find linear separations in the data points. SVM only uses the relevant points to find linear separation (support vectors) and hence works well with outliers with minimum tunning. The Cost and parameters of the kernel are the required inputs for the algorithm.

In the dataset, when the information is spread randomly, no useful information can be obtained. The KNN algorithm tries to find the k nearest points, but as the data is spread, the accuracy is questionable. Hence KNN has several limitations like –

- Does not work well with large datasets (as distance-based calculation for new points is expensive and hinders performance).
- Does not work well with high dimensions (similar hinderance of performance).
- Anomalous behavior to outliers and missing data (need to impute values and remove outliers).

Support Vectors on the other hand, take cares of outliers and missing values as compared to KNN. Hence, when training data is less as compared to the number of features, SVM is optimal.

SVM uses a kernel-based trick to solve non-linear problems. Decision Trees, on the other hand, use input data points to derive the hyper-rectangle. This allows the trees to be better utilized in categorical data and co-linearity. However, when there is no strong relationship among the features of the dataset (images, audio signals, etc.) the observations create a network graph pattern. These patterns are not classified well by hyper-rectangles.

Random forests are optimal with multi-dimensional rectangular blocks to your data and hence result in sharp edges in predictions. If input data is continuous in nature, then SVM is allows the use of support vectors which help find closest boundaries between the classes. For an audio classification problem Random Forest provides the probability of the signal belonging to a particular class. SVM calculates the distance to the boundary of the class. Hence for such problems, SVM generally outperforms Random Forest algorithm.

SVM uses maximal margin concepts to outperform non-linear and high-dimensional tasks. They identify decision boundaries based on support vectors and hence a neural network with similar set of parameters is high in complexity. Hence, for easily separable classes, SVM does not require high number of observations to train as compared to its neural network counterpart. The decision boundary learned by the neural network depends on the data present and hence require processing of the whole training dataset, whilst hindering performance. This proves SVMs are faster to train, while Neural Networks are quite expensive.

Finally, in unpredictable situations, either an SVM with an RBF kernel or a Random Forest are your best choices as they tend to perform quite well in average. Somebody will mention the NFL (No Free Lunch) Theorem and say that if we don't know the data then we don't know which algorithm will perform better, but that is only true if we consider all possible optimization problems. Taking into account that most optimization problems are only a small subset of the whole, it is valid to say that some algorithms will be better than others on average, and in such cases, we go with an SVM or a Random Forest model.

Conclusion

Our study compared the 4 algorithms on test and train data and a study on their effectiveness to classify music genre is conducted. Of the above-mentioned algorithms, the Support Vector Machine model was the most promising. Classification was conducted on over 26 features extracted from audio signals of the dataset, which related to spectral information. F-score and their analysis might suggest better observation further research, and hence indepth investigation can be done.

Firstly, introduction of time-based features will allow us to classify genre on more complex properties, by the use of RNN or Recurrent Neural Network. Secondly, other unknown features that were not used can be used in relevance to different classification. Finally, we can work more towards selecting high quality datasets with reduced size, or even combine different datasets for different genres. The possibilities are endless.

There are multiple practical applications of music genre recognition (eg: streaming services). Companies and platforms employ people to study music and annotate them for further

processing. They categorize songs with genre and other features, for improving recommendation playlists or help enhance their output. The meta-data of discovering tracks similar to those are already present is useful for providers.

From an engineering perspective, long-term expenditures could be reduced with the use of accurate, automated genre-recognition systems. Several machine learning techniques have been applied to such problems, and further research can help improve and shed some light on this domain.

References

Lansdown, B., & He, S. (2019). *Machine Learning for Music Genre Classification.*

Chillara, S., Kavitha, A.S., Neginhal, S.A., Haldia, S., & Vidyullatha, K.S. (2019). *Music Genre Classification using Machine Learning Algorithms: A comparison.*

Huang, D.A., Serafini, A.A., & Pugh, E.J. *Music Genre Classification.*

Silla Jr, C.N., Kaestner, C.A., & Koerich, A.L., (2007). Automatic music genre classification using ensemble of classifiers. *IEEE International Conference on Systems, Man and Cybernetics,* 1687-1692.

Bahuleyan, H. (2018). Music genre classification using machine learning techniques. *arXiv preprint arXiv:1804.01149.*

Oramas, S., Barbieri, F., Nieto Caballero, O., & Serra, X. (2018). Multimodal deep learning for music genre classification. *Transactions of the International Society for Music Information Retrieval, 1*(1), 4-21.

Alim, S.A., & Rashid, N.K.A. (2018). Some commonly used speech feature extraction algorithms, 2-19. IntechOpen. https://doi.org/10.5772/intechopen.80419

Pandey, P. (2018). *Music genre classification with python.* https://towardsdatascience.com/music-genre-classification-with-python-c714d032f0d8

Ahmad, F., & Sahil. Music Genre Classification using Spectral Analysis Techniques with Hybrid Convolution-Recurrent Neural Network. *International Journal of Innovative Technology and Exploring Engineering (IJITEE),* 9(1).

Mel Frequency Ceptral Coefficient (MFCC) tutorial, http://practicalcryptography.com/miscellaneous/machine-learning/guide-mel-frequency-cepstral-coefficients-mfccs/

DataFlair. (2021). *Python Project - Music Genre Classification - DataFlair*. https://data-flair.training/blogs/python-project-music-genre-classification/

Jeya, R., Rajesbabu, C., Singh, J., & Singh, A. (2020). Iot Based Stolen Vehicle Monitoring System. *International Journal of Advanced Science and Technology*, *29*(06), 472- 482.

Xu, C., Maddage, N.C., Shao, X., Cao, F., & Tian, Q. (2003). Musical genre classification using support vector machines. *In 2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, Proceedings, (ICASSP'03),* 5, 429. https://doi.org/10.1109/ICASSP.2003.1199998

Saravananathan, K. & Velmurugan, T. (2016). Analyzing diabetic data using classification algorithms in data mining. *Indian Journal of Science and Technology*, *9*(43), pp.1-6.

Deeba, K., & Amutha, B. (2016). Classification algorithms of data mining. *Indian Journal of Science and Technology*, *9*, 1-5.

Jeya, R., Amutha, B., Nikhilesh, N., & Immaculate, R.R. (2019). Signal Interferences in Wireless Communication-An Overview. *Spectrum*, *2*, 3.

Sturm, B.L. (2013). The GTZAN dataset: Its contents, its faults, their effects on evaluation, and its future use. *arXiv preprint arXiv:1306.1461*.