Predicting Origin of Music

Muhammet Subasi Harun Burkuk Mustafa Korkmazlar

Abstract

Music is one of the most important art that attracts worldwide attention. Beside, music represents emotion, culture, history and thoughts of its origin by its characteristics such as melody, rhythm, instruments and sound intensity. Therefore, it wouldn't be wrong to say that music has geographical diversity. Our aim in this project is to show that origin of a music can be predictable.

1. Introduction

It can be done by 2 different ways to show that origin of music can be predictable. First way is predicting countries and second way is predicting coordinates. We applied both of the ways.

We approached this problem as a classification before we predicted countries. We approached this problem as a regression before we predicted coordinates.

Both methods have pros and cons. They are mentioned in "The Approach" part.

2. Related Work

For the development of audio features, there is a large amount of research. e.g[1]. Audio features are being used in machine learning research areas largely related to finding clusters and estimating similar music tracks for satisfying the goal of music recommendation. e.g. [2]. In addition that, machine learning applications make use of these audio features to identify the genre of the music, identify the mood of the music, predicting the popularity of a certain music track, and many other applications. e.g.[3]-[5]. Besides audio features, there is also different kind of audio attributes are being used for applications of machine learning with music. One of them is MIDI which can be described as a digital music notation. Music is represented as MIDI messages distributed through sixteen possible MIDI channels. e.g. [6]-[7]. The other one is music audio signals. Music audio features can describe as sound waves that turned into an electrical current when captured by a microphone. As we can see from the related papers there is a lot of research has been done related to music and there are quite successful papers by using k-Nearest Neighbor, Support vector machine, Neural Network machine learning algorithms.

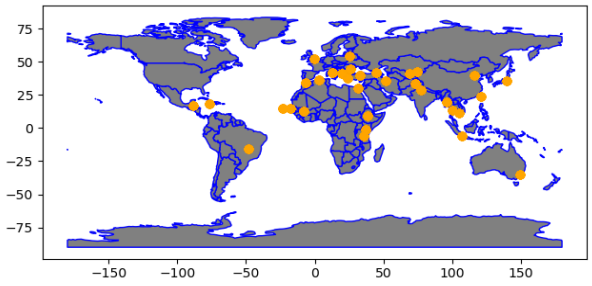
Nevertheless, there is just one related work for our topic of the project which is predicting ethnomusicology[8]. Of the related work, they are not interested in the classification of the countries. They are predicting longitudes and latitudes by using several machine learning algorithms like k-Nearest Neighbor, Random kNN and Random Forest. Even though there is just one related work for predicting latitude and longitude on the geographical location of music, there are a few papers for this problem for example this paper[9] is predicting coordinates by using a neural network for the purpose of finding the aircraft locations on the earth. Since we are planning to use a neural network for finding the coordinates, we must find papers about this approach. Luckily for us, there is a lot of examples and papers about using a neural network as a regression algorithm eg.[10]-[11]. When we back to our main related work, as a result of their study, they find that the best algorithm that gives the least error is k-Nearest Neighbor and they find the error rate as 3,113 km in terms of the mean great circle distance. Our aim will be to improve this study, so we will try to reduce the mean great circle distance error.

3. The Approach

Our aim in this study will be both predicting origin county names and predicting latitude and longitude values. For predicting origin country names, our purpose will be to investigate that a machine learning model can predict these country names by just look at the audio features and if our dataset will be suitable for this kind of research. Since there will be country names to predict, we will use classification algorithms for our machine learning model in the first task. As algorithms, we will use different classification algorithms because as we said earlier, we want to see that can a machine learning model do this job, and trying with different classification algorithms will show us our success rate clearly because in the evaluation part we can look the average evaluation metrics and this will be more accurate than just using one algorithm. Another benefit will be, we will see how is our dataset in terms of suitability when we look at the average evaluation metrics. During the first task, we will use Random Forest, Logistic Regression, and Support Vector Machines algorithms.

In the second part which is predicting latitude and longitude values of the origin of the music, we are going to use regression algorithms because the latitude and longitude values are numerical. Our purpose will be to reduce the error rate in terms of great mean circle distance in our related work. Since our related work finds this great mean circle distance error as 3,111 km, if we can achieve better scores than 3,111 km, we will consider this task as successful. As we mentioned earlier our related work uses k-Nearest Neighbor and Random Forest algorithms for this problem but we will use the neural network algorithm because we believe that neural network can reduce this error rate with the optimal hyperparameters.

3.1. Data Set



Through the study, we will use the Geographical Original of the Music Data Set from UCI [12]. This dataset contains 1059 rows, each of which represents a music track, and there are 70 columns. 68 of the columns are features. The last 2 columns contain latitude and longitude information's for geographical regions. The geographical location of origin has manually collected the information from the CD sleeve notes. The country of origin was determined by the artist's or artists' main country/area of residence. Latitude and longitude information shows the capital of the countries. There are 33 different countries. For extracting the features from audio files MARSYAS[13] is used. Collectors used the default MARSYAS settings in a single vector format (68 features) to estimate the performance with basic timbal information covering the entire length of each track. No feature weighting or pre-filtering was applied when extracting features from audio waves. All features transformed to have a mean of 0 and a standard deviation of 1 in the dataset so no need to normalize the dataset during the application of the machine learning model. Dataset also investigated the utility of adding chromatic attributes. These describe the notes of the scale being used. This is especially important as a distinguishing feature in geographical ethnomusicology. The chromatic features provided by MARSYAS are 12 per octave-Western tuning.

3.1.1. Train Test Split

Our dataset does not contain test sets in default so we have to split our data into train and test sets. Since we are going to apply 2 different approaches and will have 2 different DataFrame, we have to split our Dataframes separately. For the classification approach, we will split our data into 80% train and 20% test sets. For the regression approach, we will split into 70% train and 30% test sets.

3.2. Predicting Countries

Predicting countries is our first approach to predict origin of music. First step to predict countries was obtaining country names from longitude and latitude values. The countries and their coordinates are shown below and the coordinates are represented as (longitude, latitude).

```
{(-15.75, -47.95): 'Brasil', (14.91, -23.51): 'Cape Verde', (12.65, -8.0): 'Mali',
 (9.03, 38.74): 'Ethiopia', (34.03, -6.85): 'Morocco', (14.66, -17.41): 'Senegal',
 (52.5, -0.12): 'England', (41.26, 69.21): 'Uzbekistan', (41.9, 12.48): 'Italy',
 (28.61, 77.2): 'India', (33.66, 73.16): 'Pakistan', (54.68, 25.31): 'Lithunia',
 (44.41, 26.1): 'Romania', (36.7, 3.21): 'Algeria', (39.91, 32.83): 'Turkey',
 (19.75, 96.1): 'Myanmar', (13.75, 100.48): 'Thailand', (39.91, 116.38): 'China',
 (23.76, 121.0): 'Taiwan', (-6.17, 106.82): 'Indonesia', (17.98, -76.8): 'Jamaica',
 (35.68, 51.41): 'Iran', (30.03, 31.21): 'Egypt', (42.86, 74.6): 'Kyrgyzstan',
 (-1.26, 36.8): 'Kenya', (17.25, -88.76): 'Belize', (38.0, 23.71): 'Greece',
 (-35.3, 149.12): 'Australia', (35.7, 139.71): 'Japan', (-6.17, 35.74): 'Tanzania',
 (41.71, 44.78): 'Georgia', (11.55, 104.91): 'Cambodia', (41.33, 19.8): 'Albania'
}
```

After obtaining country label from longitude and latitude values, we splitted the dataset by 8% training and 20% test as second step. As third and the last step, we applied 3 classification algorithms which are Random Forest Classifier, Logistic Regression and Support Vector Machines to predict the countries.

3.2.1. Random Forest Classifier

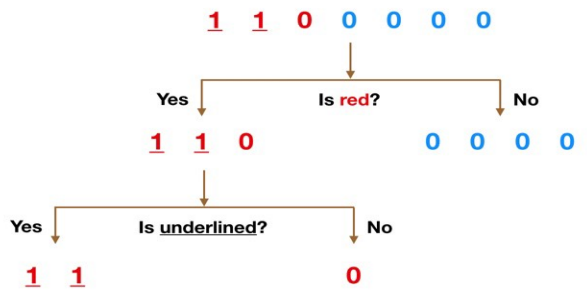
Random Forest Classifier is a estimator that fits a number of decision tree classifiers on data set and uses averaging to improve the predictive accuracy and control overfitting.[14]

3.2.1.1. Decision Trees

Before moving on to the properties of the random forest, let's take a look at the building blocks that make up the random forest.

Decision trees are a concept that we encounter frequently in our daily lives. It helps us to show nested decision-result schemes as a tree diagram.

If we want to describe decision trees with an example, in below[15], we can classify our data according to its characteristics by asking questions that will enable us to make decisions at every crossroads we encounter from top to bottom (termly, from root to leaves).

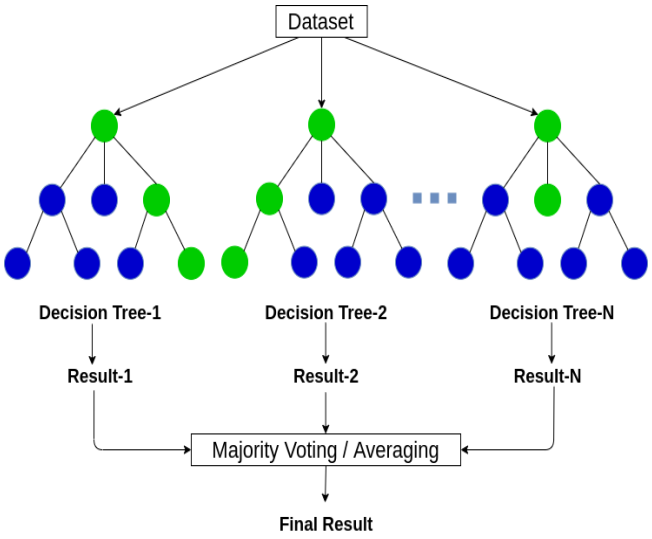


3.2.1.2. Algorithm

As the name suggests, a random forest consists of a combination of many decision trees. The idea behind the algorithm is to obtain information from a large collection of decision trees.

In fact, many decision trees generated are low correlation, but the combination of these many small correlations will give much better results than any individual estimate.

Under the success of making accurate estimation by combining trees with little correlation, although the individual decision trees are very dependent on the noise values in the data set, when they are combined and averaged they don't depend much on the data set. With this method it decreases the variance of the model, without increasing the bias.



As we mentioned in the approaches section, while trying to predict the origin of music, we are also trying to examine the suitability of predicting a large number of country names with machine learning models. For this purpose, we did not change too many parameters in the modeling of random trees.

While examining random forest modeling, we considered the most important parameters, "n_estimators" and "min_samples_leaf". Simply n_estimators indicates how many trees will be in the forest, which we have sacrificed execution time and tried to keep high for best results. min_samples_leaf parameter is the minimum number of samples required to be at a leaf node[16]. Since we saw from our experimental results that keeping this parameter high causes an under-fitting problem, we thought it would be better to keep it at low values.

3.2.2. Logistic Regression

Another algorithm that we used for classification is Logistic Regression. Logistic regression uses an equation as the representation, very much like linear regression but the difference is basically the function. Since Linear regression uses mean squared error as its cost function. If this is used for logistic regression, then it will be a non-convex function of parameters. Gradient descent will converge into global minimum only if the function is convex.

The logistic Regression algorithm is named for the function used at the core of the method, the logistic function. This logistic function also called the sigmoid function. It's an S-shaped curve that can take any real-valued number and map it into a value between 0 and 1, but never exactly at those limits. To predict which class a data belongs, a threshold can be set. Based upon this threshold, the obtained estimated probability is classified into classes. This decision boundary can be linear or non-linear. Polynomial order can be increased to get complex decision boundaries.

While applying the Logistic Regression we will consider different hyperparameters such as "solver", "penalty", "max_iter" and "multi_class". We will try with a bunch of different parameter values with different parameters and compare the evaluation metrics of each result in order to find optimal hyperparameters. But as we said earlier e are also trying to examine the suitability of predicting a large number of country names with machine learning models. Because of this, we did not change too many parameters in the modeling of random trees and we keep it simple.

3.2.3. Support Vector Machines

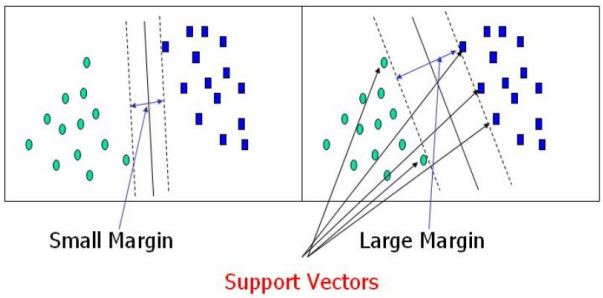
3.2.3.1. Types of Support Vector Machines

Support vector machines are supervised learning models that analyze data for classification and regression analysis. Because of predicting country names, we used Support Vector Classification.

3.2.3.2.Algorithm

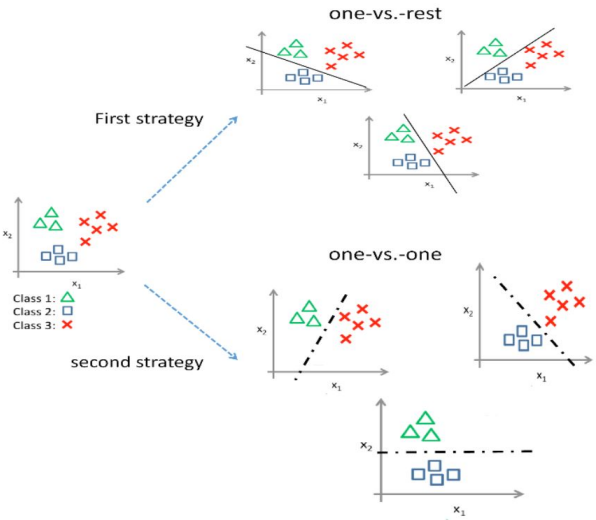
The aim of Support Vector Machines is separating classes by a maximum marginal hyper-plane. A hyper-plane is a subspace whose dimension is one less than the dimension of its ambient space. Distance between closest points to the hyper-plane determines the margin as shown in the

figure below. The closest points are called Support Vectors. Small margin causes overfit and large margin causes underfit. The margin can be changed by changing regularization parameter which is hyper-parameter C. Increasing regularization parameter makes margin smaller and vice-versa. The regularization parameter must be positive.



[17]

3.2.3.3. SVM With More Than 2 Classes



[18] When number of classes is 2, it is called binary classification and when number of classes is greater than 2, it is called multi-class classification,There are 2 strategy for determining hyper-planes for multi-class classification. First strategy is called one-vs-rest. In this strategy, each of the classes is selected once then the other classes considered as one class. So the hyper-planes can be determined as binary classification. Second strategy is called one-vs-one. In this strategy, each of the classes is selected once then the each of the other classes is also selected. After that, the unselected classes are ignored until their turn. So the hyper-planes can be determined as binary classification.

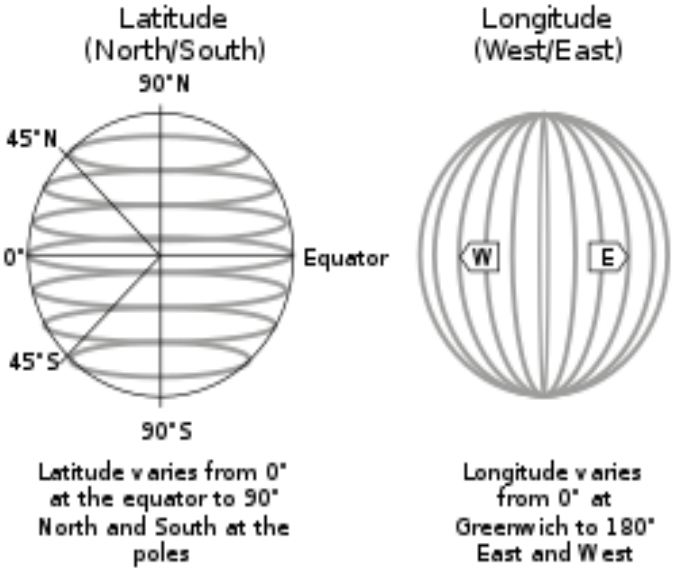
3.3. Predicting Latitude & Longitude

When we review this approach, we encounter a challenging and intriguing machine learning application. Contrary to the previous section, what we want to predict is not a class but a set of real values. This set contains the latitude and longitude information on the world map of the origin of music.

We faced so many difficulties when predicting latitude and longitude. The first difficulties was that the coordinates on Earth were limited.

At the beginning of this application, we actually did not know how to approach the limited coordinate system in the world. Frankly, we couldn't find many resources that have been studied before. Most of the past studies either estimated in a small area or ignored the boundary coordinates at all. We have come to the conclusion that we have to develop an approach from the very beginning.

We decided to move a prediction made outside the coordinate boundaries to the coordinate boundaries (-90 and +90 for latitude, -180 and +180 for longitude). Because we also want to make an application that visually shows these points on the Earth map. For this reason, it is important for us that every predicted point is within the coordinate limits.



3.3.1. Neural Network

We mentioned that we needed to use a regression algorithm to predict the latitude and longitude values because we are trying to predict a set of values as opposed to country name prediction.

So we chose Multi-Layer Perceptron Regressor from sklearn.neural_network[19]. MLPRegressor trains iteratively and each iteration gets partial derivatives of the loss function for optimization. Also an regularization term can be added to the loss function for preventing over-fitting.

Like all other MLP classes, MLPRegressor consists of at least three node layers: an input layer, a hidden layer and

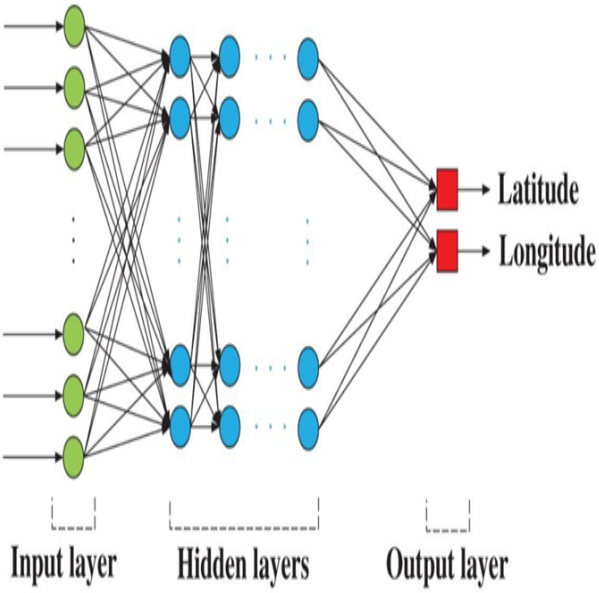
an output layer. Except for the output nodes, each node is a neuron that uses a linear or nonlinear activation function.

We used both linear and non-linear activation functions but overall we got better results with linear activation function. The reason for that the data set works more conveniently with the linear activation function. In addition, linear activation functions are more successful in giving multiple outputs(latitude and longitude values) than non-linear functions[20].

MLPRegressor uses the Mean squared error (MSE) as the loss function[21]. We didn't use any custom loss function. MSE is the most commonly used loss function for regression. The loss is the mean overseen data of the squared differences between true and predicted values.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2.$$

where $\hat{Y}$ is the $predicted$ $value.$



Another important parameter for our model is the *solver*. This parameter is used for weight optimization. As with the activation function, we tried different parameters for weight optimization. Among the ones we tried, the most robust solver was the *adam* algorithm. Adam refers to a stochastic gradient-based optimizer[22]. It uses mini-batch optimization and can make progress faster while seeing less data than the other Neural Network optimization solver like "lbfgs" or "sgd"[23].

We also used other parameters to increase the success of our neural network model. In the experimental results section, we will show the contribution of the different values of these parameters that we have tried.

3.3.2. Distance Evaluation

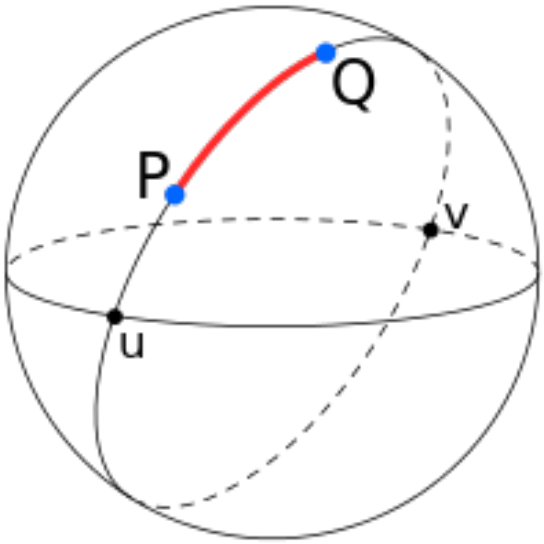
The main challenge of distance evaluation is the Earth is not flat. If the Earth were flat we would simply find the distance between the binary coordinate system as follows:

$P = (p_x, p_y)$ and $Q = (q_x, q_y)$ are two $points$ in a $two - dimensional$ $system,$ $euclidean$ $distance$ $between$ $them$ is[24] :

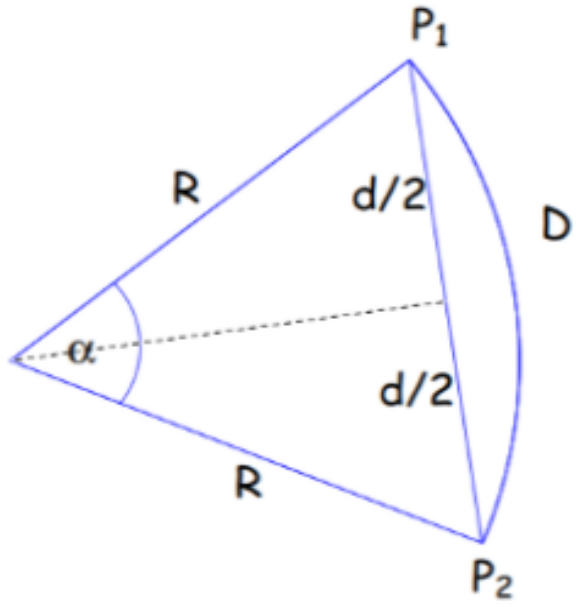
$$\sqrt{(p_x - q_x)^2 + (p_y - q_y)^2}\sqrt{(p_x - q_x)^2 + (p_y - q_y)^2}$$

But we know the Earth is not flat so we had to find another approach. And we met "Great Circle Distance" concept.

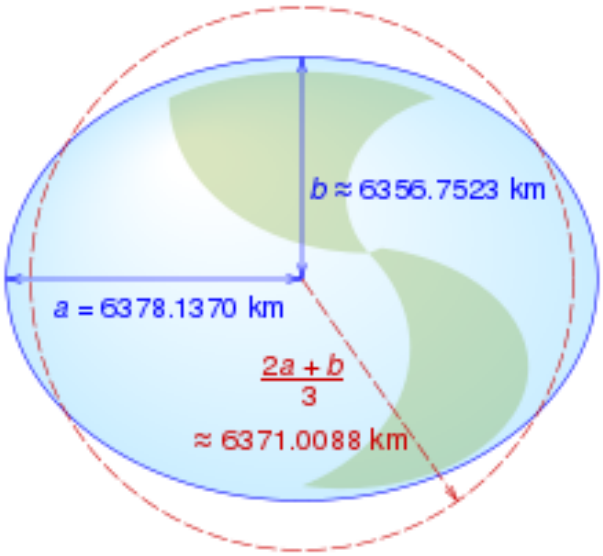
According to the Wikipedia, *"The great-circle distance, orthodromic distance, or spherical distance is the shortest distance between two points on the surface of a sphere, measured along the surface of the sphere (as opposed to a straight line through the sphere's interior)"*[25]. Unlike the Euclidean system, the distances between two points are expressed by curved lines called by *geodesics*[26], not straight lines.



Two points on the sphere can be expressed as two points on a circle—the great circle—whose radius is the radius of the sphere and whose center is the center of the sphere. If the angle between these two points can be calculated, we can also calculate the distance with the expression R*alpha due to the property of the circle .This inference can be called the application of the haversine formula on the great circle distance.



As it can be understood from the statements above, in haversine formula, the Earth was accepted as a perfect sphere and calculations were made accordingly. But we all know that the world is not a perfect sphere. According to our research, we have seen that the fact that the world is not a perfect sphere creates a 0.5 percent margin of error. this is a negligible margin of error for us because we work all over the world and this error has little effect when mapping large distances.



You can find the haversine formula as follow[27] ;

$$d = 2r \arcsin\left(\sqrt{\sin^2\left(\frac{\varphi_2 - \varphi_1}{2}\right) + \cos(\varphi_1)\cos(\varphi_2)\sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right)}\right)$$

4. Experimental Results

In the experiment, we got 2 different kind of results which are percentage of accuracy and great circle distance in kilometers. Results of Random Forest Classifier, Logistic Regression and Support Vector Machine is an accuracy. Because we used the 3 classification algorithms to predict countries. Therefore it was possible to check if the predicted country and the actual country are same or not. Result of Neural Network is great circle distance in kilometers. Because longitude and latitude values are predicted here by regression. According to the predicted coordinates and the actual coordinates, a distance value in kilometers is predicted from haversine formula.

4.1. Experimental Setup

We used data from all countries. In total, 1059 tracks were used in the experiments. For predicting countries, we used all of the 68 features. For predicting coordinates, we used feature selection according to permutation importance.

4.2. Random Forest Classifier

As we mentioned in the project progress, random forests give relatively better results with optimum parameters compared to other supervised machine learning algorithms. The reason for that RFC focuses on accuracy rather than probabilities, cut-off values, and bunch of parameters, among others. Random selection in individual decision trees of RFC can capture more complex feature patterns to provide the best accuracy. In the process of choosing one of the 33 target categories like ours, it is important to find different feature patterns.

Another reason for the success of random forests is the decision trees that it creates by choosing random subsets from the training set. In models like ours, where the data is not evenly distributed across all target classes, and some target classes are represented by very little data, over-fitting often occurs. Choosing random subsets from the training set tries to solve this over-fitting problem.[28]

As we mentioned in the Approaches section, we basically have two parameters that we focus on and that it would be useful to play with. One is ”n_estimators” and the other is ”min_samples_leaf”. We thought it would be helpful to keep the number of n_estimators high which defaults to 100 but we also tried values of 500 and 1000 and reached the optimum at 1000.

n_estimator	Accuracy
100	0.36
500	0.42
1000	0.44

We have seen that lower values are more suitable for the min_samples_leaf parameter. High min_samples_leaf values were causing under-fitting issues. We think that the reason for this is related to how the min_samples_leaf values affect the tree. When we keep the min_samples_leaf number high, we see that more samples with incompatible features are collected in each leaf. This leads to under-fitting problems.

min_samples_leaf	Accuracy
1	0.44
5	0.37
10	0.34

4.3. Logistic Regression

When we look at the result of the Logistic Regression, we can see that it is more or less similar to the other classification algorithms that we used. But also, Logistic Regression gives the worst result among these three algorithms with optimized parameters.

When we analyze this situation in which Logistic Regression gives worst results than others, we can clearly see the problems with this algorithm. Logistic Regression learns a linear relationship from the given dataset and then introduces a non-linearity in the form of the Sigmoid function. Because of that Logistic Regression constructs linear boundaries in the application phase. So, we can say those non-linear problems can't be solved with logistic regression because it has a linear decision surface. Apparently, our data is not linearly separable because our data contains 33 different classes and there are 68 different features. Also, these features are not discrete and that means our feature values for each sample do not create a linearly separable surface in a graph. As a result, we can say that it is tough to obtain complex relationships using Logistic Regression.Â

We mentioned the hyperparameters that important for optimization in the approach part. Since our study contains multi-class classification we had to optimized the parameters in regards to this situation. For "solver" parameter, lbfgs, sag, and saga used for multi-class classification. In our experiments, we found that all of the solvers give the same accuracy and other evaluation metric scores and we decided to use "lbfgs" because it is the default and more secure one than the others. Since lbfgs works with the only l2 "penalty" so we had to use that for our "penalty" parameter. Another parameter that we had to optimize was "multi-class". For this parameter "ovr" works well with binary classification tasks, since our data is not a binary classification task but multi-class. We found that the "multinomial" parameter for "multi_class" was the best way to go. Lastly, "max_iter" parameters give the best scores at 500, after 500 our evaluation metric does not change.

multi_class	Accuracy
multinomial	0.41
ovr	0.39

4.4. Support Vector Machine

Best accuracy of this algorithm is 42.4%. This accuracy is better than Logistic Regression but worse than Random Forest Classifier. To get the best result, we focussed on 2 hyper-parameters.

First hyper-parameter is kernel. Kernel determines shape of the hyper-plane.We got best accuracy when the kernel is linear. Linear kernel allows to use one-vs-rest heuristic method mentioned in 3.2.3.3.One-vs-rest heuristic method uses the most confident models. Linear kernel is also used after adding one more dimension. The additional dimension allows to separate classes by linear kernel. Results for the other kernel values are shown below.

Kernel	Accuracy
linear	0.42
poly	0.08
rbf	0.06
sigmoid	0.12

Second hyper-parameter is regularization parameter which is also known as penalty parameter and denoted as C. This parameter is explained in detail in part 3.2.3.2. We got our best accuracy when the regularization parameter is 0.1.

Some of the C values and their accuracy are shown below.

C	Accuracy
0.01	0.36
0.1	0.42
1	0.41

4.5. Neural Network

Our best mean great circle distance is 2714 kilometers while it is 3113 kilometers in the related work that we used the same dataset. To get this result, we focussed on 6 hyper-parameters.

First hyper-parameter is maximum number of iteration.When using the 'adam' solver this parameter becomes epoch size for the model. 4000 is the value that gave us the best result. Values bigger than 4000 gave the same result as expected. Some of other values we tried for maximum number of iteration and their accuracy are shown below.

Second hyper-parameter is hidden layer sizes.Since our dataset is not that complicated we choose 2 hidden layers. To decide size of hidden layers, we calculated square root of the sum of input layer size and output layer size.

activation	Great Circle Distance (km)
identity	2714
logistic	3246
tanh	4289
relu	3517

max_iter	Great Circle Distance (km)
200	2864
4000	2714

hidden_layer_sizes	Great Circle Distance (km)
12	2874
11, 11	2776
12, 12	2714

Then we add 1 to 10 to this value and compared the results. We got the best result when each of 2 hidden layer has a size of 12. Some of other size values and their results are shown below.

Third hyper-parameter is activation. We got the best result when we used 'identity'. It returns x for f(x) and useful for linear datasets. We can say that our dataset is kind of linear. We also tried 'logistic', 'tanh' and 'relu' instead of 'identity' and the results are shown below.

Forth hyper-parameter is solver. "SGD" only computes on a small subset or random selection of data examples instead of the whole dataset. Our dataset isn't that big. "l-bfgs" fits with smaller datasets than ours. That's why we used "adam".

Fifth hyper-parameter is batch size. A training dataset can be divided into one or more batches. Batch size is the size of minibatches for stochastic optimizers. We tried different numbers of batch sizes. When batch size was 64, we got the best result. Some of other size values and their results are shown below.

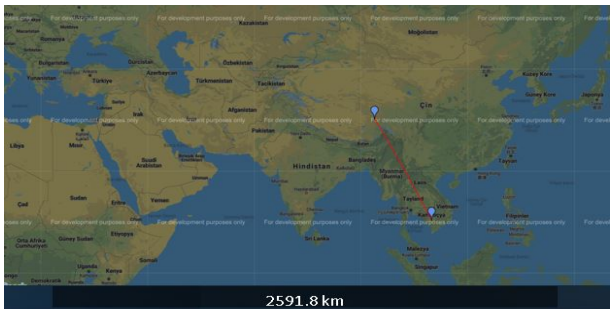
Sixth hyper-parameter is alpha. Alpha is L2 regularization penalty parameter and we got the best result when alpha was 1000.

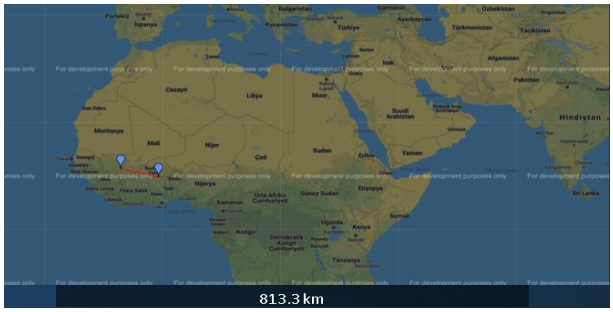
2 of neural network predictions are shown below. Their great circle distances are 813.3 kilometers and 2591.8 kilometers respectively.

batch_size	Great Circle Distance (km)
32	3016
64	2714
128	2718

solver	Great Circle Distance (km)
lbfgs	2798
sgd	2980
adam	2714

alpha	Great Circle Distance (km)
0.1	2746
10	2748
1000	2714





.

5. Conclusion

We have proposed an machine learning and neural network approach to understanding geographical diversity of music. We analyzed the geographical distribution of music - geographical ethnomusicology. This is a special technical interest of data analysis task because of the spatial data analysis. To accomplished these analysis we applied some distance calculation methodologies. To predict origin of music we applied Random Forest, Logistic Regression and Support Vector Machine algorithms. To take this further ahead we tried to predict coordinates of origin of music using Neural Network.

ALGORITHMS	RESULTS
Random Forest Classifier	%44 Accuracy
Logistic Regression Classifier	%41 Accuracy
Support Vector Machine	%42 Accuracy
Neural Network (MLPRegressor)	Mean Great Circle Distance: 2714 km

References

[1] F. Pachet and P. Roy, ”Exploring Billions of Audio Features,” 2007 International Workshop on ContentBased Multimedia Indexing, 2007, pp. 227-235, doi:10.1109/CBMI.2007.385416.

[2] Wen, X. Using deep learning approach and IoT architecture to build the intelligent music recommendation system.Soft Comput 25, 3087–3096 (2021).

[3] B. Liang and M. Gu, ”Music Genre Classification Using Transfer Learning,” 2020 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR), 2020, pp. 392-393, doi: 10.1109/MIPR49039.2020.00085.

[4] S. Chauhan and P. Chauhan, ”Music mood classification based on lyrical analysis of Hindi songs using Latent Dirichlet Allocation,” 2016 International Conference on Information Technology (InCITe) - The Next Generation IT Summit on the Theme - Internet of Things: Connect your Worlds, 2016, pp. 72-76, doi: 10.1109/INCITE.2016.7857593.

[5] J. Lee and J. Lee, ”Music Popularity: Metrics, Characteristics, and Audio-Based Prediction,” in IEEE Transactions on Multimedia, vol. 20, no. 11, pp. 3173-3182, Nov. 2018, doi: 10.1109/TMM.2018.2820903.

[6] Øland, A. (2012). Machine Learning and its Applications to Music.

[7] Jiangtao Li, Xiaohong Yang and Qingcai Chen, ”MIDI melody extraction based on improved neural network,” 2009 International Conference on Machine Learning and Cybernetics, 2009, pp. 1133-1138, doi: 10.1109/ICMLC.2009.5212378.

[8] F. Zhou, Q. Claire and R. D. King, ”Predicting the Geographical Origin of Music” 2014 IEEE International Conference on Data Mining, 2014, pp. 1115-1120, doi: 10.1109/ICDM.2014.73.

[9] D. Adesina, O. Adagunodo, X. Dong and L. Qian, ”Aircraft Location Prediction using Deep Learning,” MILCOM 2019 - 2019 IEEE Military Communications Conference (MILCOM), 2019, pp. 127-132, doi: 10.1109/MILCOM47813.2019.9020888.

[10] A. Sharaff and S. R. Roy, ”Comparative Analysis of Temperature Prediction Using Regression Methods and Back Propagation Neural Network,” 2018 2nd International Conference on Trends in Electronics and Informatics (ICOEI), 2018, pp. 739-742, doi: 10.1109/ICOEI.2018.8553803.

[11] C. Wilkins and J. Dickieson, ”Using neural network technology for predicting military attrition,” IJCNN-91-Seattle International Joint Conference on Neural Networks, 1991, pp. 913 vol.2-, doi: 10.1109/IJCNN.1991.155510.

[12] http://archive.ics.uci.edu/ml/datasets/geographical+original+of+music

[13] G. Tzanetakis and P. Cook, â€œMARSYAS: a framework for audio analysis,â€ Organised Sound, vol. 4, pp.

169â€“175, 2000.

[14] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

[15] https://towardsdatascience.com/understanding-random-forest-58381e0602d2

[16] https://medium.com/all-things-ai/in-depth-parameter-tuning-for-random-forest-d67bb7e920d

[17] Rohith Gandhi "Support Vector Machine" Jun 7, 2018

[18] Shervan Fekri Ershad "Texture image analysis and texture classification methods - A review" April 2019

[19] https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPRegressor.html

[20] https://xzz201920.medium.com/activation-functions-linear-non-linear-in-deep-learning-relu-sigmoid-softmax-swish-leaky-relu-a6333be712ea: :text=Linear/

[21] https://scikit-learn.org/stable/modules/neural_networks_supervised.html

[22] https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPRegressor.html

[23] https://docs.oracle.com/en/database/oracle/oracle-database/21/nfcon/adam-optimization-solver-for-the-neural-network-algorithm-274058466.html: :text=Adam//

[24] https://en.wikipedia.org/wiki/Euclidean$_distance_matrix$.

[25] https://en.wikipedia.org/wiki/Great-circle$_distance$

[26] https://en.wikipedia.org/wiki/Geodesic

[27] https://en.wikipedia.org/wiki/Haversine$_formula$

[28] https://towardsdatascience.com/predictive-modeling-and-multiclass-classification-a4d2c428a2eb