Predicting Origin of Music

1 Description of the dataset

This dataset presents us a task in which we need to find if there is a connection between some traits of musical pieces and the geographical location where they were composed.

The dataset contains 1059 musical pieces, each piece being encoded in 116 numerical features (using a softwear named MARSYAS). It also contains 2 features for location (expressed as coordinates for latitude and longitude). Our target being to predict latitude and longitude of some new data.

Although this task may seem a classification problem (finding the country for a track) we choose to solve it with regression algorithms, which gives us better results.

First, for getting a better idea about our data I made a heap map representing the distribution of track per country.

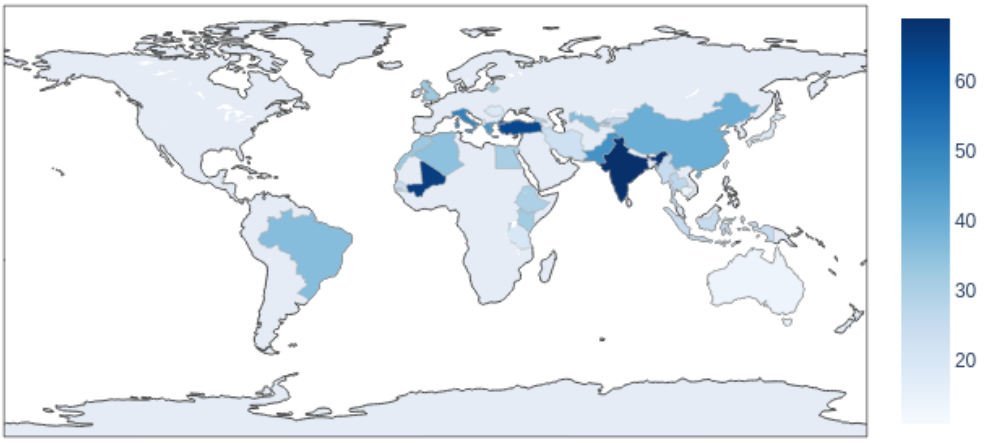

Figure 1: Distribution of tracks

Note there are some countries which have just a few examples, with a ratio close to 1:7 compered to the country with most data.

2 Model

The methods I choose to compere in this project are ElasticNet (first I tried LassoLars) and MLPRegressor.

LASSO regression is a type of linear regression. The name stands for Least Absolute Shrinkage and Selection Operator. Shrinkage is where data values are shrunk towards a central point, like the mean. This type of regression has a tuning parameter named α which controls the amount of shrinkage.

The exact type of Lasso I used is LassoLars which is a LASSO model fitted with Lars (Least Angel Regression). LARS is a model selection method for linear regression that is making decisions based on the angles between variables.

After trying several parameters for this model I did some research and I decided to change it to a ElasticNet which is more general.

ElasticNet is a model that combines LASSO Regression and Ridge Regression by using a penalty function based on both. Given that both LASSO and Ridge regressions use an α parameter ElasicNet has a parameter named *l1_ration* which specify the ratio for Lasso.

Best parameters for Elastic Net were:
ElasticNet(alpha=0.01, copy_X=True, fit_intercept=True, l1_ratio=0.3, max_iter=10000, normalize=True, positive=False, precompute=False, random_state=None, selection='cyclic', tol=0.001, warm_start=False)
MLPRegressor is a Multi-Layer Perceptron made for regression.

Beste parameters were:
MLPRegressor(activation='relu', alpha=0.0001, batch_size='auto', beta_1=0.9, beta_2=0.999, early_stopping=
epsilon=1e-08, hidden_layer_sizes=(100, 50, 50), learning_rate='adaptive', learning_rate_init=0.001, max_iter=2
momentum=0.9, n_iter_no_change=1... random_state=0, shuffle=True, solver='lbfgs', tol=0.0001, validation_fraction=0.1, verbose=False, warm_start=True)

3 Result

For evaluation, I defined a loss function (based on the geographical distance between points). For the first model I used StratifiedKFold cross-validation (with 10 splits) and the best mean result was 5971.55 km. For the MLPRegressor the best result was around 8000 km.

4 Conclusion

As we can see, our model misplaced data points that are far from the mean (for example, the tracks from Brazil aren't even in South America). But, data points from India are not so far from the true labels, given that are much more tracks from India then Brazil. So our model can do better if it gets correctly India data, even if is misplacing Brazil data, doing that.

Also, European countries that is almost not place on its continent (one reason may be that we have more data from Africa and Asia).

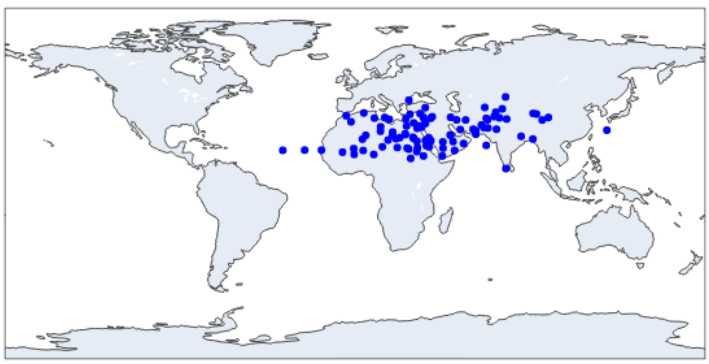

Figure 2: Prediction