Random Forest

Random forest 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 RF [1]. Random Forest can reduce the overfitting of data by introducing extra randomness when growing trees.

According to the collected dataset from Spotify, our extracted feature set = {danceability, energy, key, loudness, mode, speechiness, acousticness, instrumentalness, liveness, valence, tempo, time\_signature, genre}. There are 12000 data in the extracted dataset including 10 genres ('acoustic', 'blues', 'classical', 'dance', 'jazz','latin', 'metal', 'pop', 'rock', 'techno'). An additional dataset (7200 data) is also extracted with 6 genres ('acoustic', 'classical', 'jazz', 'metal', 'rock','tecno') to compare RF’s classification accuracy. Despite having highly correlated features seen from the correlation matrix, no features were removed in this case. For the target column ‘genre’, label encoding is implemented. The columns ‘key’, ‘mode’, and ‘time\_signature’ is encoded with one-hot encoder and the rest of the numerical columns are encoded with a MinMax scaler. Stratified train-test splitting is done with 10000 data in the train set and 2000 data in the test set. For the additional dataset with 7200 data and 6 genres, the train set contains 6000 data, and the test set contains 2000 data. Applying RF in the dataset with 12000 data gives an accuracy of 91.87% on the training set and 49% accuracy on the test set. Similar to this, RF is applied to the dataset to classify 6 genres which give 97.87% accuracy on the training set and 65.58% accuracy on the test set.

A picture containing graphical user interface

Description automatically generated

Fig 1: Random Forest accuracy on the training and test set with 12000 data and classifying 10 genres.

It is observed clearly that RF is overfitting the data. To avoid overfitting, cross-validation and hyperparameter tuning are implemented. For cross-validation, K-Folds cross-validator from scikit-learn 1.1.1 is used. 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, 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)

Graphical user interface, application, Teams

Description automatically generated  
Fig 2: Random Forest accuracy on the training and test set with 7200 data and classifying 6 genres

Analysis of Results:

When classifying 10 genres, hyperparameter tuning gives best parameters as max\_depth': 32, 'max\_features': 'sqrt', 'min\_samples\_leaf': 7, 'min\_samples\_split': 16, 'n\_estimators': 300. These parameters are different for the classification of 6 genres. While classifying 6 genres, max\_depth is found 20, and min\_samples\_split is found at 13 as the best parameters. Classification of 10 genres gives the best score of 0.49266 and classification of 6 genres gives the best score of 0.648166. These best parameters are used to find accuracy. Classification of 10 genres with Random forest gives now 49.7% so after cross-validation and hyperparameter tuning accuracy is increased from 49% to 49.7%. Classification of 6 genres gives an accuracy increase from 65.58% to 65.75%. Some genres are correlated so while predicting those genres, accuracy gets decreases. Later on in this paper, a clustering algorithm is applied to find out the best genres for classification. 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. If a few set feature is selected according to the correlation matrix, then computation time is decreased but accuracy also decreases. There is a compromise between computation time and accuracy which is described in a separate section of this paper.

Graphical user interface, text, application, email

Description automatically generated

[1] L. Breiman, “Random forests,” Mach. Learn., vol. 45, no. 1, pp. 5–32, 2001