

**School of Information Technology  
Predictive Modeling Assignment  
  
IT- Assignment 1**

**Prepared by:** Li Zheng (203774M)

**Module Group: IT2361-01**

**Submission Dateline:**

**Lecturer: Law Chee Yong**

Table of Content

### 

[**Data understanding**](#_2gtj5ktcijon) **3**

[**Data preparation**](#_av75xcddosrp) **5**

[**Modeling**](#_q0khm3gxvs81) **13**

[**Evaluation**](#_w88giufodfze) **16**

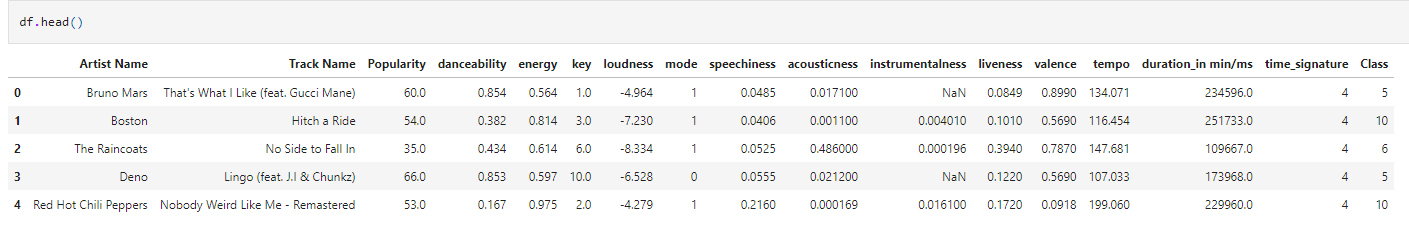
[**Prediction**](#_v7g475tnylgk) **24**

[**Summary**](#_n0qbsmlzce99) **27**

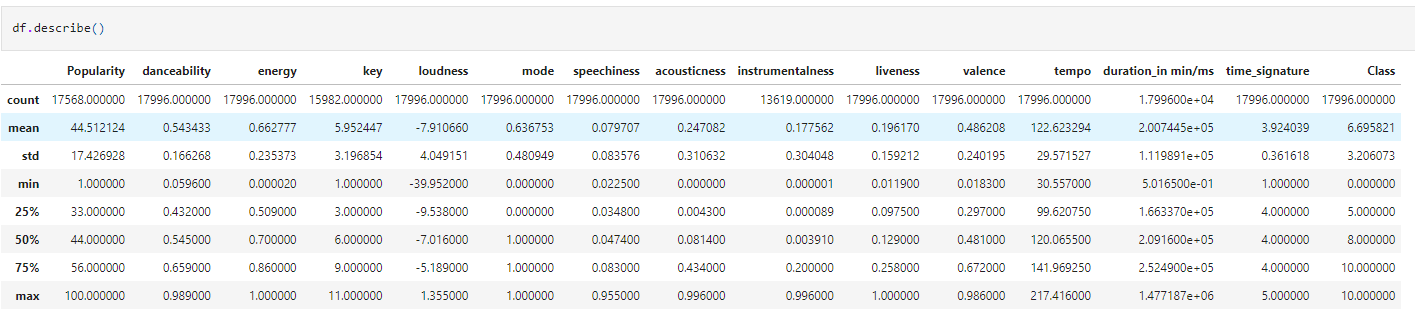
### 

### Data understanding

1. Getting to know the data



From the df.head(), i can tell that I probably need to convert the values in duration\_in min/ms to either all in minutes or all in milliseconds.

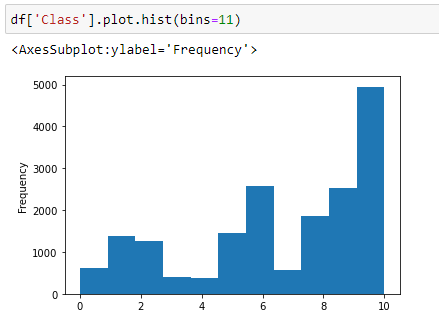


Statistical Values of the different columns to understand the mean, rows of non-nulls, max and min value of each row.

Can also tell that there are missing values in:

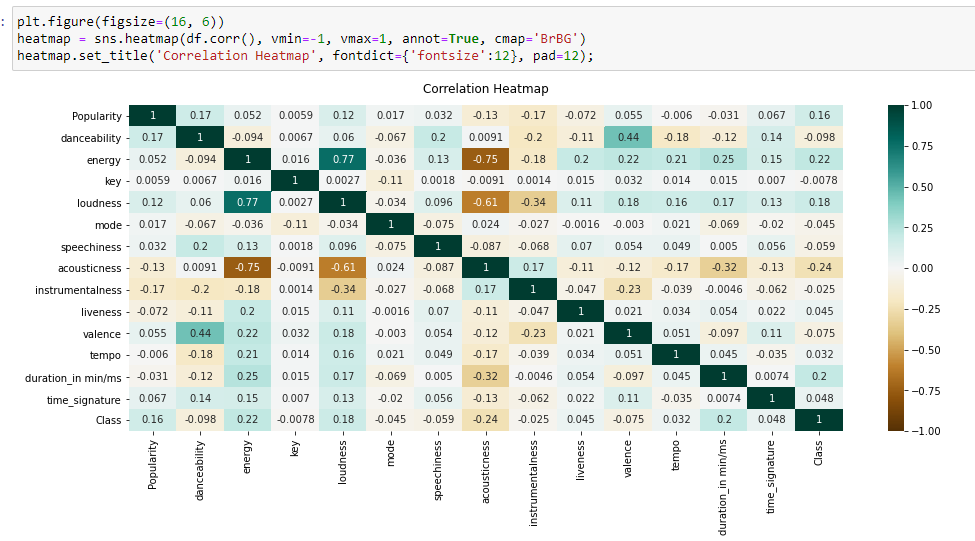
* Popularity
* Key
* instrumentalness

1. Class is the Label



There is a clear class/label imbalance as class 10 is the majority of all the samples, may need to use smote to balance the data or use stratify=y when splitting the data into train test sets.

1. Which features correlate more with the ‘Class’ label



Heatmap shows that Popularity, energy, loudness and duration has a weak correlation to the class while acousticness has a weak negative correlation.

Note: There are some missing value so it might not be fully accurate

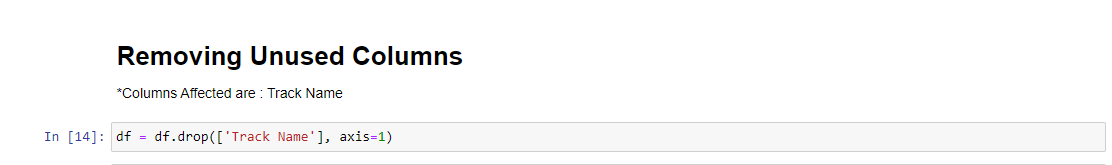
### Data preparation

1. Remove duplicates to prevent issues for model in differentiating the label



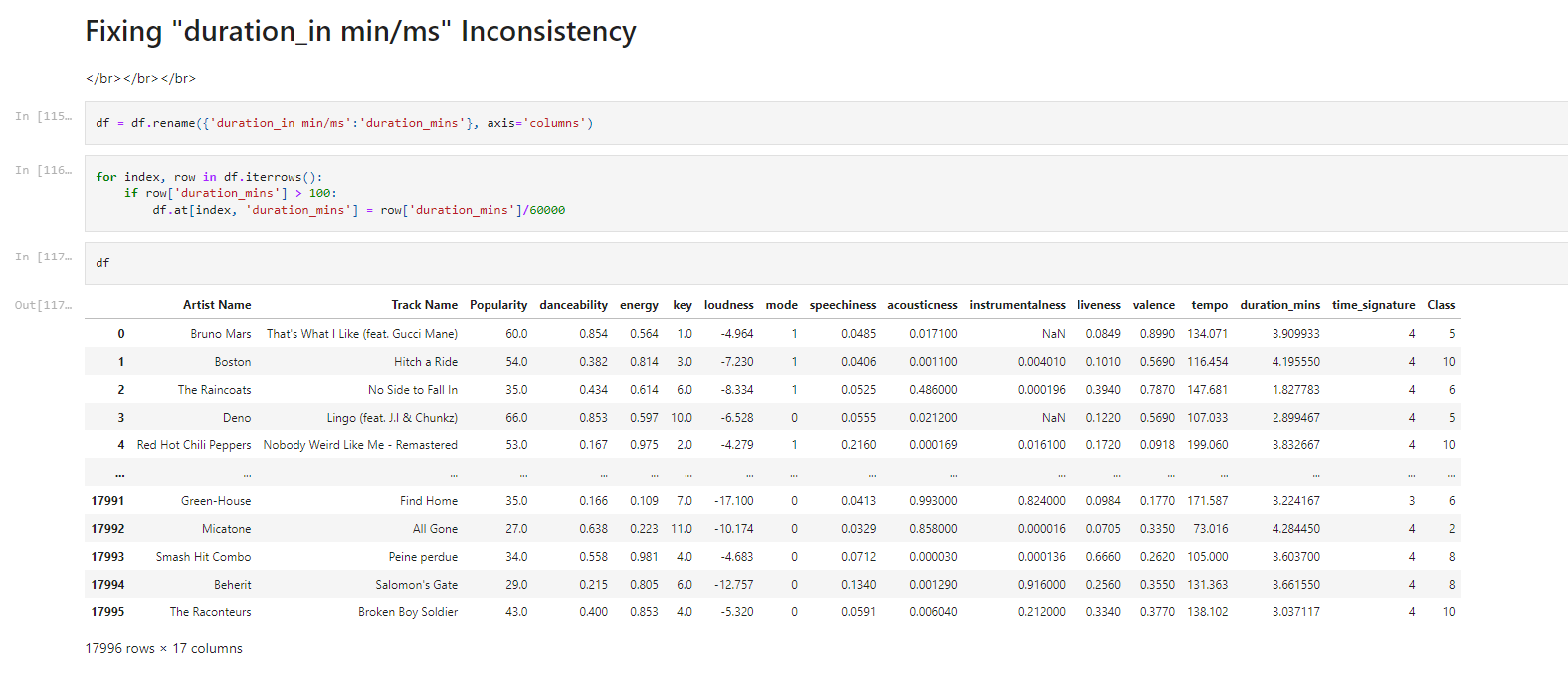
Removed a total of 17996 - 16254 = 1742 Rows

1. Dropping Unused Columns

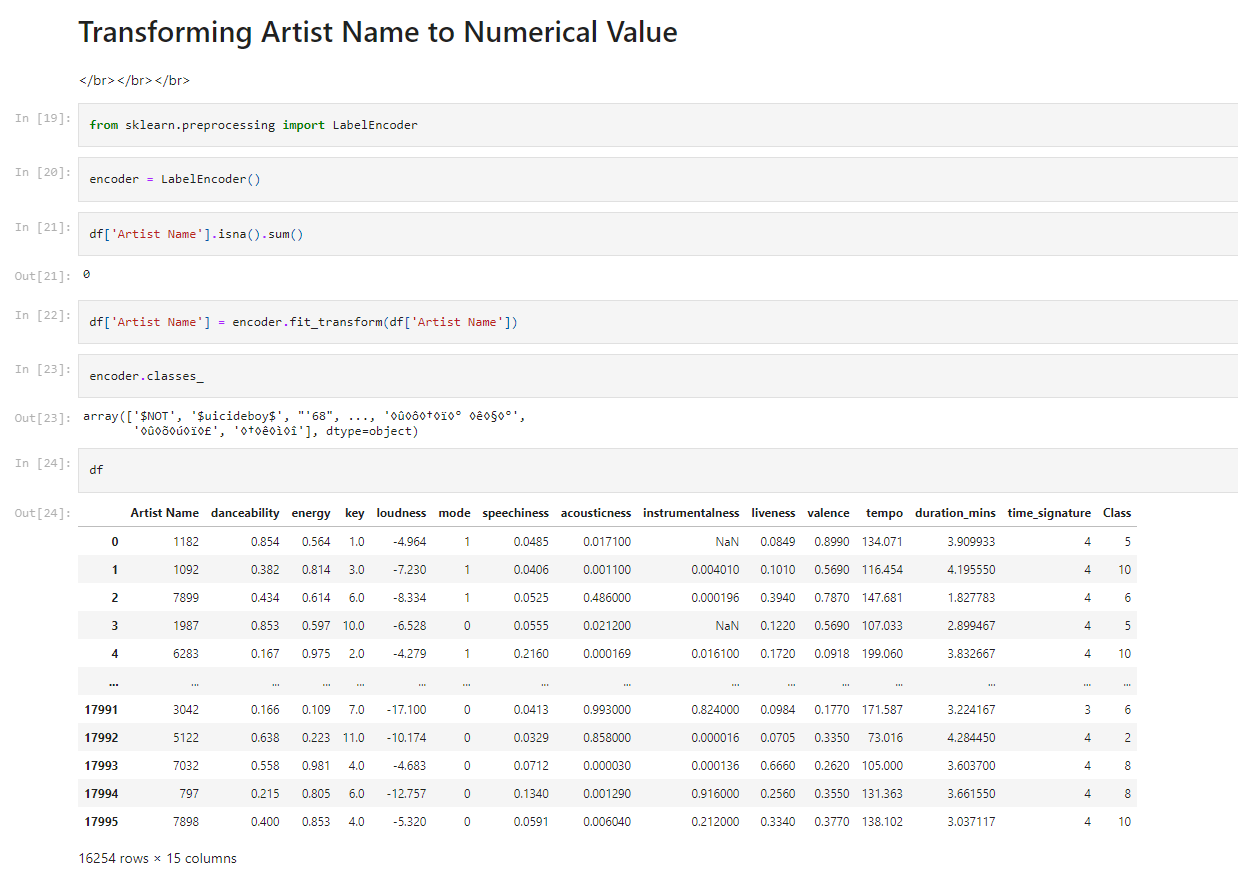


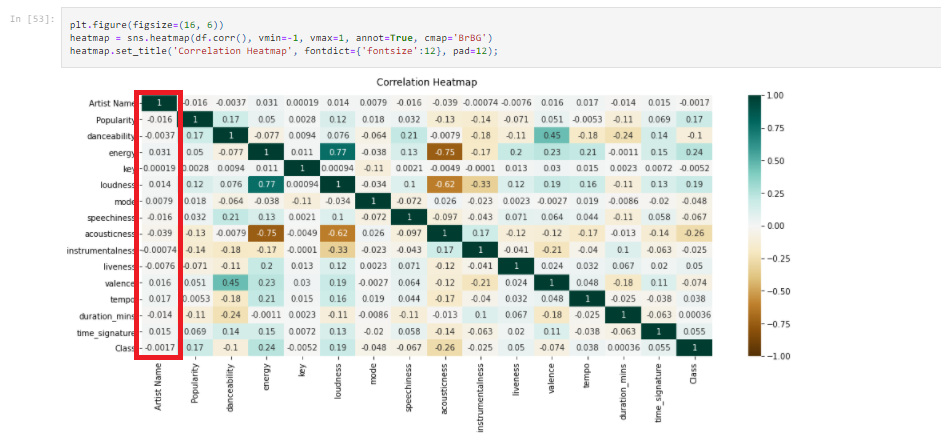
Removed Track Name as i don't plan to use it as unlike Artist Name, might have no correlation to the actual genre (eg. can be named one way, but can be a different genre). Meanwhile artist name is kept as the same artist might only sign a specific set of genres

1. Fixing Duration Column to convert all duration into minutes for consistency



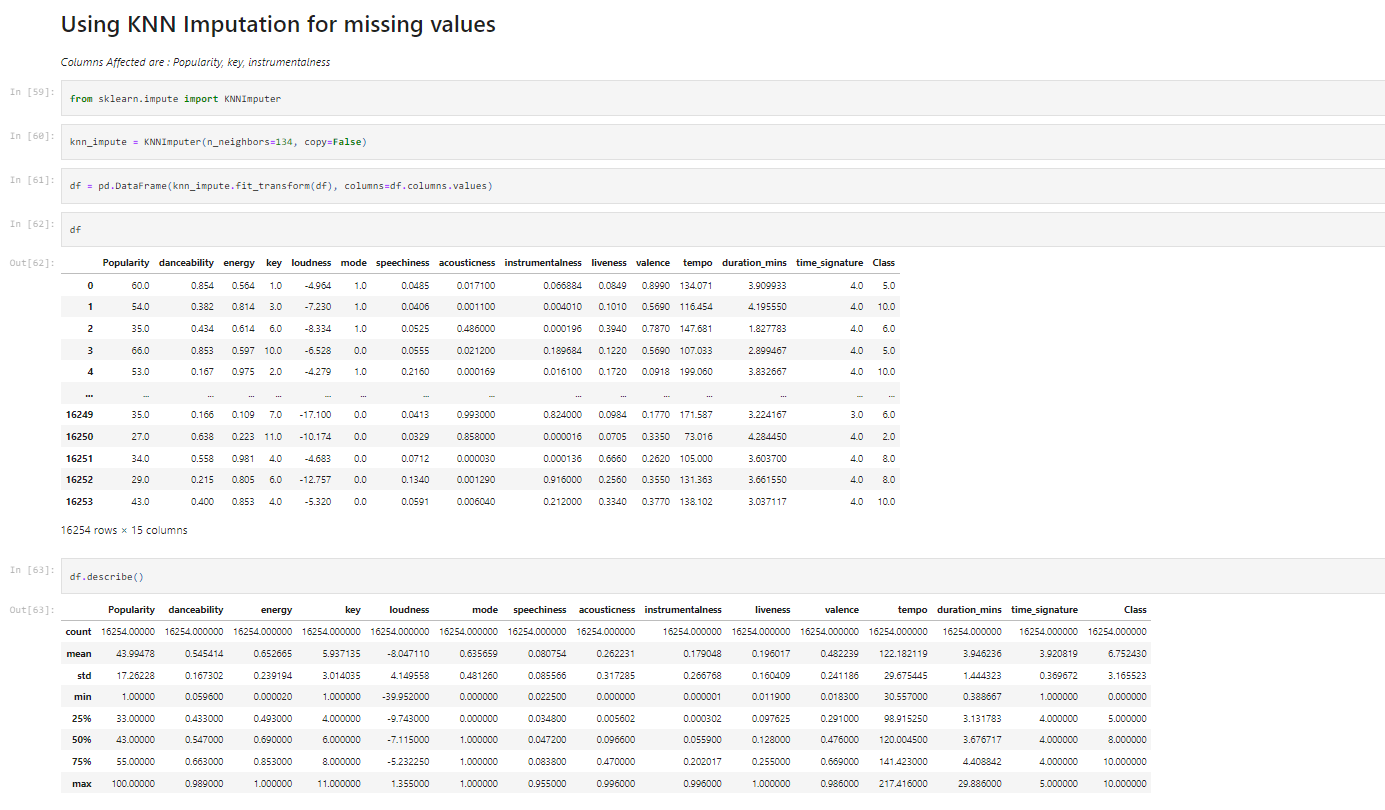
1. Transforming Artist Name into a Numerical Column so that it can be used in the predictive modeling





Decided not to include Artist Name as it has very low correlation value to Class label. Hence will be dropping the Artist Name Column

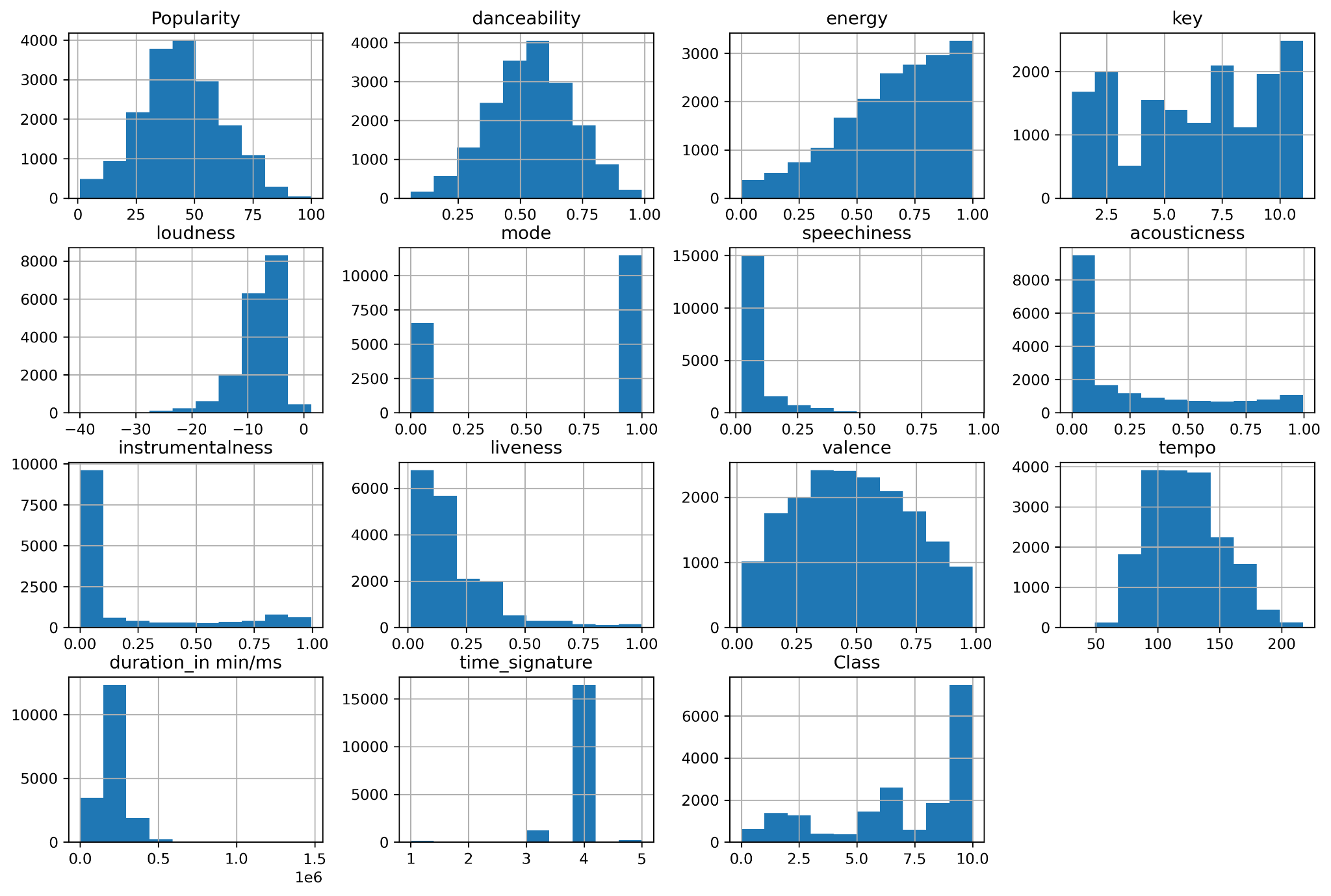
1. Tried using Isolation Forest However, due to null values, it can't run
2. Imputation of Missing Data using KNN Imputation where K = , = 134.



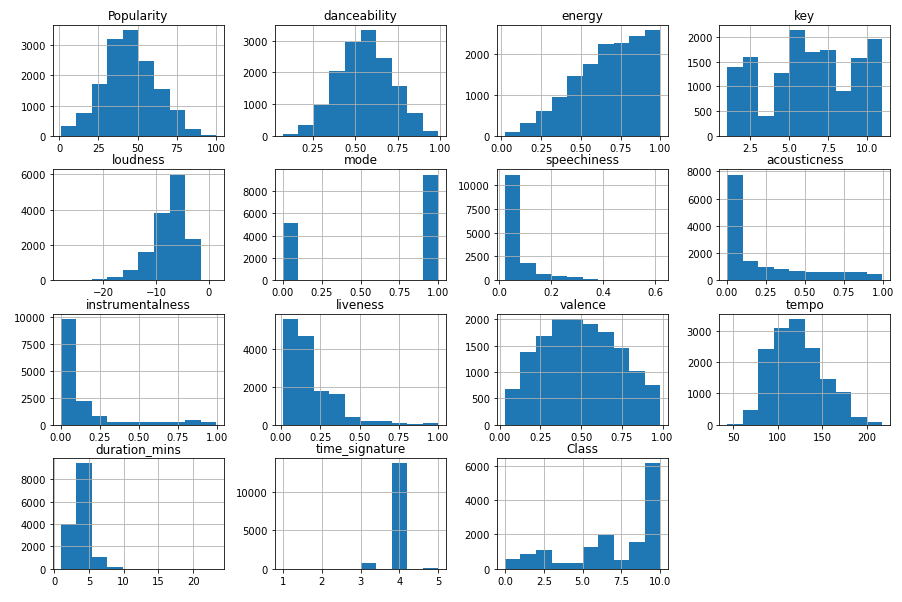
K-Nearest neighbor (K-NN) imputation algorithms are efficient methods to fill in missing data along with taking account to what classes the features belong to. Hence decided to use it instead of replacing with statistical values such as mean as it does not take in consideration of the different classes/labels

1. Outlier Removal Using Isolation Forest

Data Distribution before Outlier Removal

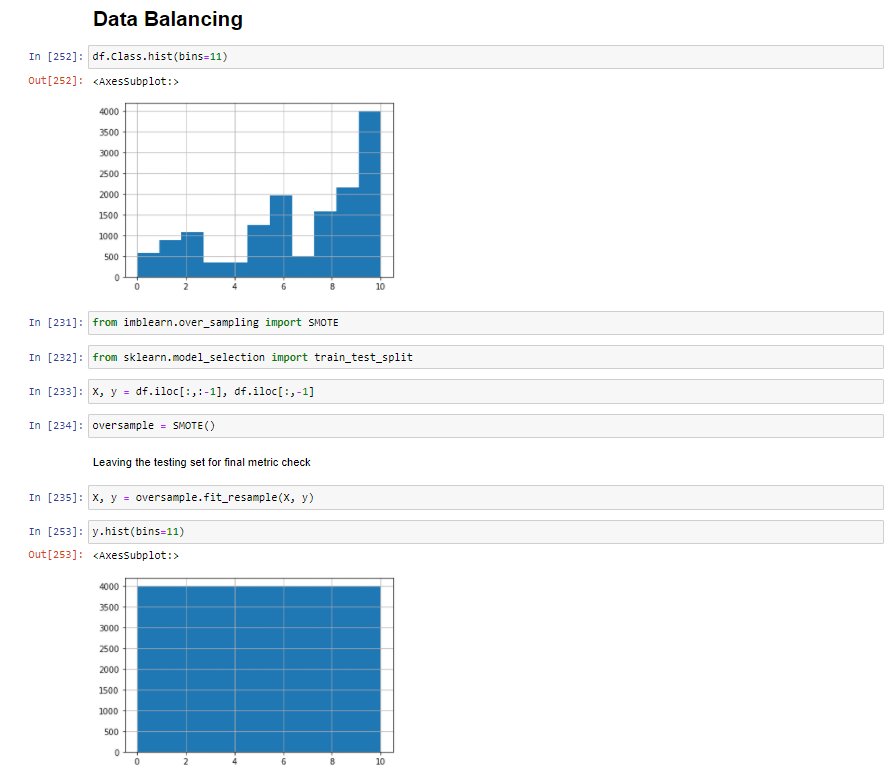


Data Distribution after Outlier Removal

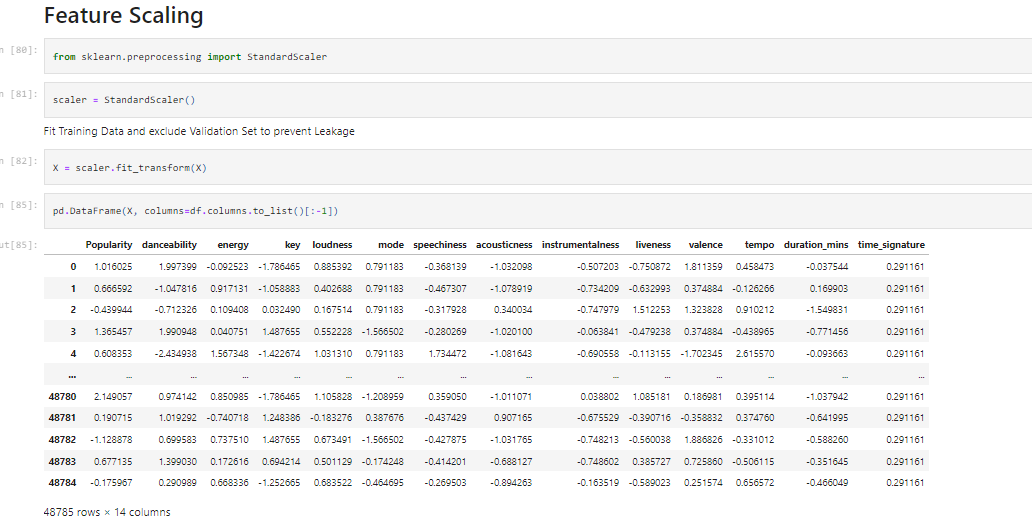


From the Histograms of the data before and after, there are some visible changes in most of the columns such as energy, loudness, acousticness, instrumentalness, tempo.

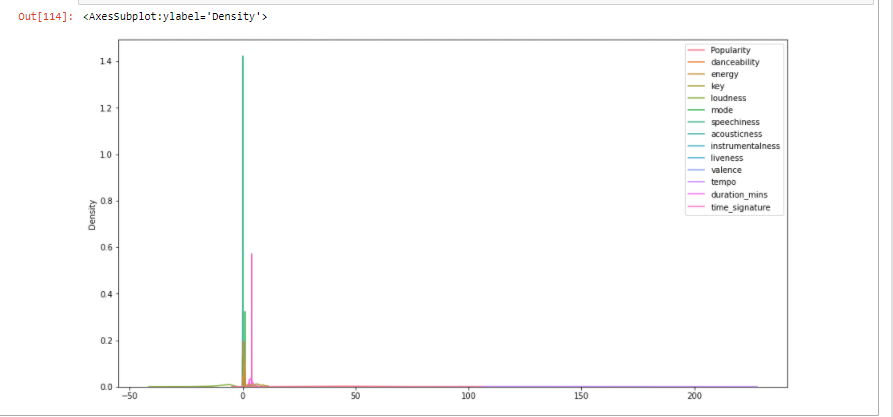
1. Data Balancing the Minority Class using SMOTE



1. Feature Scaling Thru StandardScaler

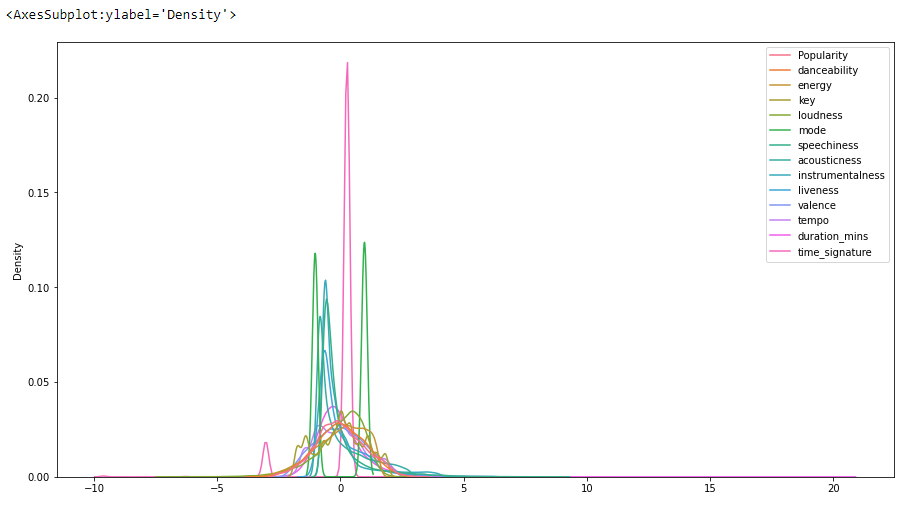


Before Scaling:



The values are more distributed as they having varying min and max values

After Scaling:

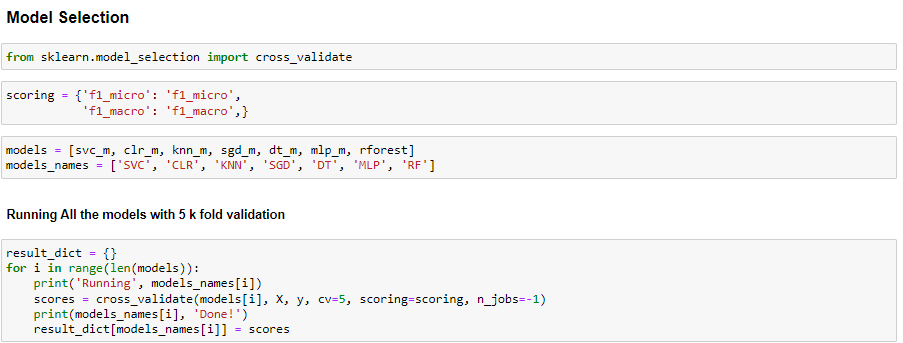


The values are more closer towards each other, with the mean being somewhat in the same positions, hence will help result in a more accurate result for models using distances between points to predict

### Modeling

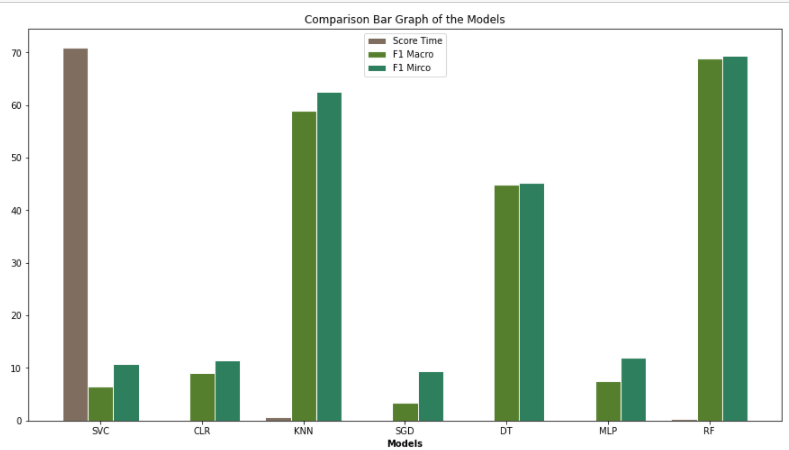
1. Use all classification model, use k fold cross validation to get result of each base models

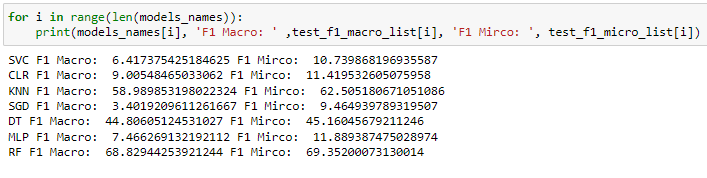






Score in a Graph





For the comparison, i’ll be using Macro F1 score as it is a balance between both recall and precision score along with equal weights on all the classes ( SMOTE balanced already)

From the graph above, Random Forest has the highest F1 score for both in macro (equal weight for each class) and in mirco score (aggregated weight for each class). Additionally, it is also one of the fastest few models via low score time (higher score time, means take longer to run)

SVM took the longest (highest score time) while providing one of the worst f1 scores while logistic regression, SGD classifier and MLP classifier all have low f1 scores

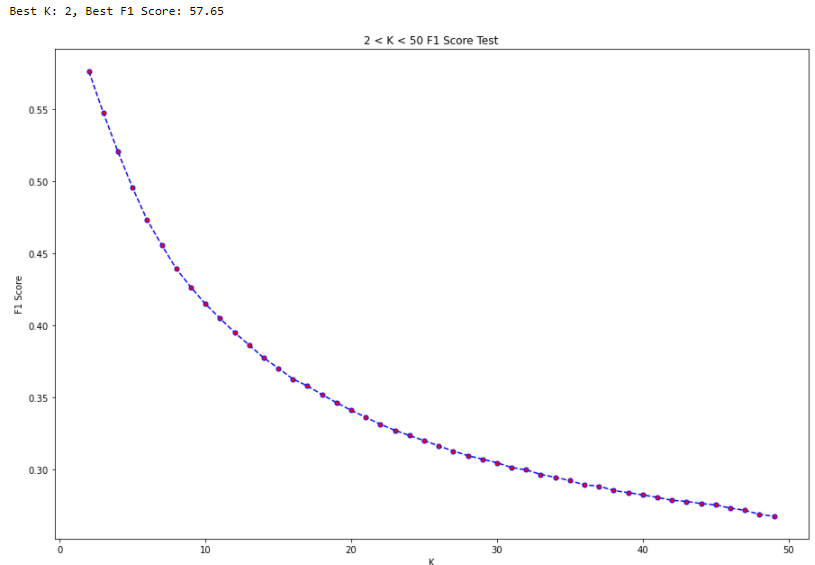
Decision tree will not be continued as i will proceed with Random Forest which uses Decision tree classifier as a base estimator

Hence, to continue to my hyperparameter tuning, i will be using KNN and Random Forest as those two provide the highest base f1 score

### Evaluation

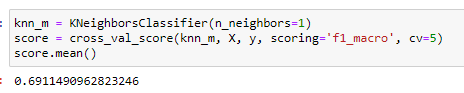
1. Testing K values of K-Neighbour Classifier to see if it will provide a more accurate result,





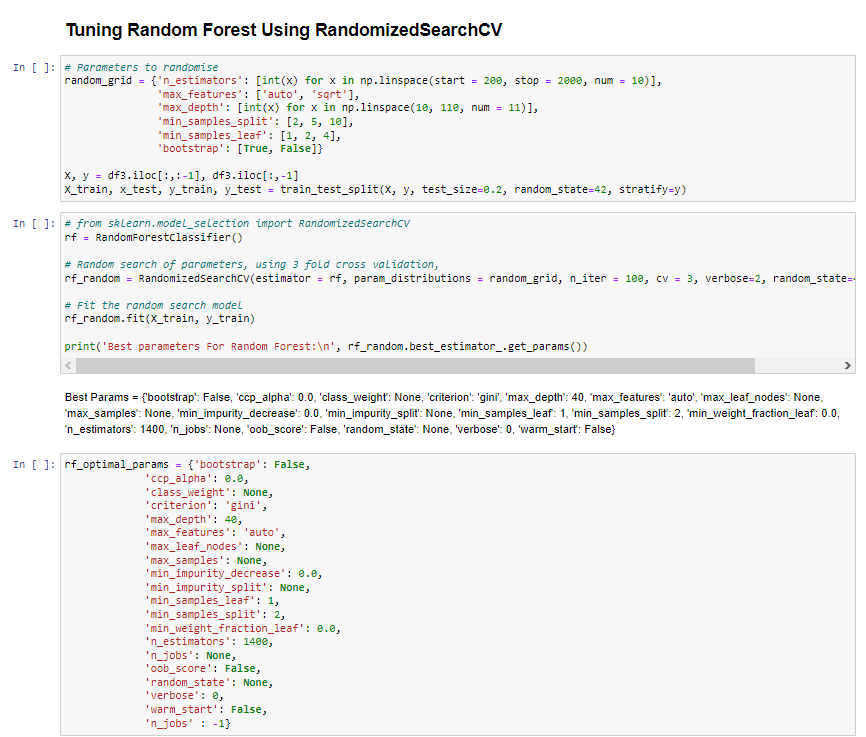
Graph of F1\_score from K=2 to K=50 whereby K=2 would be the recommended k value as it is the square root of the total number of rows (per class after SMOTE). There seems to be a trend where as the k value increases, the score decreases.

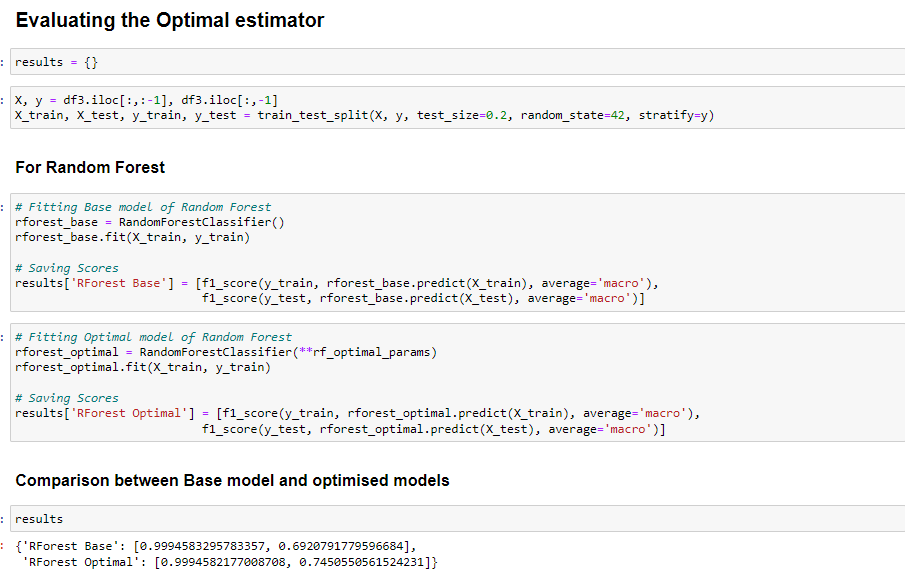
Additional: k = 1

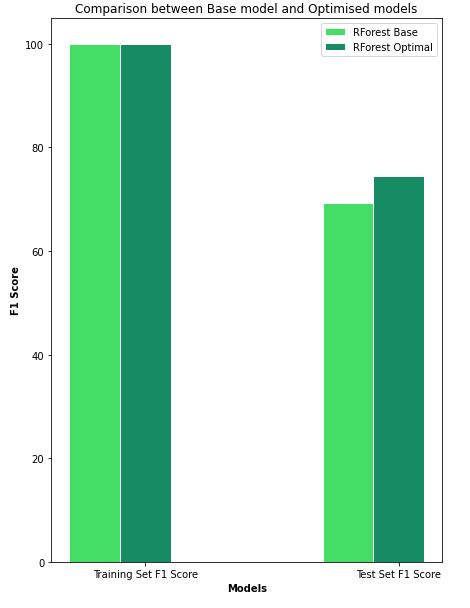


However, the F1\_score is still beaten by base version of the Random Forest, Hence will continue to optimize Random Forest to check for further improvements

1. Randomized Search to find optimal parameters for Random Forest Classifier

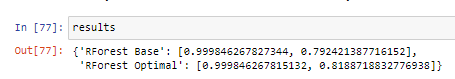




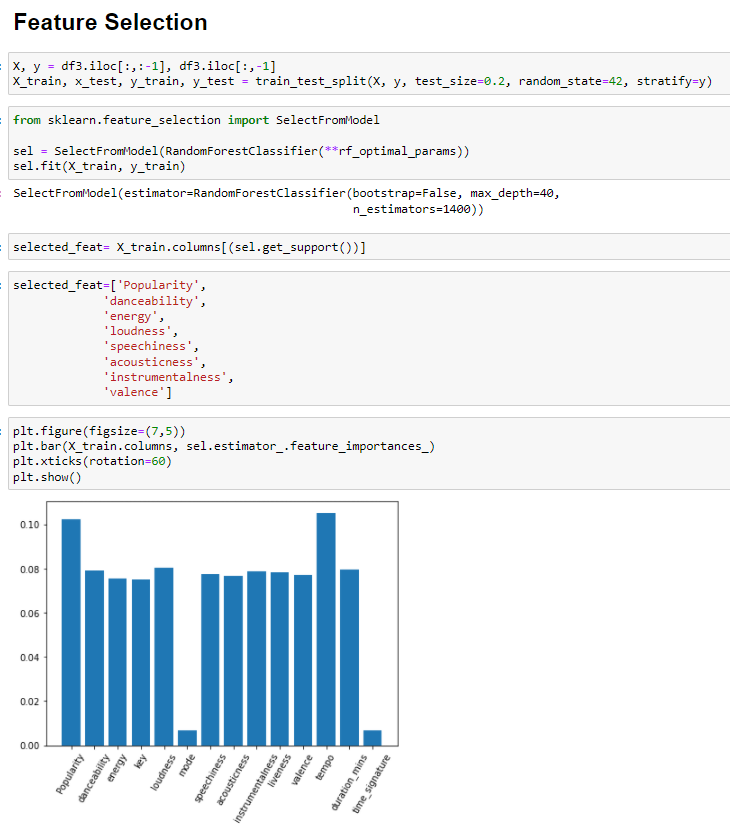


As you can see, although the training set f1 score is similar on both the Base Random Forest and Optimal Random Forest, the test data set return a higher score on the optimal Random Forest than the Base classifier

Raw Data:



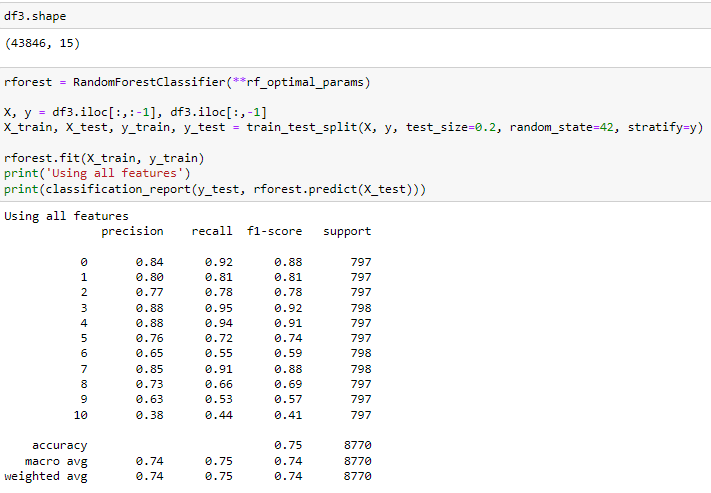
1. Feature Selection via Feature Importance in Optimal Random Forest Decision Tree



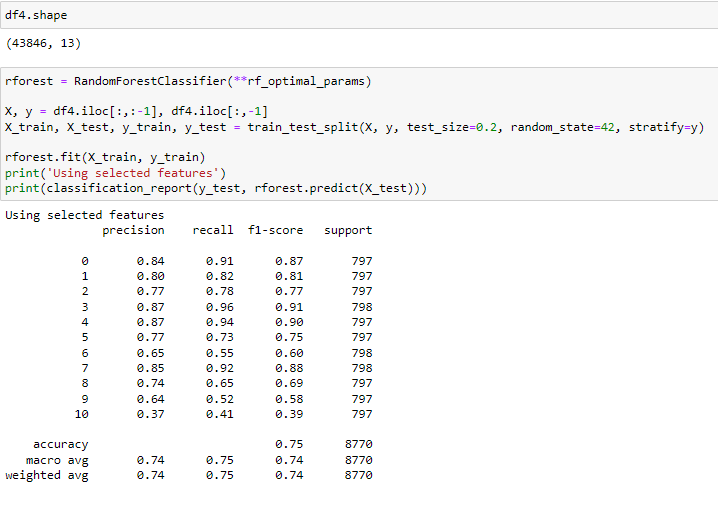
Those that features that were dropped are mode, time\_signature as they all had a importance value of less than 0.7

1. Checking if Feature Selection will affect the score of the optimal RForest

Random Forest with All Features gives macro avg f1 score of 0.75 which is similar to the results previously



Random Forest with Selected Features (Everything except for mode & time signature) provided the same result as to the model that ran with all the features.

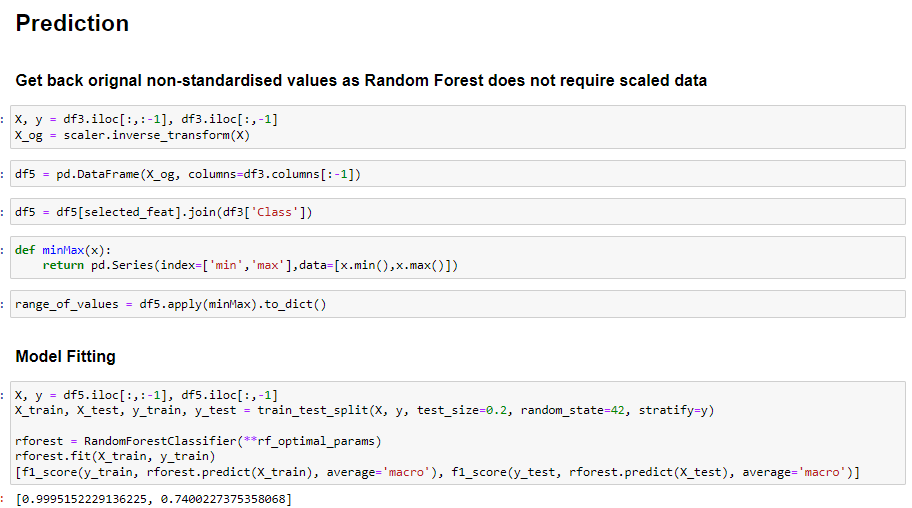


Hence, since the model using selected features provided the same f1 score as the model using all the features, this means that i can continue on using the selected features and drop the features that were not included.

Reason been is that a a lower feature count while having the same result means that resources will be saved while providing the same result

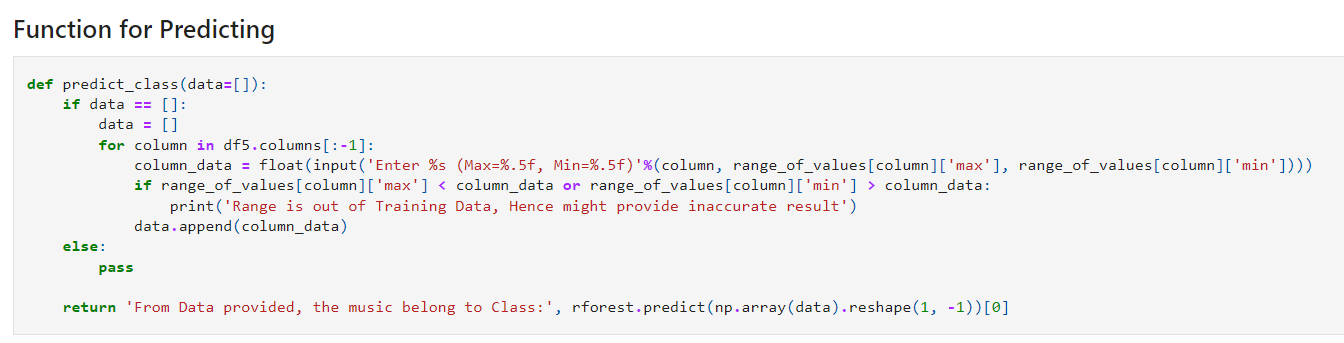
### Prediction

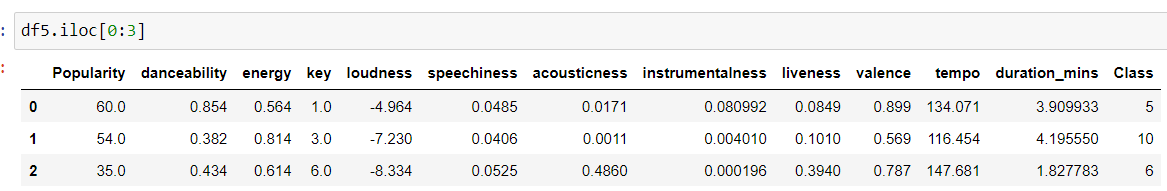
1. Preparing Final data and best Model (Random Forest with selected Features)

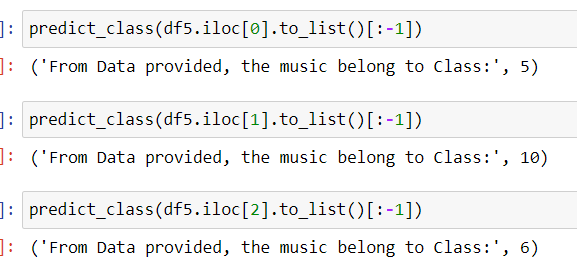


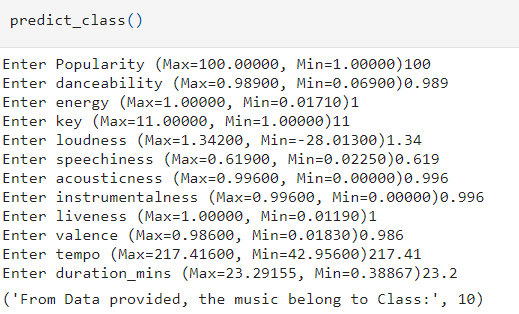
\*Range of values is for user to know what are the max and min of each feature that were used for training the model

1. Function for prediction along with prediction examples









The function created can either take in a list of variable (the features values) or can be empty, if empty, it will ask u to input the variable before trying to predict.

For the inputs, if the value enter is above the Max or below the Min values, it will give a warning that the max and min was the data used for the training of the model, hence the model might predict inaccurately if the values entered are beyond the max and min values used to train it.

### Summary

1. Identify which column is your label.

The ‘Class’ Attribute is the label that will be predicted

2. Decide if this is a classification or regression problem.

This is a Classification Problem as the goal of this assignment is to predict the class of a piece of music based on the characteristic or features of the music.

Additionally, the class that we have to predict are the integers between 0 to 10, and contains no continuous number, hence is a classification problem

3. Data Preparation Steps Used

* Removing Duplicate Values/Rows
* Remove Unused Columns (Track Name)
* Encoded Artist Name into Numerical data
* Fixing data inconsistency in “duration\_in min/ms” to only minutes
* KNN Imputation of Missing Data
* Outlier Removal using IsolationForest
* Data Balancing to balance Class Minority
* Feature Scaling using Standard Scaler

6. Modeling steps

Using 5 K- Cross validation, i tested the following algorithms with their default params and compared their macro f1 score to determine which model to do hyperparameter tuning:

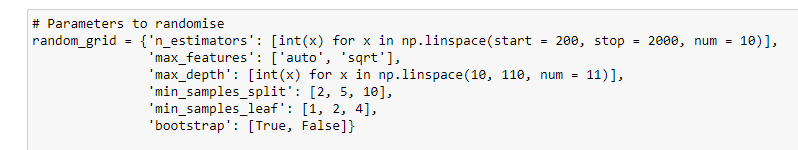
* K-Nearest Classifier
* Logistic Regression
* SVM
* SGD Classifier
* Decision Tree
* MLP Classifier
* Random Forest

Reasoning for using f1 score instead of accuracy or recall would be that there are no business implication or use for this predictive model, hence i would rather compare the models using a f1 score which provide the average of both accuracy and recall as it will provide a better and fairer comparison on the performance of the model.

Best Performing Models (Order of accuracy) : Random Forest, K-Nearest Classifier

7. Hyperparameter Tuning of selected models

For K-Nearest Classifier, i change the number of n\_neighbors from a range of 1 - 50, and found out that n\_neighbour = 1 provided the best result while the more n\_neighbours there are, the worst the f1 score. Hence the best k=1 params, provided me with a f1 score of ~69.007% (5sf)



For Random Forest, i used RandomizedSearchCV to search through the range of different settings and from there, the best params found gave a f1 score of ~74.012% (5sf)

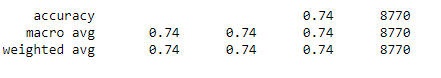
Hence, From this, i continue with the Random Forest Algorithm with the optimal parameters found to continue my evaluation and prediction steps

8. Feature Selection

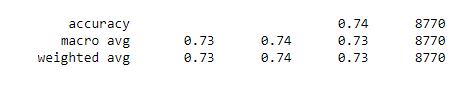
After Comparing the best params for Random Forest and K-Nearest Classifier and choosing Random Forest as it provided a better score, I move on to see if i can reduce the dimension/amount of features used in the model via using SelectFromModel to get the features to keep (which ended up removing two features, mode & time\_signature)

After the feature selection, i test it the optimal Random Forest algo with both the 14 feature dataset and the reduced 12 feature dataset, they both provided me with roughly the same score,

All Feature



Reduced Feature



Hence, since the chance in score is insignificant, i decided to drop the unused features (mode & time\_signature) to save up resources in the final prediction

Final Feature List that was used in the prediction example:

['Popularity', 'danceability', 'energy', 'key', 'loudness', 'speechiness', 'acousticness', 'instrumentalness', 'liveness', 'valence', 'tempo', 'duration\_mins']