Project Title

Designing an End-to-End Machine Learning Use Case

Name

Mohamed Gamal

1. Overview of the Dataset

The dataset contains the following columns:

popularity: The popularity score of the song.

acousticness: A confidence measure of whether the track is acoustic.

danceability: How suitable a track is for dancing.

duration_ms: The duration of the song in milliseconds.

energy: The intensity and activity measure of the song.

instrumentalness: Predicts whether a track contains no vocals.

liveness: The presence of an audience in the recording.

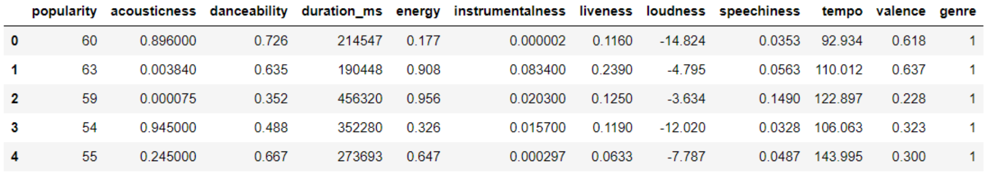
loudness: The overall loudness of the track in decibels (dB).

speechiness: The presence of spoken words in a track.

tempo: The speed or pace of the song, measured in beats per minute (BPM).

valence: The musical positiveness conveyed by a track.

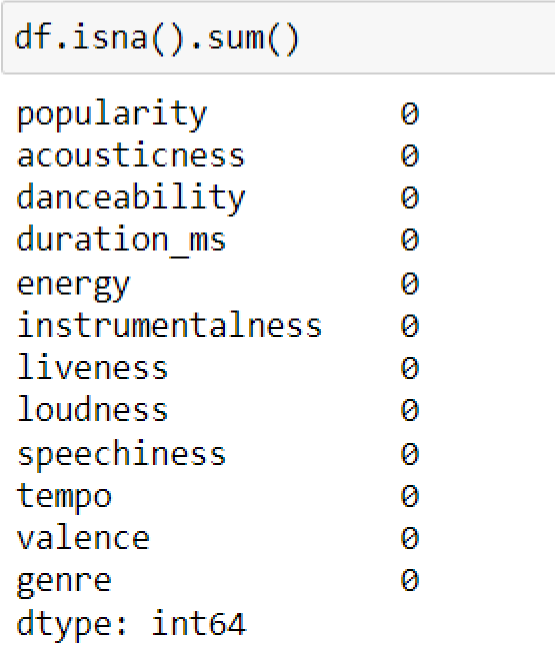
genre: The genre of the song (all entries are '1,0', which suggests a binary genre focus).



	popularity	acousticness	danceability	duration_ms	energy	instrumentalness	liveness	loudness	speechiness	tempo	valence	genre
0	60	0.896000	0.726	214547	0.177	0.000002	0.1160	-14.824	0.0353	92.934	0.618	1
1	63	0.003840	0.635	190448	0.908	0.083400	0.2390	-4.795	0.0563	110.012	0.637	1
2	59	0.000075	0.352	456320	0.956	0.020300	0.1250	-3.634	0.1490	122.897	0.228	1
3	54	0.945000	0.488	352280	0.326	0.015700	0.1190	-12.020	0.0328	106.063	0.323	1
4	55	0.245000	0.667	273693	0.647	0.000297	0.0633	-7.787	0.0487	143.995	0.300	1

2. Summary Statistics

Let's check if there is null values and compute some basic summary statistics for each column.

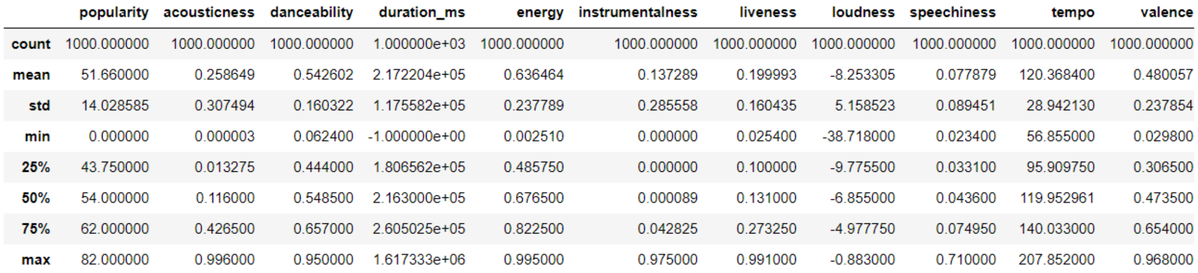


```
df.isna().sum()
```

```
popularity          0
acousticness        0
danceability        0
duration_ms         0
energy              0
instrumentalness    0
liveness            0
loudness            0
speechiness         0
tempo               0
valence             0
genre               0
dtype: int64
```

After that, I checked to see if the dataset had duplicates, and my dataset had no duplicates.

And this is my dataset Decription



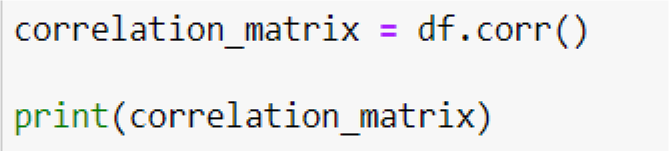
	popularity	acousticness	danceability	duration_ms	energy	instrumentalness	liveness	loudness	speechiness	tempo	valence
count	1000.000000	1000.000000	1000.000000	1.000000e+03	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000
mean	51.660000	0.258649	0.542602	2.172204e+05	0.636464	0.137289	0.199993	-8.253305	0.077879	120.368400	0.480057
std	14.028585	0.307494	0.160322	1.175582e+05	0.237789	0.285558	0.160435	5.158523	0.089451	28.942130	0.237854
min	0.000000	0.000003	0.062400	-1.000000e+00	0.002510	0.000000	0.025400	-38.718000	0.023400	56.855000	0.029800
25%	43.750000	0.013275	0.444000	1.806562e+05	0.485750	0.000000	0.100000	-9.775500	0.033100	95.909750	0.306500
50%	54.000000	0.116000	0.548500	2.163000e+05	0.676500	0.000089	0.131000	-6.855000	0.043600	119.952961	0.473500
75%	62.000000	0.426500	0.657000	2.605025e+05	0.822500	0.042825	0.273250	-4.977750	0.074950	140.033000	0.654000
max	82.000000	0.996000	0.950000	1.617333e+06	0.995000	0.975000	0.991000	-0.883000	0.710000	207.852000	0.968000

3. Correlation Analysis

We'll examine the correlation between different variables to identify any significant relationships.

Step 1: Calculate the Correlation Matrix

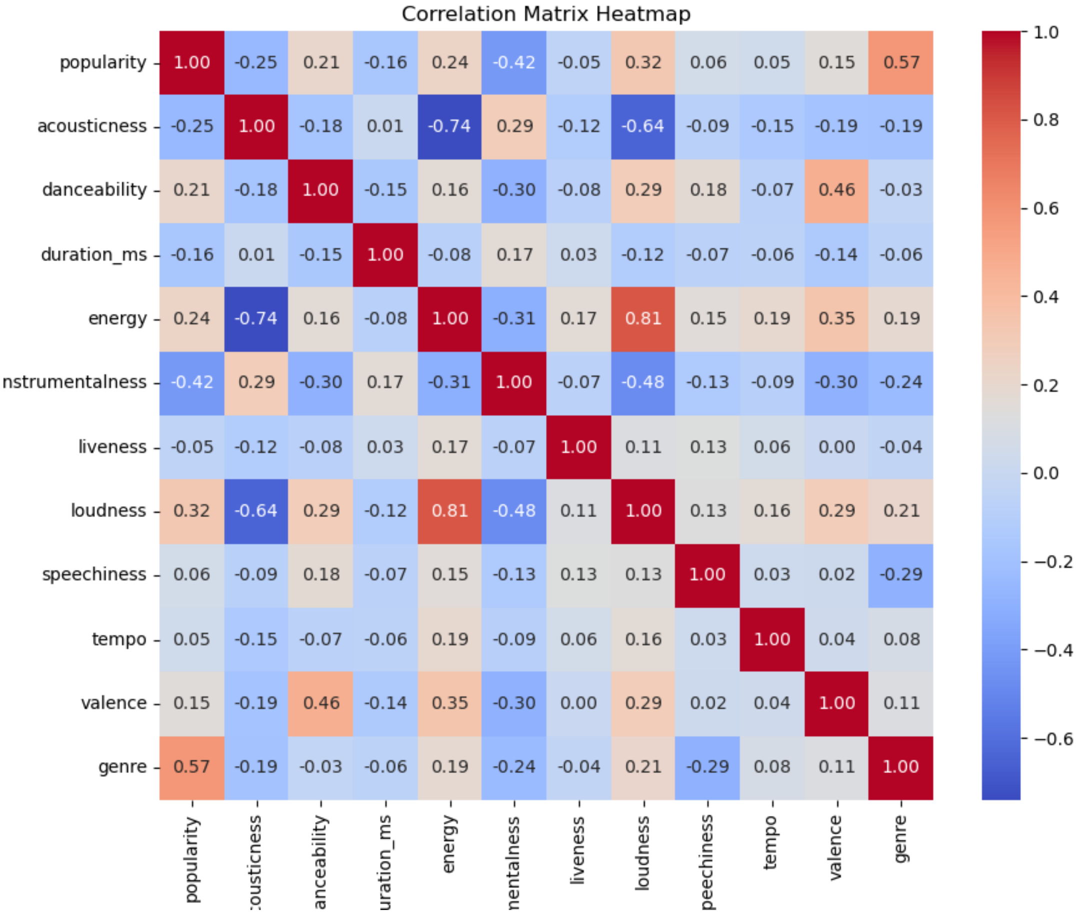
Using a tool like pandas in Python, you can calculate the correlation matrix, which shows the Pearson correlation coefficient between each pair of variables.



```python
correlation_matrix = df.corr()

print(correlation_matrix)
```

Step 2: Interpret the Correlation Matrix

The output of the above code is a correlation matrix, where each cell (i, j) represents the correlation coefficient between the ith and jth variables.



Key Observations

energy and loudness: A correlation of 0.81 indicates a strong positive relationship.

Popularity and genre: A correlation of 0.57 indicates a moderate positive relationship.

valence and danceability: A correlation of 0.46 indicates a moderate positive relationship.

Acousticness and energy: A correlation of -0.74 shows a strong negative relationship.

loudness and Acousticness: A correlation of -0.64 shows a strong negative relationship.

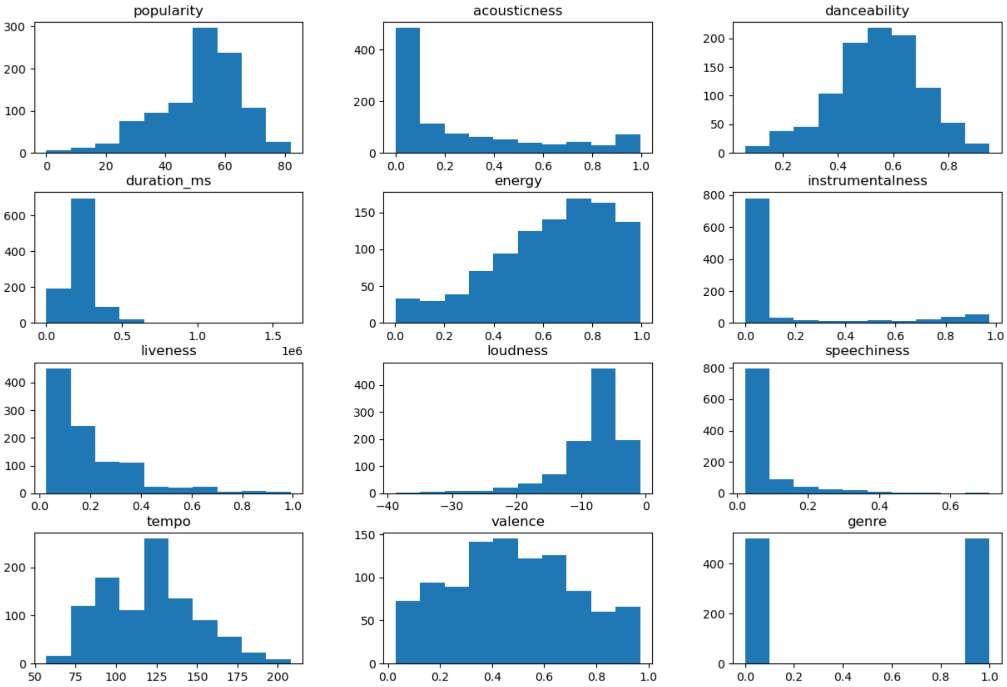
loudness and instrumentalness: A correlation of -0.48 shows a moderate negative relationship.

Popularity and instrumentalness: A correlation of -0.42 shows a moderate negative relationship.

4. Histogram

Histograms give a clear visual representation of the distribution of data in each column. You can see if the data is normally distributed or not.

Histograms can help in identifying extreme values that may need further investigation or handling.

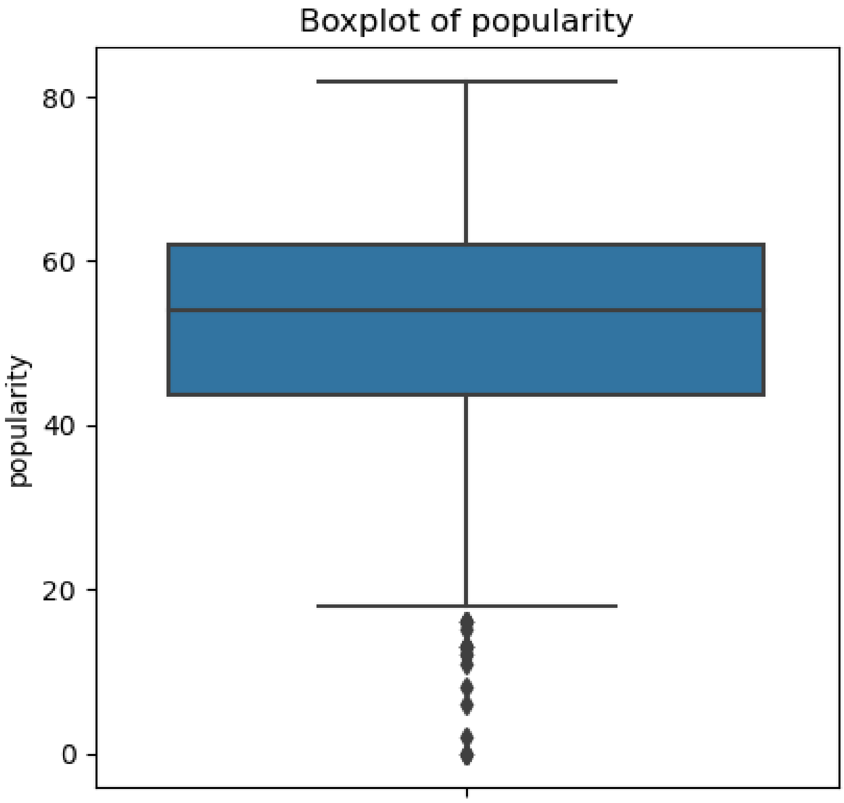


5. Outlier Handling

We will try the model with outlier handling, then try it without outlier handling. Because the outliers may be useful for my dataset, the accuracy of the model will determine that.

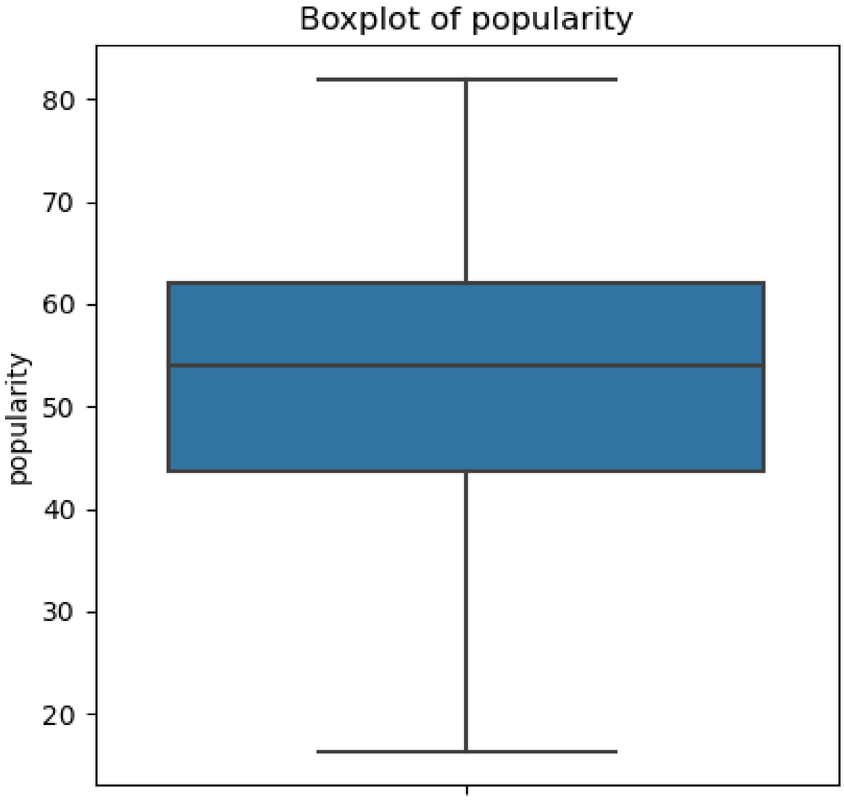
To handle the outliers, we will make a boxplot to see if there is an outlier or not, and if there are outliers, we will use the IQR method to handle them. If there are no outliers, we will ignore this column and check the next one.

Let's start with the 1st column named popularity.

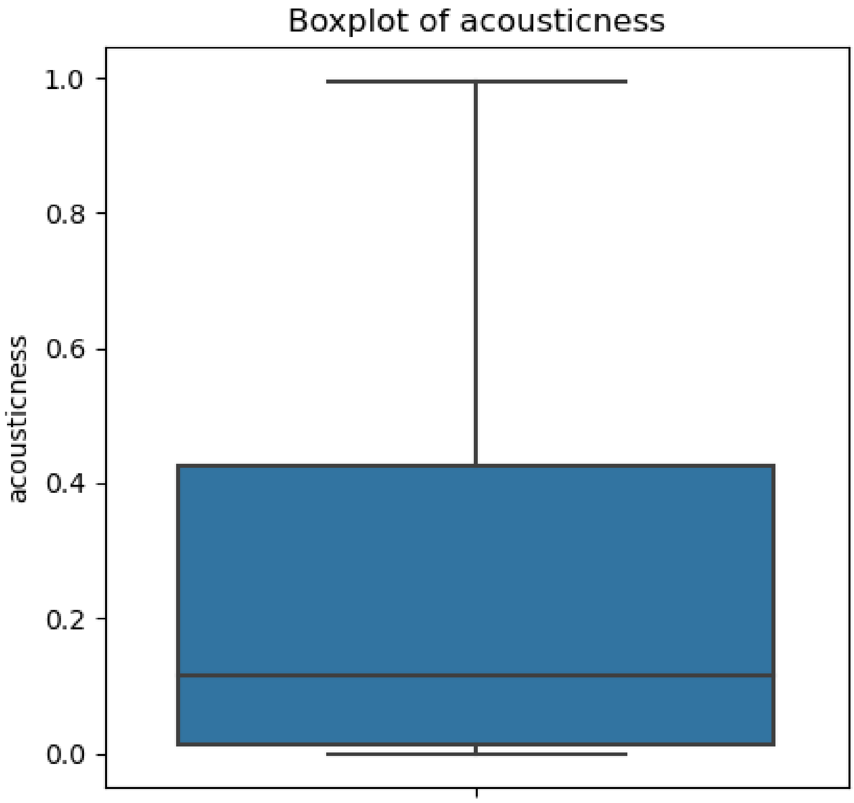


We found that there are outliers, so we will handle it using the IQR method.

After handling the outliers, the figure will change and be like this:

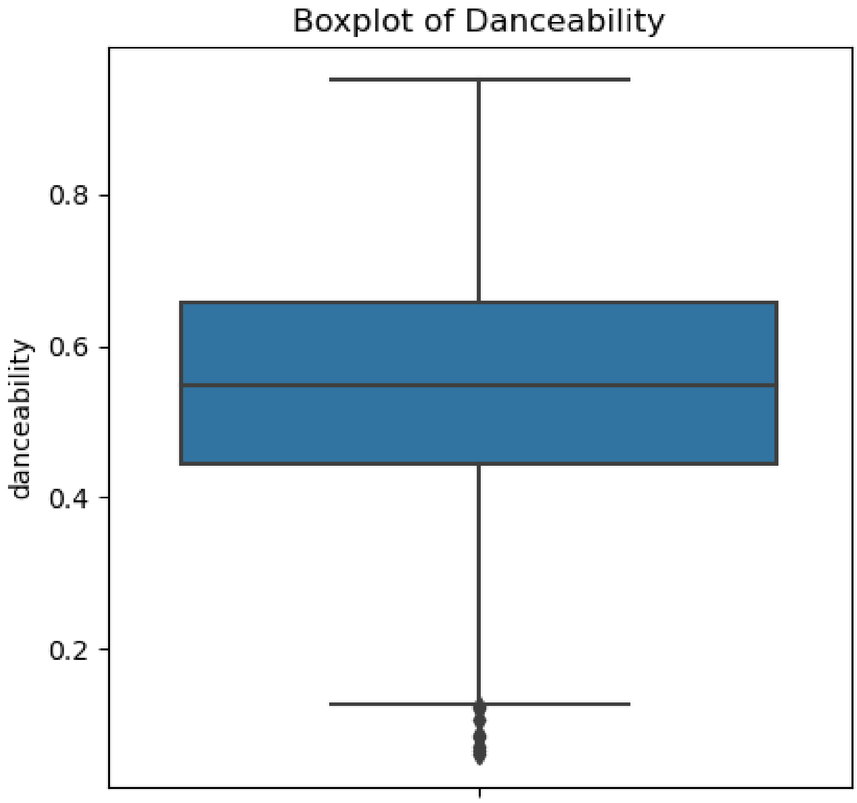


Let’s continue with the 2nd column named acousticness.



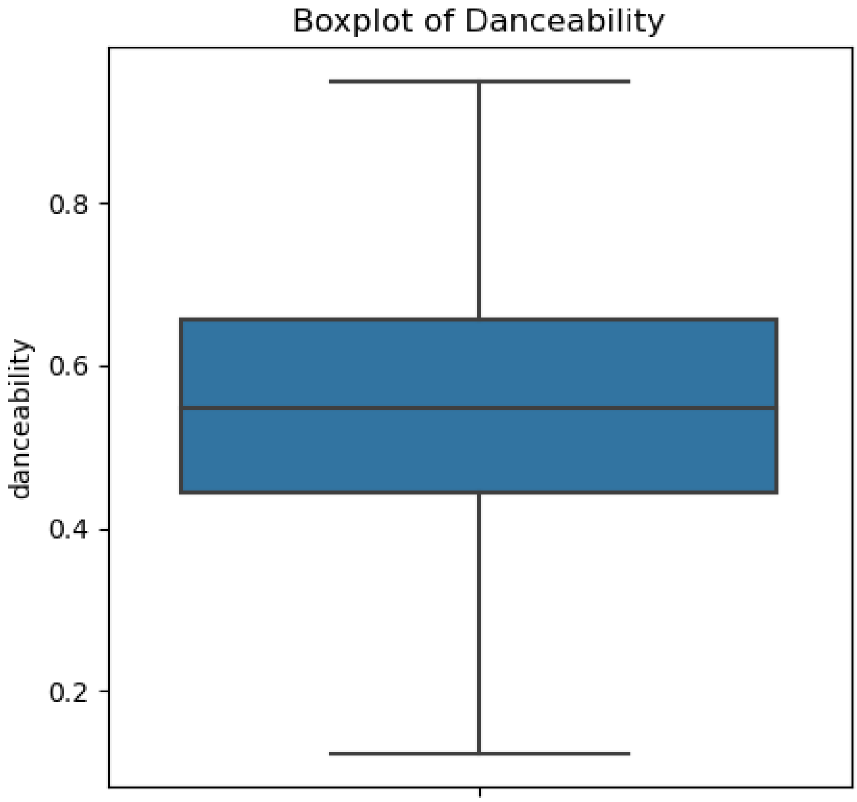
There are no outliers.

Let’s continue with the 3rd column named Danceability.

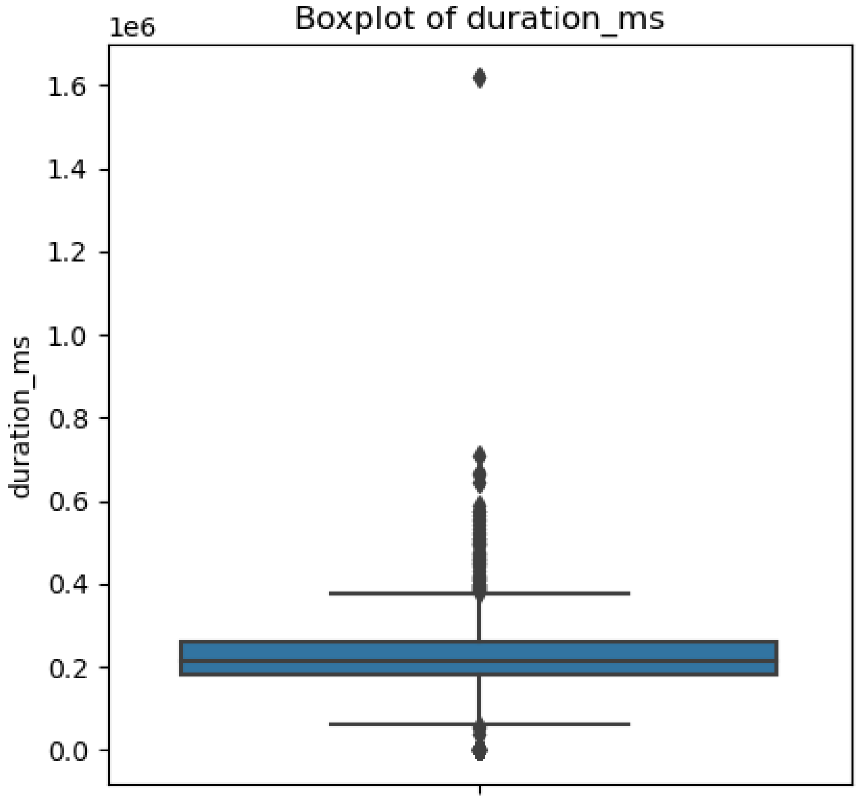


We found that there are outliers, so we will handle it using the IQR method.

After handling the outliers, the figure will change and be like this:

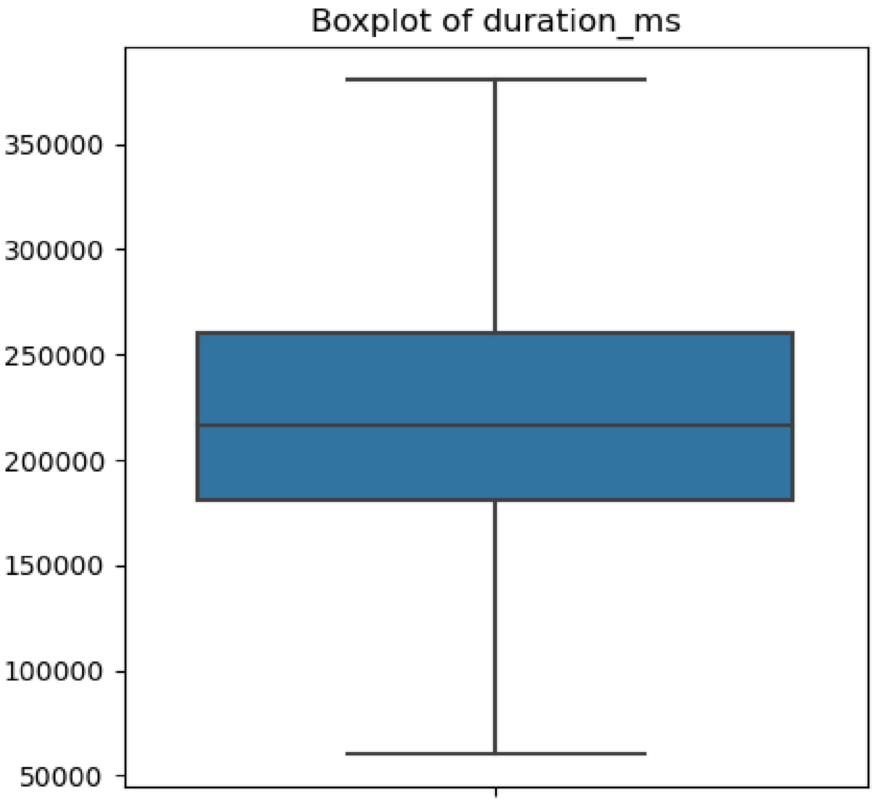


Let’s continue with the 4th column named duration_ms.

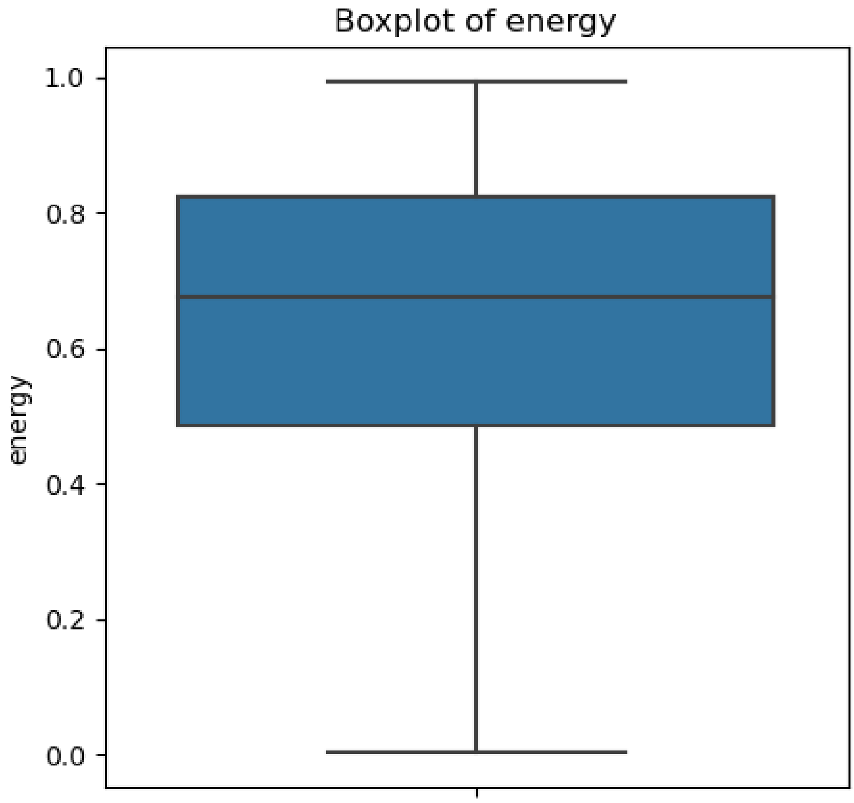


We found that there are outliers, so we will handle it using the IQR method.

After handling the outliers, the figure will change and be like this:

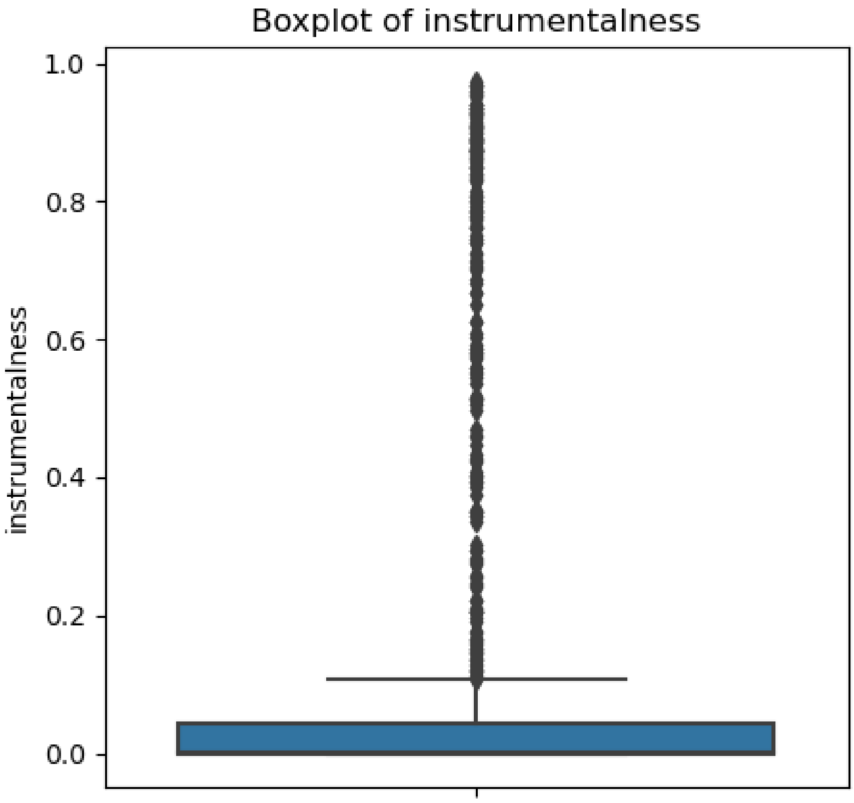


Let’s continue with the 5th column named Energy.



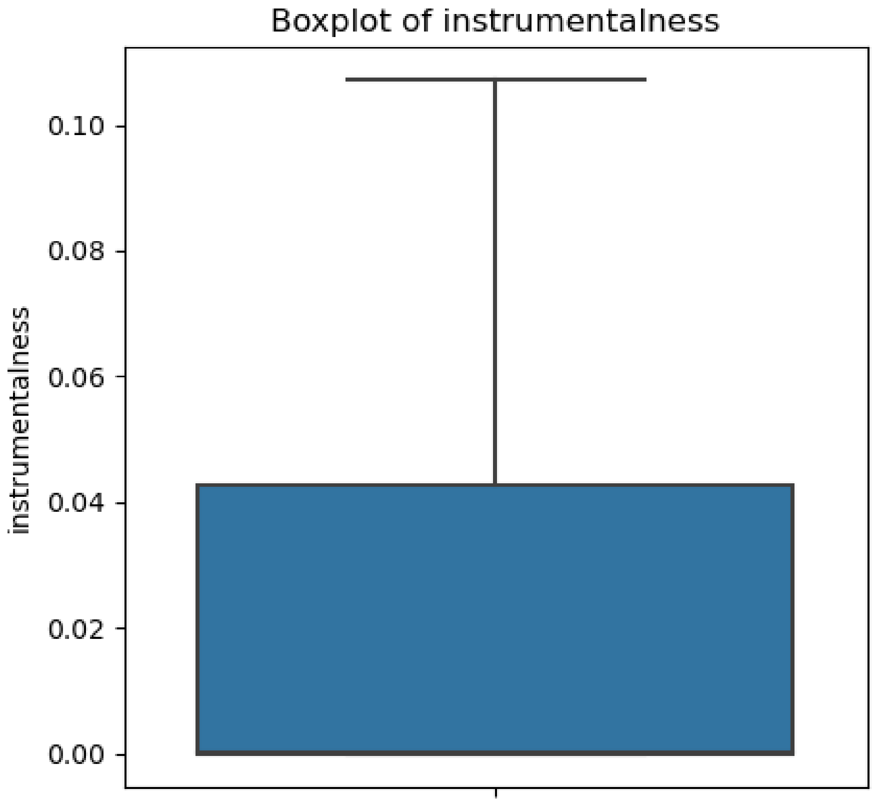
There are no outliers.

Let’s continue with the 6th column named instrumentalness.

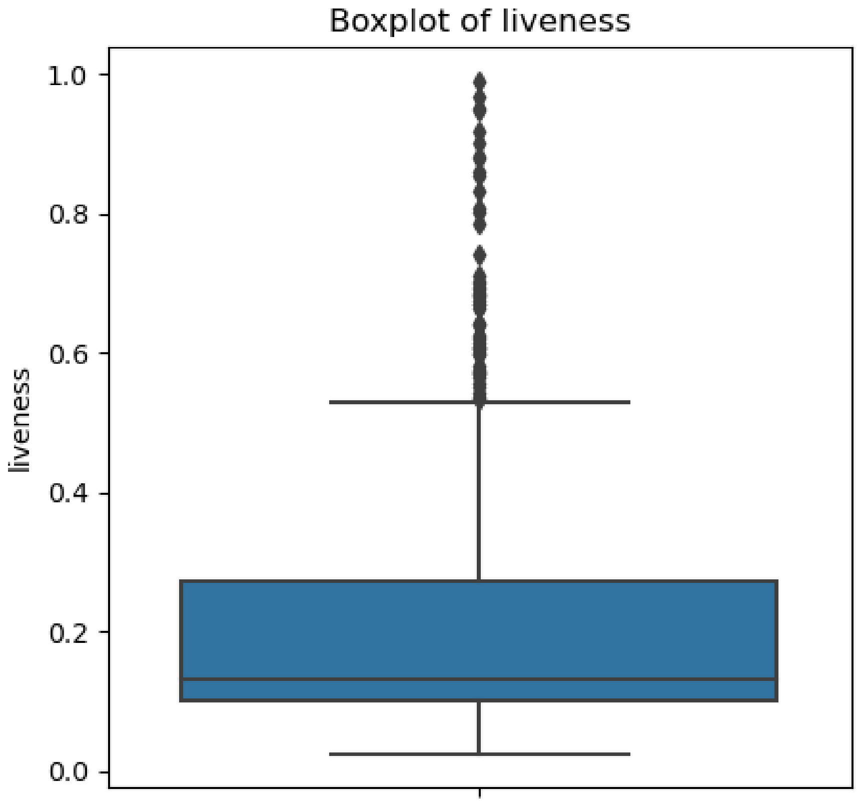


We found that there are outliers, so we will handle it using the IQR method.

After handling the outliers, the figure will change and be like this:

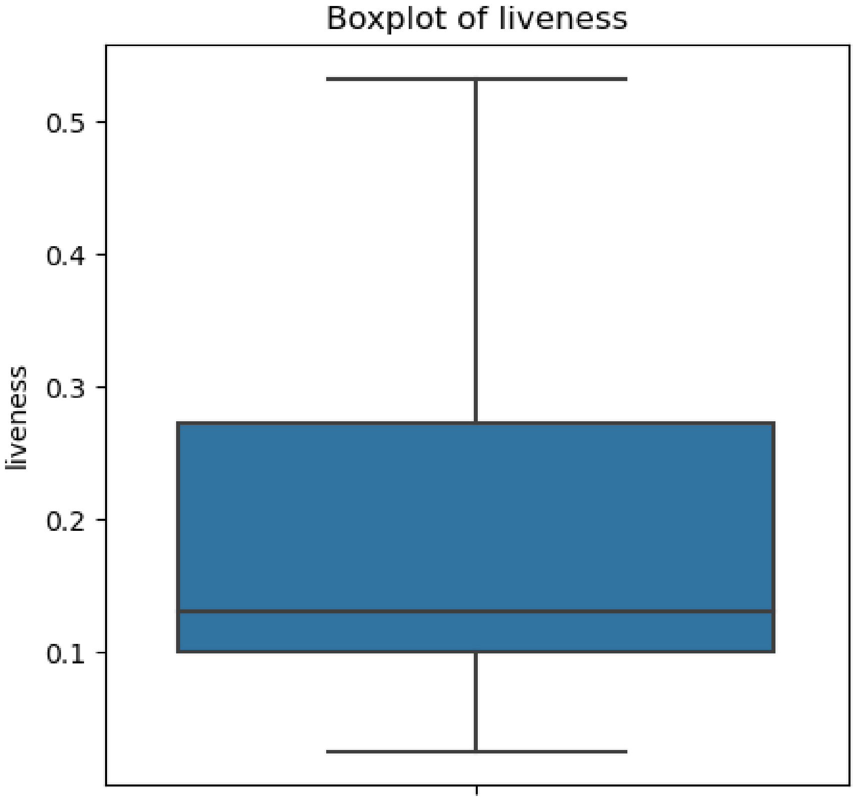


Let’s continue with the 7th column named Liveness.

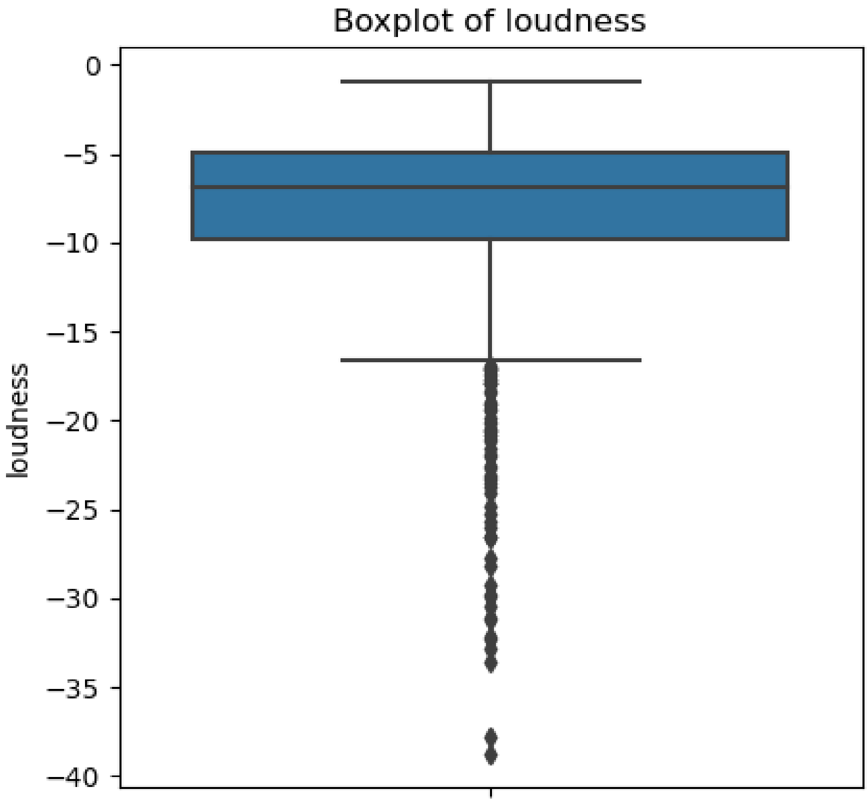


We found that there are outliers, so we will handle it using the IQR method.

After handling the outliers, the figure will change and be like this:

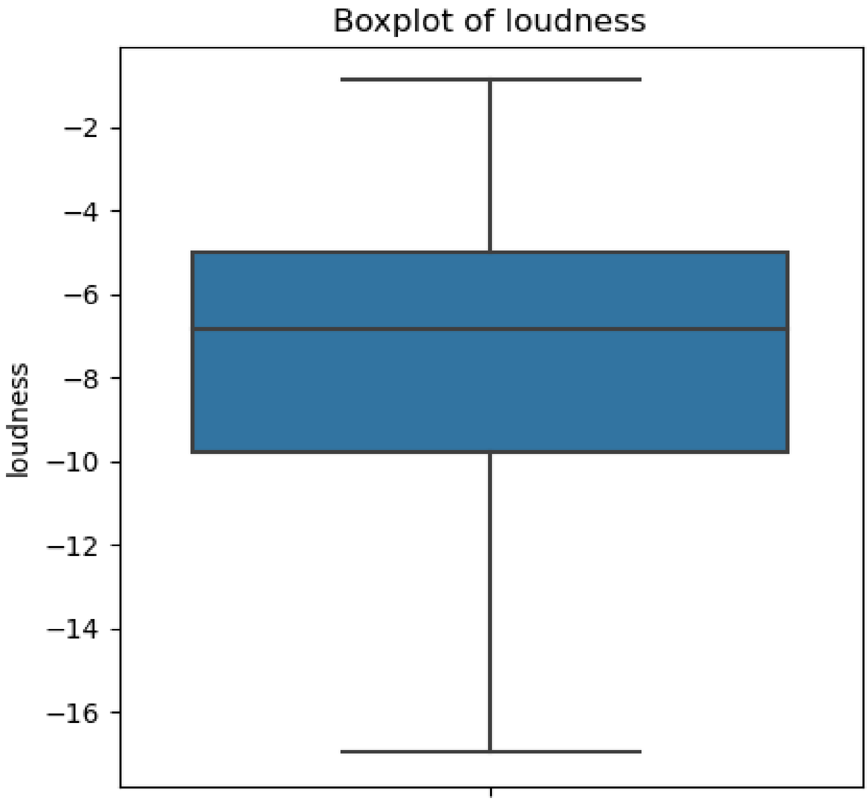


Let’s continue with the 8th column named Loudness.

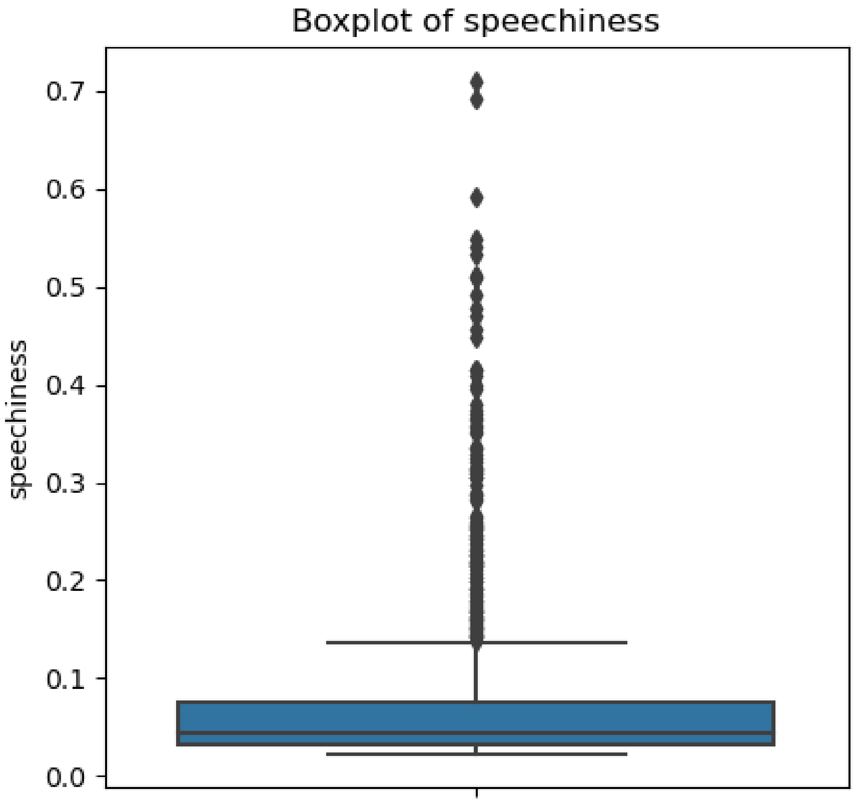


We found that there are outliers, so we will handle it using the IQR method.

After handling the outliers, the figure will change and be like this:

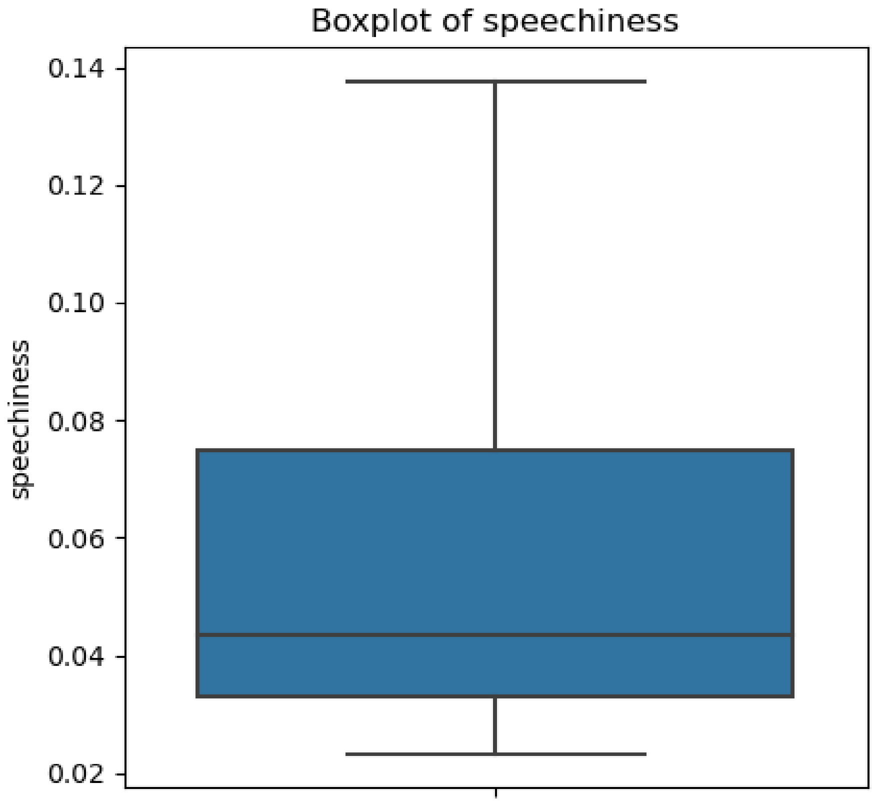


Let’s continue with the 9th column named speechiness.

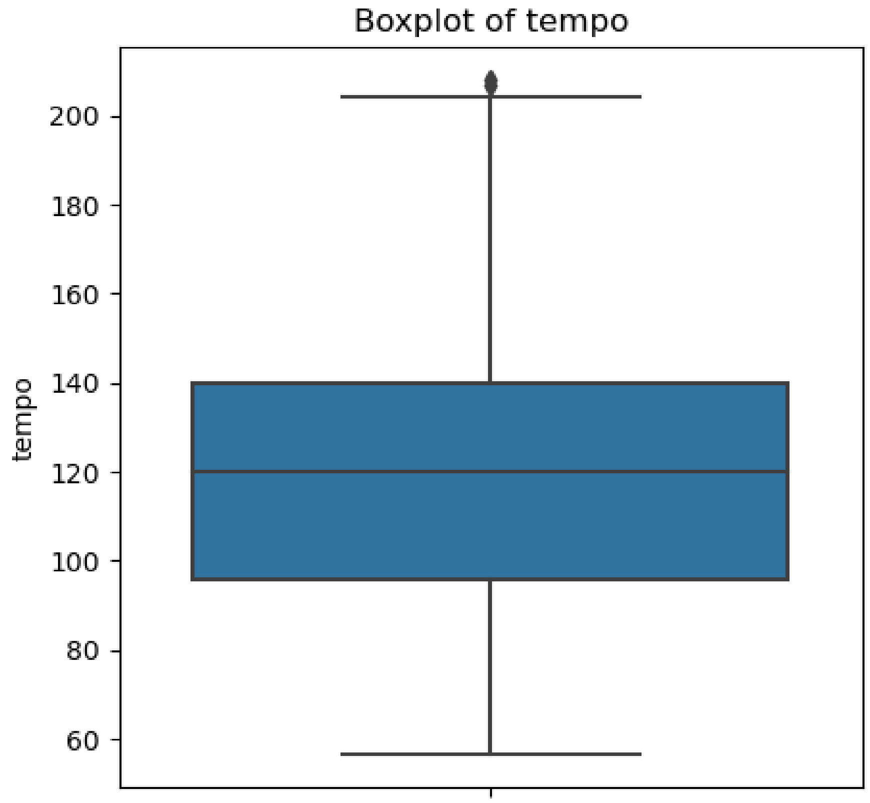


We found that there are outliers, so we will handle it using the IQR method.

After handling the outliers, the figure will change and be like this:

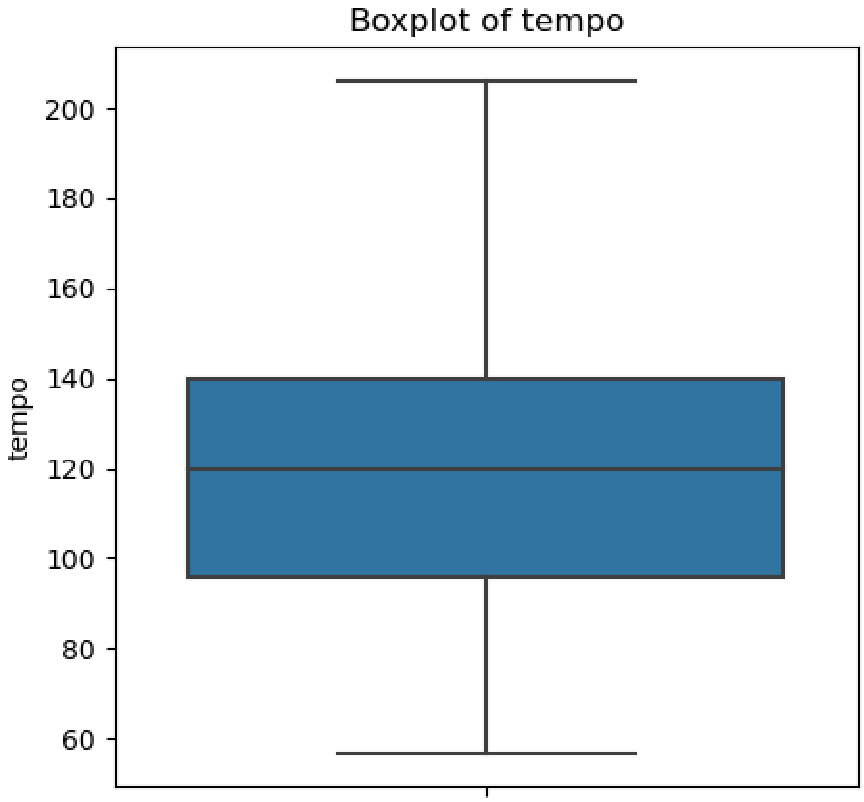


Let’s continue with the 10th column named Tempo.

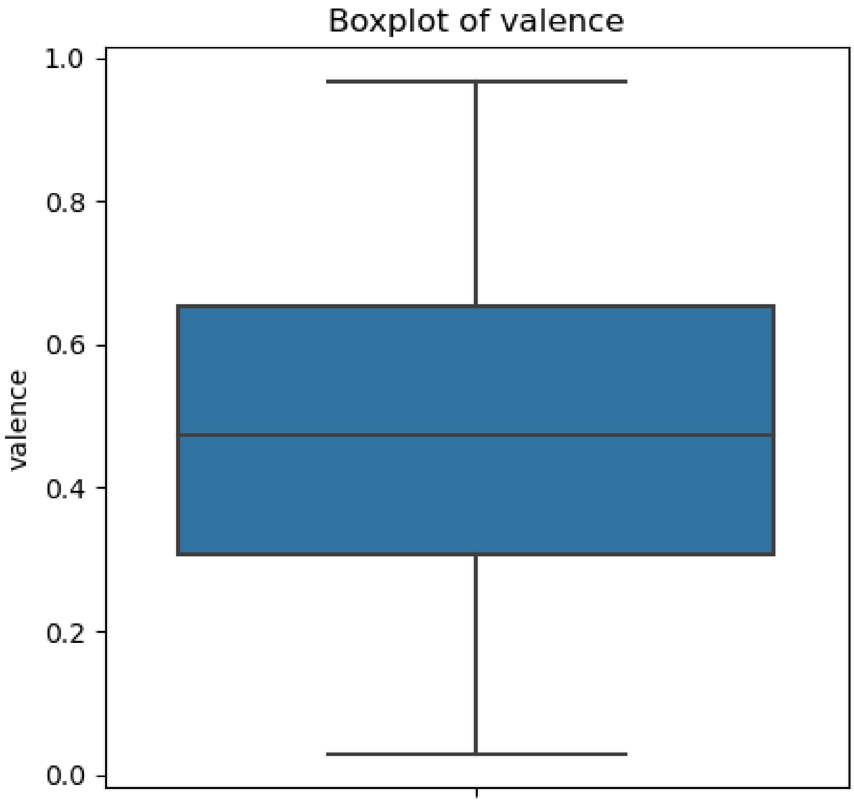


We found that there are outliers, so we will handle it using the IQR method.

After handling the outliers, the figure will change and be like this:



Let’s continue with the 11th column named Valence.



There are no outliers.

6. Preparing the data columns and target column.

7.Splitting the data to 60% for training, 20% for validation and 20% for testing.

8.Feature scaling

I used feature scaling in my data preprocessing stage to ensure that all features contribute equally to the machine learning model's performance. Feature scaling standardizes the range of independent variables or features of data, bringing them to a common scale without distorting differences in the ranges of values. This is crucial because support vector machines (SVM) are sensitive to the magnitudes of feature values. By scaling the features, the optimization process becomes more efficient, leading to faster convergence and improved model accuracy.

9. Hyper Tuning

Hyperparameter tuning is crucial because it allows the model to find the most optimal settings for its parameters, leading to improved performance and generalization on unseen data. Grid search systematically works through multiple combinations of parameter tunes, cross-validating as it goes to determine which tune gives the best performance.

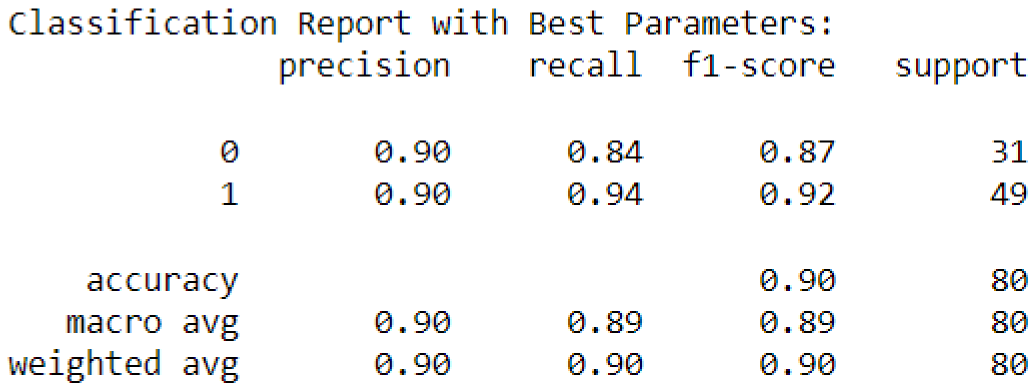
10. Train the models with outlier handling applied

10.1 Logistic regression model:

The validation accuracy is 0.88.

The test accuracy is 0.90.

This is the classification report:



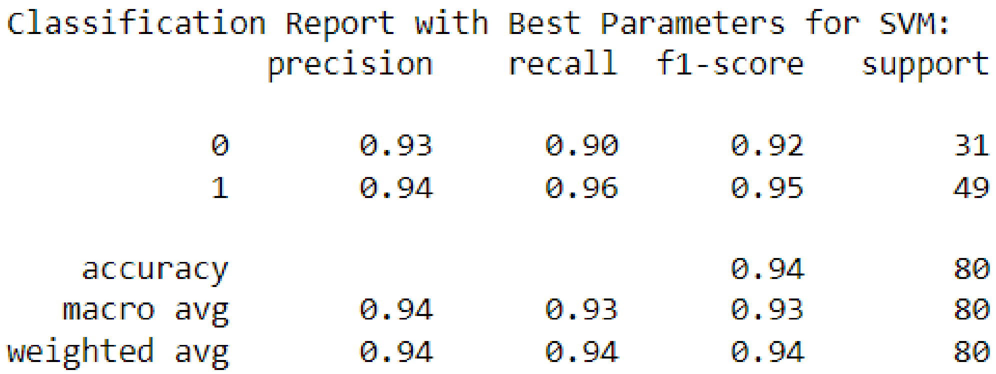
```
Classification Report with Best Parameters:
              precision    recall  f1-score   support

           0       0.90      0.84      0.87        31
           1       0.90      0.94      0.92        49

    accuracy                           0.90        80
   macro avg       0.90      0.89      0.89        80
weighted avg       0.90      0.90      0.90        80
```

10.2 Support Vector Machine model:

The validation accuracy is 0.93.

The test accuracy is 0.94.

This is the classification report:



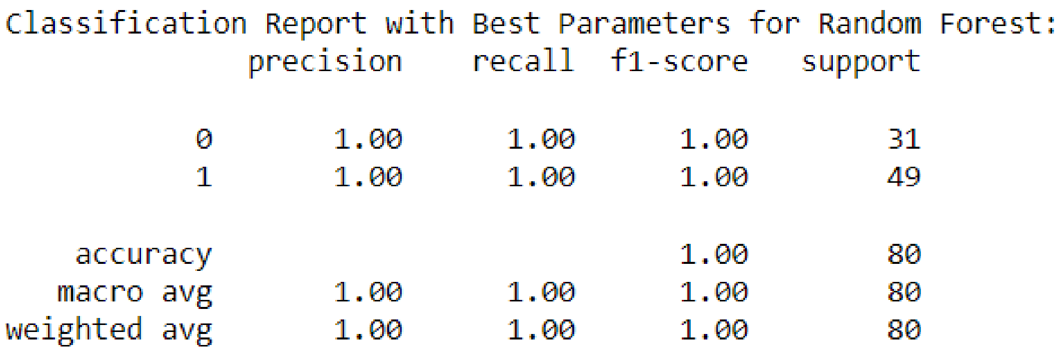
```
Classification Report with Best Parameters for SVM:
              precision    recall  f1-score   support

           0       0.93      0.90      0.92        31
           1       0.94      0.96      0.95        49

    accuracy                           0.94        80
   macro avg       0.94      0.93      0.93        80
weighted avg       0.94      0.94      0.94        80
```

10.3 Random Forest model:

The validation accuracy is 0.99.

The test accuracy is 1.0.

This is the classification report:



```
Classification Report with Best Parameters for Random Forest:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        31
           1       1.00      1.00      1.00        49

    accuracy                           1.00        80
   macro avg       1.00      1.00      1.00        80
weighted avg       1.00      1.00      1.00        80
```

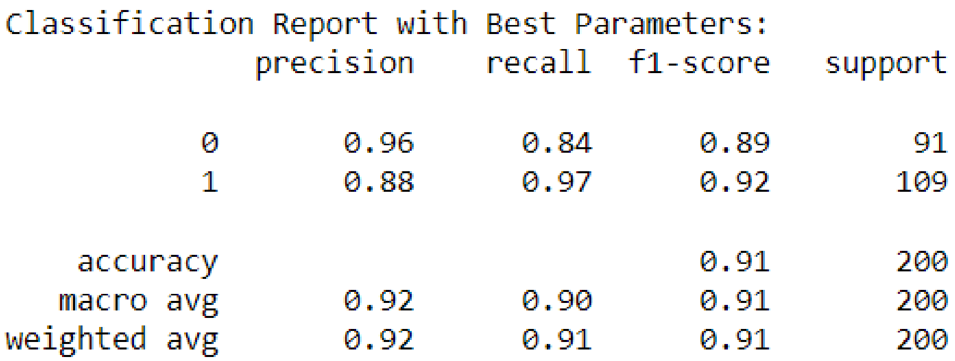
11. Train the models without outlier handling

11.1 Logistic regression model:

The validation accuracy is 0.91.

The test accuracy is 0.91.

This is the classification report:



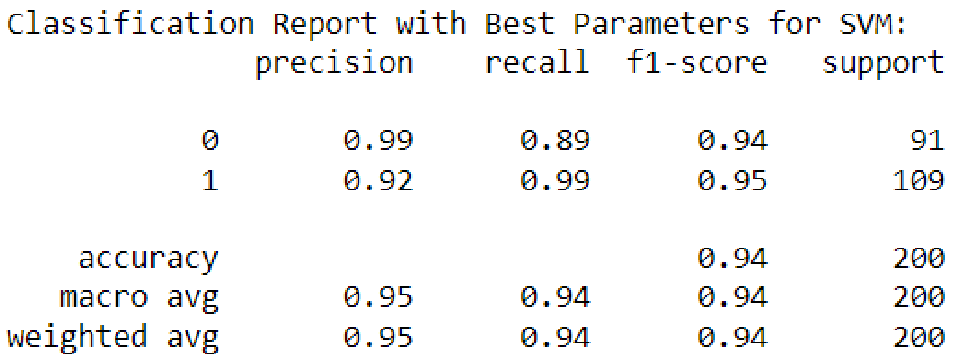
```
Classification Report with Best Parameters:
              precision    recall  f1-score   support

           0       0.96      0.84      0.89        91
           1       0.88      0.97      0.92       109

    accuracy                           0.91       200
   macro avg       0.92      0.90      0.91       200
weighted avg       0.92      0.91      0.91       200
```

11.2 Support Vector Machine model:

The validation accuracy is 0.93.

The test accuracy is 0.94.

This is the classification report:



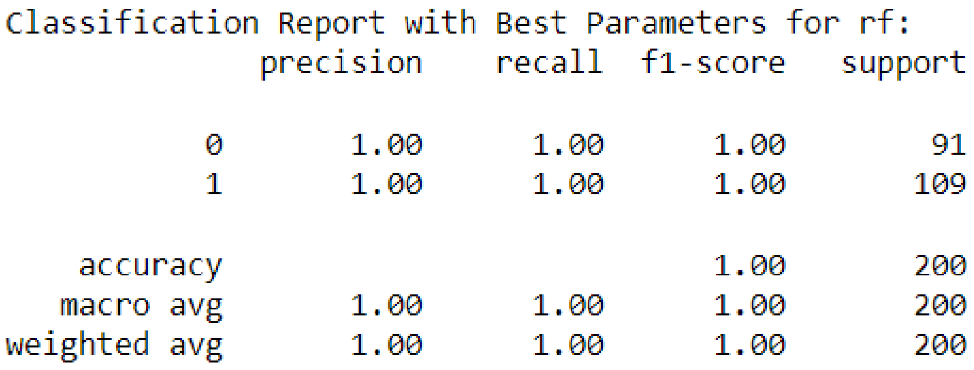
```
Classification Report with Best Parameters for SVM:
              precision    recall  f1-score   support

           0       0.99      0.89      0.94        91
           1       0.92      0.99      0.95       109

    accuracy                           0.94       200
   macro avg       0.95      0.94      0.94       200
weighted avg       0.95      0.94      0.94       200
```

11.3 Random Forest model:

The validation accuracy is 0.99.

The test accuracy is 1.00.

This is the classification report:



```
Classification Report with Best Parameters for rf:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        91
           1       1.00      1.00      1.00       109

    accuracy                           1.00       200
   macro avg       1.00      1.00      1.00       200
weighted avg       1.00      1.00      1.00       200
```