NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY (NUST)

CS-471 MACHINE LEARNING

**MUSIC AND MENTAL HEALTH**

PROJECT REPORT



**BESE-11 A**

HAMNA NAVEED (346470)

TANZEEL-UR-REHMAN (345510)

SYEDA SANA ZEHRA ZAIDI (339278)

Table of Contents

[Project Description 2](#_Toc133706181)

[Predicting Mental Severity Class Based on Musical Preferences 2](#_Toc133706182)

[Problem Statement 2](#_Toc133706183)

[Project Objective 2](#_Toc133706184)

[Data 2](#_Toc133706185)

[Data Exploration and Preprocessing 2](#_Toc133706186)

[Loading and Cleaning the Dataset 2](#_Toc133706187)

[Encoding 2](#_Toc133706188)

[Outliers 2](#_Toc133706189)

[Class Imbalance 5](#_Toc133706190)

[Distribution of Data 5](#_Toc133706191)

[Linearity of Data 6](#_Toc133706192)

[Feature Engineering and Selection 6](#_Toc133706193)

[Correlation of Features and Target Variable 6](#_Toc133706194)

[Dimensionality Reduction 7](#_Toc133706195)

[Model Selection and Training 7](#_Toc133706196)

[Comparison of Various Models 7](#_Toc133706197)

[Overfitting 8](#_Toc133706198)

[Model Selection 9](#_Toc133706199)

[Model Improvement 10](#_Toc133706200)

[Hyper-parameter Tuning 10](#_Toc133706201)

[Model Evaluation 11](#_Toc133706202)

[Model Deployment 11](#_Toc133706203)

[Model Training on Entire Dataset 11](#_Toc133706204)

## Project Description

### **Predicting Mental Severity Class Based on Musical Preferences**

### Problem Statement

Mental health disorders are a growing concern globally, affecting individuals' welfare, social interaction, and work performance. Researchers have started exploring the link between music and mental health, indicating that music may offer benefits to those with mental health issues.

### Project Objective

To develop a machine learning model that accurately predicts a person's level of mental health severity class based on their musical preferences and current mental health score

### Data

The dataset consists of 206 rows of how music preferences affects the mental health severity of various participants. Each row represents a record and each column represents a feature such as the Participant\_ID, Song\_name, Artist, Spotify\_ID, Loudness, Valence, Danceability, Acousticness, Instrumental, Audio\_class, Lyrics, Sentiment\_class, Audio + Lyrics analysis and Total\_mental\_health. The target variable is Mental\_health\_severity\_class.

## Data Exploration and Preprocessing

### Loading and Cleaning the Dataset

From human analysis, the columns containing irrelevant information were dropped. These columns included Participant\_ID, Song\_name, Artist, Spotify\_ID, and Lyrics. Four duplicate records were removed from the dataset. For each feature, we checked if the values lied within the standard range. It was discovered that mental health score can only range between 0 and 27 according to Patient Health Questionnaire (PHQ-9). Three records were detected to be out of range and were, therefore, removed.

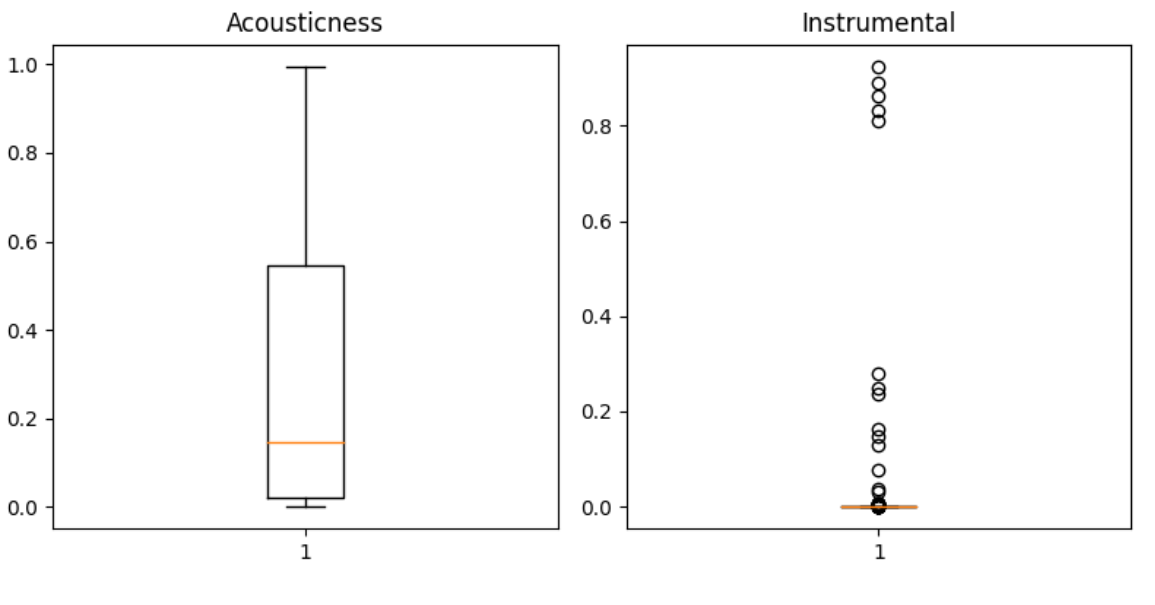
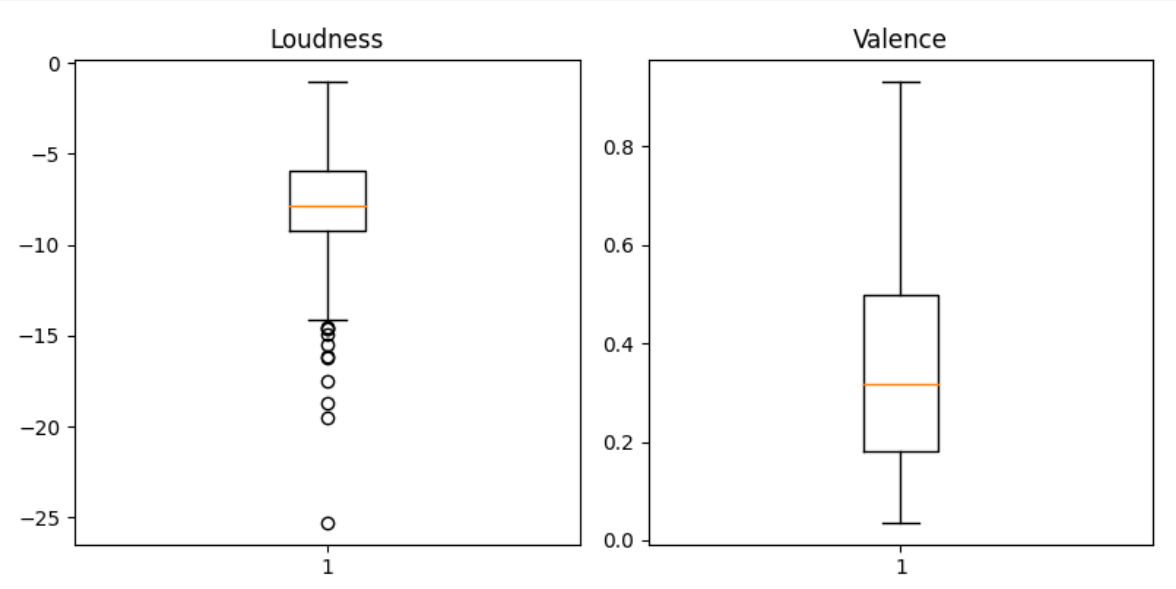
### Encoding

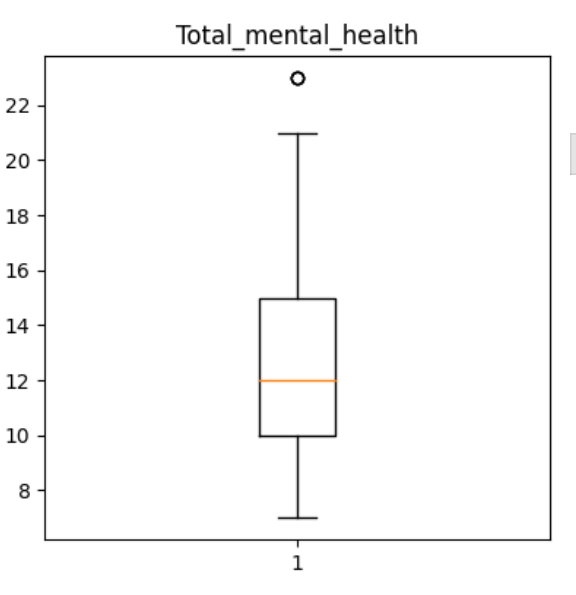
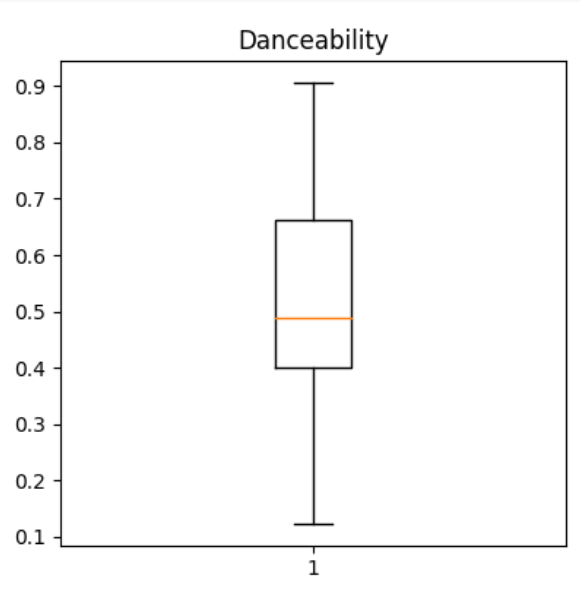
Machine learning algorithms require numerical inputs, and therefore categorical data must be converted to numerical data before it can be used for modeling. For this purpose, Label Encoding was employed. To preserve the ordinal relationship in categories with ordinal variables, we pre-defined the labels. Label encoding was preferred over One-Hot Encoding to prevent the curse of dimensionality which can be computationally expensive.

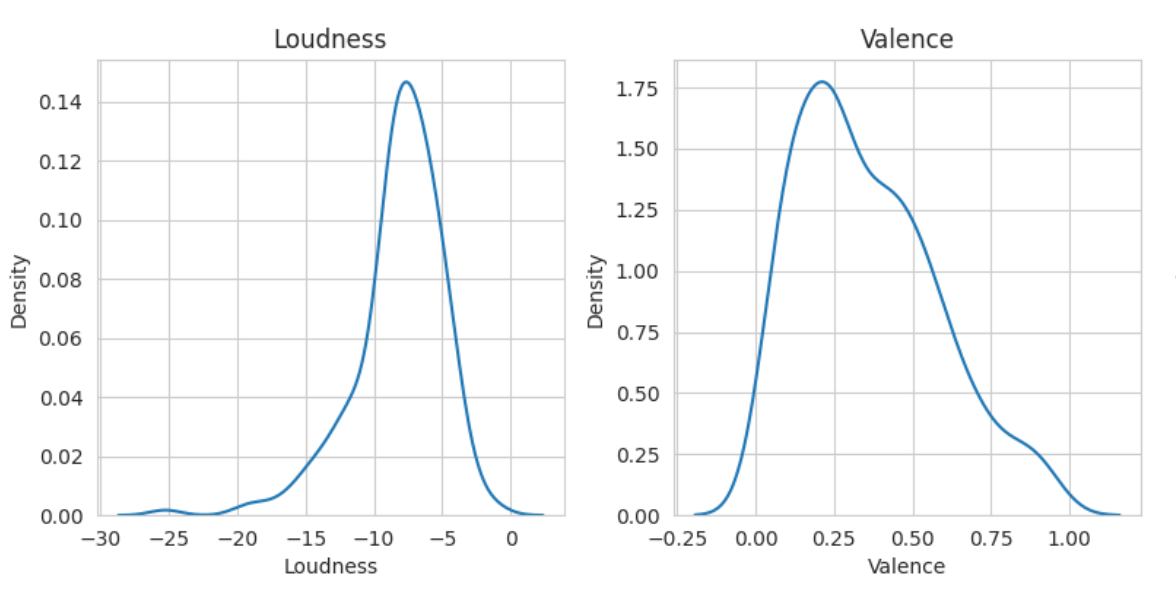
### Outliers

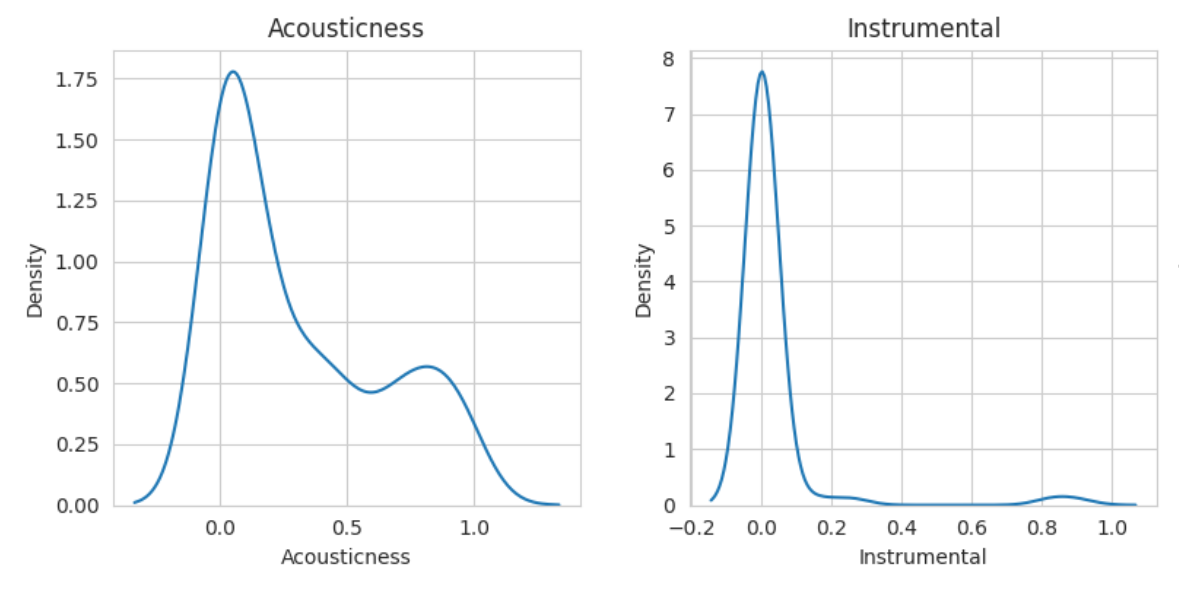
Outliers are the data points that significantly deviate from the expected or normal behavior of a dataset, and may negatively impact the accuracy and effectiveness of a model. When outliers are present in a model, they can lead to inaccurate and unreliable predictions, causing the model to either over-fit or under-fit the data. To tackle these outliers, they were first discovered using both the graphical and statistical methods i.e. boxplots, density graphs and the Z-score. The number of outliers for each feature are tabulated below.

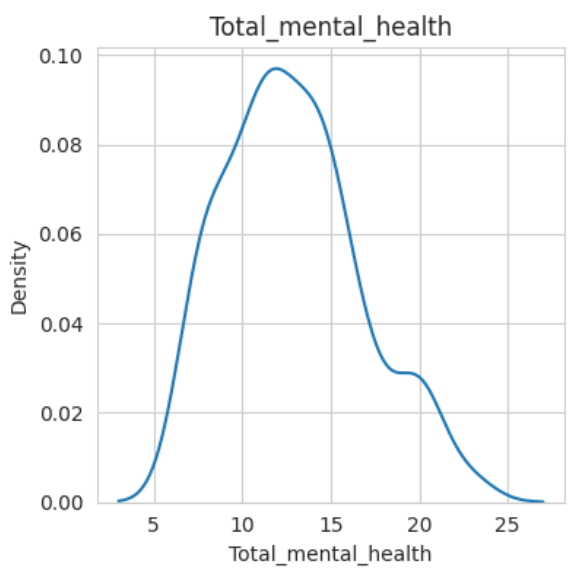
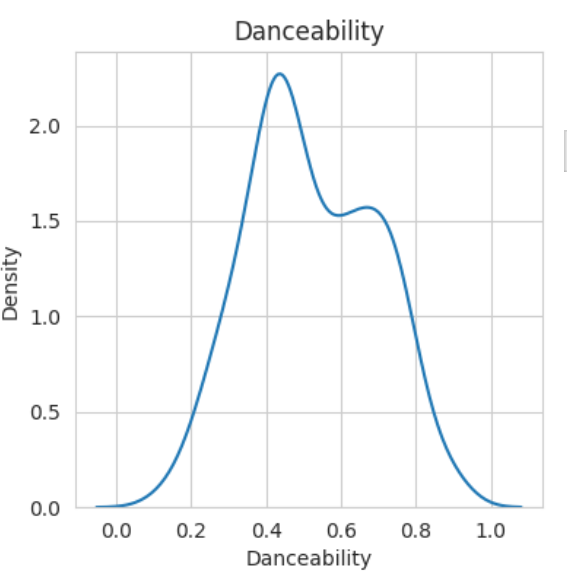
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Features | Loudness | Valence | Danceability | Acousticness | Instrumental | Total\_mental\_health |
| Outliers | 3 | 0 | 0 | 0 | 5 | 0 |





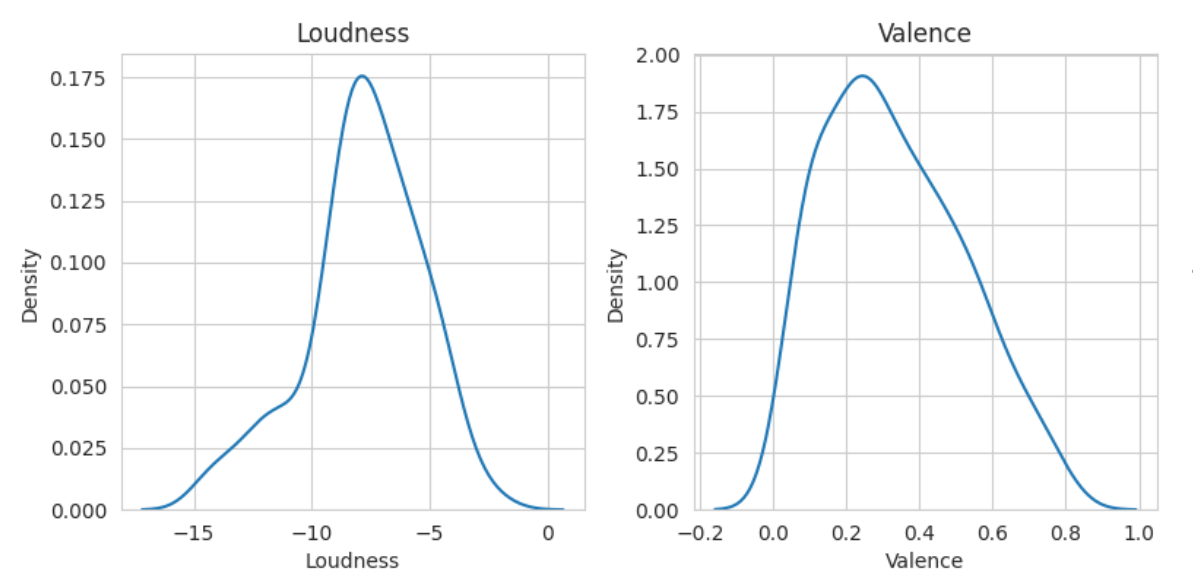


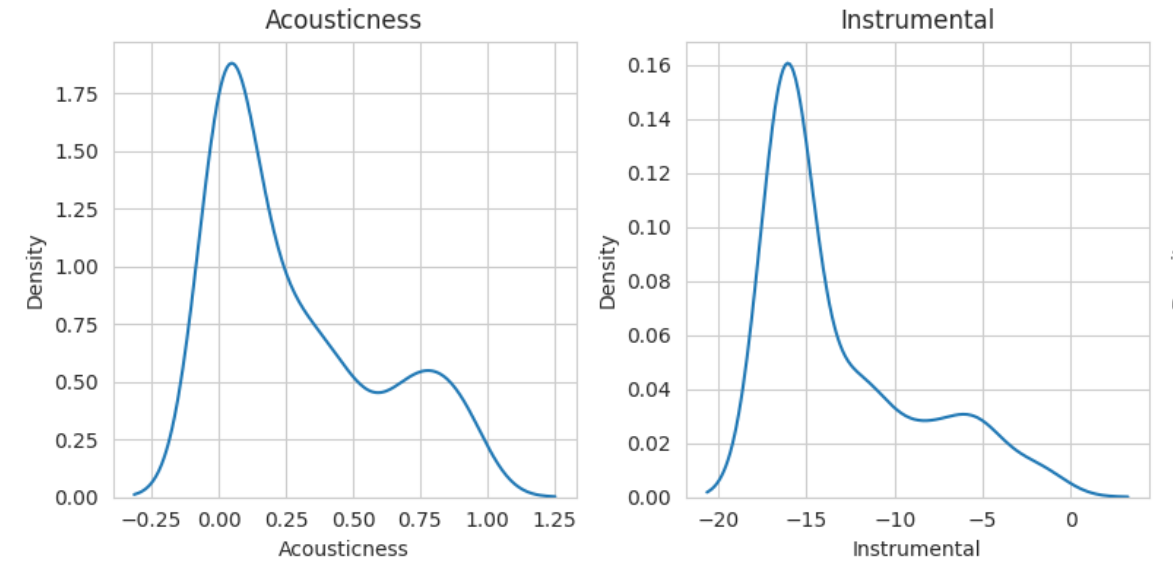


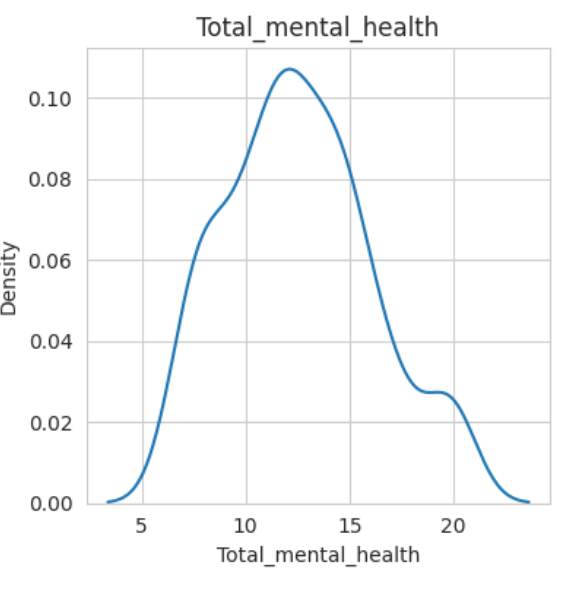
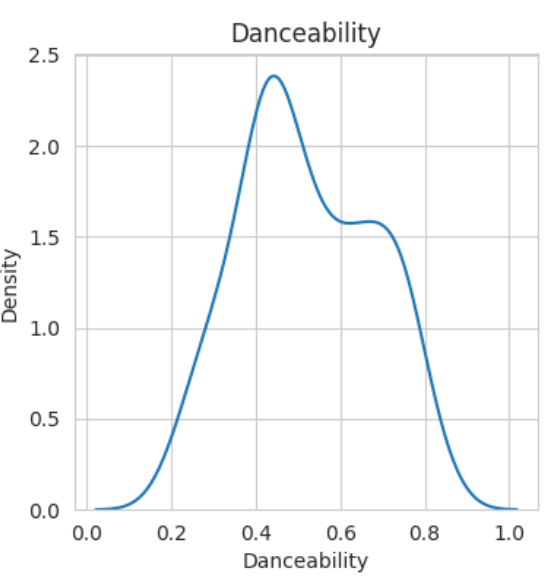


After detection, the outliers were imputed by replacing them with the mean of non-outlier values.

Even after imputing, the instrumental feature still showed traces of outliers. Since the column contains negative and zero values, logarithmic transformation cannot be directly applied. As a result, we first replaced those values with very small positive integer and then applied logarithmic adjustment. The density plots below depict the features after imputation.







### Class Imbalance

In our dataset, different classes of a target variable were not evenly distributed. This class imbalance can lead to biased models that perform poorly on underrepresented classes, as they are not given enough attention during training. The following table demonstrates the class imbalance of our dataset.

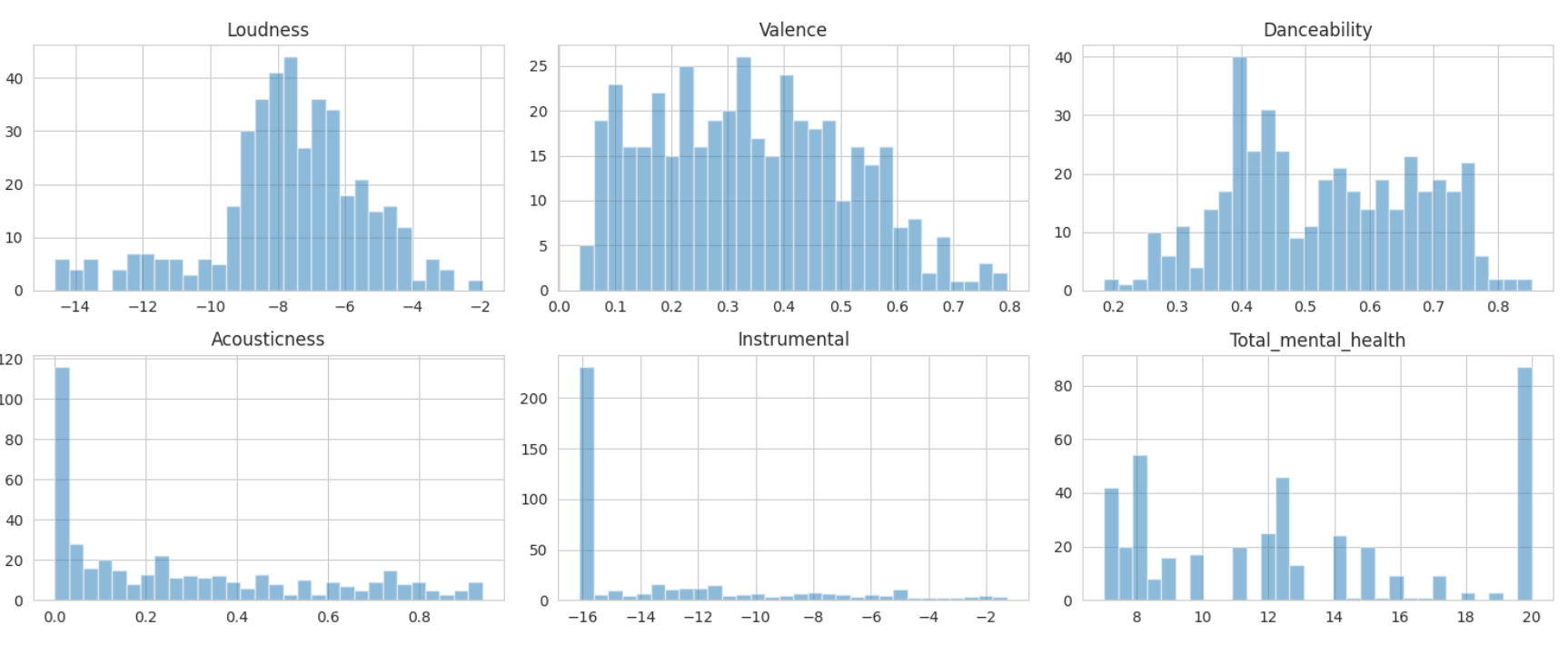
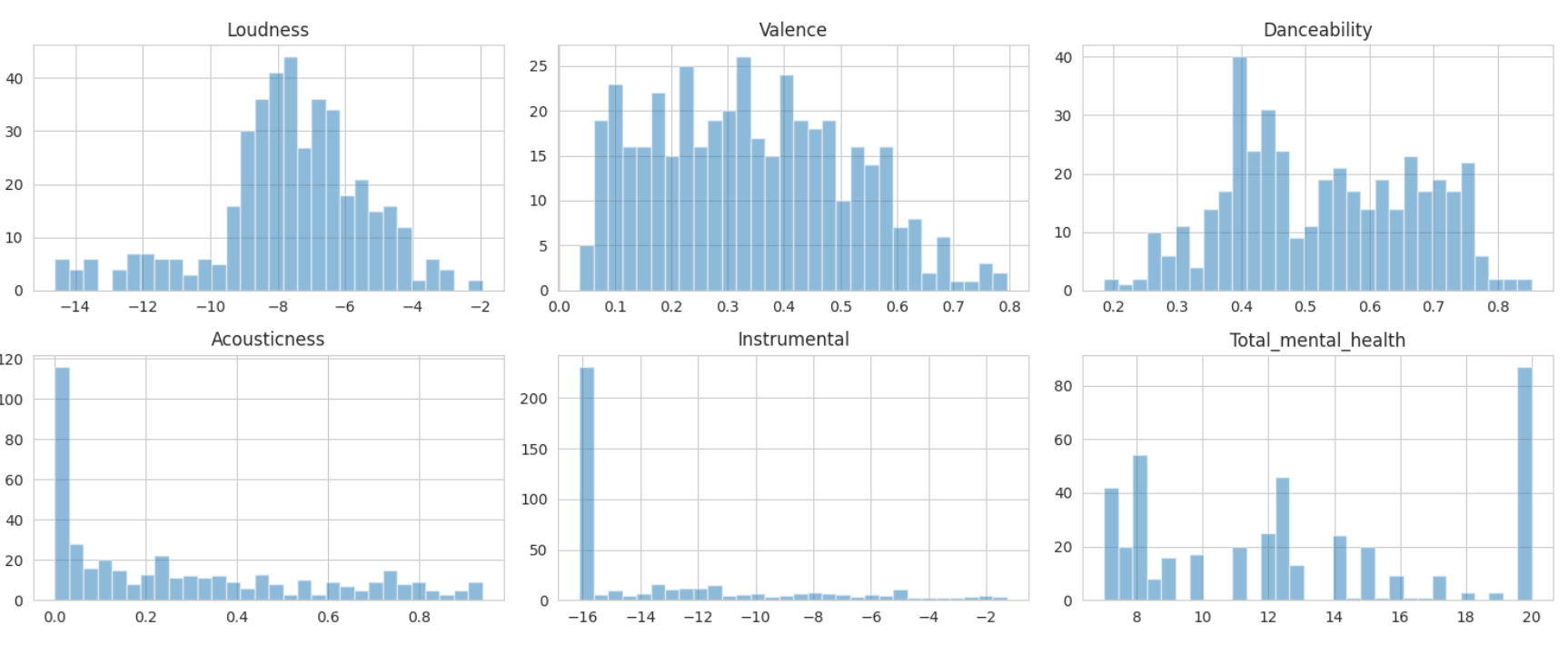
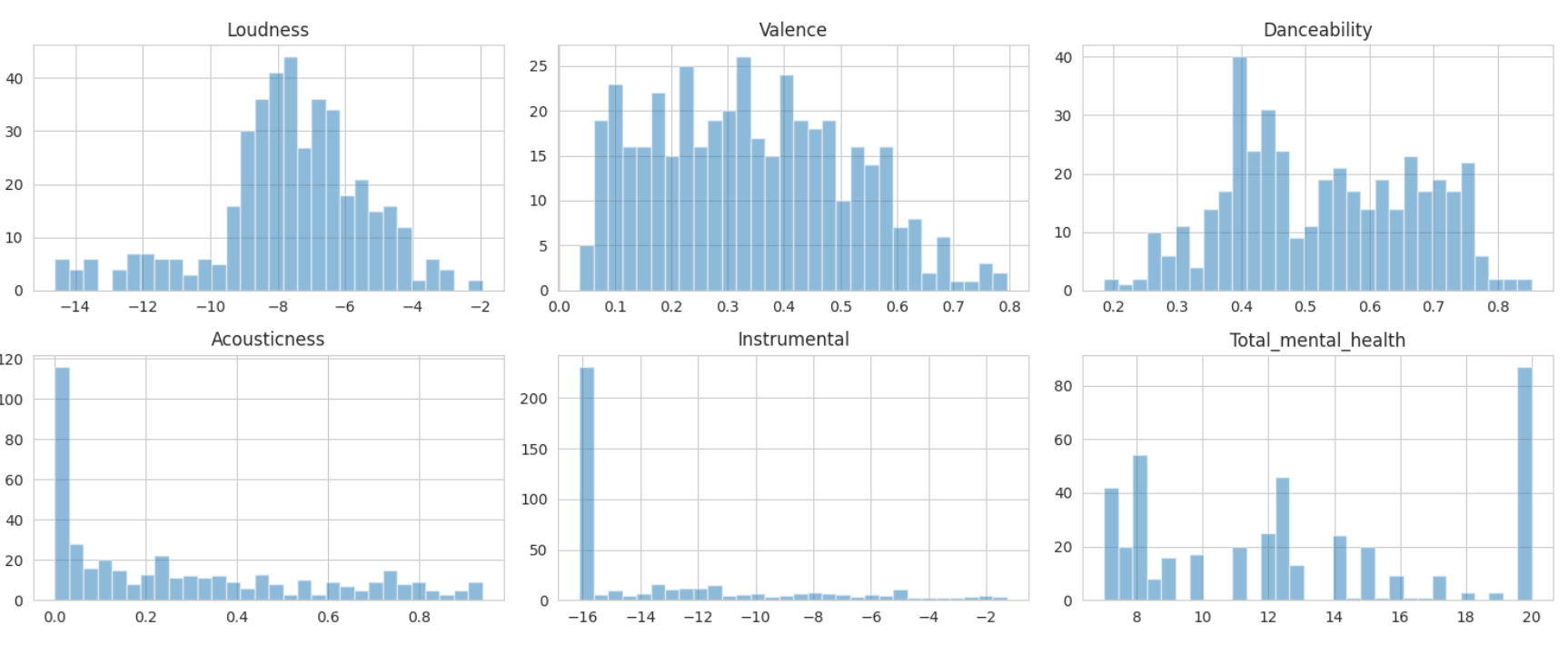
|  |  |  |  |
| --- | --- | --- | --- |
| Target | Low | Medium | High |
| Samples | 41 | 140 | 18 |

The datasets having class imbalance ration < 1 are considered to be extremely imbalanced. The ratio for the ‘Music and Mental Health’ was computed to be 0.12.

For addressing this imbalance, SMOTE algorithm (Synthetic Minority Over-Sampling Technique) is utilized. By interpolating new samples along the line segments between existing minority class samples, the SMOTE algorithm creates synthetic minority class samples.

### Distribution of Data

Jarque Bera test, based on skewness and kurtosis, suggests whether the data is normally distributed. The method portrayed non-normality in our dataset. The visual distribution of the data is shown using the histograms.



### Linearity of Data

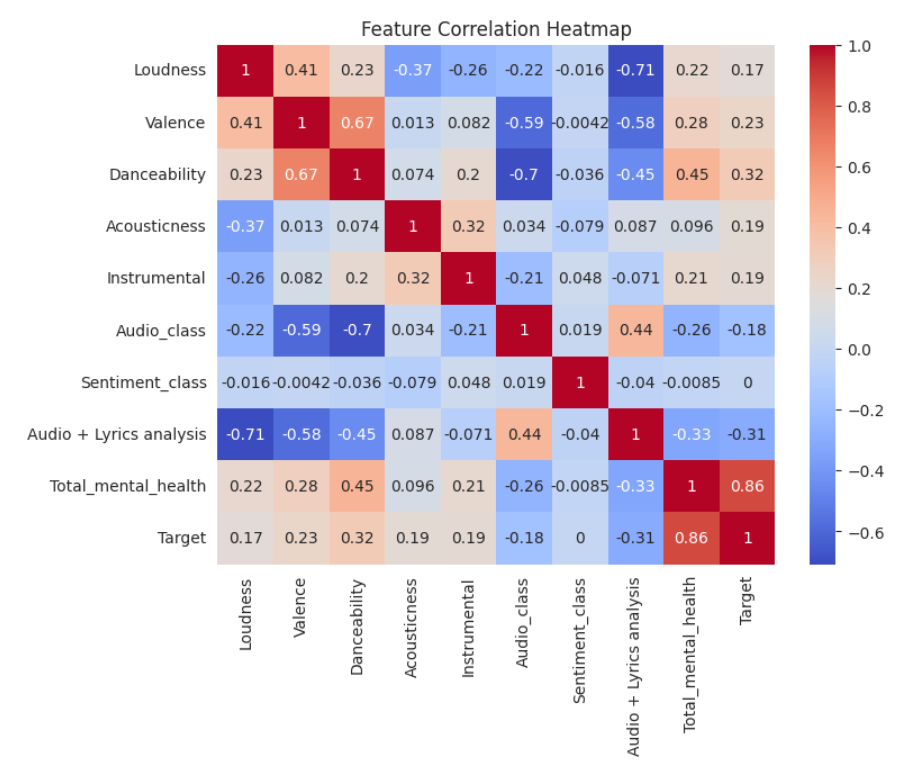
The Durbin-Watson test is a statistical test that can indicate if the dataset is linear. The test showed that our dataset is non-linear.

## Feature Engineering and Selection

### Correlation of Features and Target Variable

Finding correlations can help us identify which features have a stronger or weaker relationship with the target variable. This information can be used to determine which features to include in our model, and how much weight to give each feature when making predictions.

As ‘Music and Mental Health’ is not normal, we used the ‘Spearman’ method of the correlation matrix to understand how the features may be influencing the target variable. The correlation matrix is represented by using a ‘Heatmap’.



### Dimensionality Reduction

The Heatmap demonstrates that there is no association between the ‘Sentiment\_class’ and the target variable. As a result, the ‘Sentiment\_class’ feature can be eliminated without significantly reducing predictive power.

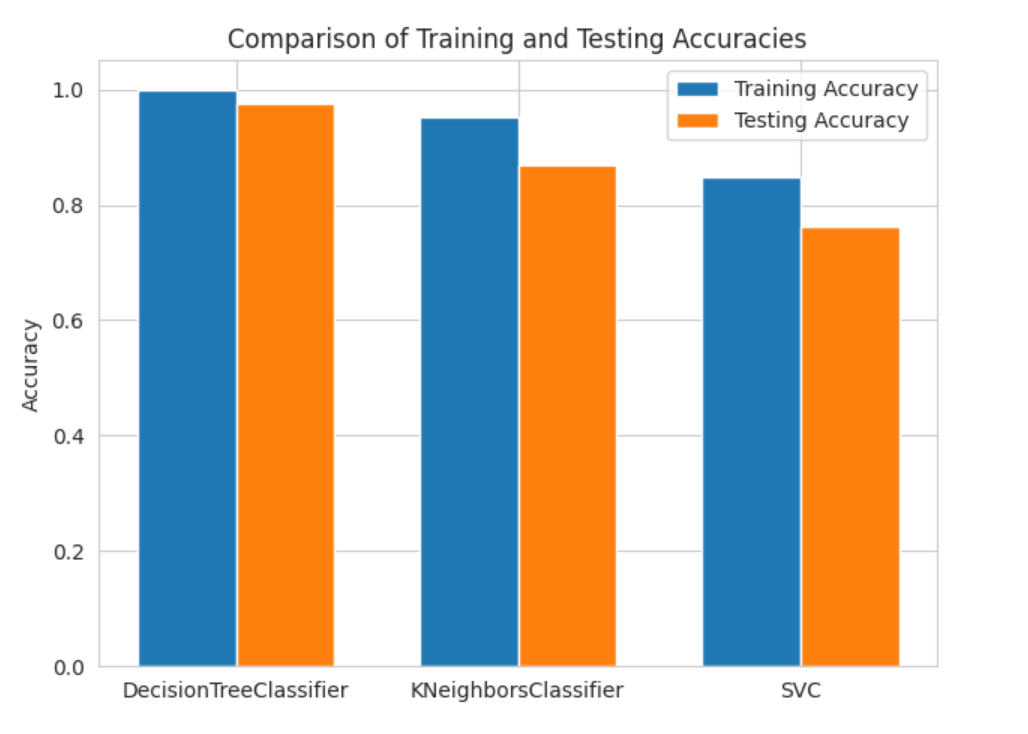
## Model Selection and Training

### Comparison of Various Models

Our dataset is small, non-linear, and non-normally distributed with multi-class target variable, as the analysis above has demonstrated. These traits compelled us to take into account the following three models that do well on a dataset like this one:

1. DecisionTreeClassifier
2. KNeighborsClassifier
3. SVC

Training accuracy is a performance statistic that gauges the accuracy of a machine learning model on the data that was used to train it. Testing accuracy, on the other hand, is a crucial indicator for evaluating how effectively a model generalizes to new, unknown data and is used to assess overall performance of the model. Both of these accuracies were calculated after training each of the aforementioned models on the training set of the 'Music and Health' dataset. The outcomes were recorded and plotted for comparison.



Each model is represented by two bars in the bar chart, one for training accuracy and one for testing accuracy. The graph compares the training and testing accuracies of the three machine learning models in order to determine which model performs best on the given dataset.

### Overfitting

Overfitting can occur when the model is too complex and starts fitting the noise in the training data, resulting in poor generalization to new and unseen data. After multiple repetitions, it was discovered that SVC and KNN regularly displayed overfitting. This was determined by the fact that the difference between training and testing accuracies was more than 0.4%.

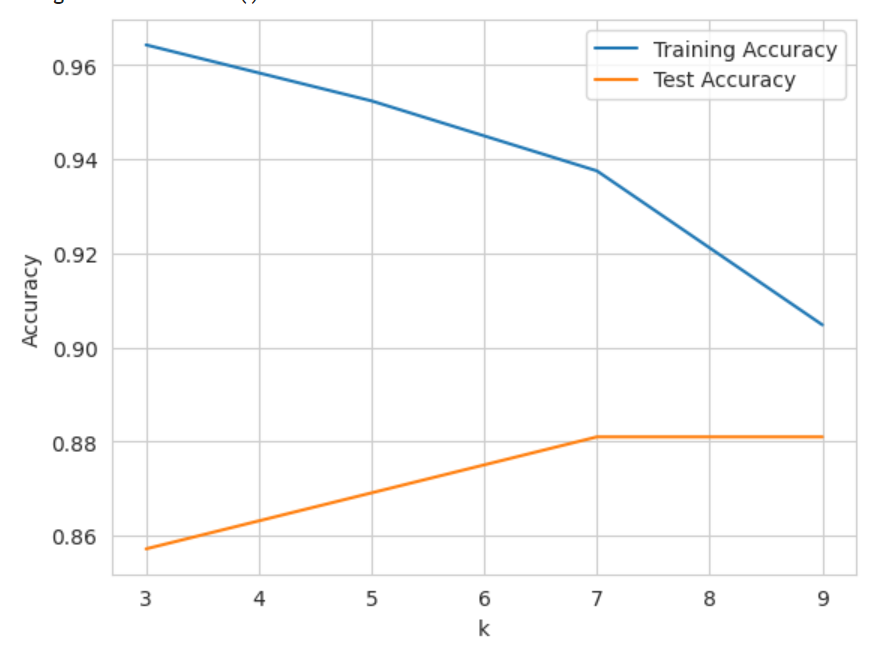
##### Improved SVC

Early stopping is a technique used to prevent overfitting by monitoring the performance of the model on a validation set during training and stopping the training process when the performance stops improving.

To implement early stopping for SVC, we set a maximum number of iterations for training, a tolerance for convergence, and a maximum number of iterations with no improvement in validation score. The SVC was then trained for a maximum number of iterations and its validation score was monitored. To prevent overfitting, the training was halted when the validation score did not improve for a certain number of iterations.

##### Improved KNN

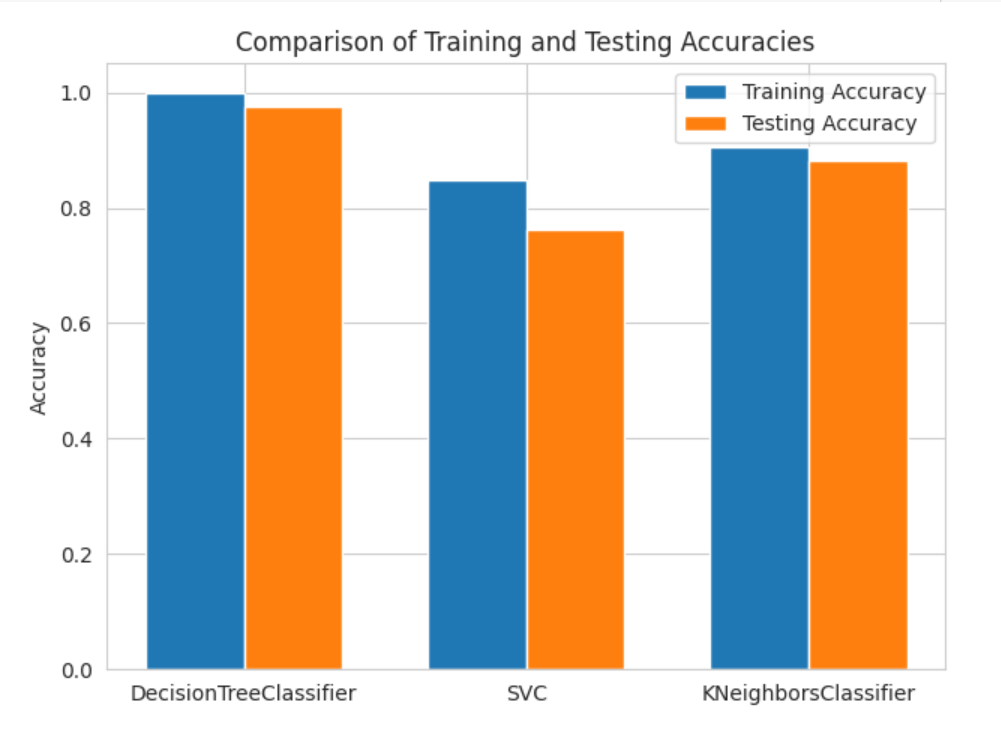
The KNN model was hyper-parameter tuned by testing different values of k i.e. number of nearest neighbors. For each value of k, the training and testing accuracy of the model was computed and their difference was recorded and plotted using line graph. The model with minimum difference between the two accuracies was selected.



Modifying the model helped to resolve overfitting by tuning the hyper-parameter k, which controls the complexity of the model.

### Model Selection

After successfully coping overfitting, we compared the performances of the three improved models by plotting a bar graph of train and test accuracies of each of them.

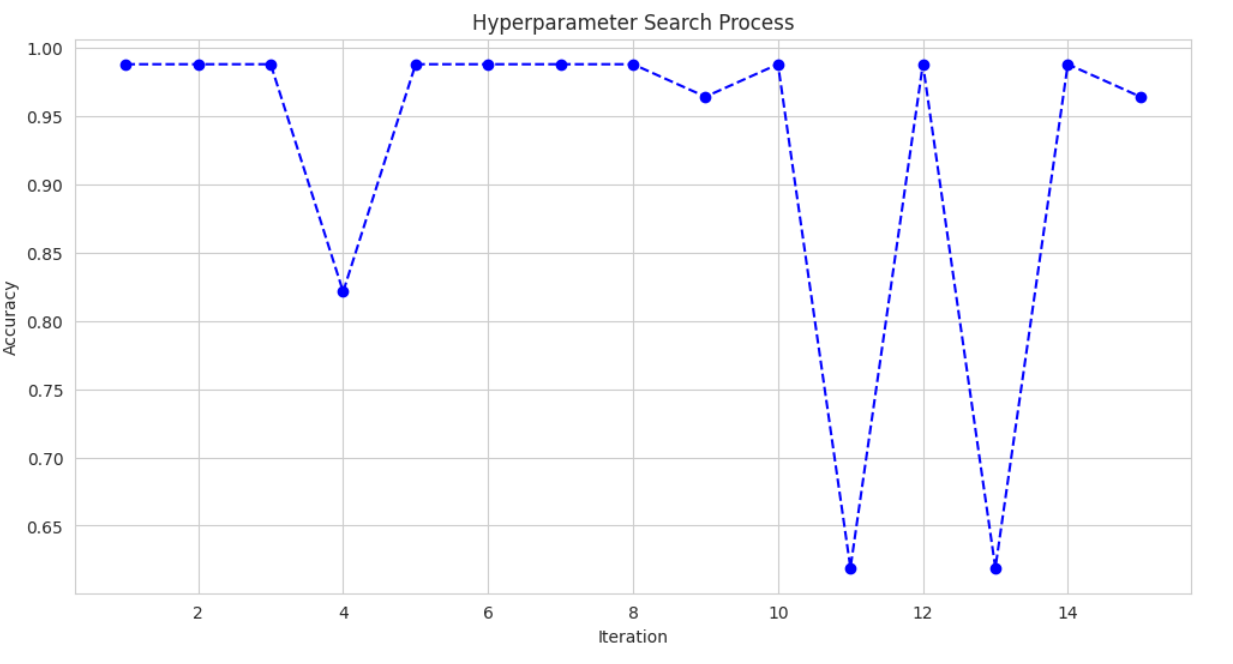


Taking into account the Low Variance-Low Bias tradeoff depicted by DecisionTree and the accuracies given by the models, we conclude that **DecisionTreeClassifier** is the best fit for our scenario.

## Model Improvement

### Hyper-parameter Tuning

After dealing with the outliers, we used Bayesian Optimization to hyper-parameter tune the Decision Tree Classifier. Bayesian Optimization used cross-validation to find the optimal combination of hyper-parameters over a defined range grid. The best model was chosen after a graph of test accuracy on the y-axis was plotted against multiple hyper-parameter optimized Decision Tree models.



### Model Evaluation

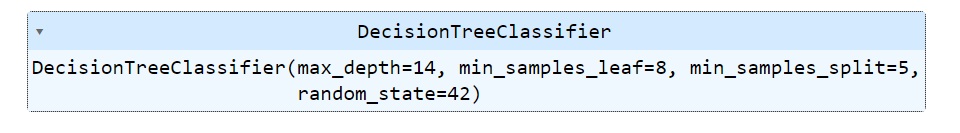
The best Decision Tree model was evaluated on the test set by computing several evaluation metrics. The results are tabulated in the table below.

|  |  |  |  |
| --- | --- | --- | --- |
| Accuracy | Precision | Recall | F1-Score |
| 0.98809 | 0.98845 | 0.98809 | 0.98807 |

## Model Deployment

### Model Training on Entire Dataset

Our final hyper-parameter optimized Decision Tree was trained on the entire dataset. The trained model was then saved as .joblib file.



### Deployment

The XGBoost model was integrated with the web application using Flask, a popular Python framework. After running and testing the model on localhost, it was deployed to a production server to make it accessible to end-users. This was done using ngrok. Ngrok allowed us to expose the Flask development server running on the local computer to the internet, making the web application accessible to users from anywhere in the world.