

# CS4622

## Machine Learning

### Lab 1 - Feature Engineering Report Submission

Name : K. G. Akila Induranga  
Index No: 190239A  
Submission Date : 25/08/2023

# Introduction

The primary objective of this lab exercise is to get familiar with the techniques related to feature selection and feature engineering in order to decrease the number of features within a provided dataset, while preserving the dataset's accuracy at a considerable range. The dataset includes two CSV files: "train.csv" and "valid.csv." Both of these files consist of 256 features (columns for x-variables) alongside 4 target labels (columns for y-variables). Additionally, there exists a "test.csv" file designed for evaluating the accuracy of the reduced-featured model.

In the initial dataset, the first 256 columns contain speaker embedding vectors linked to each audio file. These vectors were generated through the "wav2vec-base model". The last 4 columns include labels associated with the following findings of the speaker:

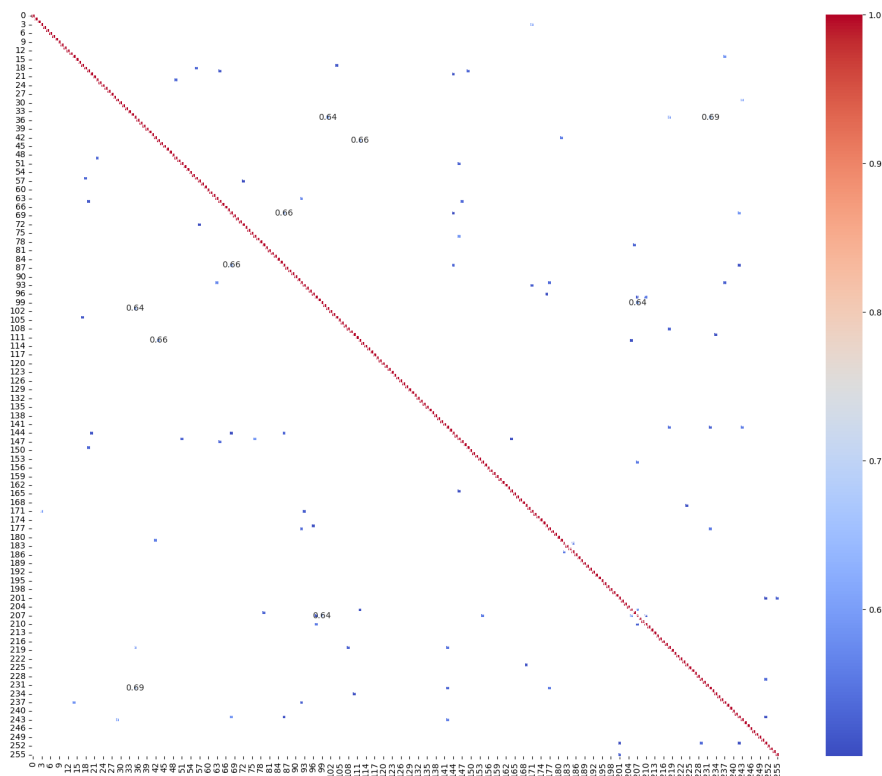
- Label 1: Speaker ID
- Label 2: Speaker's age
- Label 3: Speaker's gender
- Label 4: Speaker's accent

The process of feature reduction (through the processes of feature selection, and other feature engineering techniques) is executed independently for each label. Although the general procedure remains consistent across the labels, specific adjustments are made to accommodate variations in behaviours associated with different labels. To demonstrate the feature engineering process for each label, four separate Jupyter Notebooks were used (linked in the sources section).

Within each label category, after the feature reduction is executed, the accuracy of the outcome is assessed using the KNN (K-Nearest-Neighbors) model. Although many different models can be used to evaluate each label independently, only the kNN model is used in this exercise for the consistency of the process.

## Correlation of Features

First the correlation between the features was observed as a general rule of thumb in feature engineering. As there are 256 features, the correlation matrix came pretty compact to get any idea of the features. Therefore, it is decided to remove the features that have correlation less than 0.5 between them, as they can be considered as more independent than the others. Here is the heatmap of the correlation matrix, where the loosely correlated features are opted out.



As the figure states, there are a considerable number of features with correlation greater than 0.5. They can be eliminated using the feature engineering techniques. The following techniques were tried on the data set.

## Feature Scaling

Scaling data in features is a critical step that ensures fair and accurate analysis. It involves adjusting the values of features in a dataset to make them comparable.

In data, features often have different scales, which can lead to biased results. Scaling addresses this by normalizing the values of features, ensuring they are treated equally in analysis. Standard scaling, a method where features are transformed to have a mean value of 0 and a consistent standard deviation equal to 1, is especially useful.

Scaling is particularly important before applying techniques like Principal Component Analysis (PCA). By scaling data before PCA, we ensure that the analysis isn't skewed by features with different scales. This helps PCA accurately represent the data's patterns.

For the exercises in this lab, StandardScaler from sklearn.preprocessing is used. Although there are some better scaling tools such as RobustScaler, which would make the scaling more robust to the outliers in data, StandardScaler is a simple and intuitive tool to use at first.

## Feature Selection

Feature selection is the process of selecting a subset of features from the given dataset in order to reduce the number of features that are going to be provided to the model. There are different types of feature selection tools available and Sci-Kit Learn library provides a number of feature selection tools out of the box. SelectKBest and SelectPercentile from sklearn.feature\_selection were used in this lab.

### Select K-Best Features

SelectKBest method selects features according to the k highest scores, where k is a hyperparameter we provide into the method. Since there are 256 features, selecting 100 out of them was tried as the first exercise of the feature engineering process. Those selected features were then provided into the model and following are the accuracy scores obtained.

Label 1 - KNN Accuracy: 0.9826666666666667  
Label 2 - KNN Accuracy: 0.9877717391304348  
Label 3 - KNN Accuracy: 1.0  
Label 4 - KNN Accuracy: 0.992

### Select Percentile Features

The SelectPercentile method selects features according to a percentile of the highest scores, where it needs to give the percentile as the hyper parameter. As a feature engineering technique separate from the SelectKBest method, percentile was set as 40 in this lab and followings are the accuracy scores obtained.

Label 1 - KNN Accuracy: 0.9813333333333333  
Label 2 - KNN Accuracy: 0.9891304347826086  
Label 3 - KNN Accuracy: 1.0  
Label 4 - KNN Accuracy: 0.9946666666666667

## Principal Component Analysis

Principal Component Analysis (PCA) is a technique used to simplify the complexity of a dataset while preserving its crucial patterns. It begins by making the data comparable through standardization. Then, it calculates a matrix that shows how different features change together. This matrix is broken down into two important things: eigenvectors and eigenvalues. The eigenvectors point out the directions with the most variation, and the

eigenvalues tell us how much variation there is in those directions. By picking the most important eigenvectors based on their eigenvalues, we capture the significant patterns in the data. These selected eigenvectors are used to project the original data onto them, creating a new representation with fewer dimensions. PCA is immensely valuable in feature engineering because it helps us get rid of less important features, tackles the problem of having too many dimensions, makes models work faster, and aids in understanding how data patterns appear.

In this lab PCA from sklearn.decomposition was used to do the principal component analysis and to obtain the newly generated set of features. After providing those labels into our model, the following are the accuracy scores obtained.

Label 1 - KNN Accuracy: 0.9866666666666667

Label 2 - KNN Accuracy: 0.9891304347826086

Label 3 - KNN Accuracy: 1.0

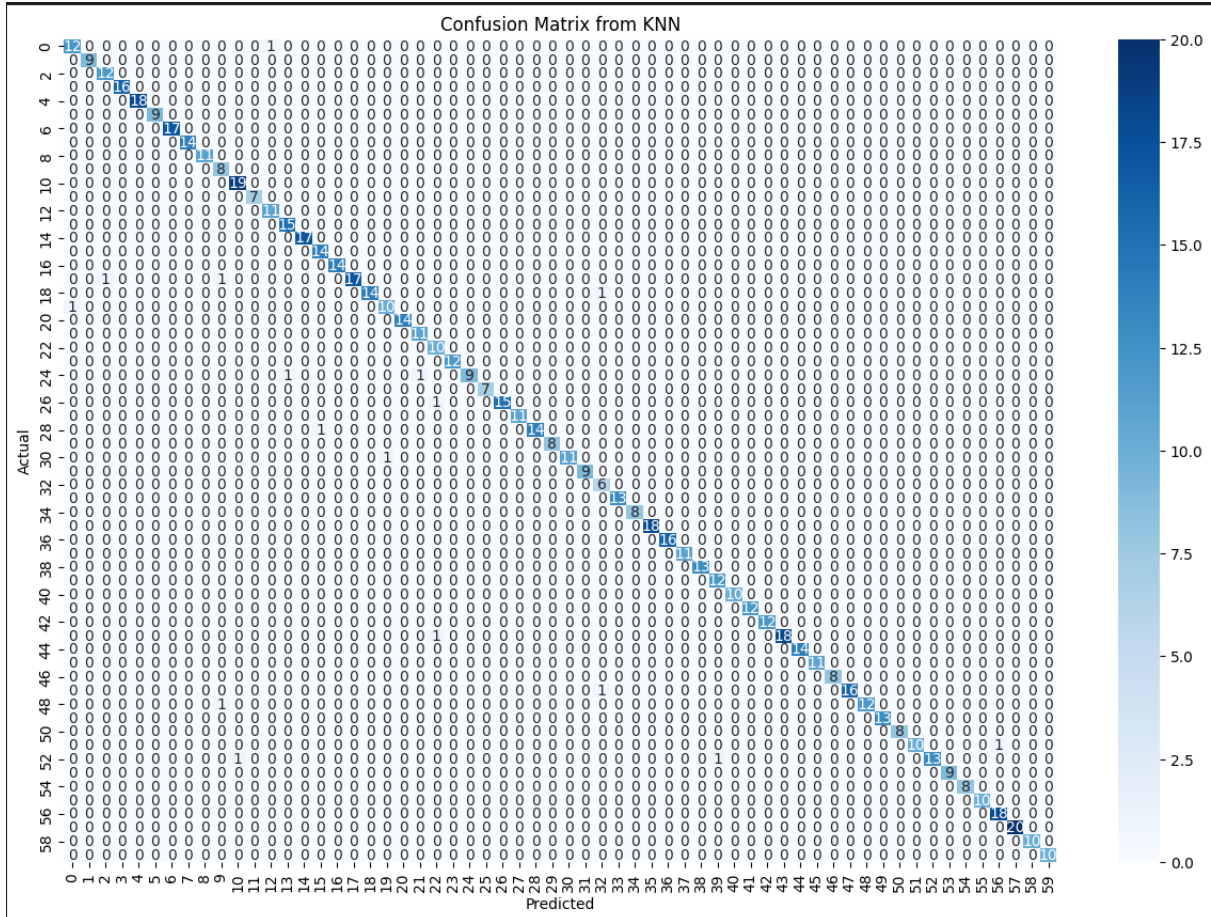
Label 4 - KNN Accuracy: 0.9946666666666667

## Observations with respect to each label

After trying out different feature engineering techniques separately, a combination of all three techniques was used to further reduce the number of features while maintaining a good accuracy score. The order of techniques was first to apply the SelectKBest feature selection method with different k values for each label, and do principal component analysis on top of the selected features, and then apply the SelectPercentile method, again with different percentile values for different labels. Here is a summary of the hyper parameters used for each label, and the confusion matrix obtained for each label.

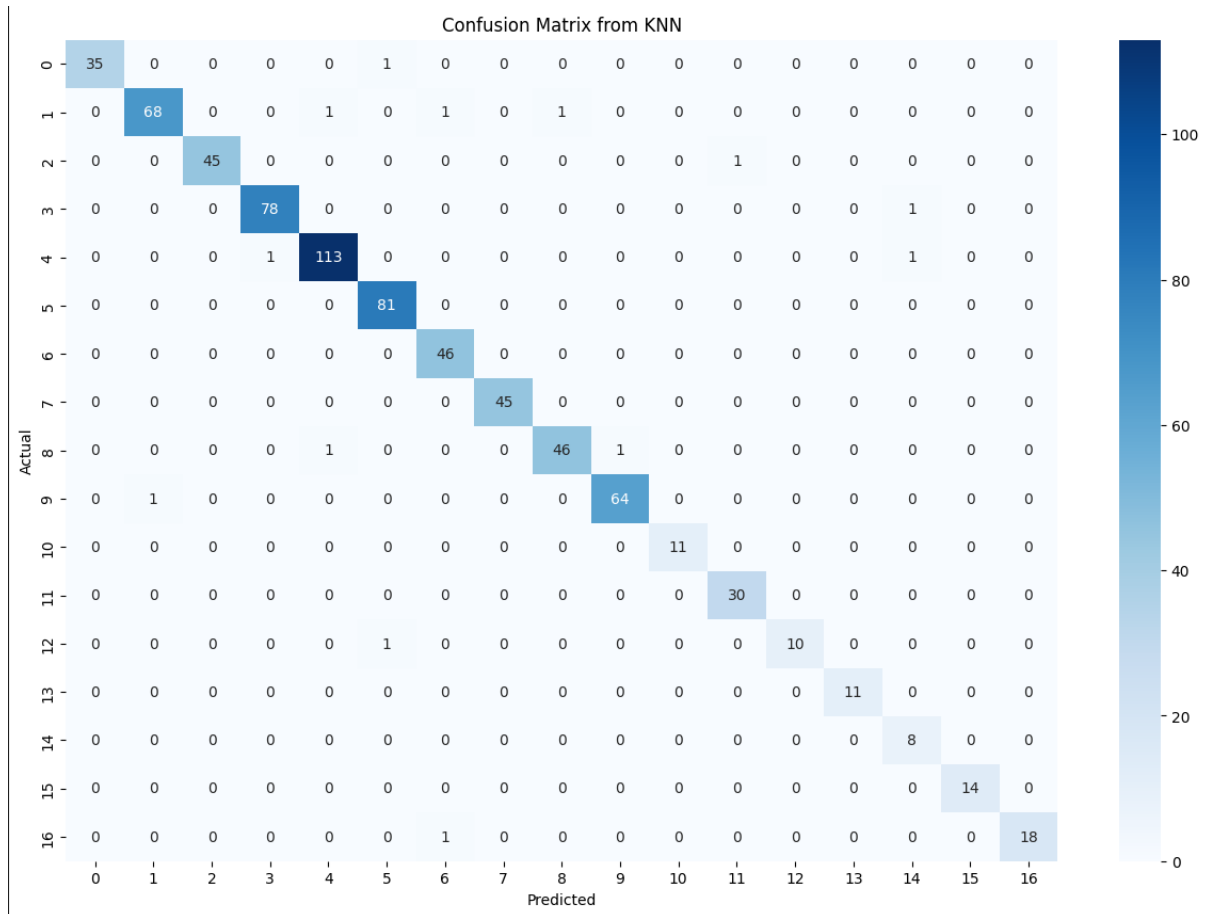
## Label 1

SelectKBest - k = 100  
PCA - variation = 0.99  
SelectPercentile - percentile = 80  
Accuracy score - KNN Accuracy: 0.9786666666666667



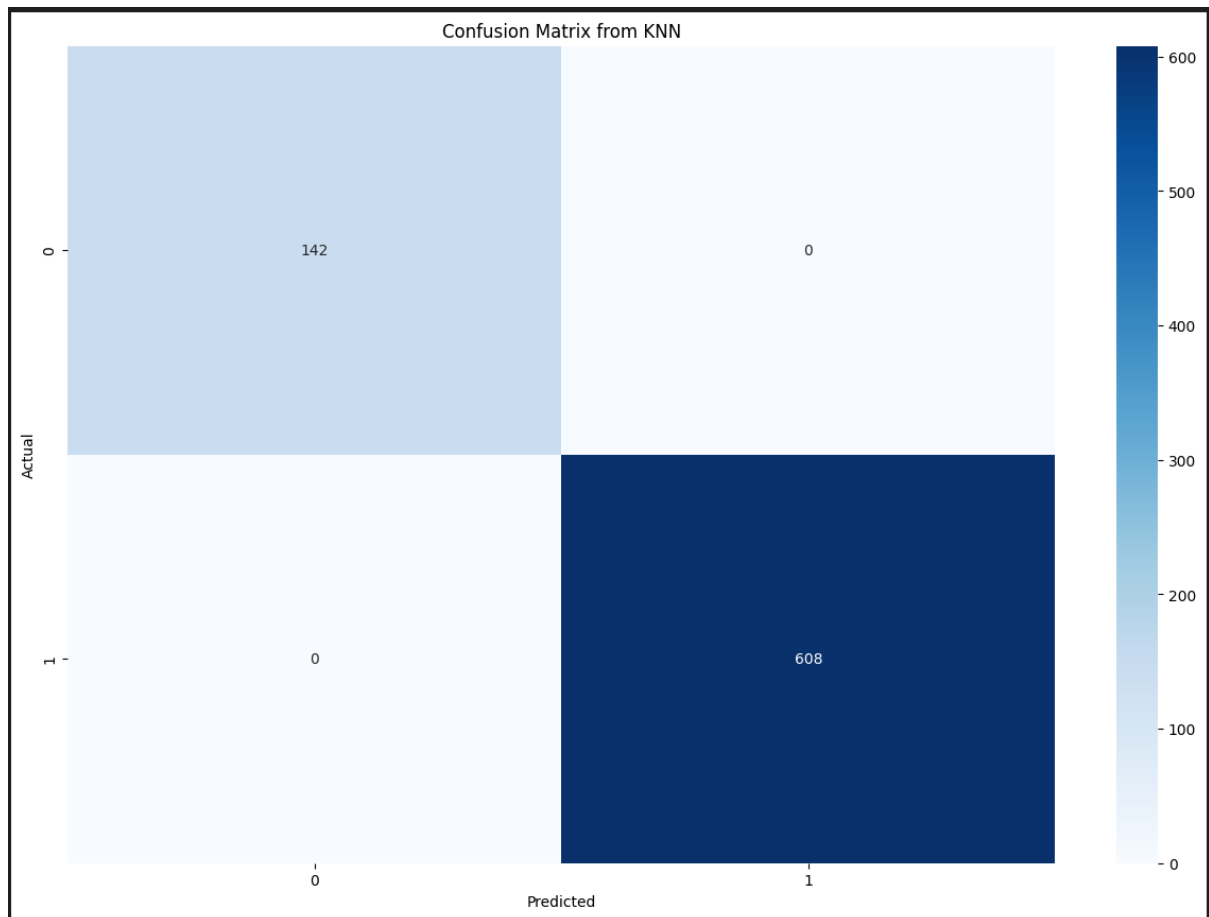
## Label 2

SelectKBest - k = 100  
PCA - variation = 0.98  
SelectPercentile - percentile = 80  
Accuracy score - KNN Accuracy: 0.9823369565217391



## Label 3

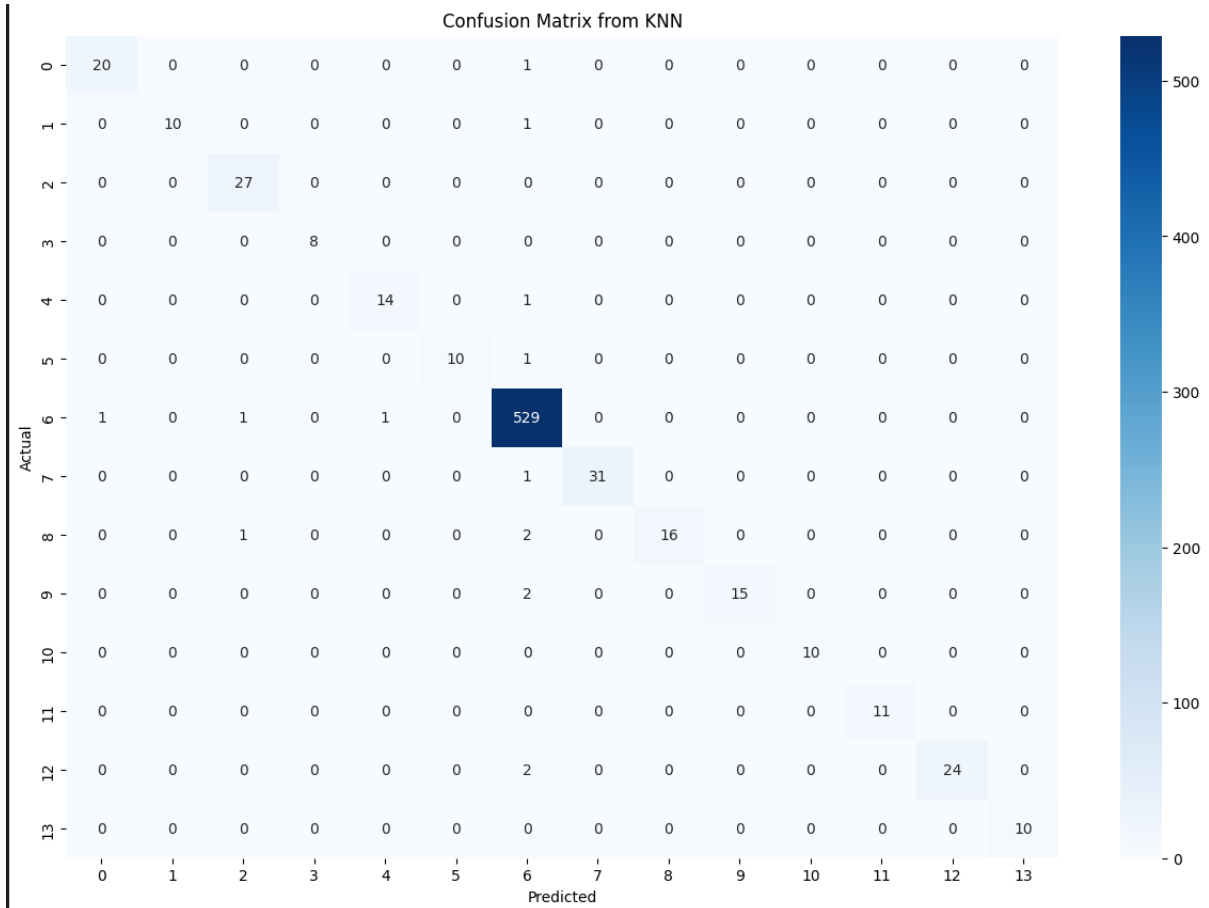
SelectKBest      - k = 100  
PCA                - variation = 0.95  
SelectPercentile   - percentile = 50  
Accuracy score    - KNN Accuracy: 1.0







## Label 4


SelectKBest - k = 100  
PCA - variation = 0.95  
SelectPercentile - percentile = 60  
Accuracy score - KNN Accuracy: 0.98




# Sources

Jupyter Notebook for label 1 -  label1.ipynb

Jupyter Notebook for label 2 -  label2.ipynb

Jupyter Notebook for label 3 -  label3.ipynb

Jupyter Notebook for label 4 -  label4.ipynb