Project Report – Spoken Language Processing 2022

# Authors

Yazid Muaket1, Qossay Jawad ZeinEddin2, Abdelghafur Sbaih3

[1181710@student.birzeit.edu](mailto:1181710@student.birzeit.edu), [1180235@student.birzeit.edu](mailto:1180235@student.birzeit.edu), [1180628@student.birzeit.edu](mailto:1180628@student.birzeit.edu).

# Abstract

The aim of this project is to implement a Speech Recognition System of ethnic groups, which identifies two main ethnic groups in the city of Birmingham, UK, namely Asian and White.

# Introduction

In this project we will use the most common feature extraction techniques in speech processing in order to recognize the ethnicity of the speaker. Some of the features extracted include 12 Mel-Frequency Cepstral Coefficients (MFCCs), 1 Energy Feature, 12 Delta MFCC features, Delta Energy Feature, and 12 Double Delta MFCC features and Double Delta Energy features. Two Machine Learning techniques will be used to train a model for each group and identify the speaker ethnicity as “Asian” or “White”, which are the K-Nearest Neighbors (KNN) and the Support Vector Model (SVM). We will use WEKA tool in order to implement the models and test them.

# Background

Since speech signals are unstable in nature, statistical representations should be generated for compressing the speech signal variability which is achieved by performing feature extraction. The most widely used extraction technique is MFCC. [1]

## Mel-frequency Cepstral Coefficients (MFCC)

The block diagram of MFCC is shown in Fig. 1.

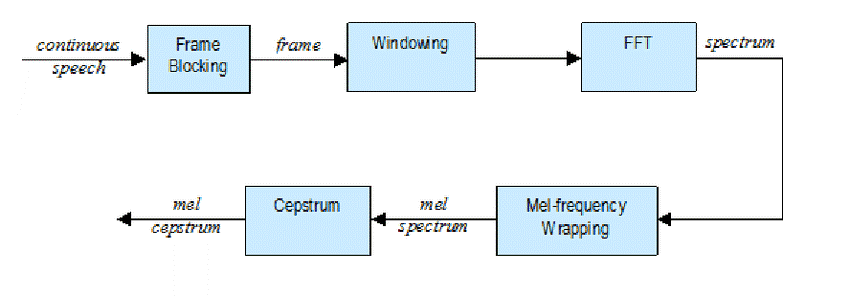


Figure 1: MFCC block diagram.

The MFCC processor involves the following steps:

### Pre-emphasis:

In the pre-processing, the speech signal increases the amplitude of high frequency bands and decrease the amplitudes of lower bands which is implemented by FIR filter.

### Framing and windowing:

The speech signal is split into number of frames. The frame size considered as 25 ms, hamming windowing is applied in order to minimize the signal discontinuities at the starting each edge of the frames.

### Fast Fourier Transformer (FFT):

Each frame of N samples is converted in to time domain into frequency domain

### Mel Filter Bank:

The scale of frequency is converted from linear to mel-scale which is called mel-filter bank.

### Logarithm:

Logarithm is taken for the mel-filter bank which is known as log mel-spectrum

### Discrete cosine Transform (DCT):

The log mel-scale is again converted in to frequency domain to time domain which produces the feature of MFCC.

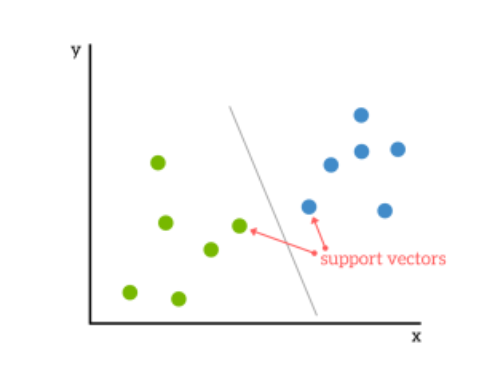
## K-Nearest Neighbor (KNN)

K-nearest neighbors (kNN) is a supervised machine learning algorithm that can be used to solve both classification and regression tasks. kNN works as the value of a data point is determined by the data points around it. [2]

KNN works by finding the distances between a centroid and all the examples in the data, selecting the specified number examples (K) closest to the centroid, then votes for the most frequent label (in the case of classification) or averages the labels (in the case of regression). [3]

## Support Vector Model (SVM)

Support vector Model is one of the most used algorithms that based on statistical theory in machine learning. Support vector machines are used to sort two data groups by like classification. The algorithms draw lines (hyperplanes) to separate the groups according to patterns, as shown in Fig. 2.



z

# Methodology

In this part, feature extraction for both Asian and White speaker speeches was used. A dataset containing .wav files for both Asian and White speaker was used. The dataset was divided to two sets: Training set and a test set. Each set contains the recorded files of the Asian and White speaker separately. The files used can be found in the references section number 5.

## Feature Extraction Process

The feature extraction process should’ve been done using Python coding language directly, but due to lack of a strong feature extraction library in python that will be used to extract the features of the .wav files from the training set, a Linux command was used in order to convert the extension of the .wav files to be more usable. The command used is shown in Fig. 3.

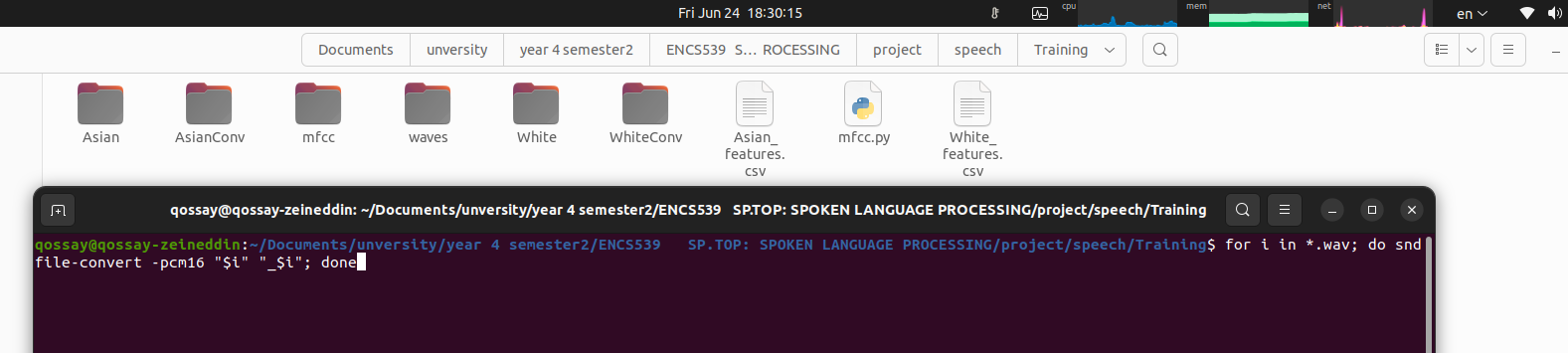


Figure 3: Linux command used to convert extension of the .wav files.

Using Python code on Linux, the MFCC feature extraction was applied to the .wav training files pointed earlier. Fig. 4 shows the code used for getting the MFCC features from the speech files.

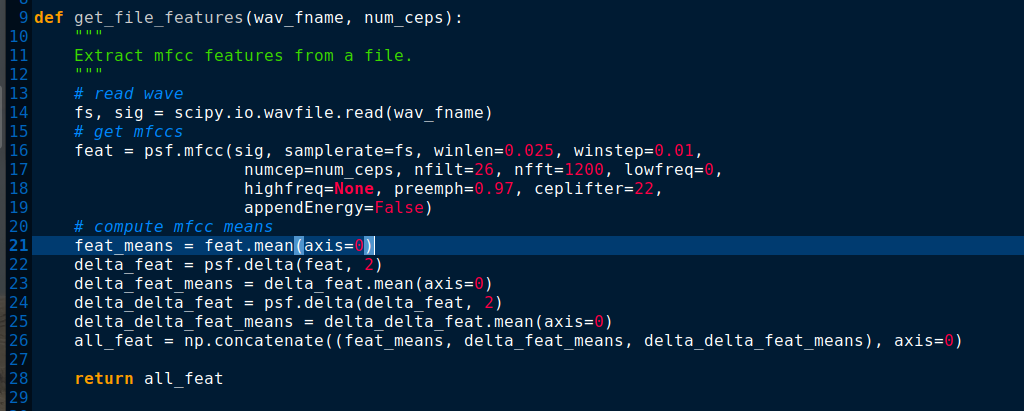


Figure 4: MFCC feature extraction python code.

The total number of features extracted was 39 features which include 12 MFCC features, 1 energy feature, 12 MFCC delta features, 1 energy delta feature, 12 MFCC double delta features and 1 energy double delta feature.

The following function shown in Fig. 5 takes the sound from each file, and applies the feature extraction, then saves the features vectors in a .csv file. This was applied to all files which include the training set and the test set.

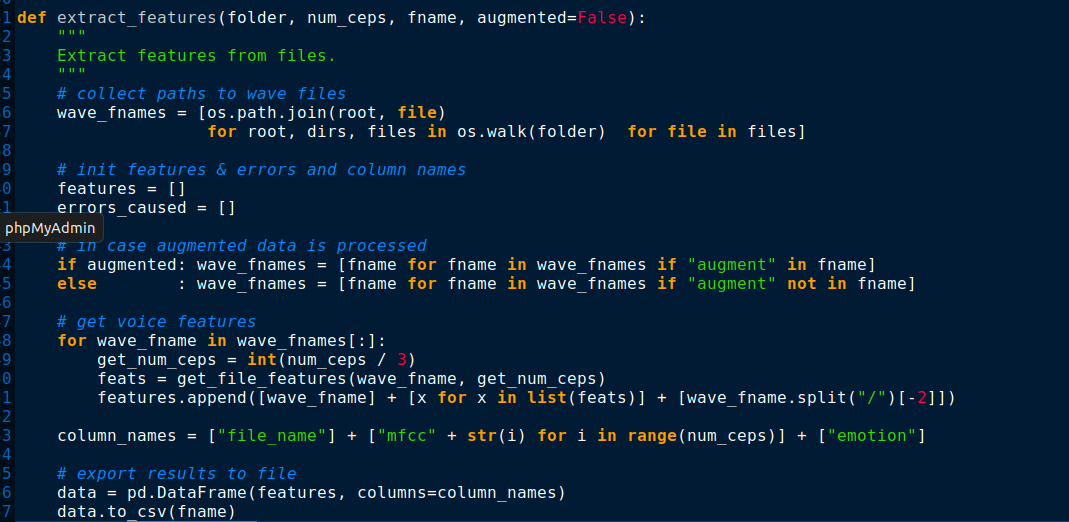


Figure 5: Saving the features into a .csv file.

The “Asian” .csv training file and the “White” .csv training file were merged together as one .csv file as shown in Fig. 6.

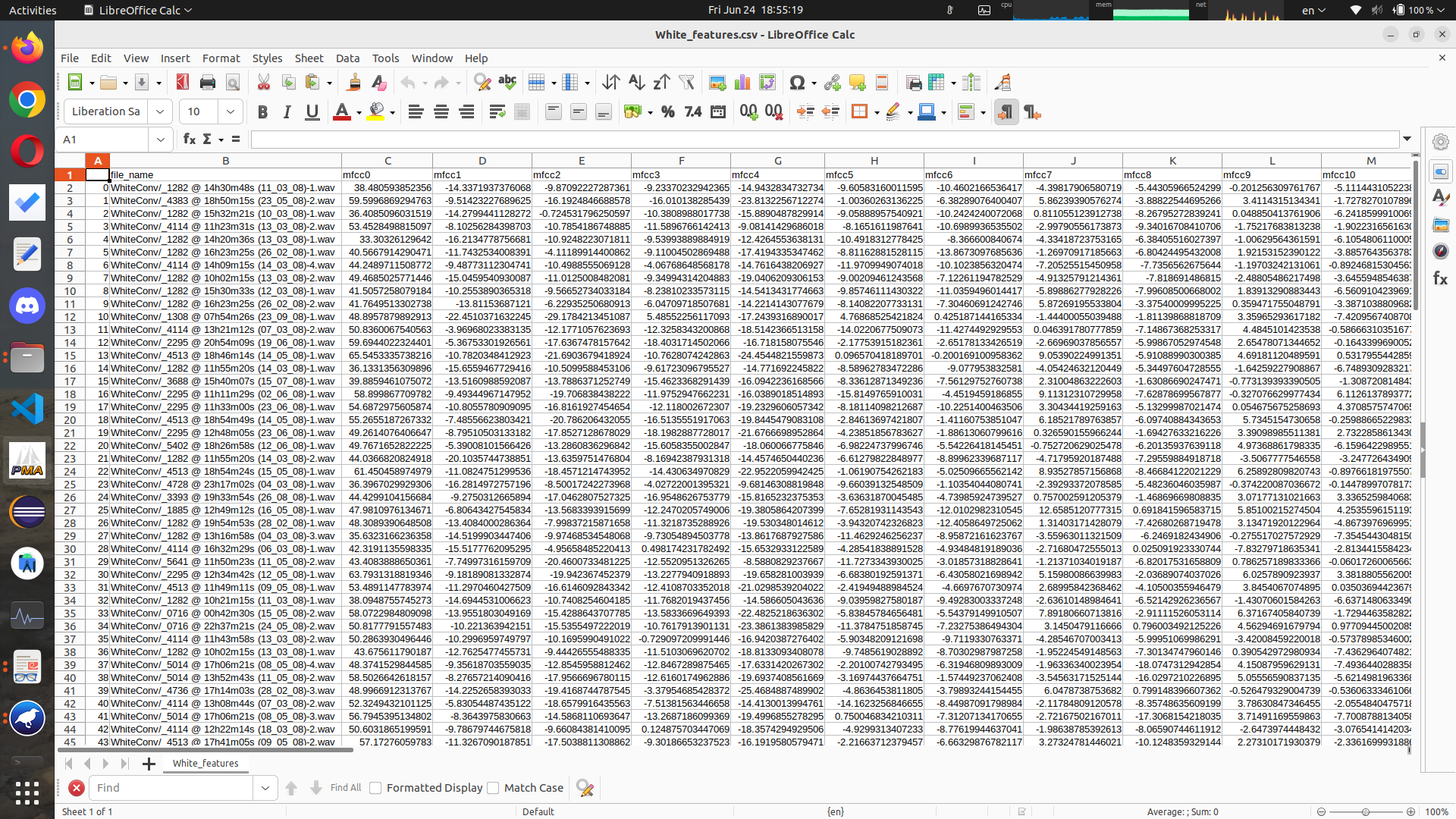


Figure 6: Features .csv file.

The pandas python library was used in order to get the 39 features from each file using array view for each speech file.

## Building the models

### KNN model

The WEKA machine learning tool was used to build the KNN model. The .csv file mentioned earlier was used to build that model as shown in Fig. 7.

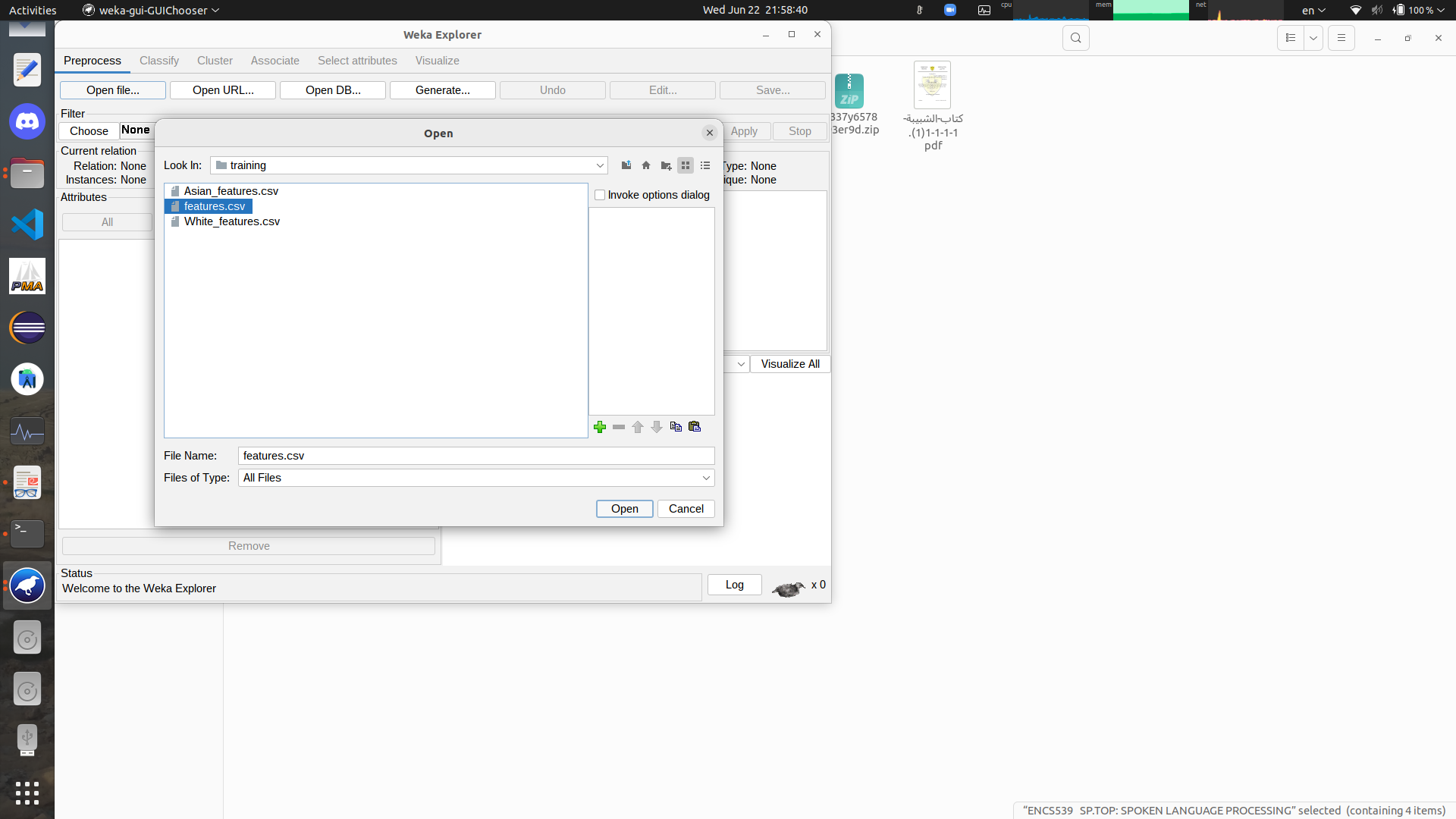


Figure 7: Using .csv file to build KNN model.

From classify lazy → IBK, KNN was chosen. The KNN model was built as shown in Fig. 8.

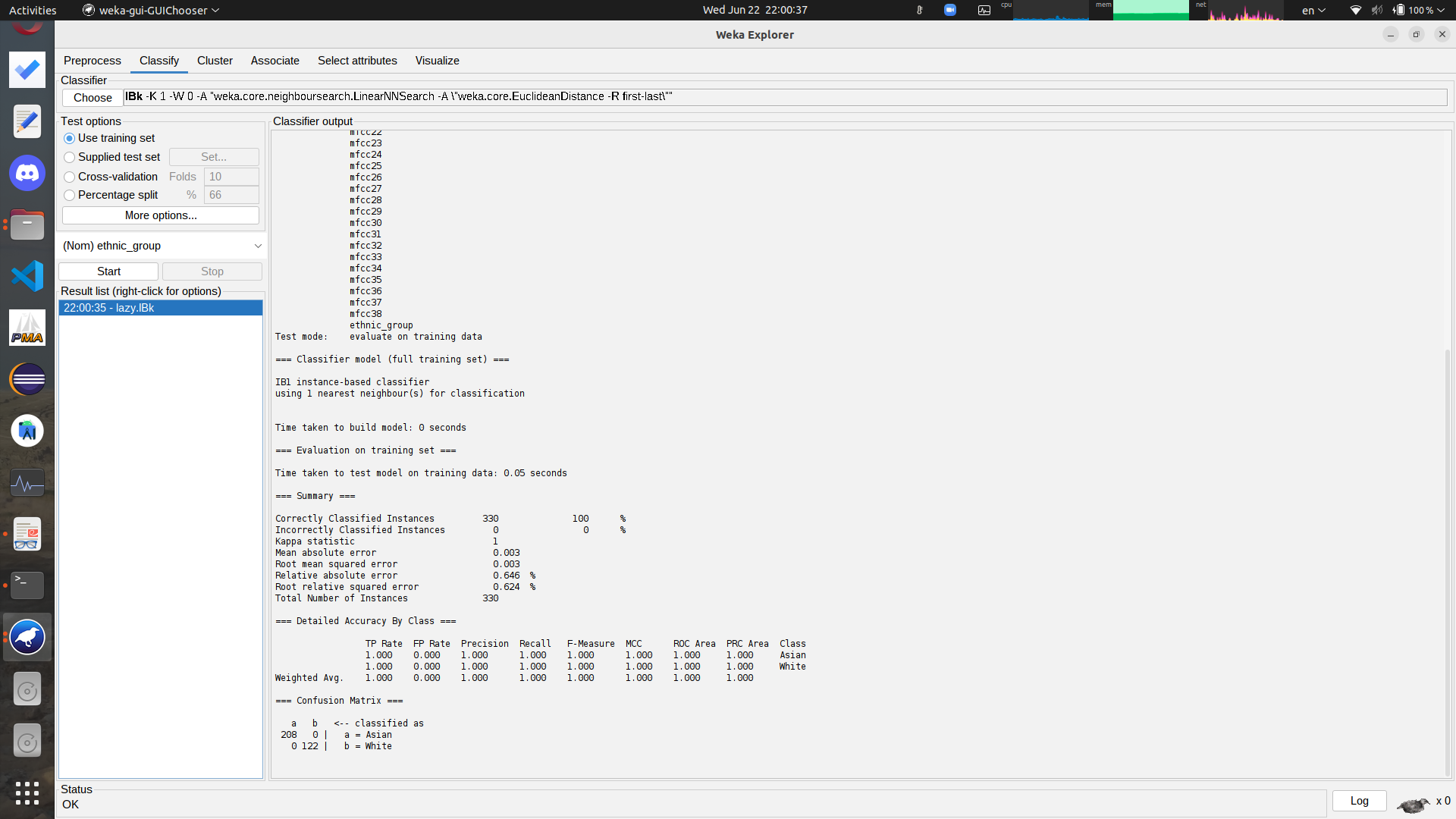


Figure 8: KNN model.

As shown the accuracy is 100%, which means that its overfitting. This happened because the dataset given was not big enough, and the model fits the data given exactly.

### SVM model

The same approach was repeated for the SVM model, but in order to build the model libSVM was chosen from lazy from classify. The SVM model build results are shown in Fig. 9.

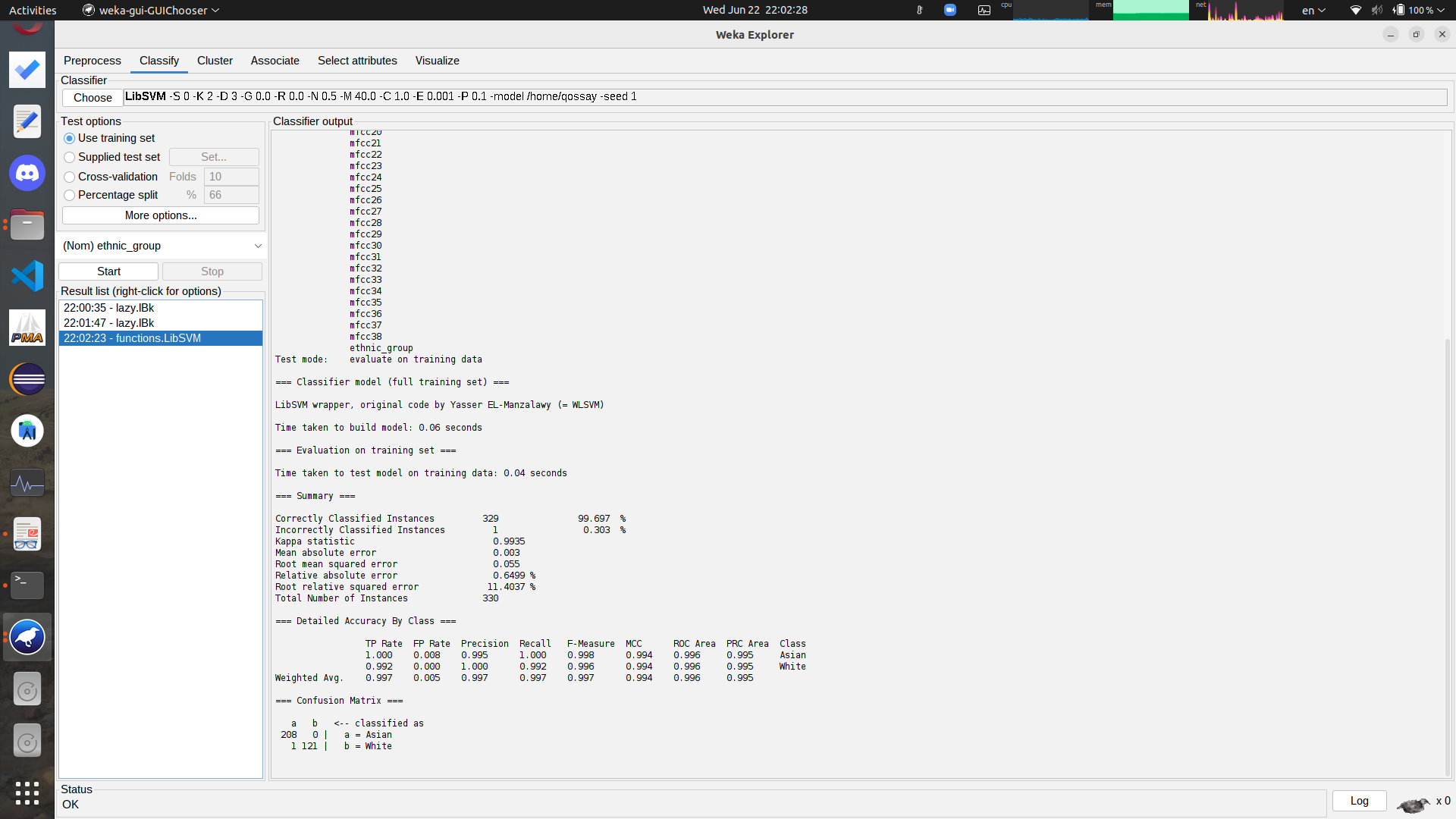


Figure 9: SVM model.

The accuracy for this model was 99.6% which means that the model is a bit overfitting.

# Experiments and Results

We used the same tool in order to test the models that have been built in the previous section, we have a testing data that consists of 40 different voice samples.

## Testing KNN Model

Starting with the K-Nearest Neighbor Model, as seen in the testing results below, the correctly classified instances are 23 i.e., **accuracy 57.5%**

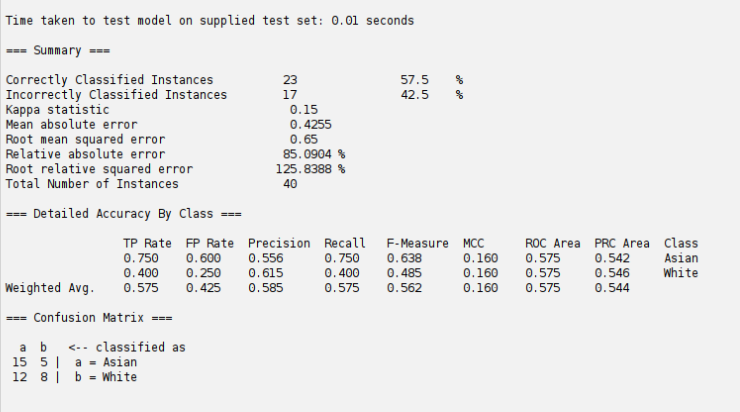


Figure 10:KNN testing results.

## Testing SVM Model

Moving towards the Support Vector Model, as seen in the testing results below, the correctly classified instances are 23 i.e., **accuracy 45%**

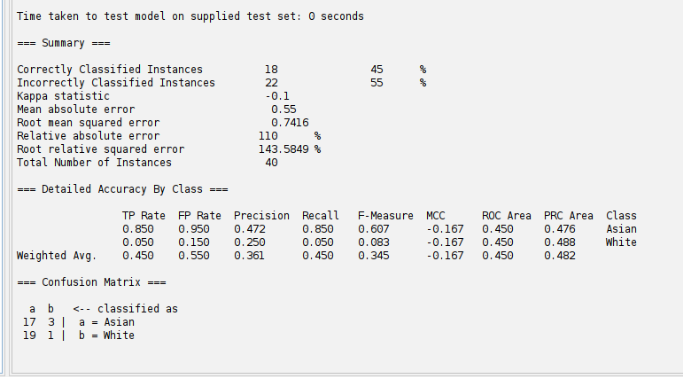


Figure 11: SVM testing results.

## Summary

We obviously see the that the KNN model has slightly better accuracy than the SVM model as Table 1 suggests.

Table 1: Models Comparison.

|  |  |  |
| --- | --- | --- |
|  | KNN model | SVM model |
| Accuracy | 57.5% | 45% |

The lack of overall accuracy is proportional to the size of the training data set, in other words if we increased the size of given training data set the accuracy will increase also.

# Conclusion

In this project, we’ve explored the procedure of Mel-Frequency Cepstrum Coefficients (MFCCs) extraction. The motivations and implementation of each step was discussed. We also intent to improve the system to take more than two ethnicities other that Asian and White, we can also improve by adding another extraction technique of the same speech signal this will insure more accuracy to each model (more features 🡪 less error).

# References

1. <http://www.ijetajournal.org/volume-4/issue-2/IJETA-V4I2P9.pdf>
2. <https://towardsdatascience.com/k-nearest-neighbors-knn-explained-cbc31849a7e3#:~:text=K%2Dnearest%20neighbors%20(kNN)%20is%20a%20supervised%20machine%20learning,friends%20we%20grew%20up%20with>
3. <https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761>
4. <https://www.kdnuggets.com/2016/07/support-vector-machines-simple-explanation.html>
5. <https://drive.google.com/drive/folders/1cpobyvIAW0KxIEeoRII0DRbkxMzZXHaN?usp=sharing>