

**ENGINNERING FACULTY - COMPUTER ENGINEERING DEPARTMENT**

**Biomedical Signal Analysis and Machine Learning**

**2023-2024**

**STUDENT NAME** : Semih Utku POLAT

**STUDENT ID** : 191805060

**TUDENT NAME** : Burak Tüzel

**STUDENT ID** : 191805057

TABLE OF CONTENTS

Contents

[2 Dataset Description 1](#_Toc157196340)

[3 Audio Recording 2](#_Toc157196341)

[3.1 Record Audio Files For Model Training 2](#_Toc157196342)

[3.2 Record Audio Files For Model Testing With Background Noise 2](#_Toc157196343)

[3.3 Record Audio Files For Model Testing Without Background Noise. 3](#_Toc157196344)

[4 Audio Labeling 3](#_Toc157196345)

[5 Audio Model Training 5](#_Toc157196346)

[6 Audio Model Testing 10](#_Toc157196347)

INTRODUCTION

Voice or speaker recognition is the ability of a machine or program to receive and interpret dictation or to understand and perform spoken commands. Voice recognition can be defined as a technology that allows a device to understand and analyse a human voice and then transcribe each of the dictated words into usable text. Specifically, the voice is captured via the device’s microphone in sound frequencies and then transcribed into written text. Voice recognition can be seen as an alternative to keyboard/handwritten entry and is often praised for being faster and saving time in everyday tasks.

# Dataset Description

The dataset comprises a collection of sound recordings generated through a scripted series of events. Each recording is designed to capture various audio cues, including specific verbal instructions and non-verbal actions. The events within each recording are as follows:

Wait: A period of silence to capture ambient background noise.

Read "computer": The participant reads the word "computer" aloud.

Wait: Another silent interval.

Read "engineering": The participant reads the word "engineering" aloud.

Wait: Another silent interval.

Say your name: The participant verbally states their first name.

Wait: Another silent interval.

Say your last name: The participant verbally states their last name.

Wait: Another silent interval.

Cough once: The participant coughs audibly.

Wait: Another silent interval.

Clap your hands once: The participant claps their hands.

Wait: Another silent interval.

Snap your fingers once: The participant snaps their fingers.

Wait: Another silent interval.

This sequence of events is repeated three times to provide variability and ensure robustness in the dataset. Additionally, each recording is created in two variations: with background sound and without background sound. In this project, the dataset accounts for multiple members, generating a total of 12 recordings for our group (6 with background sound and 6 without background sound).

# Audio Recording

This project has three different scripts used to generate audio files for training the model, audio files with background noise for testing the model with background noise, and audio files without background noise for testing the model without background noise.

## Record Audio Files For Model Training

metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturulduThis script used to record 15 sec long audio files for training the model with and without background noise.

## Record Audio Files For Model Testing With Background Noise

metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturulduThis script used to record 2 sec long audio files for testing the model with background noise.

## Record Audio Files For Model Testing Without Background Noise.

This script used to record 2 sec long audio files for testing the model without background noise.

metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

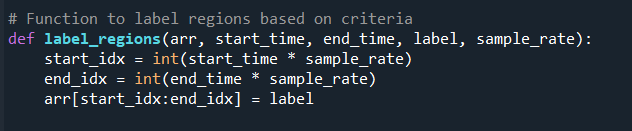
# Audio Labeling

In this part, sound\_signal\_labeling.py script has written to analyze an audio file, specifically applying Short-Time Fourier Transform (STFT) to generate a spectrogram. The script also allows interactive labeling of specific segments of the audio.

Importing necessary libraries and loading the audio file with its sample rate. apply\_stft\_and\_plot function applies STFT to audio file to generate a spectrogram then plots it. The on\_select function allows the user to label a spectrogram graph based on the range they choose for an audio file. Finally saves the spectrogram in decibels and the labels array into the sounds folder when the user pressed “Return” keymetin, ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldu

This is the function that is used in on\_select function to label on the spectrogram graph.



The example of executing:

metin, yazılım, ekran görüntüsü, multimedya yazılımı içeren bir resim

Açıklama otomatik olarak oluşturuldu

The label file when pressed “Return” key:

metin, ekran görüntüsü, sayı, numara, yazılım içeren bir resim

Açıklama otomatik olarak oluşturuldu

The spec\_db file when pressed “Return” key:

metin, ekran görüntüsü, sayı, numara, yazılım içeren bir resim

Açıklama otomatik olarak oluşturuldu

# Audio Model Training

In this part, sound\_signal\_model\_training.py script has written to train 4 different models named as RandomForestClassifier, KNeighborsClassifier, DecisionTreeClassifier, SVC.

Importing necessary libraries, defining required paths, defining seed to make sure the results are not different from each other when executing, loading audio\_signal and labels, lastly splitting audio signal into train and test parts to be used in training phase.

metin, ekran görüntüsü, yazılım, multimedya yazılımı içeren bir resim

Açıklama otomatik olarak oluşturuldu

Defining commands recorded, and defining arrays to save the scores after the training.

metin, ekran görüntüsü içeren bir resim

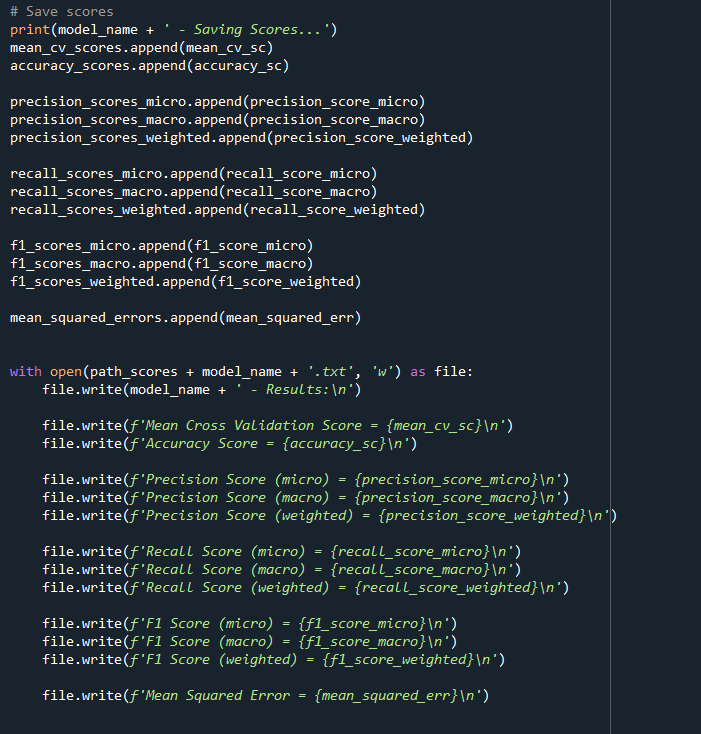
Açıklama otomatik olarak oluşturuldu

###### Training Part

Model training and calculating scores.  
  
metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

Saving scores into \*txt.



Plotting Confusion matrix.

metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

Plotting Time Performance Graphs.

metin, elektronik donanım, ekran görüntüsü, yazılım içeren bir resim

Açıklama otomatik olarak oluşturuldu

Results of execution.

metin, ekran görüntüsü, diyagram, dikdörtgen içeren bir resim

Açıklama otomatik olarak oluşturuldumetin, ekran görüntüsü, diyagram, dikdörtgen içeren bir resim

Açıklama otomatik olarak oluşturulduDT KNN

metin, ekran görüntüsü, diyagram, paralel içeren bir resim

Açıklama otomatik olarak oluşturuldumetin, ekran görüntüsü, diyagram, paralel içeren bir resim

Açıklama otomatik olarak oluşturulduRF SVC

Accuracy Score

ekran görüntüsü, dikdörtgen, kare, çizgi içeren bir resim

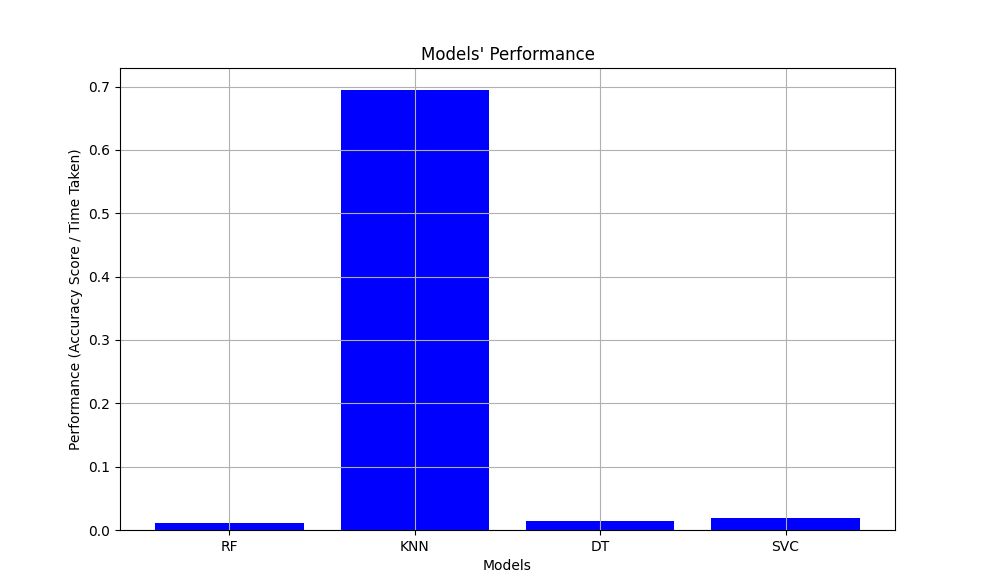
Açıklama otomatik olarak oluşturuldu

Time

çizgi, dikdörtgen, ekran görüntüsü, kare içeren bir resim

Açıklama otomatik olarak oluşturuldu

Performance



# Audio Model Testing

In this part, sound\_signal\_model\_testing.py script has written to predict recorded audios. Two types of voice recordings were used in this script: pre-recorded and recorded during the execution of the code.

###### PRE-RECORDED AUDIO FILES

Load pre-recorded audio files with and without background sound. Then, add audio files to the list to use in the loop.

metin, ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldu

Each sound file was adapted using short term fourier transform, label prediction was made. Then the most guessed label printed.

metin, ekran görüntüsü, yazılım içeren bir resim

Açıklama otomatik olarak oluşturuldu

Results of Execution:

10 prediction out of 12 prediction is true..

metin, ekran görüntüsü, yazılım, multimedya yazılımı içeren bir resim

Açıklama otomatik olarak oluşturuldu

###### RECORDING DURING THE EXECUTION

In this part, everything is the same except that we record and use it to predict while the code is running.

Results of execution:

Cough record result from recording during the execution

metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu