**Name: Smitha Reddy Kondakalla**

**Student ID: U80307854**

**README**

**CAP 4628/5627 Affective Computing  
Project 1 – 3D Expression Recognition**

import sys

import os

import math

**import numpy as np:**

* NumPy arrays are used to store and manipulate the 3D coordinates of facial landmarks.
* Trigonometric functions like **np.radians()** and **np.cos()** are used to convert angles to radians and compute cosine values, respectively.
* NumPy arrays are used to represent rotation matrices for 3D rotations.
* Matrix multiplication (**np.dot()**) is used to apply rotations to the original landmarks.
* NumPy arrays are used to convert data between different formats, such as reshaping data from a flat representation to a 3D representation and vice versa.
* NumPy's random module (**np.random.choice()**) is used to select random samples from the dataset.

**import pandas as pd:**

* Pandas (**import pandas as pd**) is primarily used for data manipulation and organization, especially for handling tabular data.
* Pandas is used to read and load data from files into DataFrames, such as reading BND files in the **read\_bnd\_file()** function and loading processed data from directories in the **process\_and\_create\_dataframe()** function.
* DataFrames are used to organize the facial landmark data, where each row represents a sample (e.g., facial landmark coordinates) and each column represents a feature or attribute.
* DataFrames are used to preprocess and transform data, such as translating landmarks to the origin (**translate\_to\_origin()**) and rotating landmarks (**rotateX()**, **rotateY()**, **rotateZ()** functions).

**import matplotlib.pyplot as plt, from mpl\_toolkits.mplot3d import Axes3D:**

* **matplotlib.pyplot**: This module is imported as **plt**. It is used to create plots and visualizations, such as scatter plots, histograms, line plots, etc.
* **mpl\_toolkits.mplot3d**: This submodule is imported to enable 3D plotting functionality. It allows the creation of 3D axes for plotting 3D data.

**from sklearn.model\_selection import KFold:**

* The **KFold** class from **sklearn.model\_selection** is used to split the dataset into k folds for cross-validation. It provides methods to generate indices for training and testing sets, facilitating the evaluation of machine learning models across multiple subsets of the data. Here we have done 10 fold cross-validation where the dataset is divided into 10 subsets.

**from sklearn.ensemble import RandomForestClassifier**

**from sklearn.svm import SVC**

**from sklearn.tree import DecisionTreeClassifier**

* the imported classifiers from scikit-learn (**RandomForestClassifier**, **SVC**, and **DecisionTreeClassifier**) are used to create instances of machine learning models for classification tasks. These models are trained on input data and evaluated using cross-validation

**from sklearn.metrics import confusion\_matrix, accuracy\_score, precision\_score, recall\_score:**

* The imported functions from scikit-learn (confusion\_matrix, accuracy\_score, precision\_score, and recall\_score) are used to compute evaluation metrics such as confusion matrices, accuracy, precision, and recall, respectively, for evaluating the performance of machine learning models