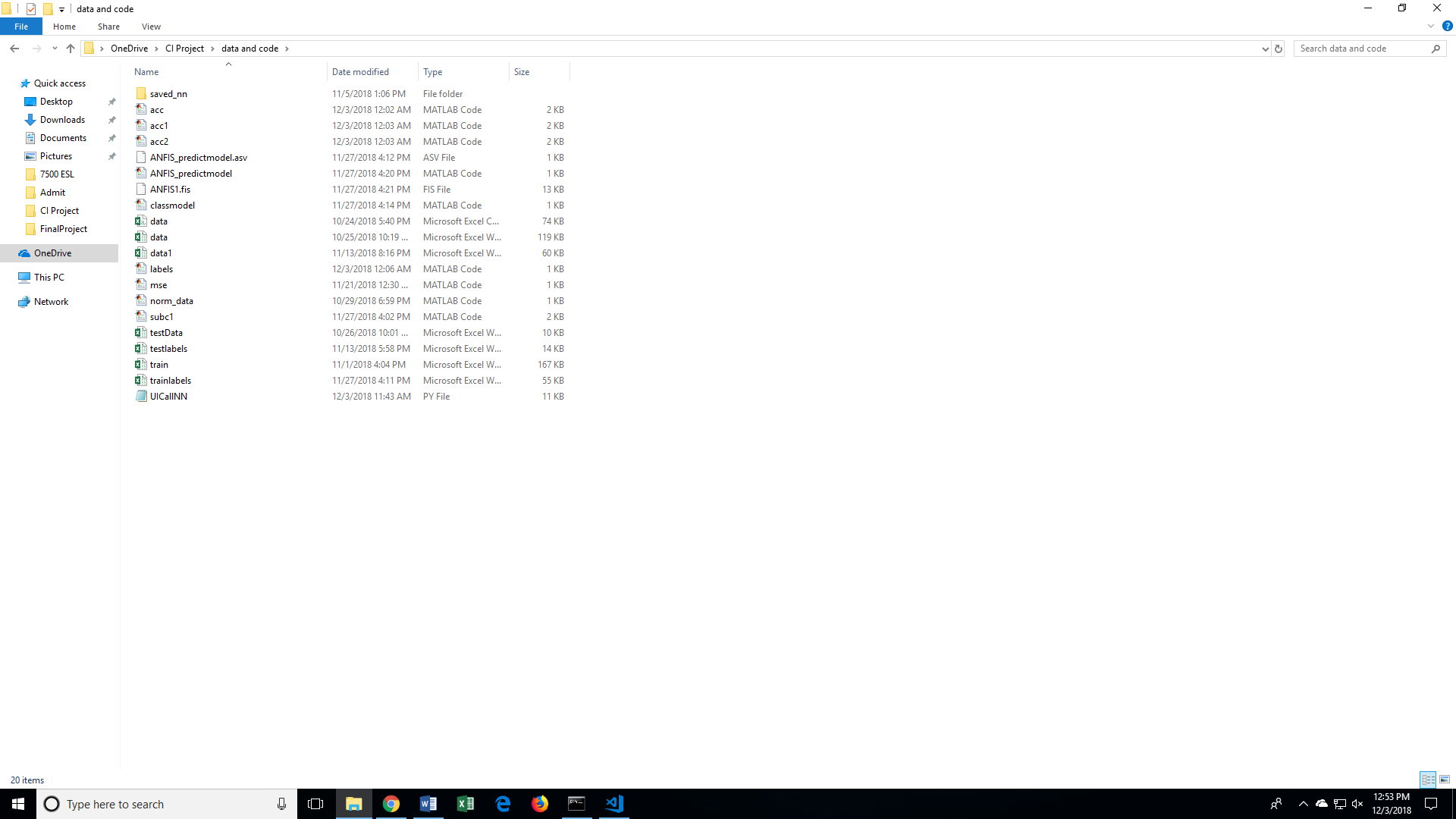
README

Student Performance Prediction using NN and ANFIS

The “data and code” folder consists of the data we used for training and testing , the labels and the matlab code for ANFIS and python code for the UI and ANN .

Copy the files onto your computer. The file structure should look like this:



**GUI -- APPLICATION**

The project can be seen using the GUI application we built. Please follow the steps below:

* Code to run : UICallNN.py
* Environment : Python3
* Packages: Tkinter,Numpy,pandas,TensorFlow

1. If you are using an IDE, please open the "UICallNN.py", and run it directly from the IDE.
2. If you run it from the command line, please cd into the correct folder and then type: python UICallNN.py
3. You should get a GUI application window with five input boxes. Please enter the Assignment#1 score, Assignment#2 score, Assignment #3 score, and Quiz #1 score, and Quiz#2 score that you want to predict the final grade for.
4. Click the "Predict by NN" button to get the result from the trained Neural Network.
5. Click the "Predict by ANFIS" button to get the result from the trained ANFIS.

The above steps are using already trained models.

**Training a new NN model:**

1. Please change the **code line 146 in UICallNN.py** , from **do\_train = 0 to do\_train = 1**
2. Run the GUI application again. When you click “Predict by NN” button, a new model will be trained, saved, and used to predict the result.

**Training a new ANFIS model:**

1. Run the code **subc1.m** in MATLAB.
2. Save “**fismat2**” model from the workspace as ‘**ANFIS1.fis**’ in the current directory.
3. Run the GUI application again. When you click on “Predict by ANFIS”, a new model will be used to predict the values

**Code Directory:**

**acc.m, acc1.m, acc2.m** – calculates accuracy of ANFIS+ Classification model for different data splits

**subc1.m** – training an ANFIS model

**mse.m** – calculates the MSE for the ANFIS model

**norm\_data.m** – does Min –Max Normalization

**labels.m** – assigns labels to the original data after normalizing

**classmodel.m** – contains the code for the classification model

**ANFIS\_predictmodel** – takes the user input from the GUI and returns the predicted label

**UICallNN.py –** GUI and code for ANN

**Saved\_nn –** folder where the trained NN is saved and is used later to predict labels