Importing Libraries:

1. import pandas as pd: Imports the pandas library with an alias pd.
2. import numpy as np: Imports the numpy library with an alias np.
3. import tensorflow as tf: Imports the TensorFlow library with an alias tf.

Loading Data (load\_data function):

1. load\_data(file\_path): A function to load normalized data from an Excel file located at file\_path.
2. pd.read\_excel(file\_path): Reads the Excel file into a pandas DataFrame df.
3. df.iloc[:, 1:14].values: Extracts the input parameters and the target (TMRF) from the DataFrame and converts them to a numpy array.
4. return data[:, :-1], data[:, -1]: Returns the input parameters and the target separately.

Training the Model (train\_model function):

1. This function trains a neural network model using TensorFlow.
2. It takes the input data, target values, weights, biases, learning rate, momentum, and epochs as parameters.
3. It initializes velocity variables for weights and biases.
4. It iterates through each epoch and each data point, performing forward and backward passes to update the weights and biases using the momentum optimizer.
5. Finally, it returns the predictions made by the trained model.

Calculating Metrics (calculate\_metrics function):

1. This function calculates Mean Absolute Deviation (MAD), Standard Deviation (SD), Pearson correlation coefficient (CC), and Mean Squared Error (MSE) between true and predicted target values.
2. It takes true target values (y\_true) and predicted values (y\_pred) as parameters.
3. It returns MAD, SD, CC, and MSE.

Outputting Results (output\_results function):

1. This function saves the predicted target values and metrics to Excel files.
2. It creates two DataFrames, one for predictions and one for metrics.
3. It saves these DataFrames to Excel files named predictions.xlsx and metrics.xlsx.
4. It prints messages indicating that the predictions and metrics have been saved.

Main Function (main):

1. This is the main function where the entire workflow is orchestrated.

It loads the data, sets hyperparameters, defines weights and biases, trains the model, calculates metrics, and outputs results.

It calls the load\_data, train\_model, calculate\_metrics, and output\_results functions with appropriate parameters.

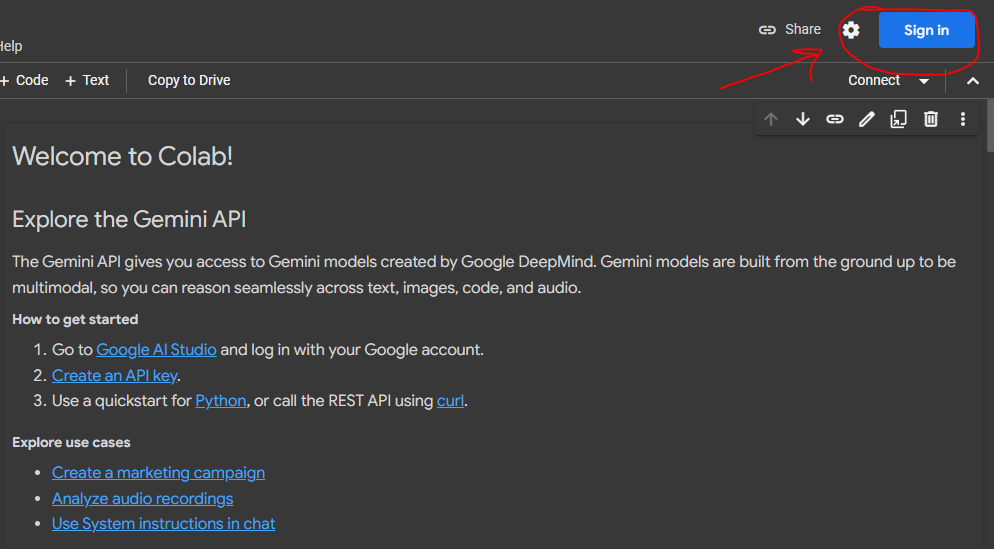
The comments in the code also provide additional context and explanation for each step. Let me know if you need further clarification on any specific part!

**How to run the program**

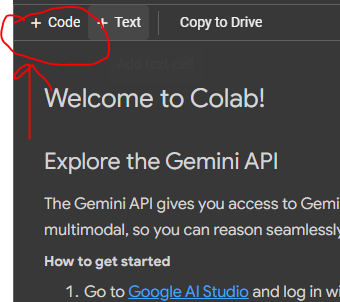
1. **Go to Google Colab**

**https://colab.research.google.com/**

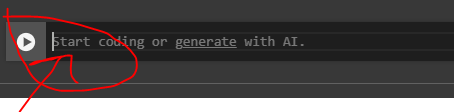
1. **Signin using gmail**

****

1. **After Signin click on +Code**

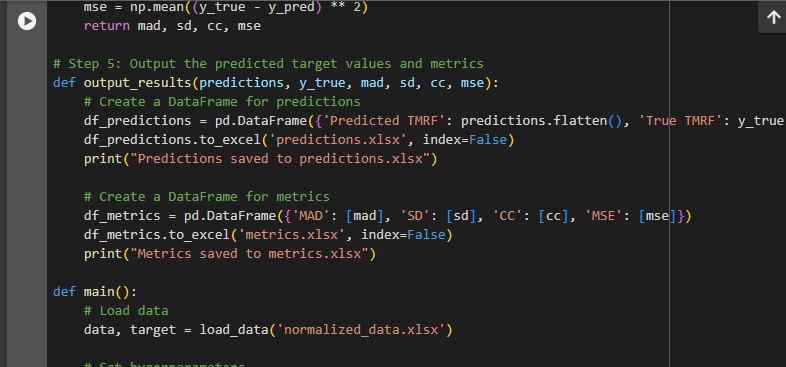


1. **Something like this appearance will come**

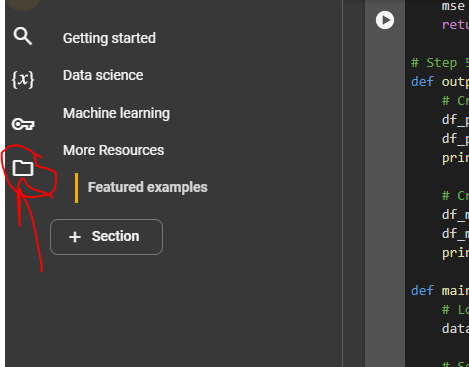
****

1. **Paste the main code named *main program.py* which is present in this location**

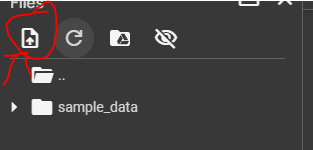
**Neural network ML\CODES\main program.py**

****

1. **Go to this element of Google Colab**

****

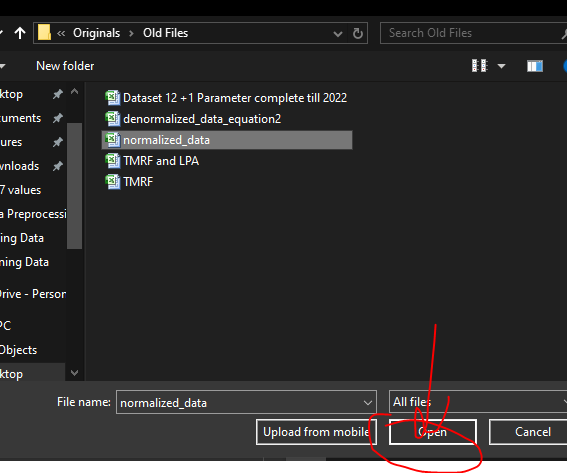
1. **And upload the noramalized dataset file which will be used by the program by clicking this icon**



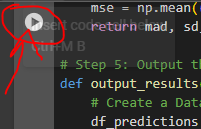
1. **This is the location to which you need to go to upload the file**

**Neural network ML\Originals\Old Files\normalized\_data.xlsx**

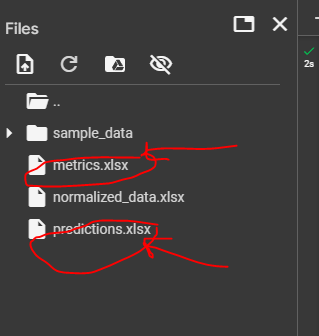
1. **Select normalized\_data.xlsx and press open**

****

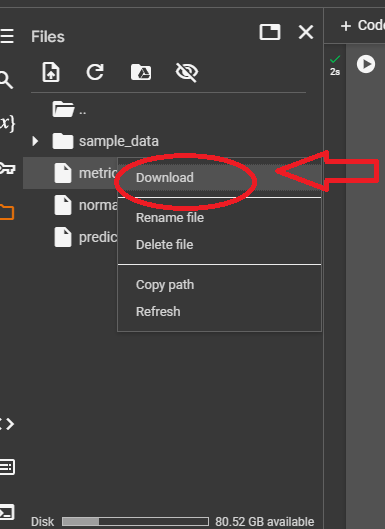
1. **Press the run button which looks like (play) option after upload the normalized\_data.xlsx**

****

1. **After approx 6-7 mins two files will be generated namely metrics and predictions download it**

****

1. **Download metrics and predictions files which are the output files**

****