Neural Network Deep Learning

Assignment – 4

Name: Kishor Kumar Andekar

Student ID: 700744713

Github Link : https://github.com/kishorreyansh/Neural-Network-Deep-Learning/tree/main/Assignment-4

1. Data Manipulation

a. Read the provided CSV file ‘data.csv’.

b. https://drive.google.com/drive/folders/1h8C3mLsso-R-sIOLsvoYwPLzy2fJ4IOF?usp=sharing

c. Show the basic statistical description about the data.

d. Check if the data has null values.

i. Replace the null values with the mean

e. Select at least two columns and aggregate the data using: min, max, count, mean.

f. Filter the dataframe to select the rows with calories values between 500 and 1000.

g. Filter the dataframe to select the rows with calories values > 500 and pulse < 100.

h. Create a new “df_modified” dataframe that contains all the columns from df except for “Maxpulse”.

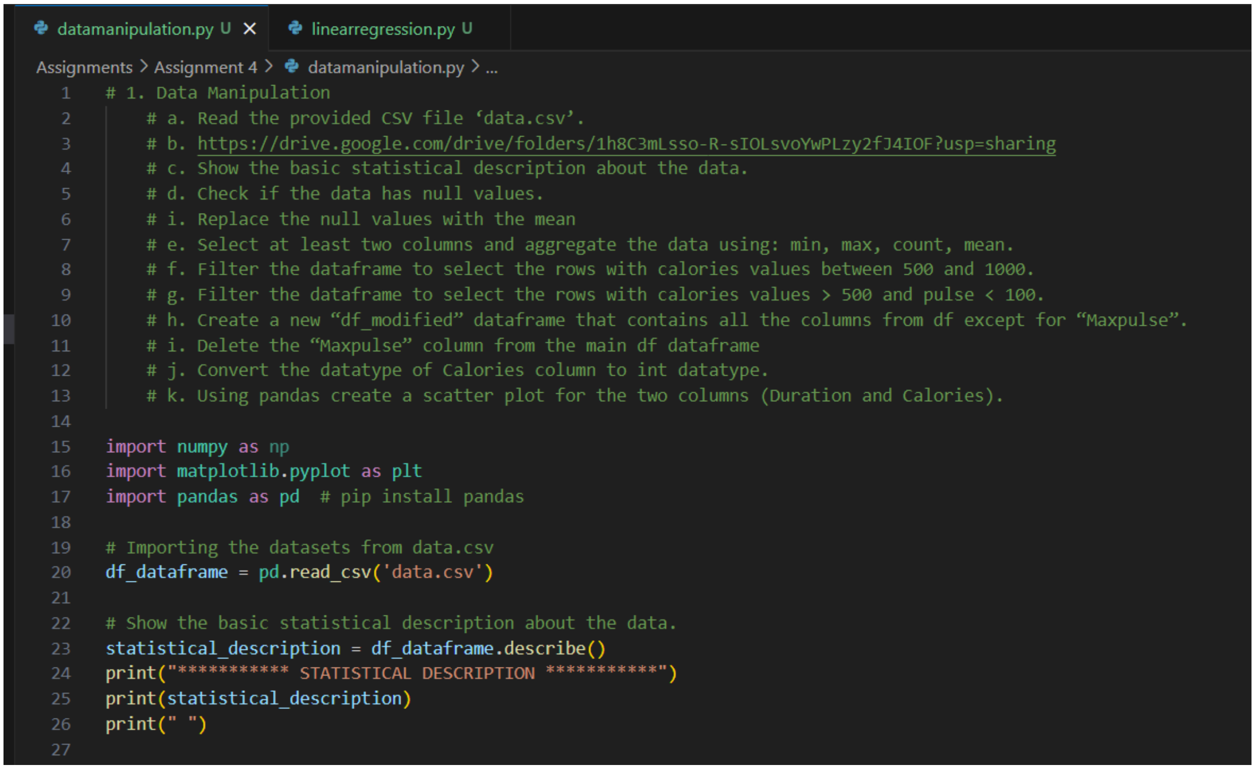
i. Delete the “Maxpulse” column from the main df dataframe

j. Convert the datatype of Calories column to int datatype.

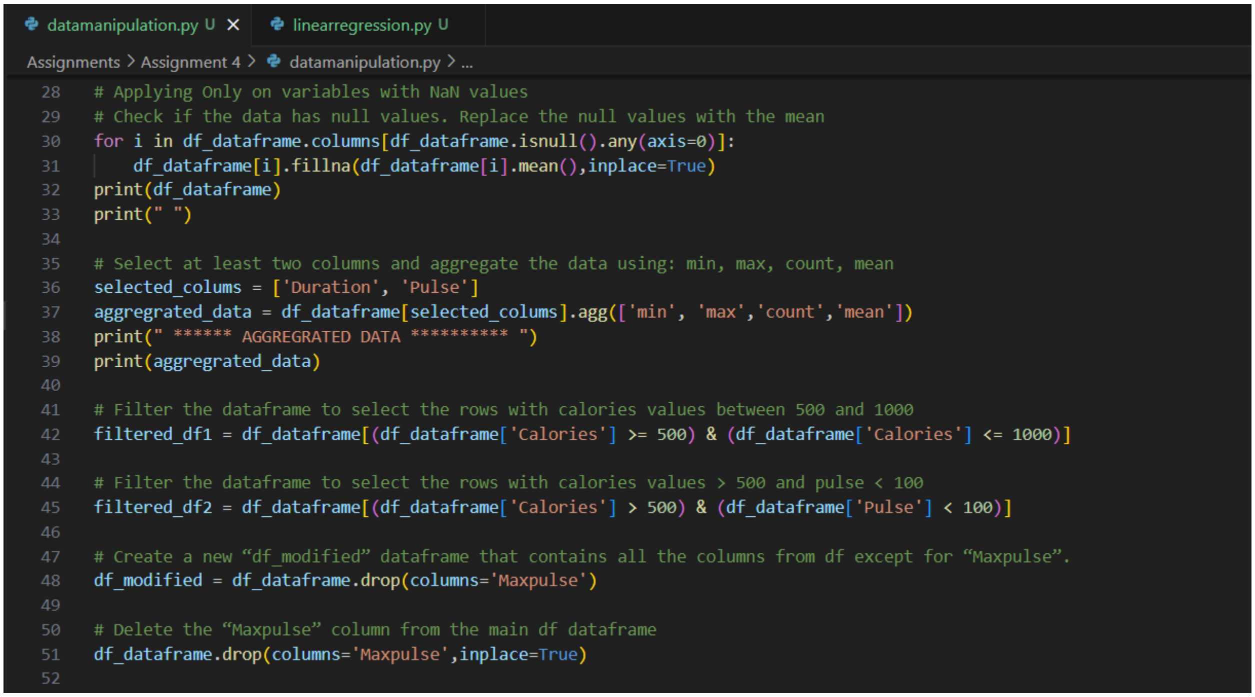
k. Using pandas create a scatter plot for the two columns (Duration and Calories).

In the below code snippet, we are doing data manipulation using Pandas:

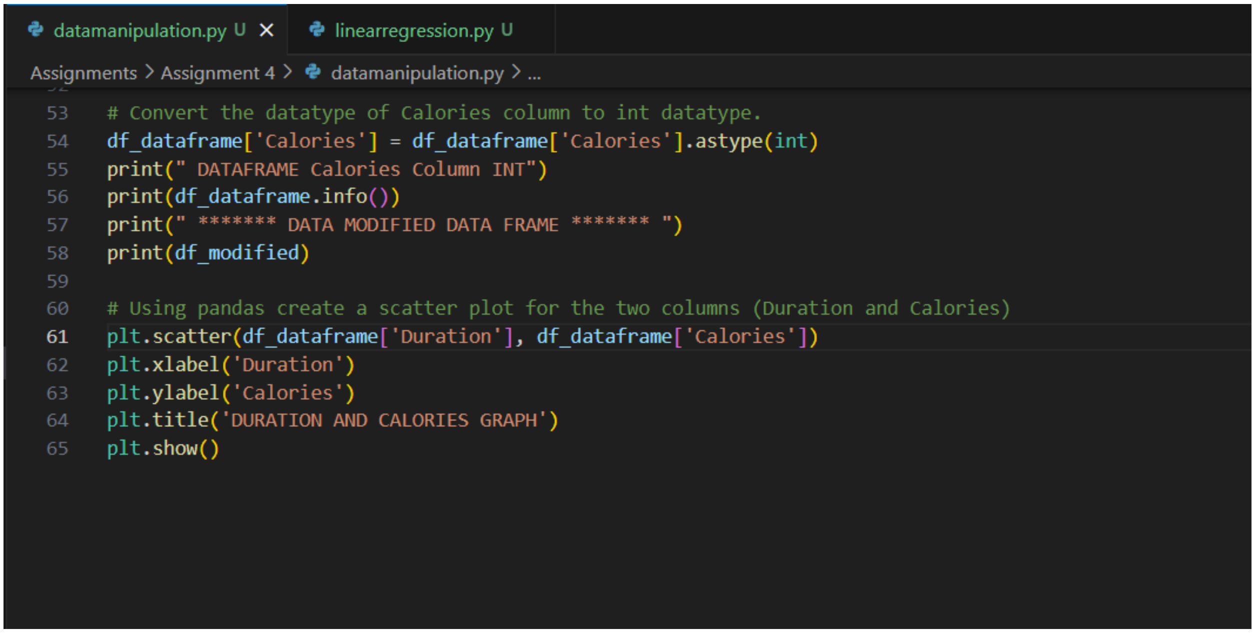
Reading the provided CSV file ‘data.csv' using the read_csv function, assigning a variable df_dataframe to it, and showing the statistical description of the dataframe using the description() function. Replacing the null values in all the columns with the mean. Selecting two columns (duration and pulse) and aggregating the data using the.agg() function and using separate variables to filter the dataframe to select the rows with calorie values between 500 and 1000 and to select the rows with calorie values > 500 and pulse <100. Next, create a new “df_modified” dataframe that contains all the columns from the dataframe except for “Maxpulse” and delete the “Maxpulse” column from the main df_dataframe dataframe. Converting the datatype of the Calories column to an int datatype using the.astype() function and finally using Pandas to create a scatter plot for the two columns (Duration and Calories).



```python
# 1. Data Manipulation
    # a. Read the provided CSV file ‘data.csv’.
    # b. https://drive.google.com/drive/folders/1h8C3mLsso-R-sIOLsvoYwPLzy2fJ4IOF?usp=sharing
    # c. Show the basic statistical description about the data.
    # d. Check if the data has null values.
    # i. Replace the null values with the mean
    # e. Select at least two columns and aggregate the data using: min, max, count, mean.
    # f. Filter the dataframe to select the rows with calories values between 500 and 1000.
    # g. Filter the dataframe to select the rows with calories values > 500 and pulse < 100.
    # h. Create a new “df_modified” dataframe that contains all the columns from df except for “Maxpulse”.
    # i. Delete the “Maxpulse” column from the main df dataframe
    # j. Convert the datatype of Calories column to int datatype.
    # k. Using pandas create a scatter plot for the two columns (Duration and Calories).

import numpy as np
import matplotlib.pyplot as plt
import pandas as pd  # pip install pandas

# Importing the datasets from data.csv
df_dataframe = pd.read_csv('data.csv')

# Show the basic statistical description about the data.
statistical_description = df_dataframe.describe()
print("*********** STATISTICAL DESCRIPTION ***********")
print(statistical_description)
print(" ")

```

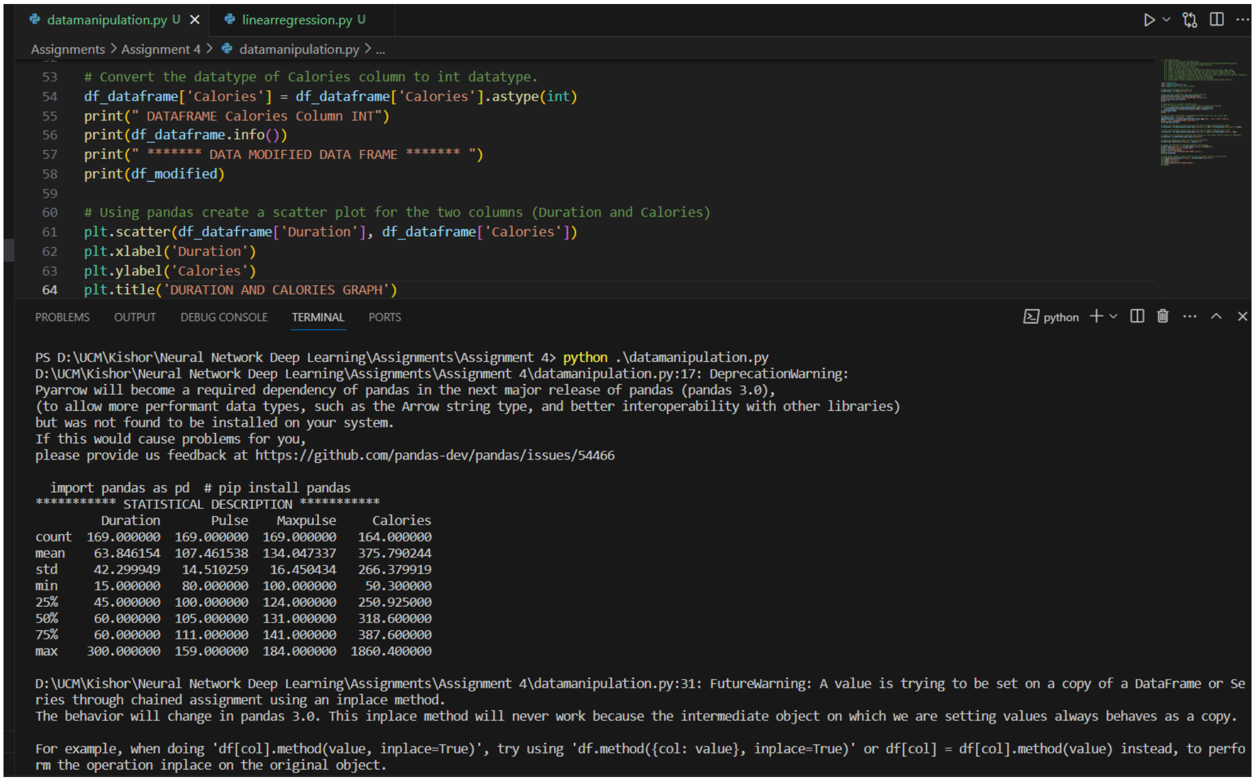


```python
# Applying Only on variables with NaN values
# Check if the data has null values. Replace the null values with the mean
for i in df_dataframe.columns[df_dataframe.isnull().any(axis=0)]:
    df_dataframe[i].fillna(df_dataframe[i].mean(),inplace=True)
print(df_dataframe)
print(" ")

# Select at least two columns and aggregate the data using: min, max, count, mean
selected_colums = ['Duration', 'Pulse']
aggregrated_data = df_dataframe[selected_colums].agg(['min', 'max','count','mean'])
print(" ****** AGGREGRATED DATA ********** ")
print(aggregrated_data)

# Filter the dataframe to select the rows with calories values between 500 and 1000
filtered_df1 = df_dataframe[(df_dataframe['Calories'] >= 500) & (df_dataframe['Calories'] <= 1000)]

# Filter the dataframe to select the rows with calories values > 500 and pulse < 100
filtered_df2 = df_dataframe[(df_dataframe['Calories'] > 500) & (df_dataframe['Pulse'] < 100)]

# Create a new “df_modified” dataframe that contains all the columns from df except for “Maxpulse”.
df_modified = df_dataframe.drop(columns='Maxpulse')

# Delete the “Maxpulse” column from the main df dataframe
df_dataframe.drop(columns='Maxpulse',inplace=True)

```



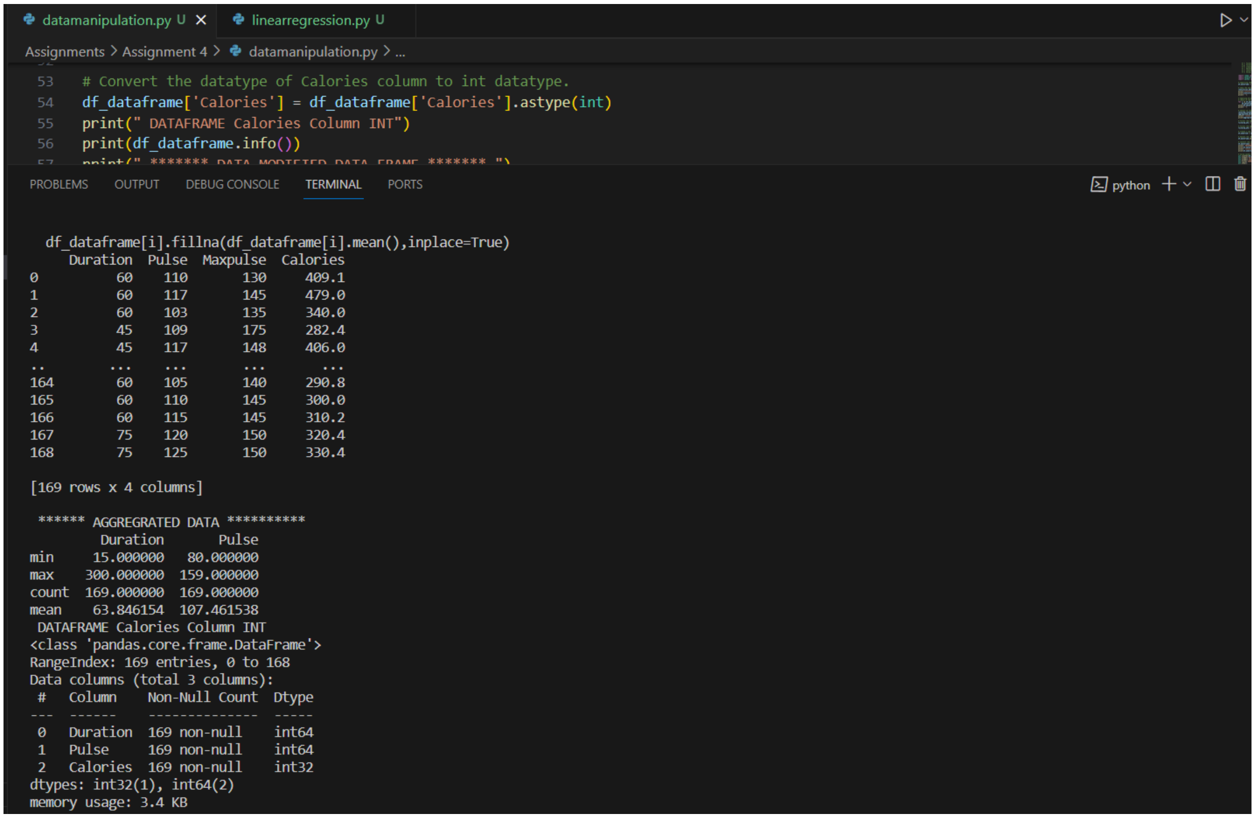
```python
# Convert the datatype of Calories column to int datatype.
df_dataframe['Calories'] = df_dataframe['Calories'].astype(int)
print(" DATAFRAME Calories Column INT")
print(df_dataframe.info())
print(" ******* DATA MODIFIED DATA FRAME ******* ")
print(df_modified)

# Using pandas create a scatter plot for the two columns (Duration and Calories)
plt.scatter(df_dataframe['Duration'], df_dataframe['Calories'])
plt.xlabel('Duration')
plt.ylabel('Calories')
plt.title('DURATION AND CALORIES GRAPH')
plt.show()
```

Output:



```
# Convert the datatype of Calories column to int datatype.
df_dataframe['Calories'] = df_dataframe['Calories'].astype(int)
print(" DATAFRAME Calories Column INT")
print(df_dataframe.info())
print(" ******* DATA MODIFIED DATA FRAME ******* ")
print(df_modified)

# Using pandas create a scatter plot for the two columns (Duration and Calories)
plt.scatter(df_dataframe['Duration'], df_dataframe['Calories'])
plt.xlabel('Duration')
plt.ylabel('Calories')
plt.title('DURATION AND CALORIES GRAPH')
```

```
PS D:\UCM\Kishor\Neural Network Deep Learning\Assignments\Assignment 4> python .\datamanipulation.py
D:\UCM\Kishor\Neural Network Deep Learning\Assignments\Assignment 4\datamanipulation.py:17: DeprecationWarning:
Pyarrow will become a required dependency of pandas in the next major release of pandas (pandas 3.0),
(to allow more performant data types, such as the Arrow string type, and better interoperability with other libraries)
but was not found to be installed on your system.
If this would cause problems for you,
please provide us feedback at https://github.com/pandas-dev/pandas/issues/54466

  import pandas as pd  # pip install pandas
*********** STATISTICAL DESCRIPTION ***********
         Duration       Pulse    Maxpulse     Calories
count  169.000000  169.000000  169.000000   164.000000
mean    63.846154  107.461538  134.047337   375.790244
std     42.299949   14.510259   16.450434   266.379919
min     15.000000   80.000000  100.000000    50.300000
25%     45.000000  100.000000  124.000000   250.925000
50%     60.000000  105.000000  131.000000   318.600000
75%     60.000000  111.000000  141.000000   387.600000
max    300.000000  159.000000  184.000000  1860.400000

D:\UCM\Kishor\Neural Network Deep Learning\Assignments\Assignment 4\datamanipulation.py:31: FutureWarning: A value is trying to be set on a copy of a DataFrame or Se
ries through chained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting values always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value) instead, to perfo
rm the operation inplace on the original object.
```



```
# Convert the datatype of Calories column to int datatype.
df_dataframe['Calories'] = df_dataframe['Calories'].astype(int)
print(" DATAFRAME Calories Column INT")
print(df_dataframe.info())
print(" ******* DATA MODIFIED DATA FRAME ******* ")
```

```
  df_dataframe[i].fillna(df_dataframe[i].mean(),inplace=True)
     Duration  Pulse  Maxpulse  Calories
0          60    110       130     409.1
1          60    117       145     479.0
2          60    103       135     340.0
3          45    109       175     282.4
4          45    117       148     406.0
..        ...    ...       ...       ...
164        60    105       140     290.8
165        60    110       145     300.0
166        60    115       145     310.2
167        75    120       150     320.4
168        75    125       150     330.4

[169 rows x 4 columns]

 ****** AGGREGRATED DATA **********
         Duration       Pulse
min     15.000000   80.000000
max    300.000000  159.000000
count  169.000000  169.000000
mean    63.846154  107.461538
 DATAFRAME Calories Column INT
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 169 entries, 0 to 168
Data columns (total 3 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   Duration  169 non-null    int64
 1   Pulse     169 non-null    int64
 2   Calories  169 non-null    int32
dtypes: int32(1), int64(2)
memory usage: 3.4 KB
```

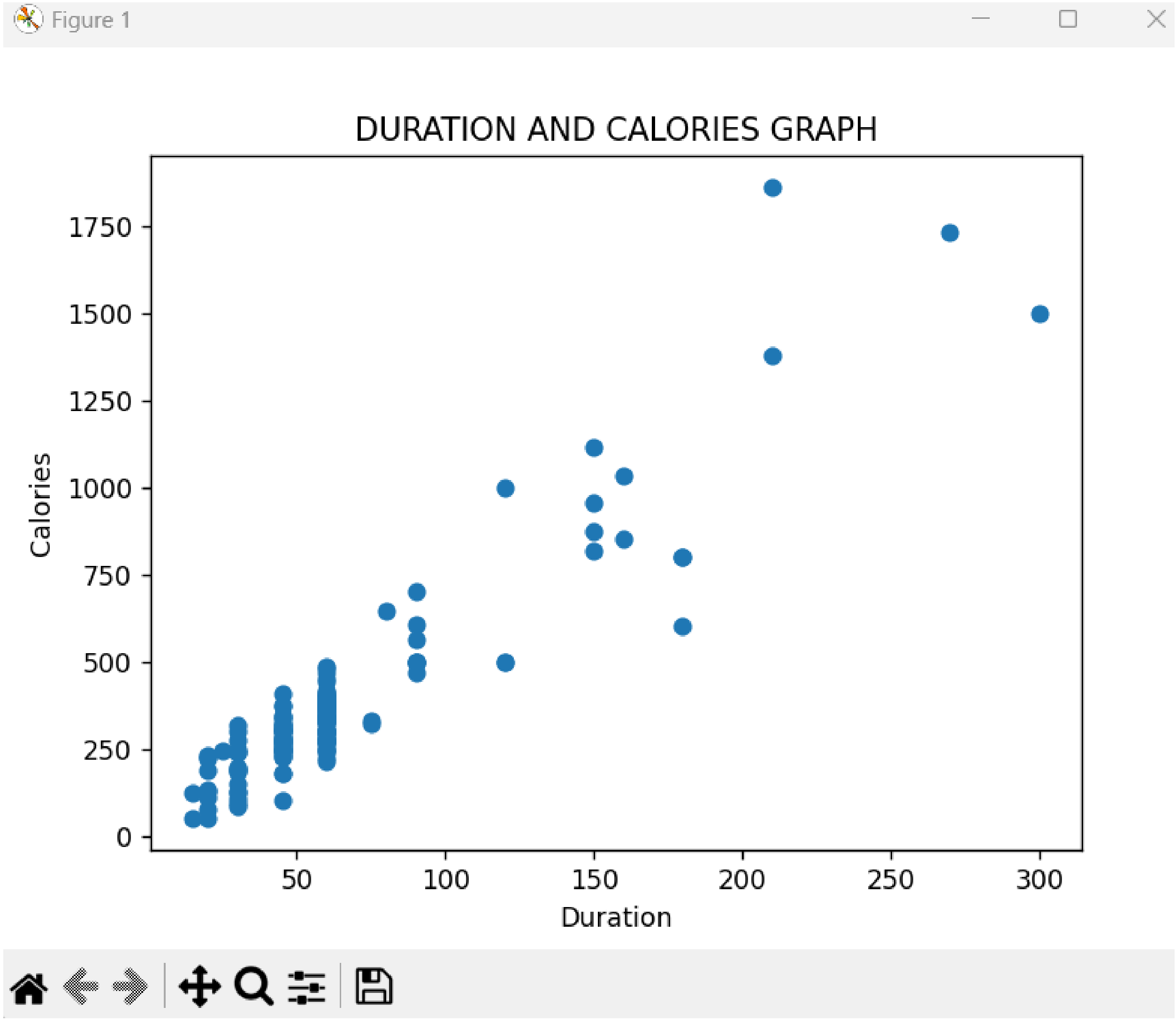
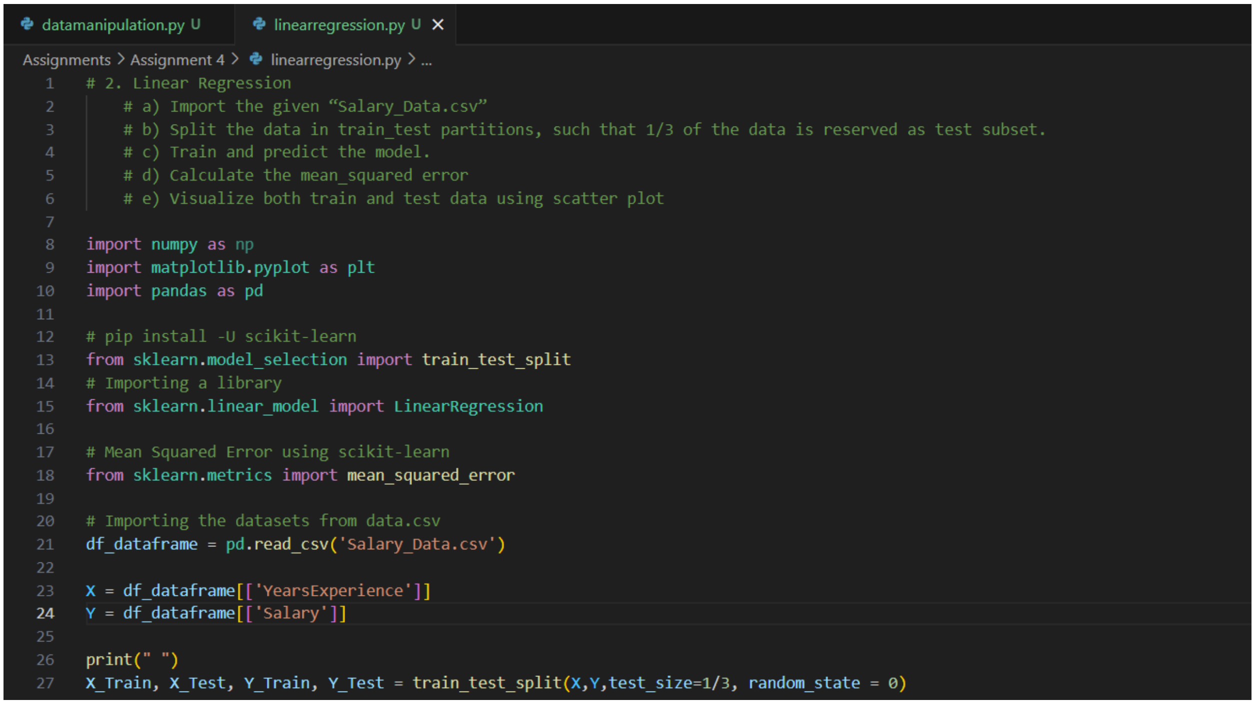


Figure 1
DURATION AND CALORIES GRAPH
Calories
0
250
500
750
1000
1250
1500
1750
Duration
50
100
150
200
250
300

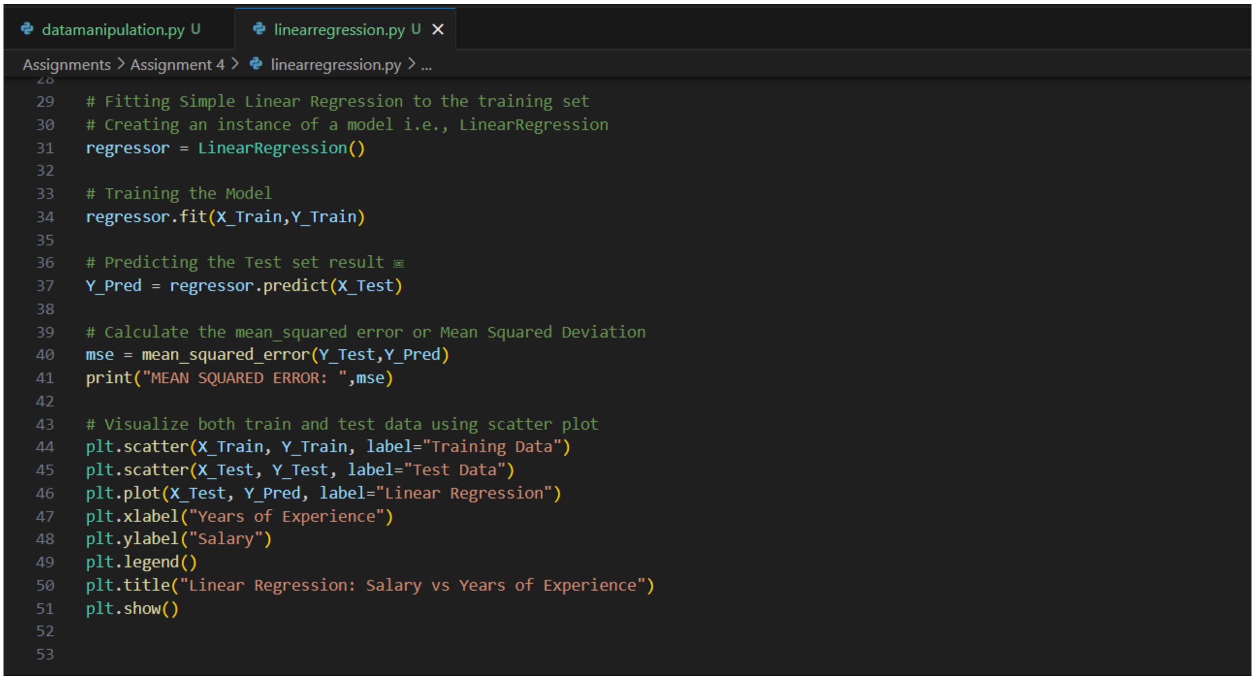
2. Linear Regression
   a) Import the given “Salary_Data.csv”
   b) Split the data in train_test partitions, such that 1/3 of the data is reserved as test subset.
   c) Train and predict the model.
   d) Calculate the mean_squared error
   e) Visualize both train and test data using scatter plot

In the below code snippet, we are training data using linear regression:

Importing the given “Salary_Data.csv” into a variable called df_dataframe. Splitting the data using the train_test_split() function, such that 1/3 of the data is reserved as a test subset. Train the model using the LinearRegression() function, predict the values using the predict() function, calculate the mean squared error of the predicted set, and use Pandas to visualize both the train and test data using a scatter plot.

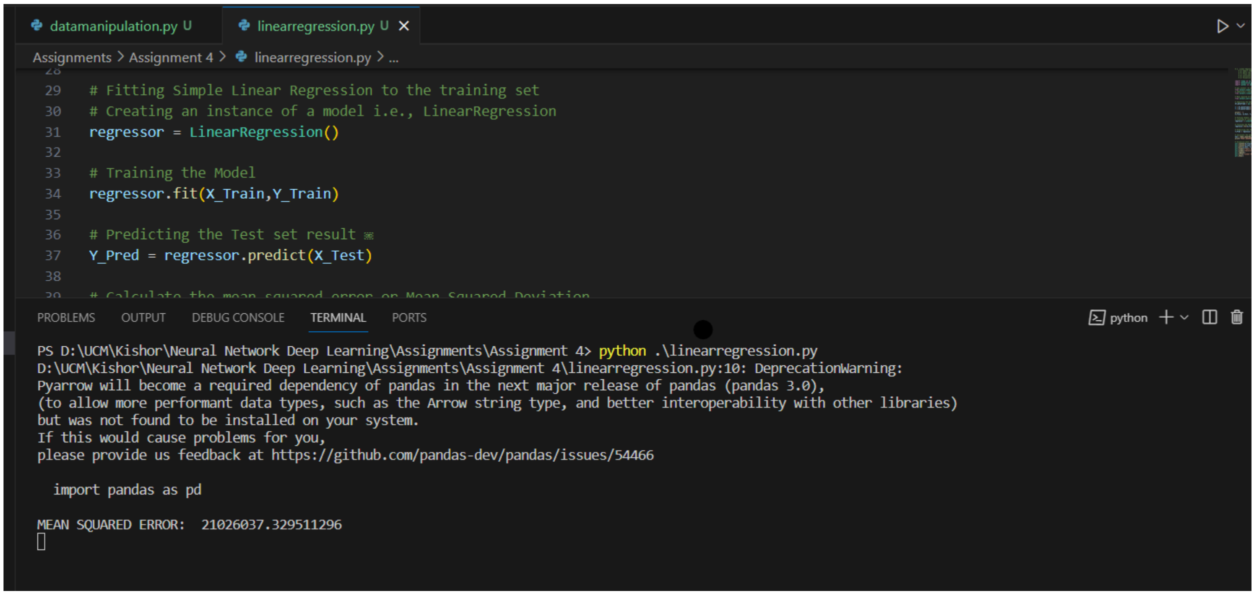


```
# 2. Linear Regression
    # a) Import the given “Salary_Data.csv”
    # b) Split the data in train_test partitions, such that 1/3 of the data is reserved as test subset.
    # c) Train and predict the model.
    # d) Calculate the mean_squared error
    # e) Visualize both train and test data using scatter plot

import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

# pip install -U scikit-learn
from sklearn.model_selection import train_test_split
# Importing a library
from sklearn.linear_model import LinearRegression

# Mean Squared Error using scikit-learn
from sklearn.metrics import mean_squared_error

# Importing the datasets from data.csv
df_dataframe = pd.read_csv('Salary_Data.csv')

X = df_dataframe[['YearsExperience']]
Y = df_dataframe[['Salary']]

print(" ")
X_Train, X_Test, Y_Train, Y_Test = train_test_split(X,Y,test_size=1/3, random_state = 0)
```



```python
# Fitting Simple Linear Regression to the training set
# Creating an instance of a model i.e., LinearRegression
regressor = LinearRegression()

# Training the Model
regressor.fit(X_Train,Y_Train)

# Predicting the Test set result
Y_Pred = regressor.predict(X_Test)

# Calculate the mean_squared error or Mean Squared Deviation
mse = mean_squared_error(Y_Test,Y_Pred)
print("MEAN SQUARED ERROR: ",mse)

# Visualize both train and test data using scatter plot
plt.scatter(X_Train, Y_Train, label="Training Data")
plt.scatter(X_Test, Y_Test, label="Test Data")
plt.plot(X_Test, Y_Pred, label="Linear Regression")
plt.xlabel("Years of Experience")
plt.ylabel("Salary")
plt.legend()
plt.title("Linear Regression: Salary vs Years of Experience")
plt.show()


```

Output:



```python
# Fitting Simple Linear Regression to the training set
# Creating an instance of a model i.e., LinearRegression
regressor = LinearRegression()

# Training the Model
regressor.fit(X_Train,Y_Train)

# Predicting the Test set result
Y_Pred = regressor.predict(X_Test)

# Calculate the mean squared error or Mean Squared Deviation
```

```
PS D:\UCM\Kishor\Neural Network Deep Learning\Assignments\Assignment 4> python .\linearregression.py
D:\UCM\Kishor\Neural Network Deep Learning\Assignments\Assignment 4\linearregression.py:10: DeprecationWarning:
Pyarrow will become a required dependency of pandas in the next major release of pandas (pandas 3.0),
(to allow more performant data types, such as the Arrow string type, and better interoperability with other libraries)
but was not found to be installed on your system.
If this would cause problems for you,
please provide us feedback at https://github.com/pandas-dev/pandas/issues/54466

  import pandas as pd

MEAN SQUARED ERROR:  21026037.329511296
```

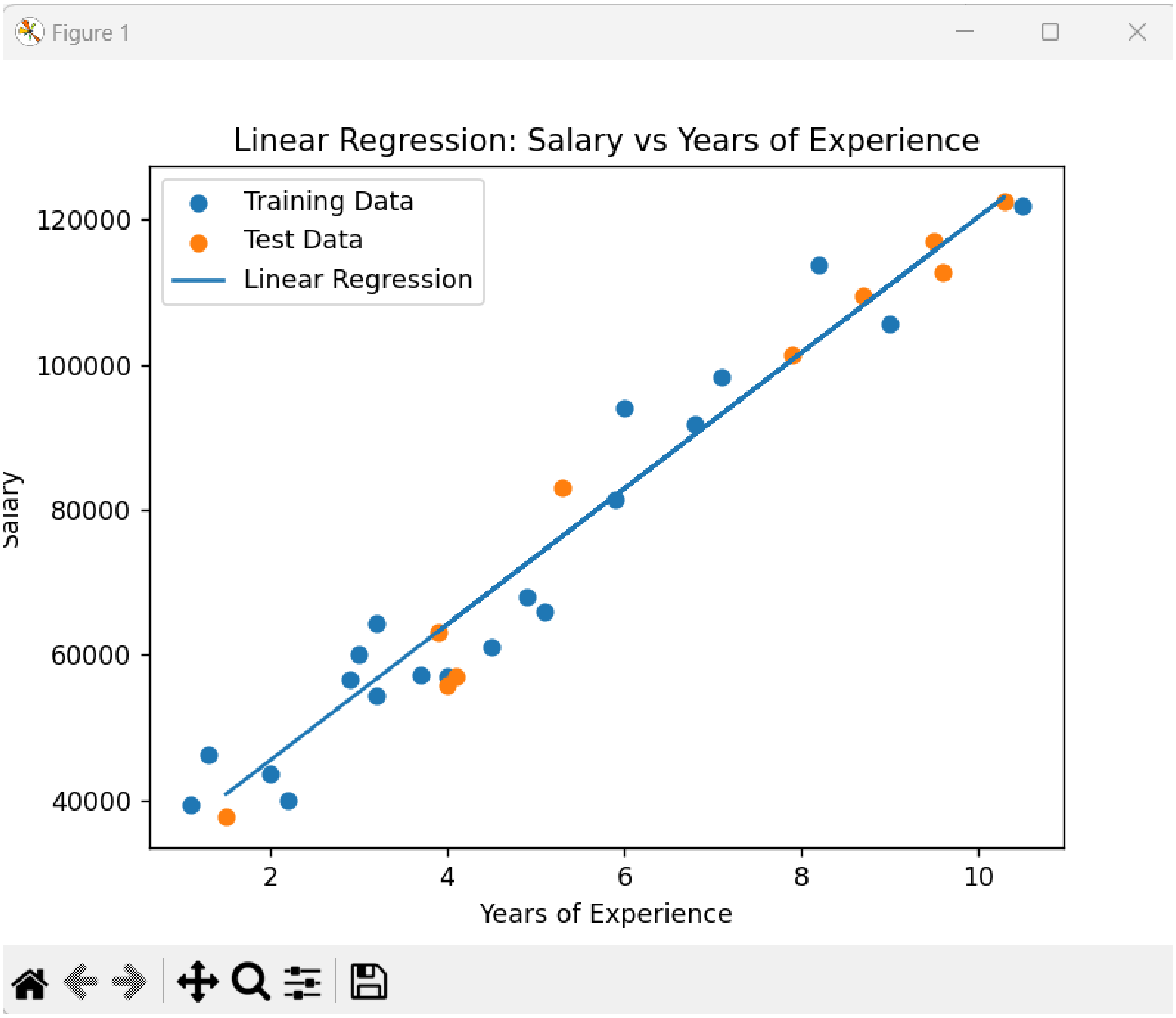


Figure 1
Linear Regression: Salary vs Years of Experience
● Training Data
● Test Data
Linear Regression
120000
100000
80000
60000
40000
Salary
2
4
6
8
10
Years of Experience