ICP – 4

Naga Phaneendra Kumara Gupta Mogili

700757977

GitHub link: https://github.com/nagaphaneendra2001/Deep_Learning_Neural_Networks.git

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).

Source Code:

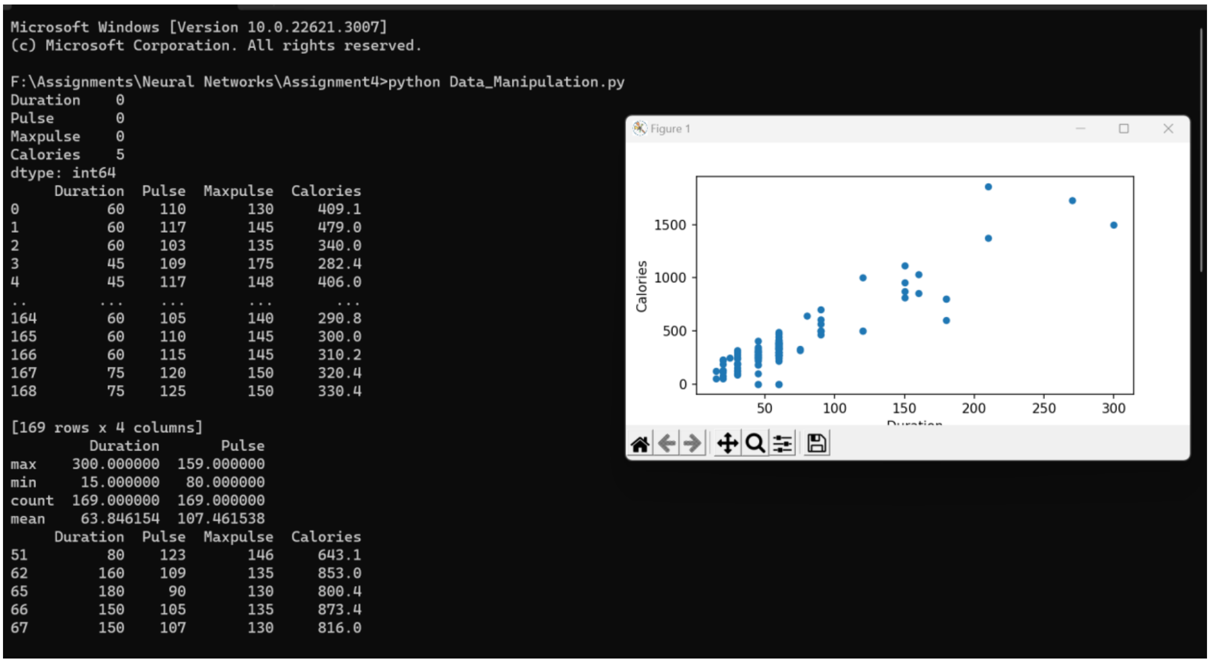
```
import pandas as pandas
import pandas as pd

# Reading CSV file
data_value = pd.read_csv("data.csv")

# Statistical description
data_value.describe()

#Checking for null values
null_values = data_value.isnull().sum()
print(null_values)
```

```
#replacing nullvalues with the mean
data_value.fillna(data_value.mean(), inplace=True)
print(data_value)

#Selecting two columns
data_value = data_value[["Duration", "Pulse" ]]

#Aggregating the data
agg_dict = {"Duration": ["max", "min", "count", "mean"],
            "Pulse": ["max", "min", "count", "mean"]}
agg_data_value = data_value.agg(agg_dict)
print(agg_data_value)

# Filtering the dataframe to select the rows with calories values between 500 and 1000.
data_value = pd.read_csv("data.csv")
Calories_filter = (data_value["Calories"] >= 500) & (data_value["Calories"] <= 1000)
filtered_data_value = data_value[Calories_filter]
print(filtered_data_value)

#Filtering the dataframe to select the rows with calories values > 500 and pulse < 100.
data_value = pd.read_csv("data.csv")
Calories_filter = (data_value["Calories"] > 500) & (data_value["Pulse"] < 100)
filtered_data_value = data_value[Calories_filter]
print(filtered_data_value)

#Creating a new “df_modified” dataframe that contains all the columns from df except for “Maxpulse” and deleting that maxpulse column from the main df fataframe
df_modified = data_value.drop(columns=["Maxpulse"])
print(df_modified)

# Converting the datatype of calories column to int datatype
data_value['Calories'] = data_value['Calories'].fillna(0).astype(int)
print(data_value)

# Plotting the output
```

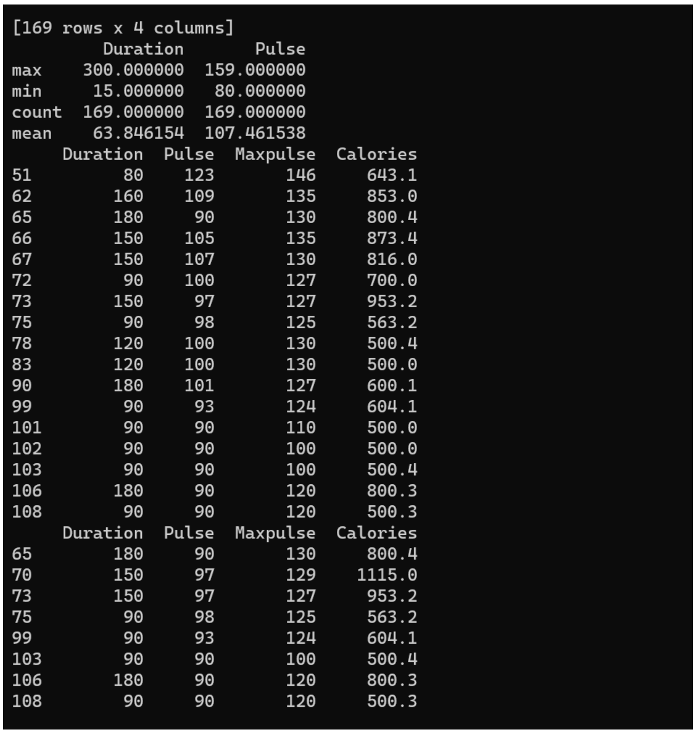
import matplotlib.pyplot as plt

data_value.plot(kind='scatter', x='Duration', y='Calories', figsize=(6,3))
plt.show()

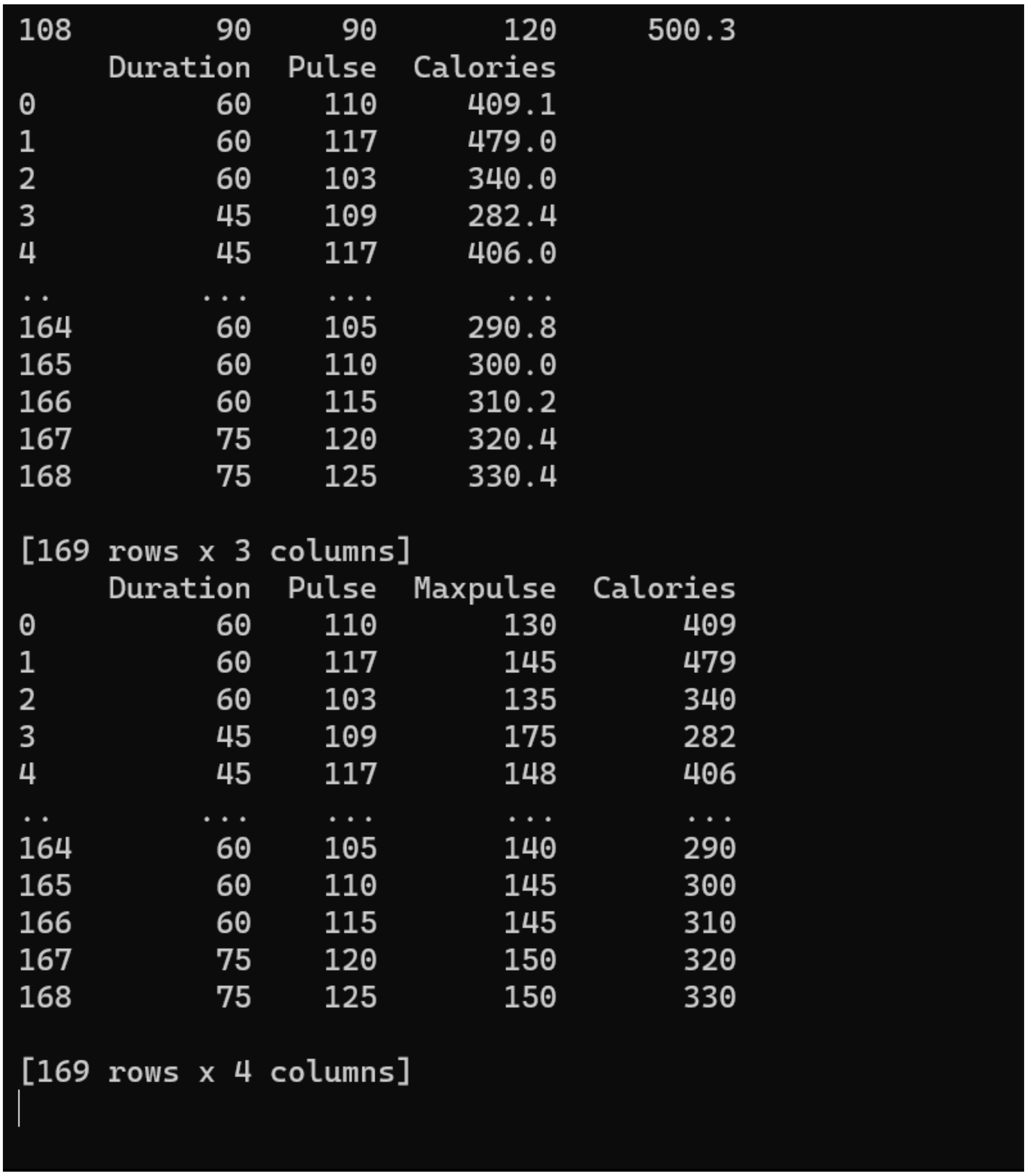
Output:



```
Microsoft Windows [Version 10.0.22621.3007]
(c) Microsoft Corporation. All rights reserved.

F:\Assignments\Neural Networks\Assignment4>python Data_Manipulation.py
Duration     0
Pulse        0
Maxpulse     0
Calories     5
dtype: int64
     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]
         Duration       Pulse
max    300.000000  159.000000
min     15.000000   80.000000
count  169.000000  169.000000
mean    63.846154  107.461538
    Duration  Pulse  Maxpulse  Calories
51        80    123       146     643.1
62       160    109       135     853.0
65       180     90       130     800.4
66       150    105       135     873.4
67       150    107       130     816.0
```



```
[169 rows x 4 columns]
         Duration       Pulse
max    300.000000  159.000000
min     15.000000   80.000000
count  169.000000  169.000000
mean    63.846154  107.461538
     Duration  Pulse  Maxpulse  Calories
51         80    123       146     643.1
62        160    109       135     853.0
65        180     90       130     800.4
66        150    105       135     873.4
67        150    107       130     816.0
72         90    100       127     700.0
73        150     97       127     953.2
75         90     98       125     563.2
78        120    100       130     500.4
83        120    100       130     500.0
90        180    101       127     600.1
99         90     93       124     604.1
101        90     90       110     500.0
102        90     90       100     500.0
103        90     90       100     500.4
106       180     90       120     800.3
108        90     90       120     500.3
     Duration  Pulse  Maxpulse  Calories
65        180     90       130     800.4
70        150     97       129    1115.0
73        150     97       127     953.2
75         90     98       125     563.2
99         90     93       124     604.1
103        90     90       100     500.4
106       180     90       120     800.3
108        90     90       120     500.3
```



```
108        90     90        120      500.3
     Duration  Pulse  Calories
0          60    110     409.1
1          60    117     479.0
2          60    103     340.0
3          45    109     282.4
4          45    117     406.0
..        ...    ...       ...
164        60    105     290.8
165        60    110     300.0
166        60    115     310.2
167        75    120     320.4
168        75    125     330.4

[169 rows x 3 columns]
     Duration  Pulse  Maxpulse  Calories
0          60    110       130       409
1          60    117       145       479
2          60    103       135       340
3          45    109       175       282
4          45    117       148       406
..        ...    ...       ...       ...
164        60    105       140       290
165        60    110       145       300
166        60    115       145       310
167        75    120       150       320
168        75    125       150       330

[169 rows x 4 columns]
```

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.

Source Code:

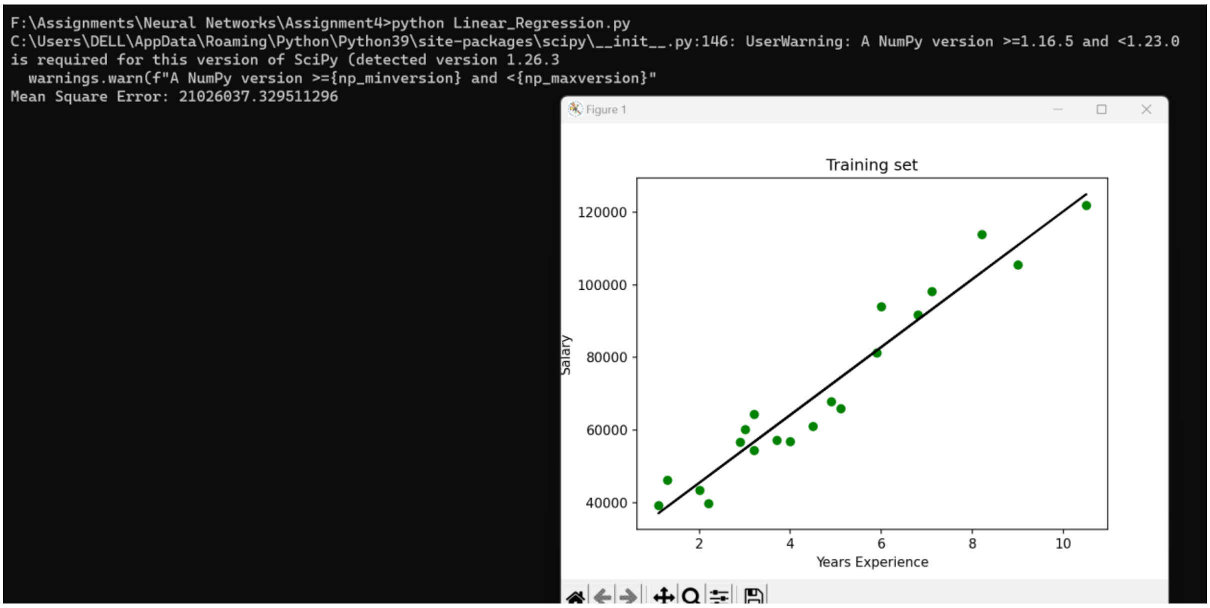
# Simple Linear Regression

# Importing the libraries

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import sklearn as sk

#Importing the datasets
dataset = pd.read_csv("Salary_Data.csv")

X = dataset.iloc[:, :-1].values
y = dataset.iloc[:, 1]

# Splitting the dataset into the Training set and Test set

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=1/3, random_state=0)

# Fitting Simple Linear Regression to the training set

from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(X_train, y_train)

# Predicting the Test set result

y_pred = regressor.predict(X_test)

from sklearn.metrics import mean_squared_error
meansquareerror = mean_squared_error(y_true=y_test,y_pred=y_pred)
```

```
print("Mean Square Error:",meansquareerror)

# Visualizing the training set results

viz_train = plt
viz_train.scatter(X_train, y_train, color='green')
viz_train.plot(X_train, regressor.predict(X_train), color='black')
viz_train.title('Training set')
viz_train.xlabel('Years Experience')
viz_train.ylabel('Salary')
viz_train.show()

viz_test = plt
viz_test.scatter(X_test, y_test, color='green')
viz_test.plot(X_train, regressor.predict(X_train), color='black')
viz_test.title('Test set')
viz_test.xlabel('Years Experience')
viz_test.ylabel('Salary')
viz_test.show()
```

Output:



```
F:\Assignments\Neural Networks\Assignment4>python Linear_Regression.py
C:\Users\DELL\AppData\Roaming\Python\Python39\site-packages\scipy\__init__.py:146: UserWarning: A NumPy version >=1.16.5 and <1.23.0
is required for this version of SciPy (detected version 1.26.3
  warnings.warn(f"A NumPy version >={np_minversion} and <{np_maxversion}"
Mean Square Error: 21026037.329511296
```

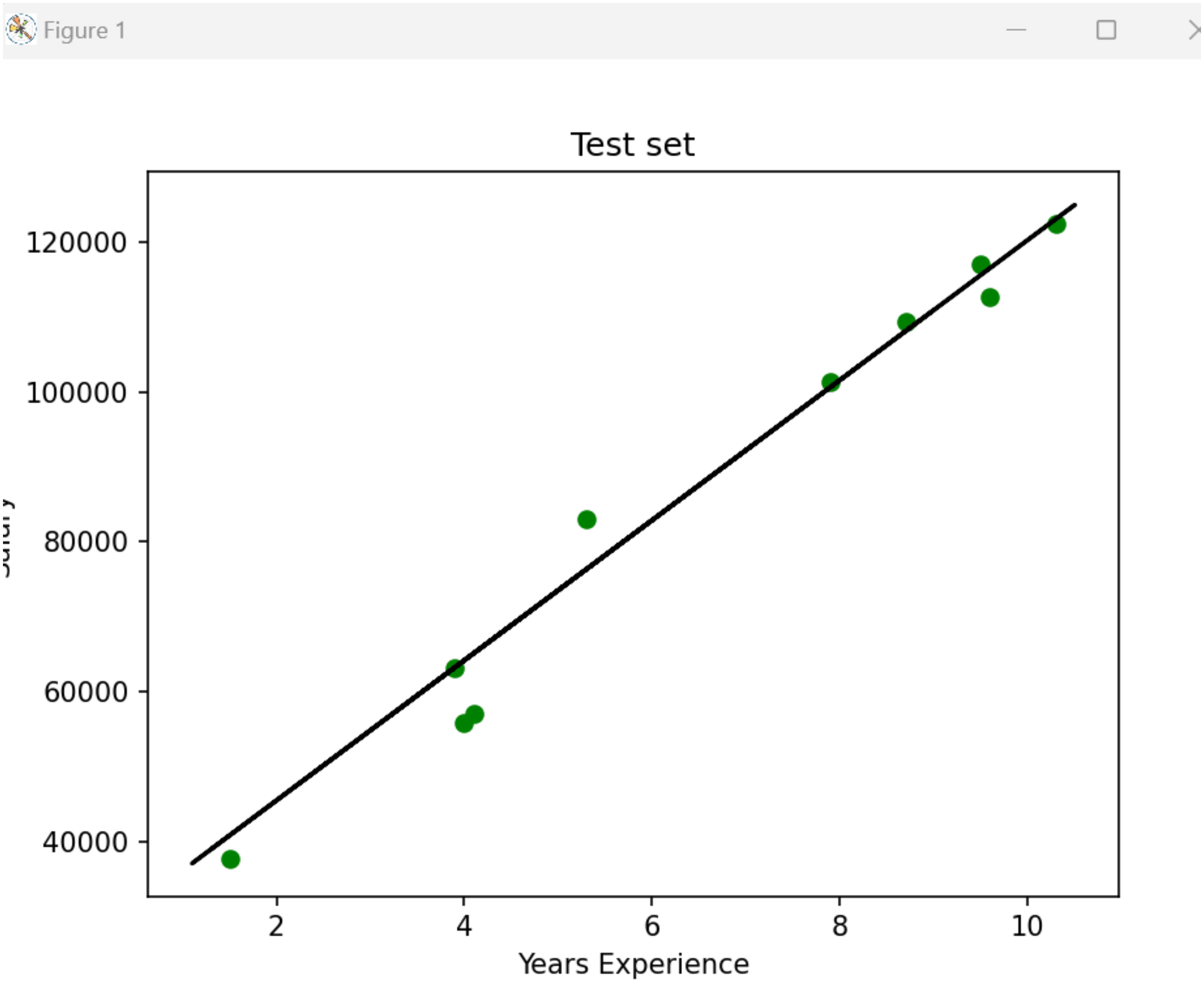


Figure 1
Test set
Years Experience