Spring 2023: CS5720 – NN &DL
In-Class Programming Assignment-2

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.

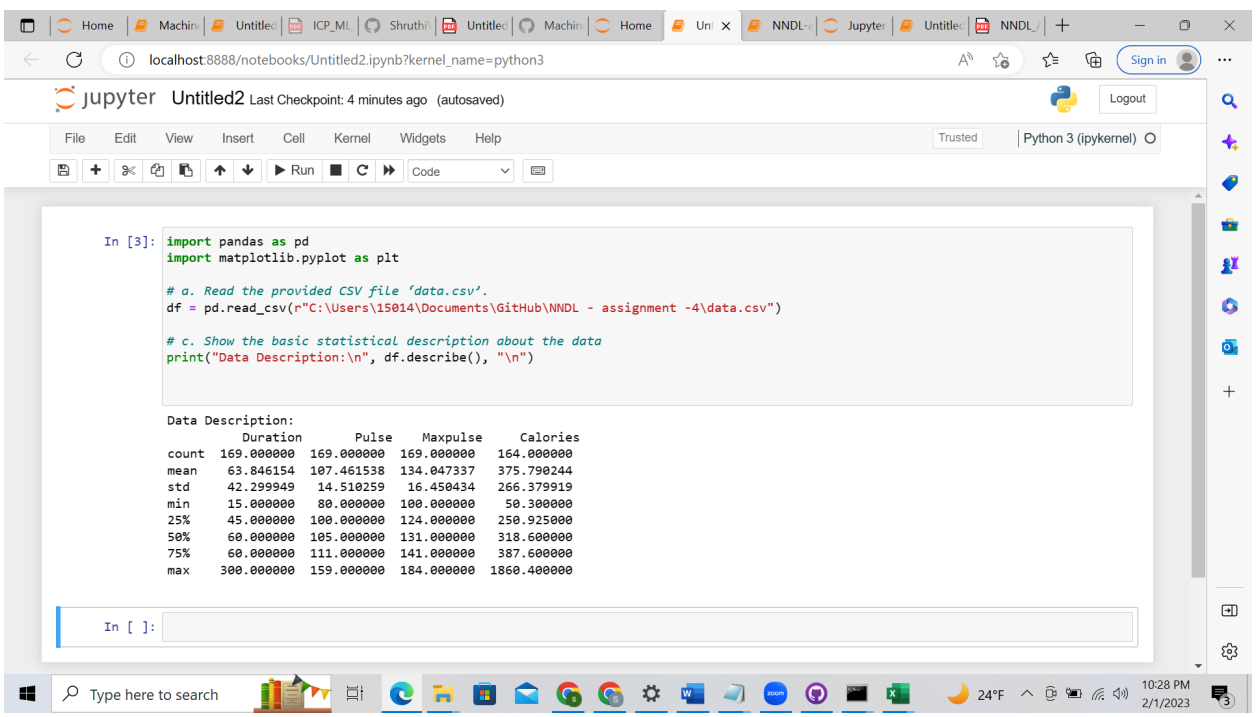
h. Create a new “df_modified” dataframe that contains all the columns from df except for “Maxpulse”.

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

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

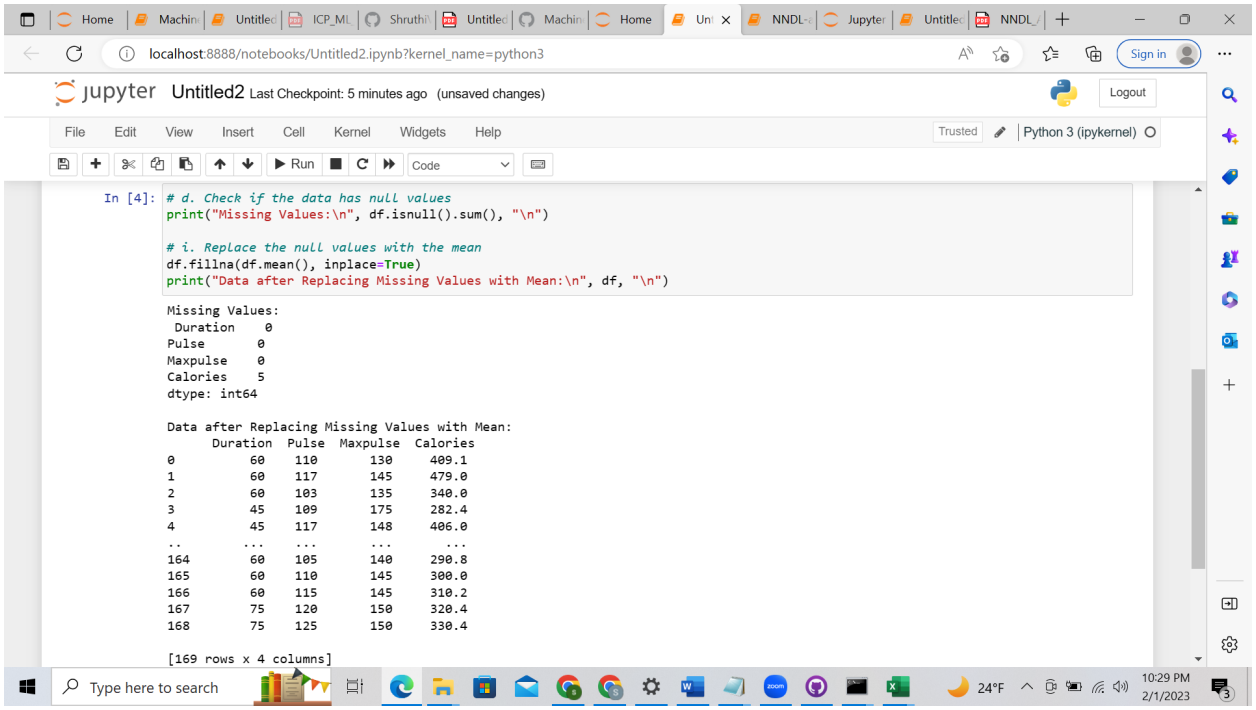
k. Using pandas create a scatter plot for the two columns (Duration and Calories).

Ans



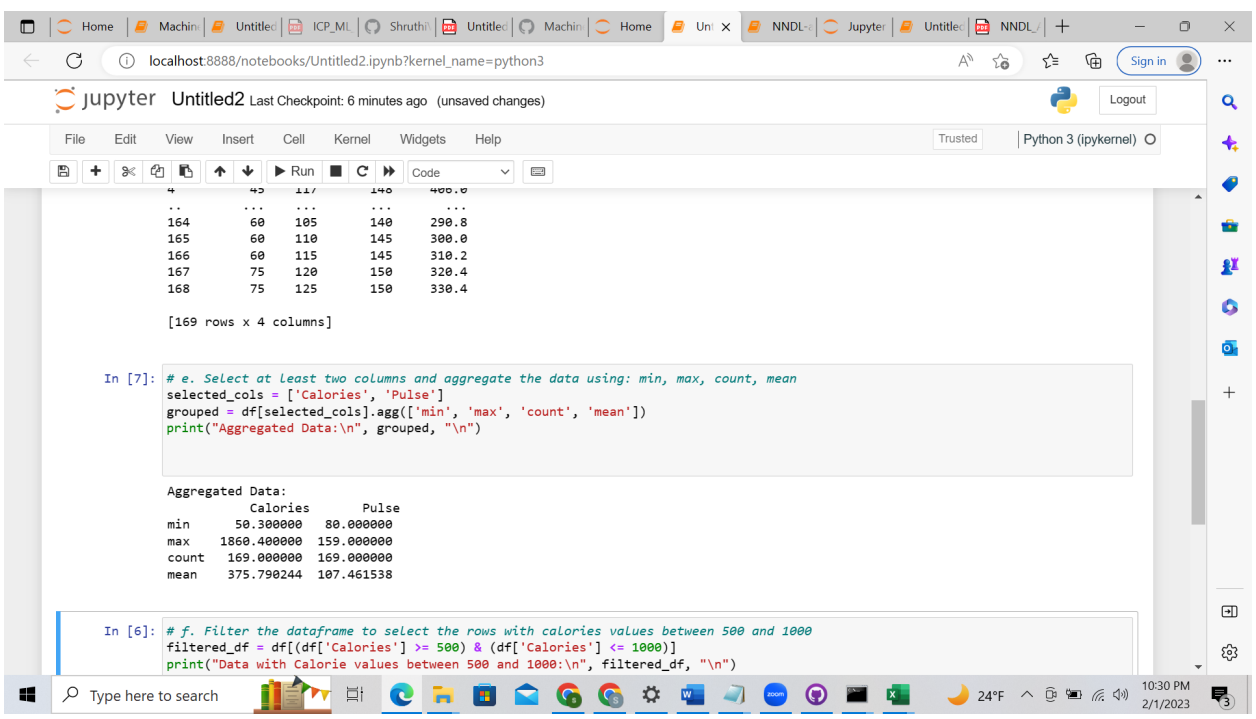
a) Read the data from data.csv at the path C:\Users\15014\Documents\GitHub\NNDL - assignment -4\data.csv

c) Using describe() method got the data description & printed the data description.



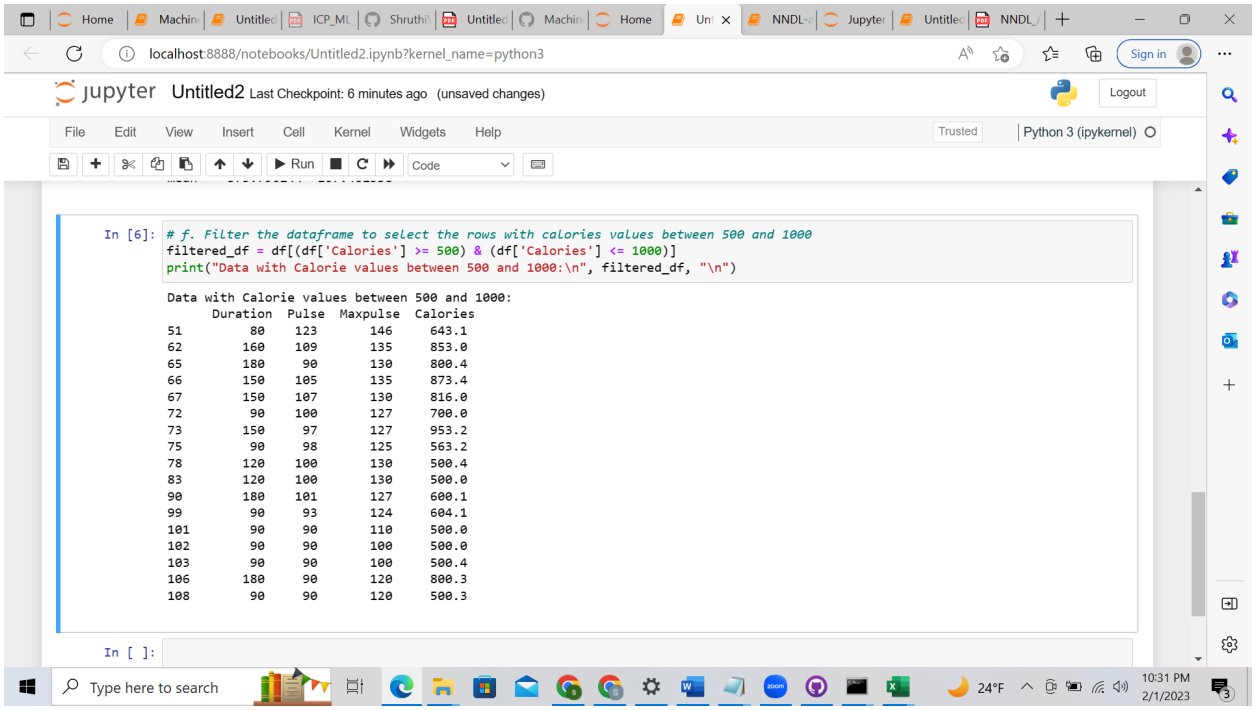
```
In [4]: # d. Check if the data has null values
        print("Missing Values:\n", df.isnull().sum(), "\n")

        # i. Replace the null values with the mean
        df.fillna(df.mean(), inplace=True)
        print("Data after Replacing Missing Values with Mean:\n", df, "\n")

Missing Values:
 Duration    0
Pulse       0
Maxpulse    0
Calories    5
dtype: int64

Data after Replacing Missing Values with Mean:
      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]
```

d) Checked if data has null values and replaced the null values with the mean using .fillna() method



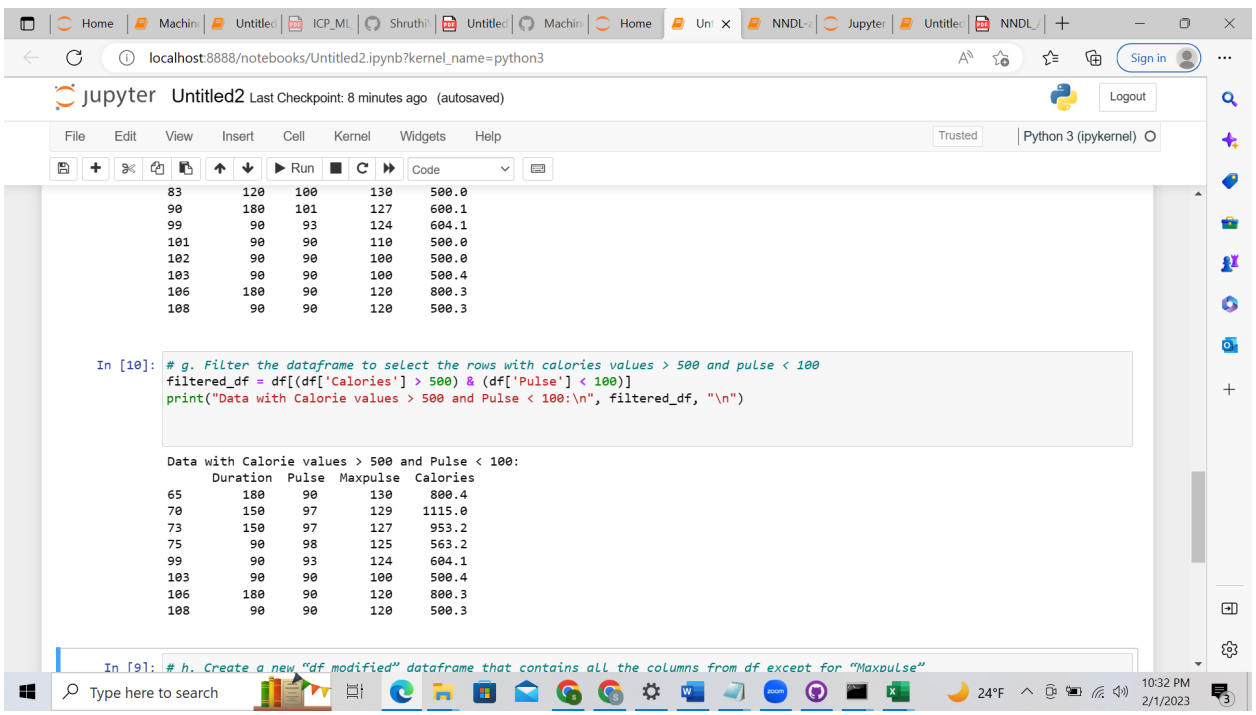
```
164         60    105       140     290.8
165         60    110       145     300.0
166         60    115       145     310.2
167         75    120       150     320.4
168         75    125       150     330.4

[169 rows x 4 columns]

In [7]: # e. Select at least two columns and aggregate the data using: min, max, count, mean
        selected_cols = ['Calories', 'Pulse']
        grouped = df[selected_cols].agg(['min', 'max', 'count', 'mean'])
        print("Aggregated Data:\n", grouped, "\n")

Aggregated Data:
            Calories       Pulse
min       50.300000   80.000000
max     1860.400000  159.000000
count    169.000000  169.000000
mean     375.790244  107.461538

In [6]: # f. Filter the dataframe to select the rows with calories values between 500 and 1000
        filtered_df = df[(df['Calories'] >= 500) & (df['Calories'] <= 1000)]
        print("Data with Calorie values between 500 and 1000:\n", filtered_df, "\n")
```

e) Selected the two columns and aggregated the data using min , max,count , mean using .agg() method



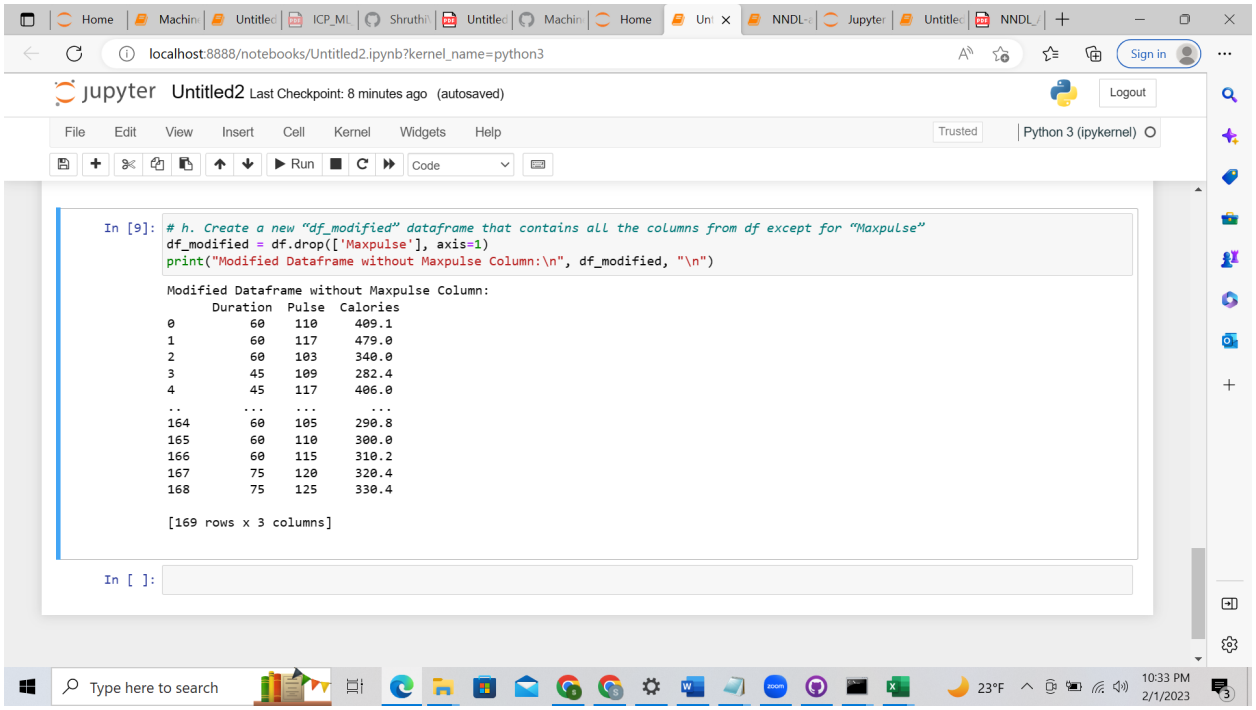
```
In [6]: # f. Filter the dataframe to select the rows with calories values between 500 and 1000
        filtered_df = df[(df['Calories'] >= 500) & (df['Calories'] <= 1000)]
        print("Data with Calorie values between 500 and 1000:\n", filtered_df, "\n")

Data with Calorie values between 500 and 1000:
      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

In [ ]:
```

f) Filtered the data frame to select the rows with calories values between 500 & 1000

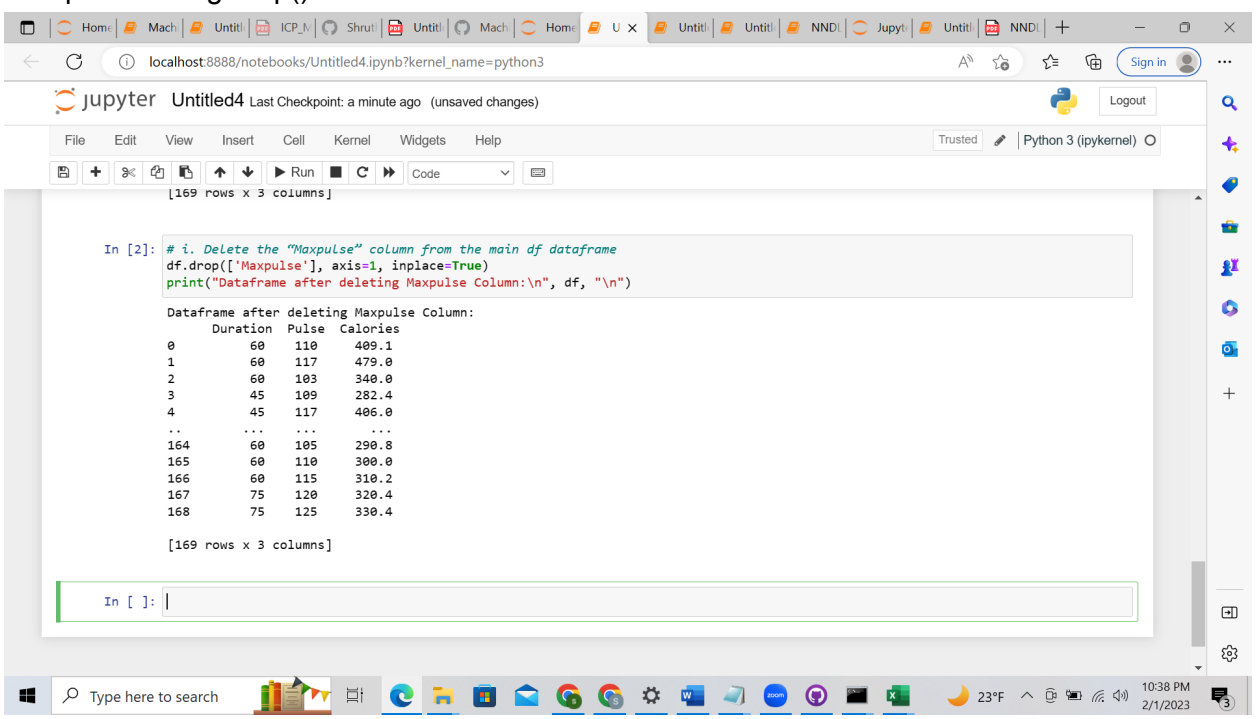


```
83        120    100       130     500.0
90        180    101       127     600.1
99         90     93       124     604.1
101        90     90       110     500.0
102        90     90       100     500.0
103        90     90       100     500.4
106       180     90       120     800.3
108        90     90       120     500.3

In [10]: # g. Filter the dataframe to select the rows with calories values > 500 and pulse < 100
         filtered_df = df[(df['Calories'] > 500) & (df['Pulse'] < 100)]
         print("Data with Calorie values > 500 and Pulse < 100:\n", filtered_df, "\n")

Data with Calorie values > 500 and Pulse < 100:
      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

In [9]: # h. Create a new "df_modified" dataframe that contains all the columns from df except for "Maxpulse"
```

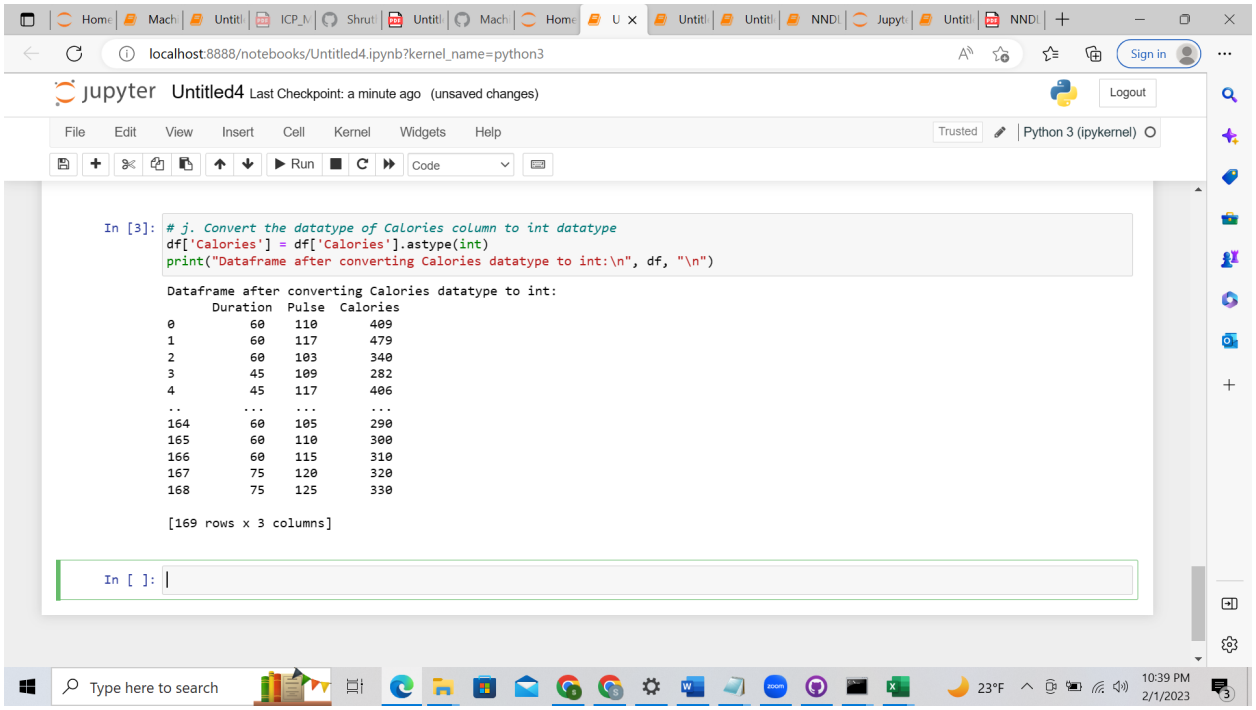
g) Filtered the data frame to select the rows with calories values > 500 & pulse < 100



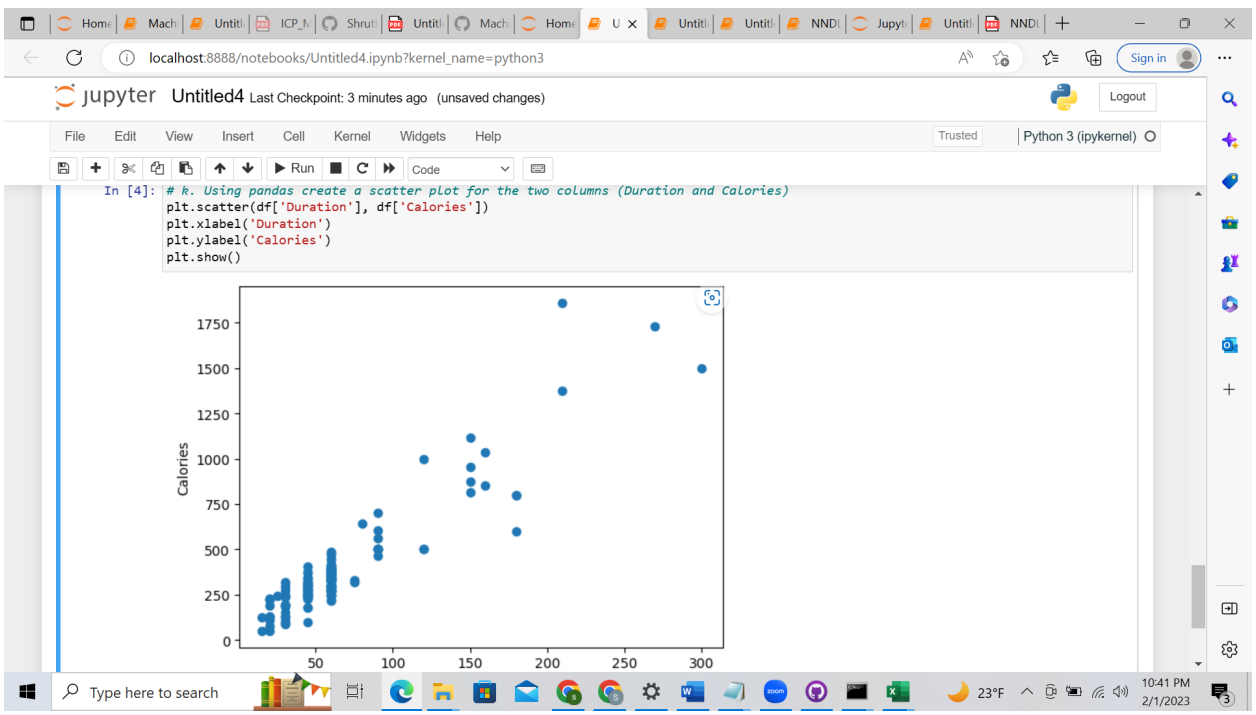
'h) created the df_modified dataframe that contains all the columns from df except for "Maxpulse" using drop() function.



i) Deleted the Maxpulse column from the main df dataframe and printed the output using drop() function.



j) converted the datatype of calories column to int datatype using astype() function



k) Using pandas created a scatter plot for the 2 columns Duration & calories using Scatter() function

2. Linear Regression

a) Import the given “Salary_Data.csv”

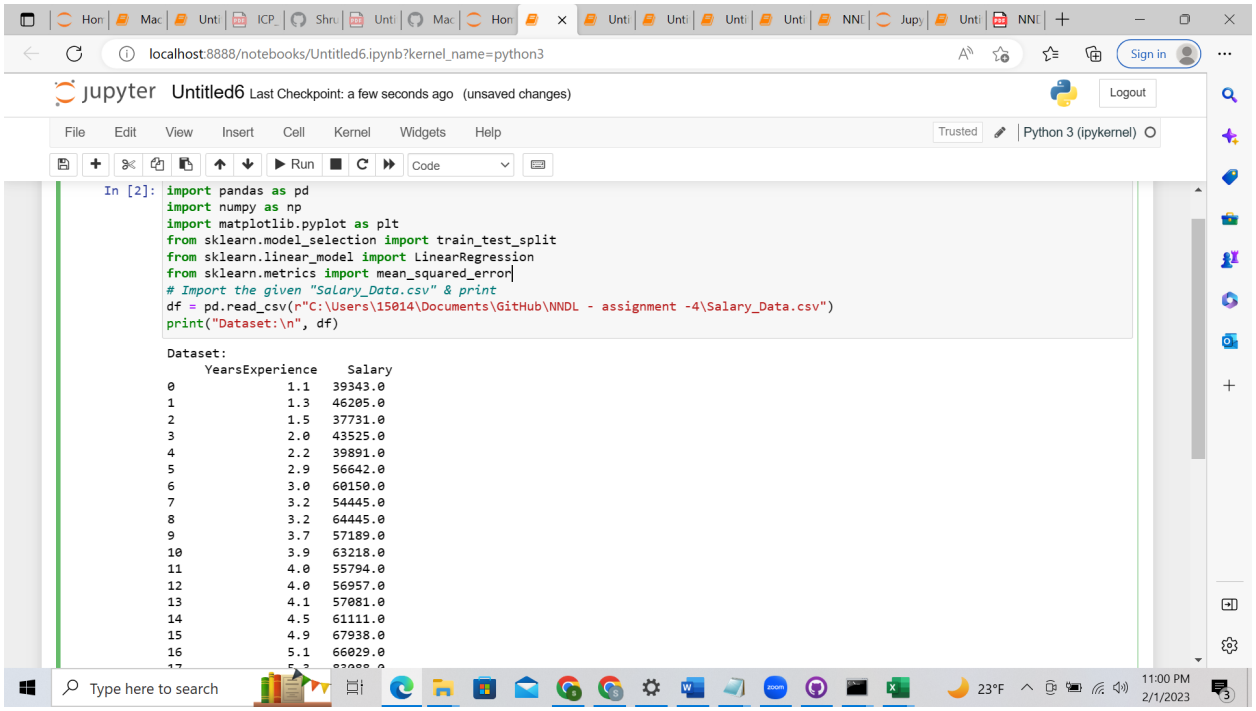
b) Split the data in train_test partitions, such that 1/3 of the data is reserved as test subset.

c) Train and predict the model.

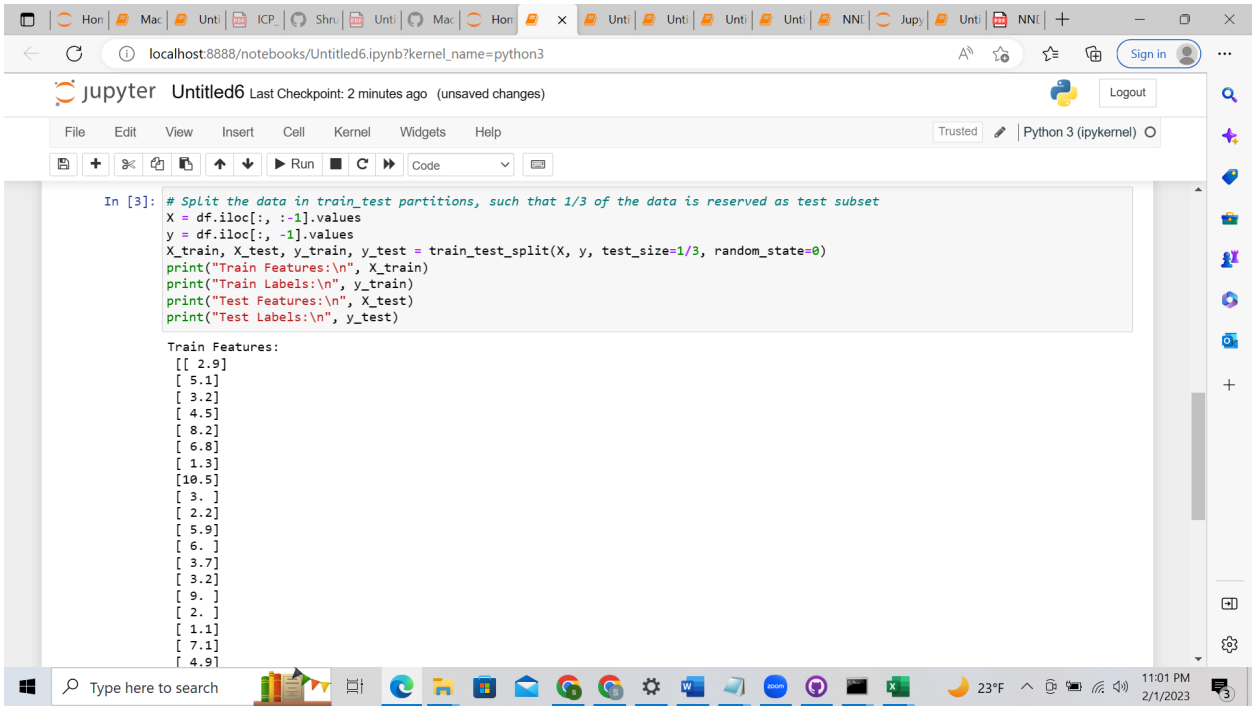
d) Calculate the mean_squared error

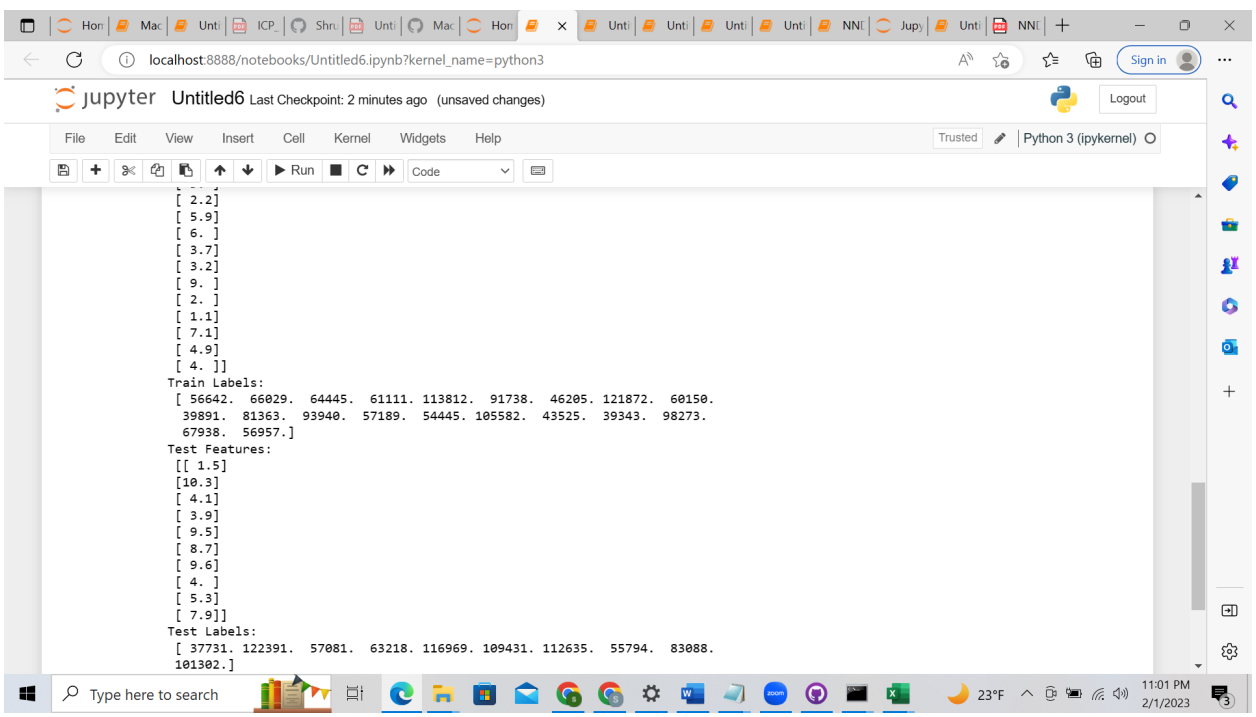
e) Visualize both train and test data using scatter plot

Ans

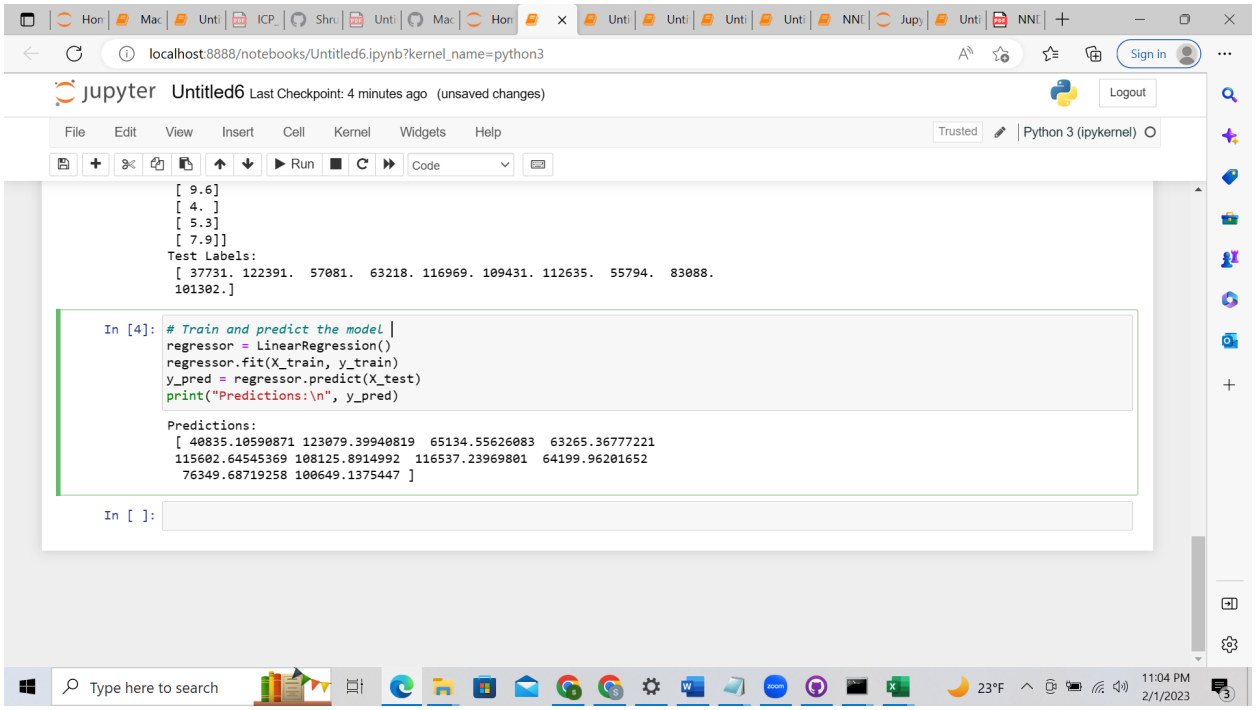


a) Imported the data set using pd.read_csv() function

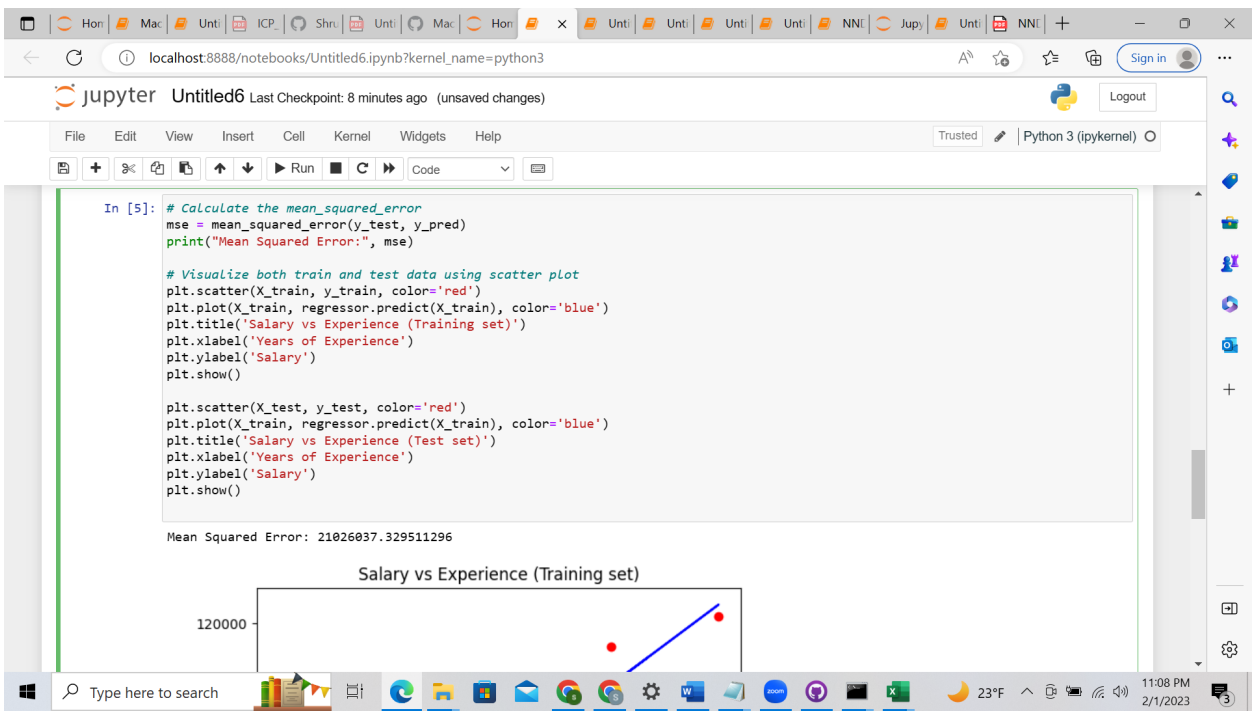




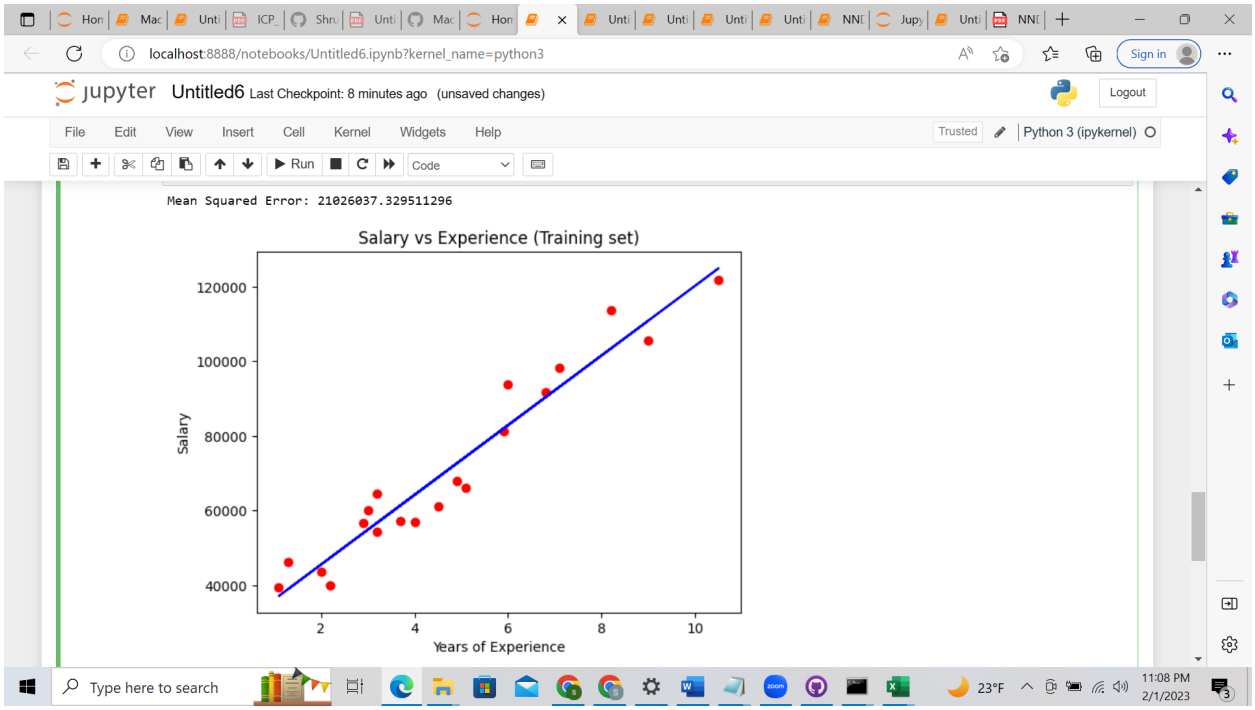
b) Splitted the data in to train_test partitions, such that 1/3 of the data is reserved as test subset using iloc() function

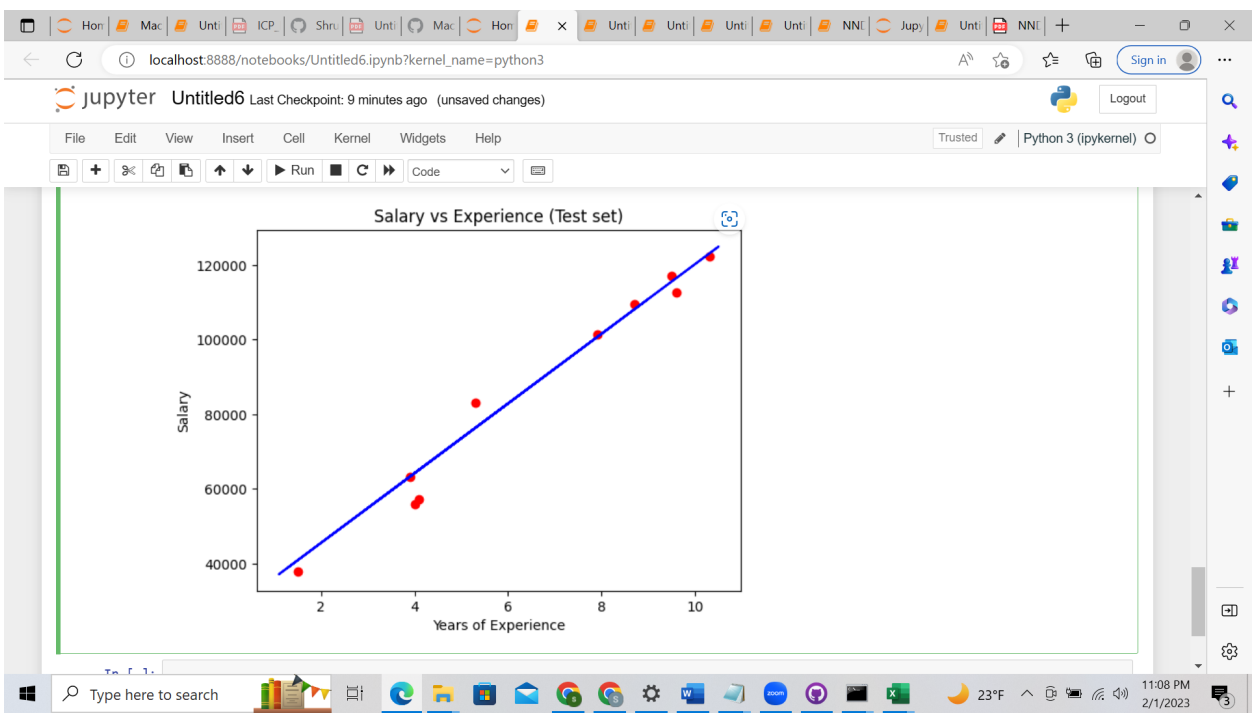


c) Trained the model using Linear Regression and predicted the model



d) Calculated the mean Squared Error using mean_squared_error() using function





e) Visualized the data set and train set using the scatter() function

Git hub repo : https://github.com/ShruthiVallapReddy/NNDL-assignment---4.git