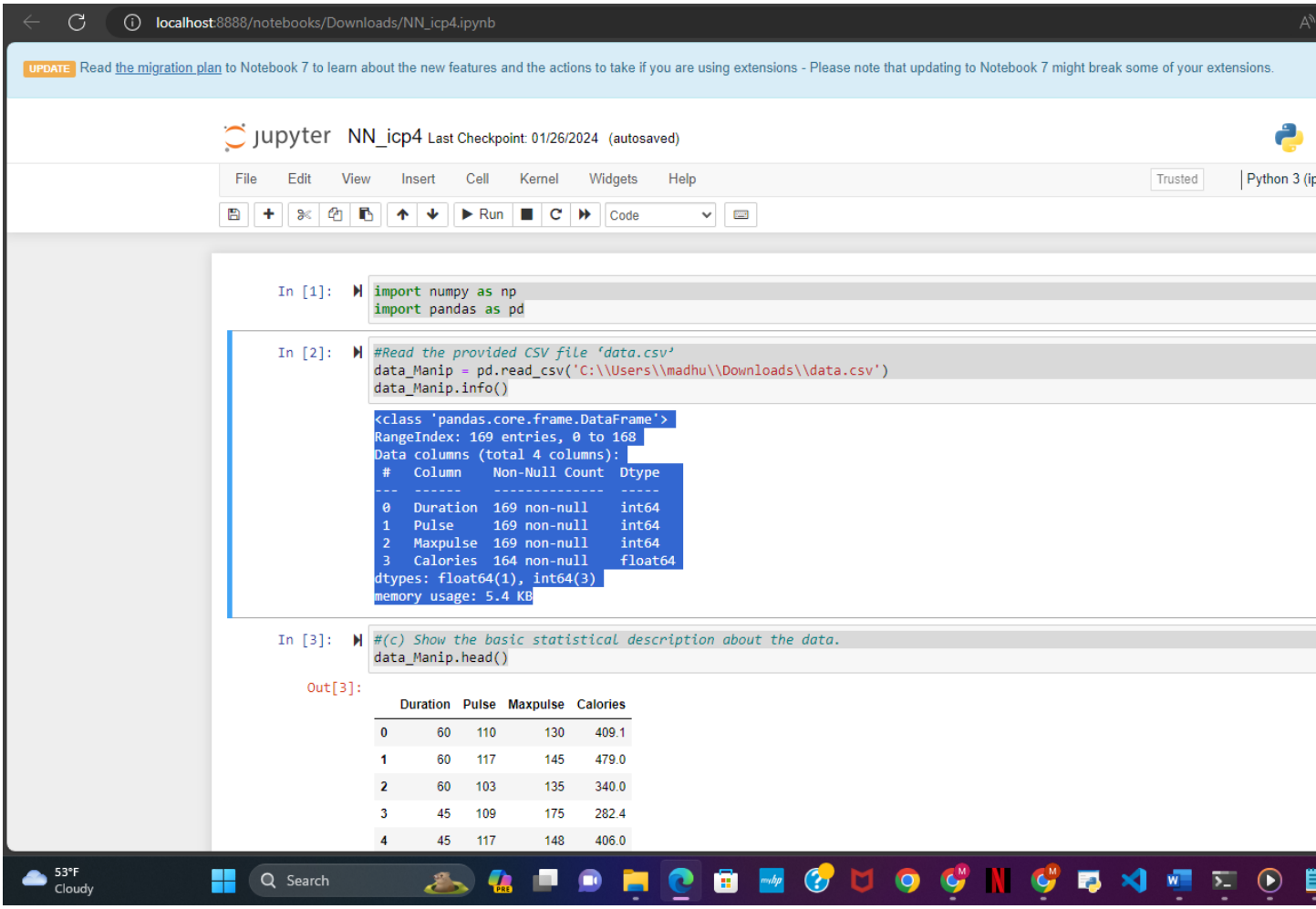
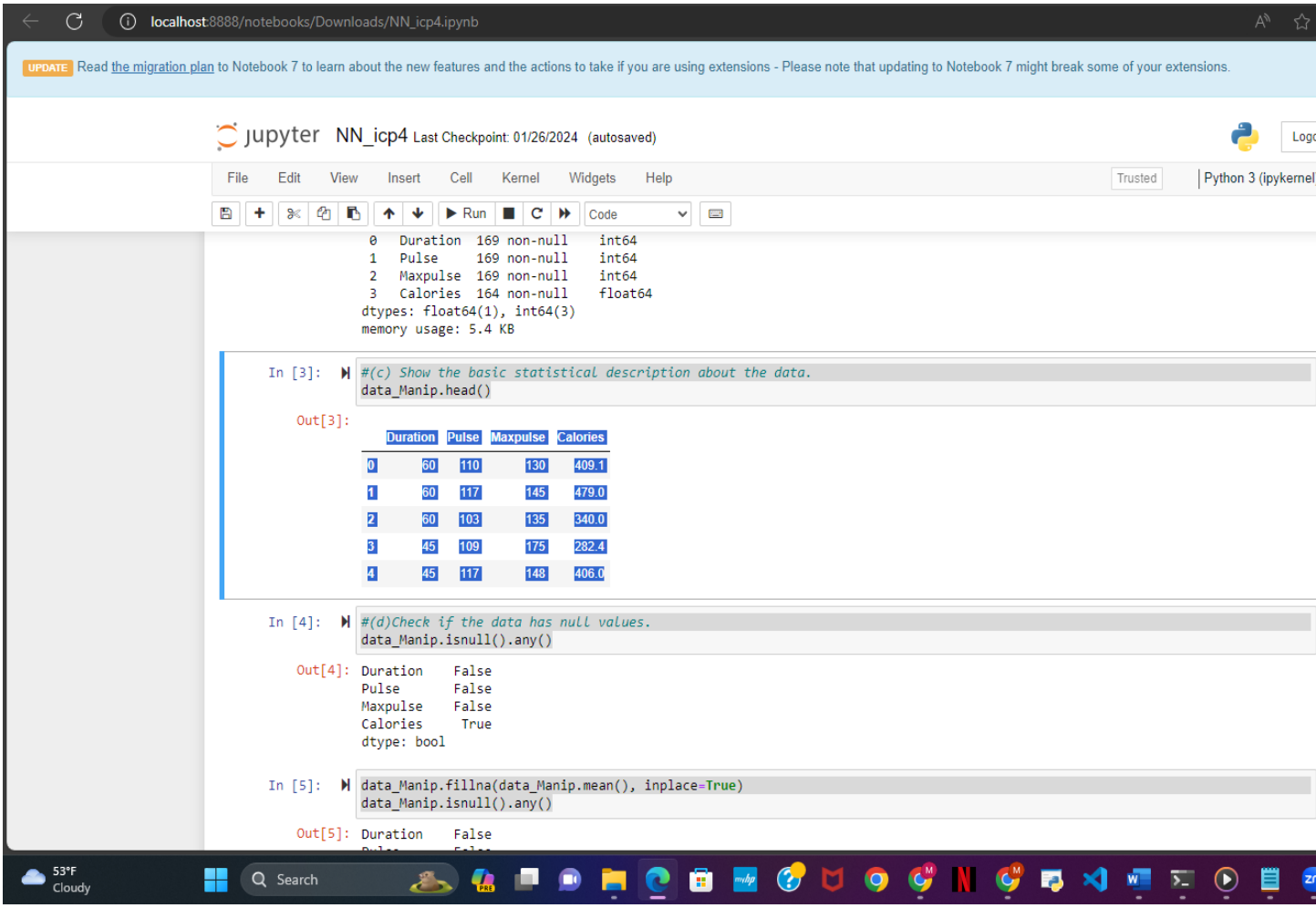
Madhulika Dayal

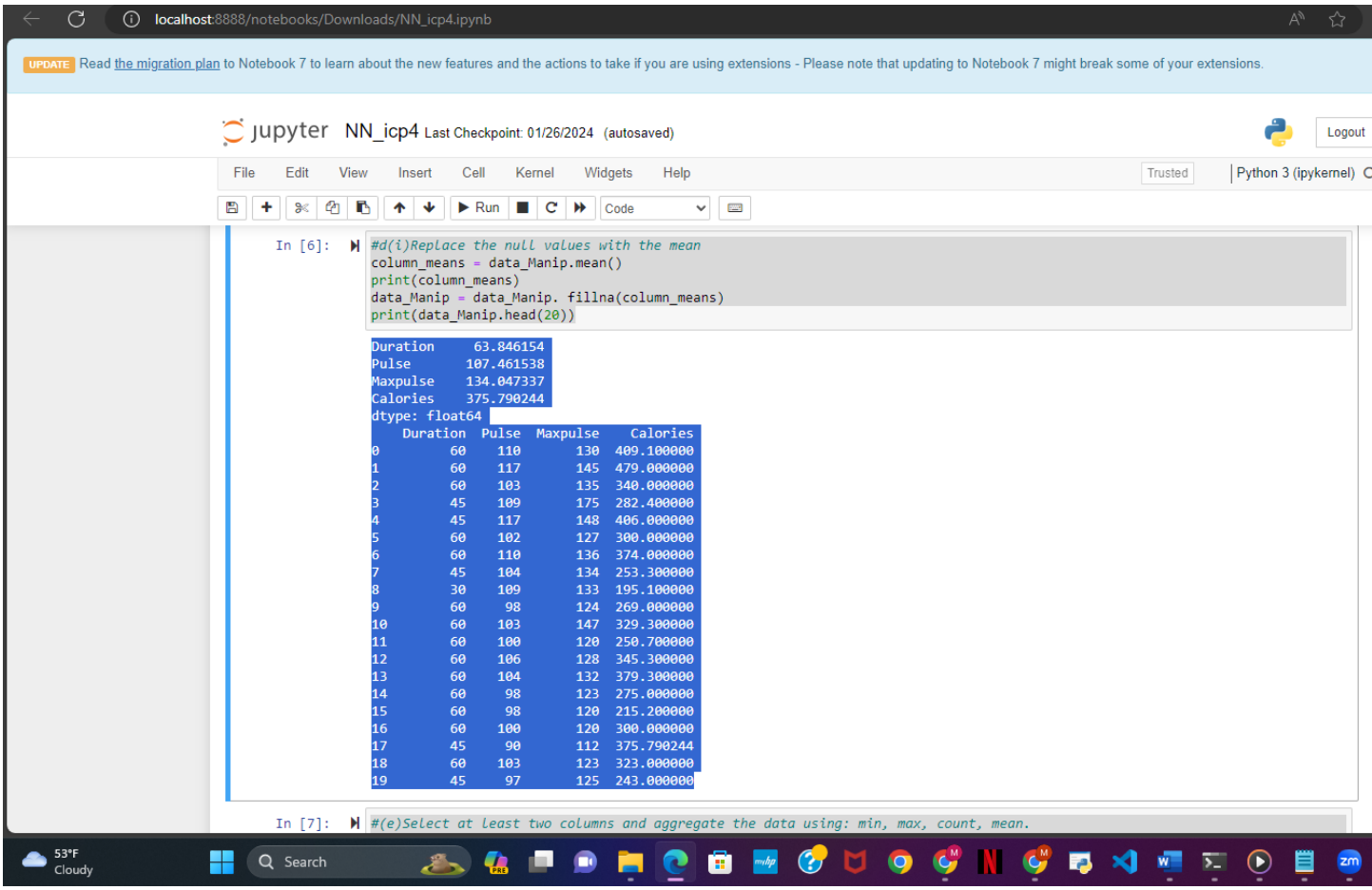
700743206

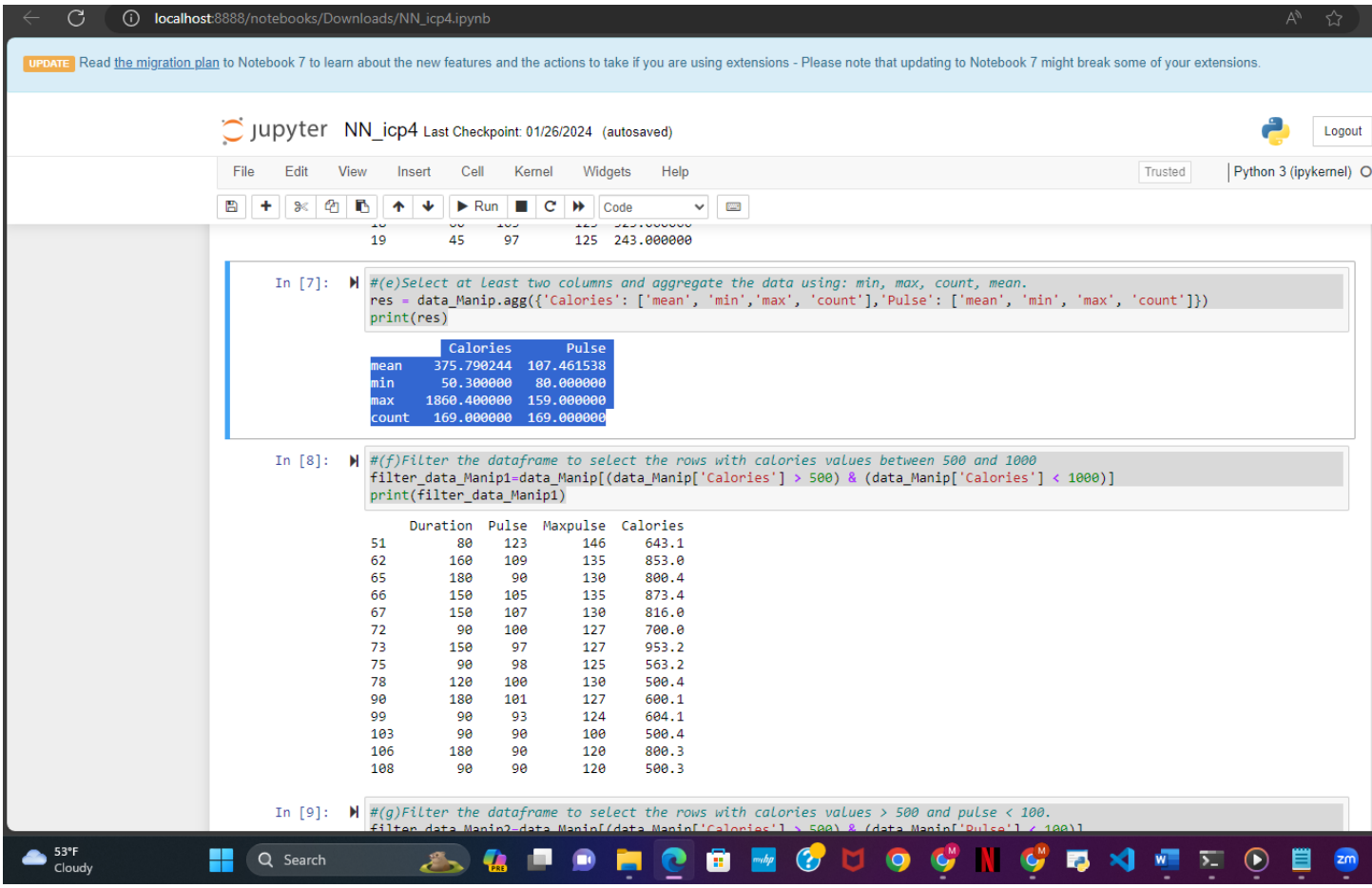
Assignment – 4

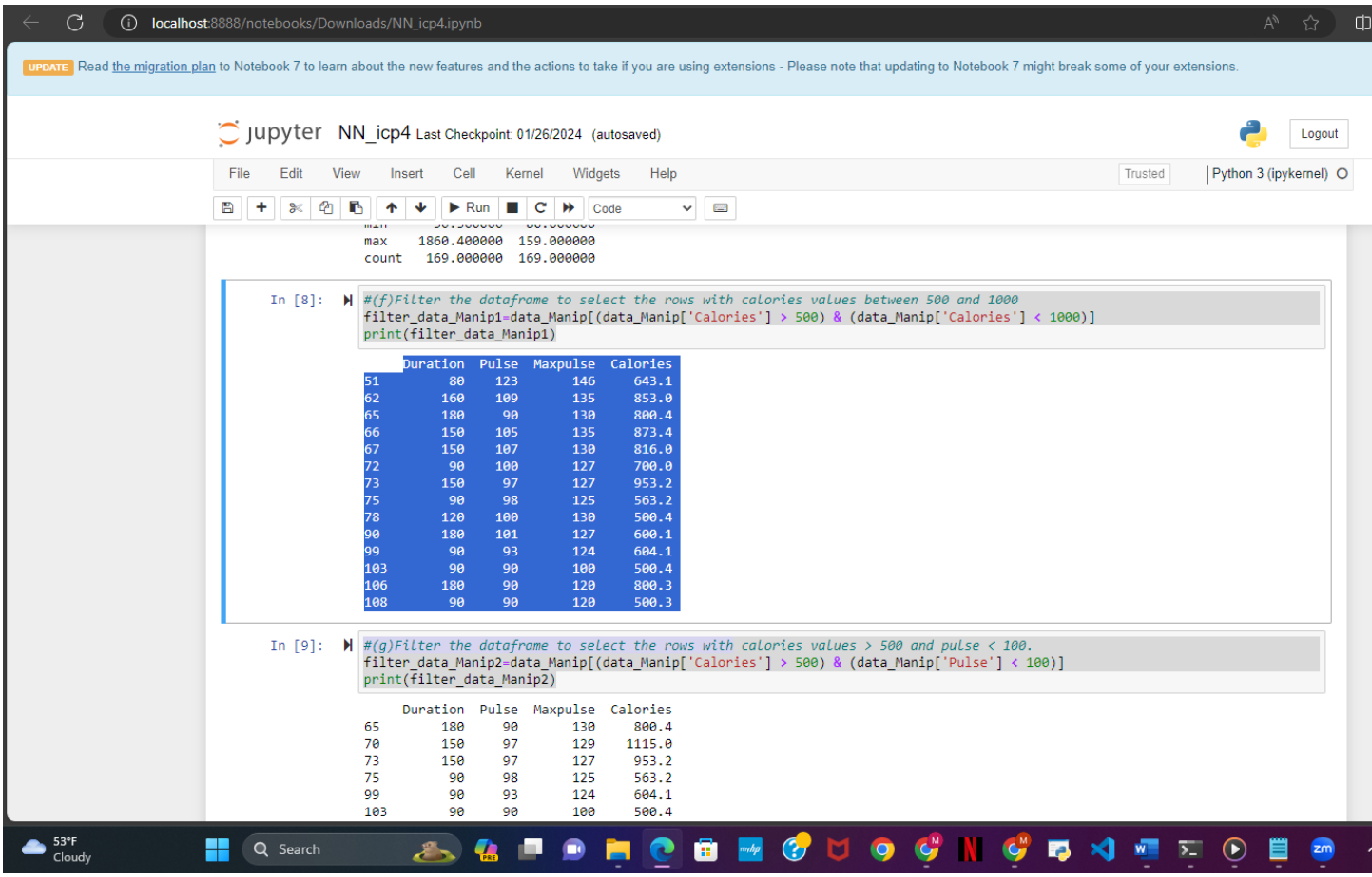
Github Link : https://github.com/Madhulika014/Assignment-4

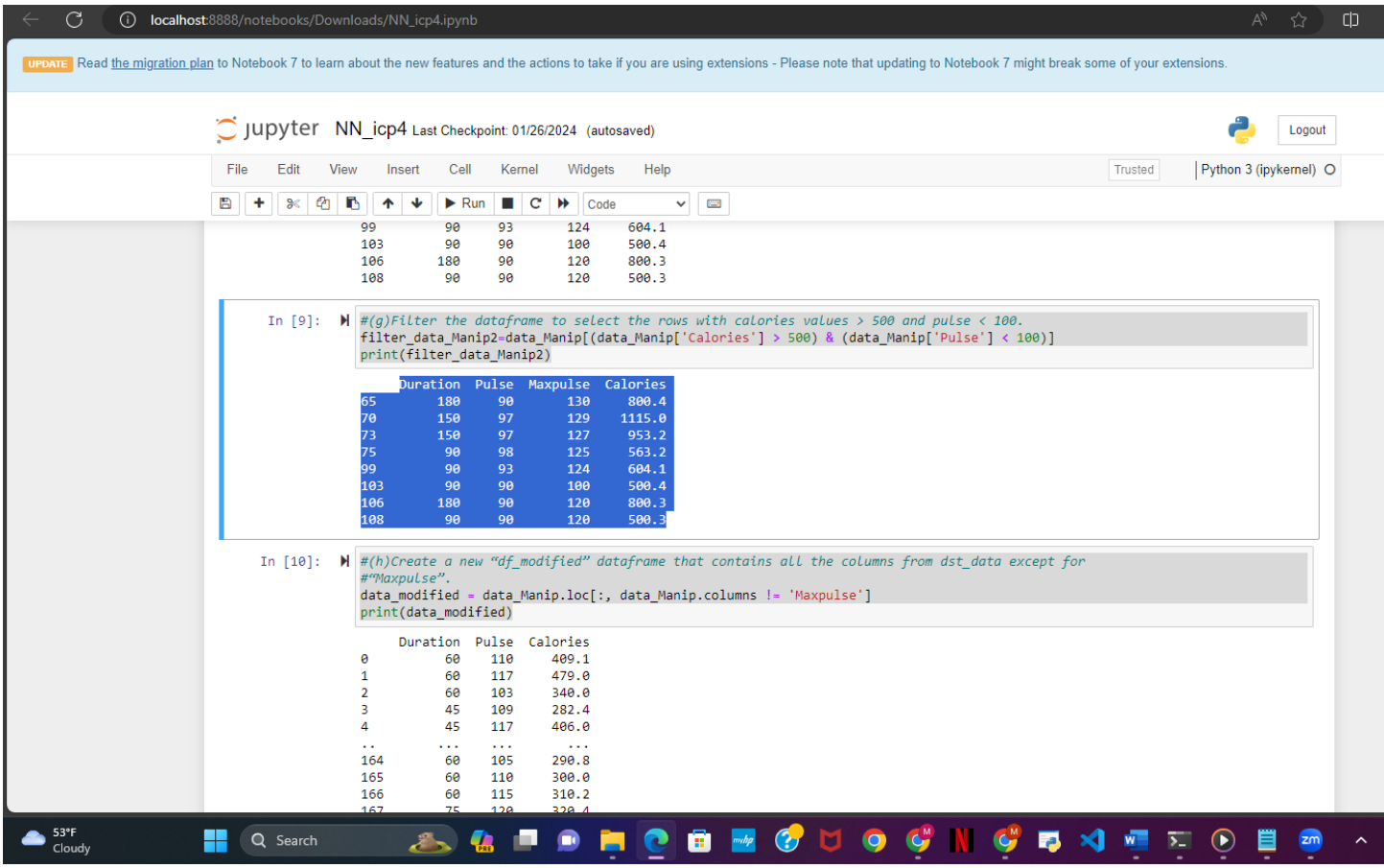


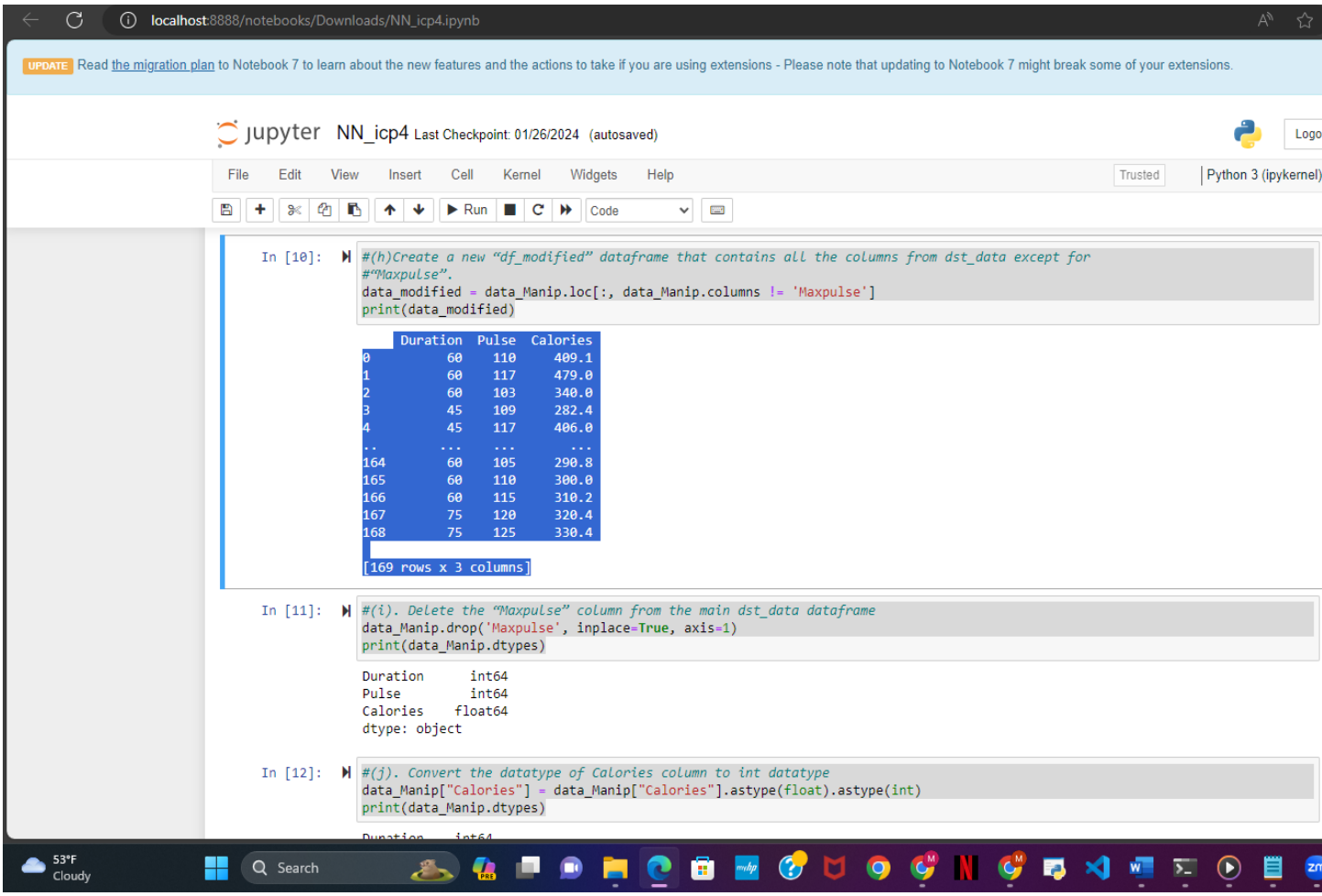

localhost:8888/notebooks/Downloads/NN_icp4.ipynb
UPDATE Read the migration plan to Notebook 7 to learn about the new features and the actions to take if you are using extensions - Please note that updating to Notebook 7 might break some of your extensions.
jupyter NN_icp4 Last Checkpoint: 01/26/2024 (autosaved)
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code
0 Duration 169 non-null int64
1 Pulse 169 non-null int64
2 Maxpulse 169 non-null int64
3 Calories 164 non-null float64
dtypes: float64(1), int64(3)
memory usage: 5.4 KB
In [3]:
#(c) Show the basic statistical description about the data.
data_Manip.head()
Out[3]:
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
In [4]:
#(d)Check if the data has null values.
data_Manip.isnull().any()
Out[4]:
Duration False
Pulse False
Maxpulse False
Calories True
dtype: bool
In [5]:
data_Manip.fillna(data_Manip.mean(), inplace=True)
data_Manip.isnull().any()
Out[5]:
Duration False
53°F
Cloudy
Search

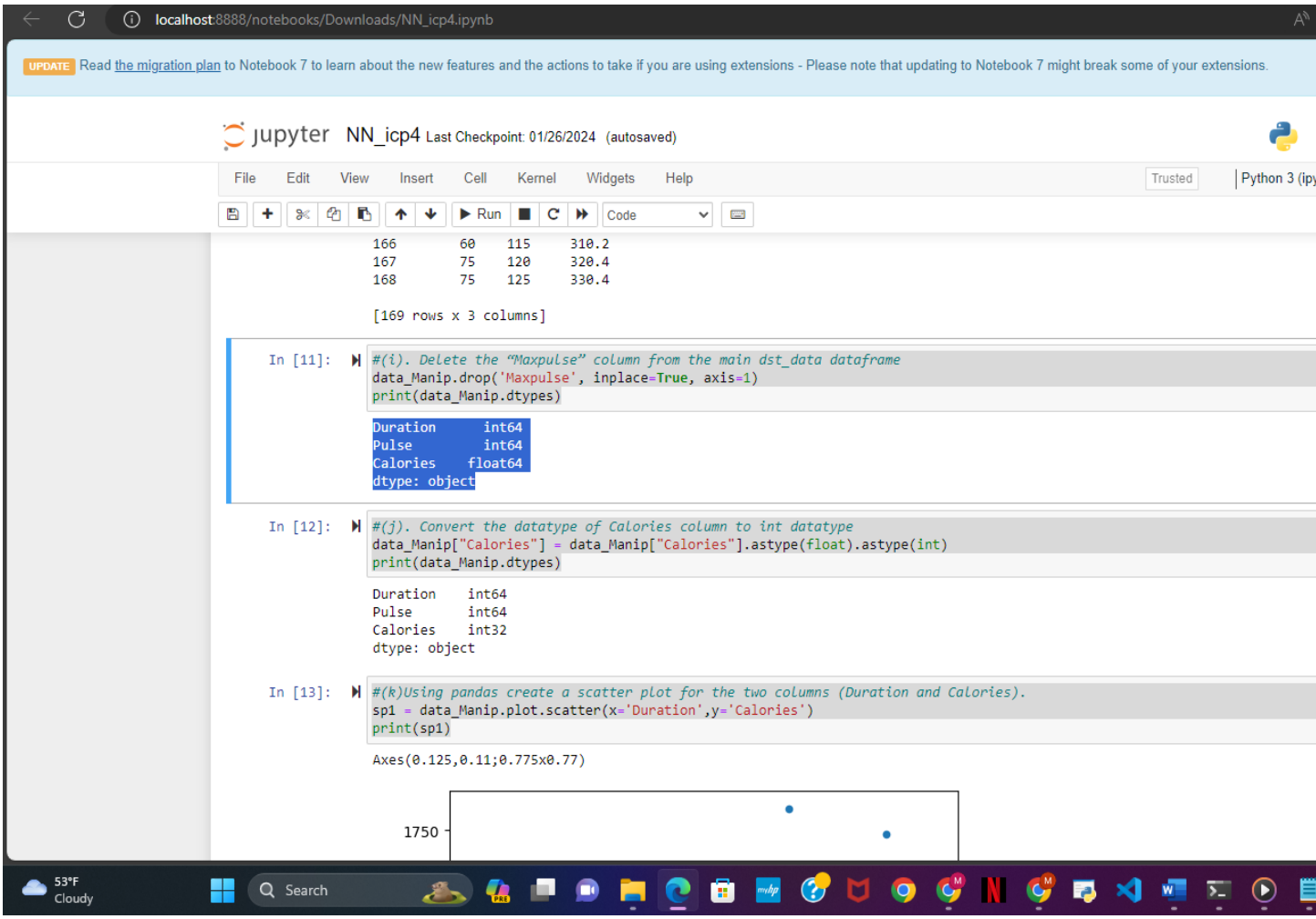

localhost:8888/notebooks/Downloads/NN_icp4.ipynb
UPDATE Read the migration plan to Notebook 7 to learn about the new features and the actions to take if you are using extensions - Please note that updating to Notebook 7 might break some of your extensions.
jupyter NN_icp4 Last Checkpoint: 01/26/2024 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code
In [6]:
#d(i)Replace the null values with the mean
column_means = data_Manip.mean()
print(column_means)
data_Manip = data_Manip. fillna(column_means)
print(data_Manip.head(20))
Duration 63.846154
Pulse 107.461538
Maxpulse 134.047337
Calories 375.790244
dtype: float64
Duration Pulse Maxpulse Calories
0 60 110 130 409.100000
1 60 117 145 479.000000
2 60 103 135 340.000000
3 45 109 175 282.400000
4 45 117 148 406.000000
5 60 102 127 300.000000
6 60 110 136 374.000000
7 45 104 134 253.300000
8 30 109 133 195.100000
9 60 98 124 269.000000
10 60 103 147 329.300000
11 60 100 120 250.700000
12 60 106 128 345.300000
13 60 104 132 379.300000
14 60 98 123 275.000000
15 60 98 120 215.200000
16 60 100 120 300.000000
17 45 90 112 375.790244
18 60 103 123 323.000000
19 45 97 125 243.000000
In [7]:
#(e)Select at least two columns and aggregate the data using: min, max, count, mean.
53°F Cloudy
Search


localhost:8888/notebooks/Downloads/NN_icp4.ipynb
UPDATE Read the migration plan to Notebook 7 to learn about the new features and the actions to take if you are using extensions - Please note that updating to Notebook 7 might break some of your extensions.
jupyter NN_icp4 Last Checkpoint: 01/26/2024 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code
19 45 97 125 243.000000
In [7]:
#(e)Select at least two columns and aggregate the data using: min, max, count, mean.
res = data_Manip.agg({'Calories': ['mean', 'min','max', 'count'],'Pulse': ['mean', 'min', 'max', 'count']})
print(res)
Calories Pulse
mean 375.790244 107.461538
min 50.300000 80.000000
max 1860.400000 159.000000
count 169.000000 169.000000
In [8]:
#(f)Filter the dataframe to select the rows with calories values between 500 and 1000
filter_data_Manip1=data_Manip[(data_Manip['Calories'] > 500) & (data_Manip['Calories'] < 1000)]
print(filter_data_Manip1)
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
90 180 101 127 600.1
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]:
#(g)Filter the dataframe to select the rows with calories values > 500 and pulse < 100.
53°F Cloudy
Search


localhost:8888/notebooks/Downloads/NN_icp4.ipynb
UPDATE Read the migration plan to Notebook 7 to learn about the new features and the actions to take if you are using extensions - Please note that updating to Notebook 7 might break some of your extensions.
jupyter NN_icp4 Last Checkpoint: 01/26/2024 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code
max 1860.400000 159.000000
count 169.000000 169.000000
In [8]:
#(f)Filter the dataframe to select the rows with calories values between 500 and 1000
filter_data_Manip1=data_Manip[(data_Manip['Calories'] > 500) & (data_Manip['Calories'] < 1000)]
print(filter_data_Manip1)
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
90 180 101 127 600.1
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]:
#(g)Filter the dataframe to select the rows with calories values > 500 and pulse < 100.
filter_data_Manip2=data_Manip[(data_Manip['Calories'] > 500) & (data_Manip['Pulse'] < 100)]
print(filter_data_Manip2)
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
53°F
Cloudy
Search


localhost:8888/notebooks/Downloads/NN_icp4.ipynb
UPDATE Read the migration plan to Notebook 7 to learn about the new features and the actions to take if you are using extensions - Please note that updating to Notebook 7 might break some of your extensions.
jupyter NN_icp4 Last Checkpoint: 01/26/2024 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code
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]:
#(g)Filter the dataframe to select the rows with calories values > 500 and pulse < 100.
filter_data_Manip2=data_Manip[(data_Manip['Calories'] > 500) & (data_Manip['Pulse'] < 100)]
print(filter_data_Manip2)
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 [10]:
#(h)Create a new "df_modified" dataframe that contains all the columns from dst_data except for
#"Maxpulse".
data_modified = data_Manip.loc[:, data_Manip.columns != 'Maxpulse']
print(data_modified)
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
53°F Cloudy
Search


localhost:8888/notebooks/Downloads/NN_icp4.ipynb
UPDATE Read the migration plan to Notebook 7 to learn about the new features and the actions to take if you are using extensions - Please note that updating to Notebook 7 might break some of your extensions.
jupyter NN_icp4 Last Checkpoint: 01/26/2024 (autosaved)
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code
In [10]:
#(h)Create a new "df_modified" dataframe that contains all the columns from dst_data except for
#"Maxpulse".
data_modified = data_Manip.loc[:, data_Manip.columns != 'Maxpulse']
print(data_modified)
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]
In [11]:
#(i). Delete the "Maxpulse" column from the main dst_data dataframe
data_Manip.drop('Maxpulse', inplace=True, axis=1)
print(data_Manip.dtypes)
Duration int64
Pulse int64
Calories float64
dtype: object
In [12]:
#(j). Convert the datatype of Calories column to int datatype
data_Manip["Calories"] = data_Manip["Calories"].astype(float).astype(int)
print(data_Manip.dtypes)
53°F
Cloudy
Search


localhost:8888/notebooks/Downloads/NN_icp4.ipynb
UPDATE Read the migration plan to Notebook 7 to learn about the new features and the actions to take if you are using extensions - Please note that updating to Notebook 7 might break some of your extensions.
jupyter NN_icp4 Last Checkpoint: 01/26/2024 (autosaved)
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipy
Run
Code
166 60 115 310.2
167 75 120 320.4
168 75 125 330.4
[169 rows x 3 columns]
In [11]:
#(i). Delete the "Maxpulse" column from the main dst_data dataframe
data_Manip.drop('Maxpulse', inplace=True, axis=1)
print(data_Manip.dtypes)
Duration int64
Pulse int64
Calories float64
dtype: object
In [12]:
#(j). Convert the datatype of Calories column to int datatype
data_Manip["Calories"] = data_Manip["Calories"].astype(float).astype(int)
print(data_Manip.dtypes)
Duration int64
Pulse int64
Calories int32
dtype: object
In [13]:
#(k)Using pandas create a scatter plot for the two columns (Duration and Calories).
sp1 = data_Manip.plot.scatter(x='Duration',y='Calories')
print(sp1)
Axes(0.125,0.11;0.775x0.77)
1750
53°F
Cloudy
Search

localhost:8888/notebooks/Downloads/NN_icp4.ipynb

UPDATE Read the migration plan to Notebook 7 to learn about the new features and the actions to take if you are using extensions - Please note that updating to Notebook 7 might break some of your extensions.

jupyter NN_icp4 Last Checkpoint: 01/26/2024 (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

Run Code

```
Duration      int64
Pulse         int64
Calories    float64
dtype: object
```

In [12]:

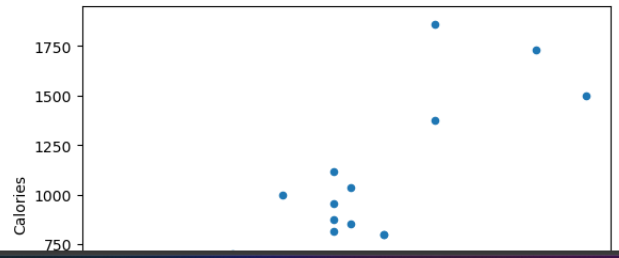
```
#(j). Convert the datatype of Calories column to int datatype
data_Manip["Calories"] = data_Manip["Calories"].astype(float).astype(int)
print(data_Manip.dtypes)
```

```
Duration    int64
Pulse       int64
Calories    int32
dtype: object
```

In [13]:

```
#(k)Using pandas create a scatter plot for the two columns (Duration and Calories).
sp1 = data_Manip.plot.scatter(x='Duration',y='Calories')
print(sp1)
```

```
Axes(0.125,0.11;0.775x0.77)
```



53°F Cloudy Search

localhost:8888/notebooks/Downloads/NN_icp4.ipynb

UPDATE Read the migration plan to Notebook 7 to learn about the new features and the actions to take if you are using extensions - Please note that updating to Notebook 7 might break some of your extensions.

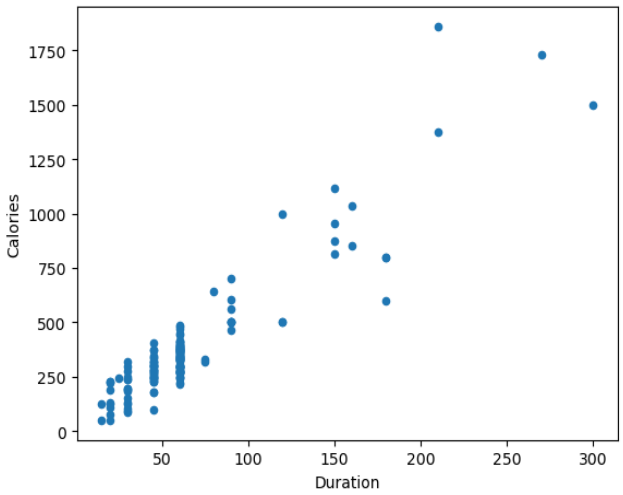
jupyter NN_icp4 Last Checkpoint: 01/26/2024 (autosaved)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

Run Code

```
sp1 = data_Manip.plot.scatter(x='Duration',y='Calories')
print(sp1)
```

```
Axes(0.125,0.11;0.775x0.77)
```



In [14]:

```
# 2(a) Import the given "Salary_Data.csv"
Lin_Re = pd.read_csv('C:\\Users\\madhu\\Downloads\\Salary_Data.csv')
Lin_Re.info()
```

53°F Cloudy

Search

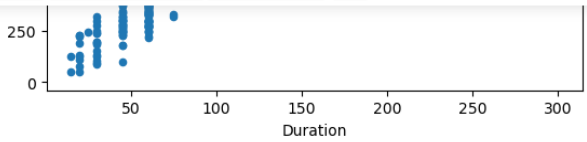
localhost:8888/notebooks/Downloads/NN_icp4.ipynb

UPDATE Read the migration plan to Notebook 7 to learn about the new features and the actions to take if you are using extensions - Please note that updating to Notebook 7 might break some of your extensions.

jupyter NN_icp4 Last Checkpoint: 01/26/2024 (autosaved)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

Run Code



In [14]:
```
# 2(a) Import the given "Salary_Data.csv"
Lin_Re = pd.read_csv('C:\\Users\\madhu\\Downloads\\Salary_Data.csv')
Lin_Re.info()
Lin_Re.head()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30 entries, 0 to 29
Data columns (total 2 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   YearsExperience  30 non-null     float64
 1   Salary           30 non-null     float64
dtypes: float64(2)
memory usage: 612.0 bytes
```

Out[14]:

	YearsExperience	Salary
0	1.1	39343.0
1	1.3	46205.0
2	1.5	37731.0
3	2.0	43525.0
4	2.2	39891.0

53°F Cloudy

Search

localhost:8888/notebooks/Downloads/NN_icp4.ipynb

UPDATE Read the migration plan to Notebook 7 to learn about the new features and the actions to take if you are using extensions - Please note that updating to Notebook 7 might break some of your extensions.

jupyter NN_icp4 Last Checkpoint: 01/26/2024 (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3 (ipykernel)

Run Code

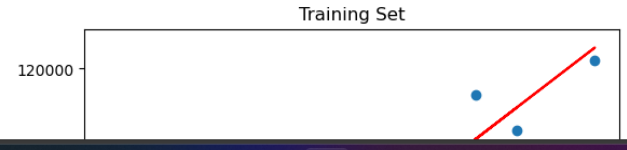
In [17]:
```
# (c) Train and predict the model.
from sklearn.linear_model import LinearRegression
lRegression = LinearRegression()
lRegression.fit(A_train, B_train)
B_Pred = lRegression.predict(A_test)
B_Pred
```

Out[17]:
```
array([ 40835.10590871, 123079.39940819,  65134.55626083,  63265.36777221,
       115602.64545369, 108125.8914992 , 116537.23969801,  64199.96201652,
        76349.68719258, 100649.1375447 ])
```

In [18]:
```
# (d) Calculate the mean_squared error
Sum_Serror = np.sum((B_Pred - B_test) ** 2)
mean_squared_error = Sum_Serror / B_test.size
mean_squared_error
```

Out[18]:
```
21026037.329511303
```

In [19]:
```
# (e) Visualize both train and test data using scatter plot.
import matplotlib.pyplot as plt
# Training Data set
plt.scatter(A_train, B_train)
plt.plot(A_train, lRegression.predict(A_train), color='red')
plt.title('Training Set')
plt.show()
```



53°F Cloudy

Search

localhost:8888/notebooks/Downloads/NN_icp4.ipynb

UPDATE Read the migration plan to Notebook 7 to learn about the new features and the actions to take if you are using extensions - Please note that updating to Notebook 7 might break some of your extensions.

jupyter NN_icp4 Last Checkpoint: 01/26/2024 (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

Run Code

```
Out[17]: array([ 40835.10590871, 123079.39940819,  65134.55626083,  63265.36777221,
                 115602.64545369, 108125.8914992 , 116537.23969801,  64199.96201652,
                  76349.68719258, 100649.1375447 ])
```

In [18]:
```
# (d) Calculate the mean_squared error
Sum_Serror = np.sum((B_Pred - B_test) ** 2)
mean_squared_error = Sum_Serror / B_test.size
mean_squared_error
```

Out[18]: 21026037.329511303

In [19]:
```
# (e) Visualize both train and test data using scatter plot.
import matplotlib.pyplot as plt
# Training Data set
plt.scatter(A_train, B_train)
plt.plot(A_train, lRegression.predict(A_train), color='red')
plt.title('Training Set')
plt.show()
```

53°F Cloudy Search

localhost:8888/notebooks/Downloads/NN_icp4.ipynb

UPDATE Read the migration plan to Notebook 7 to learn about the new features and the actions to take if you are using extensions - Please note that updating to Notebook 7 might break some of your extensions.

jupyter NN_icp4 Last Checkpoint: 01/26/2024 (autosaved)

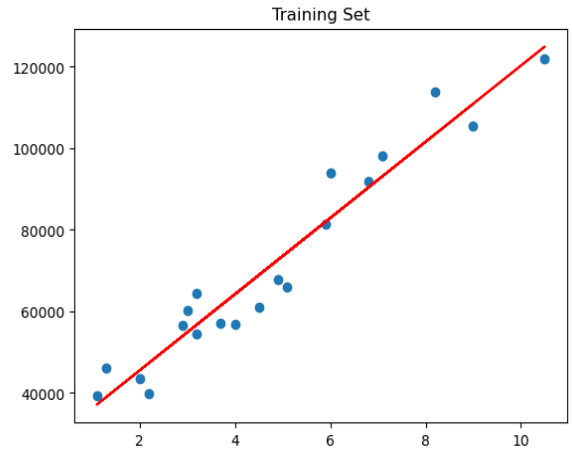
Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3 (ipykernel)

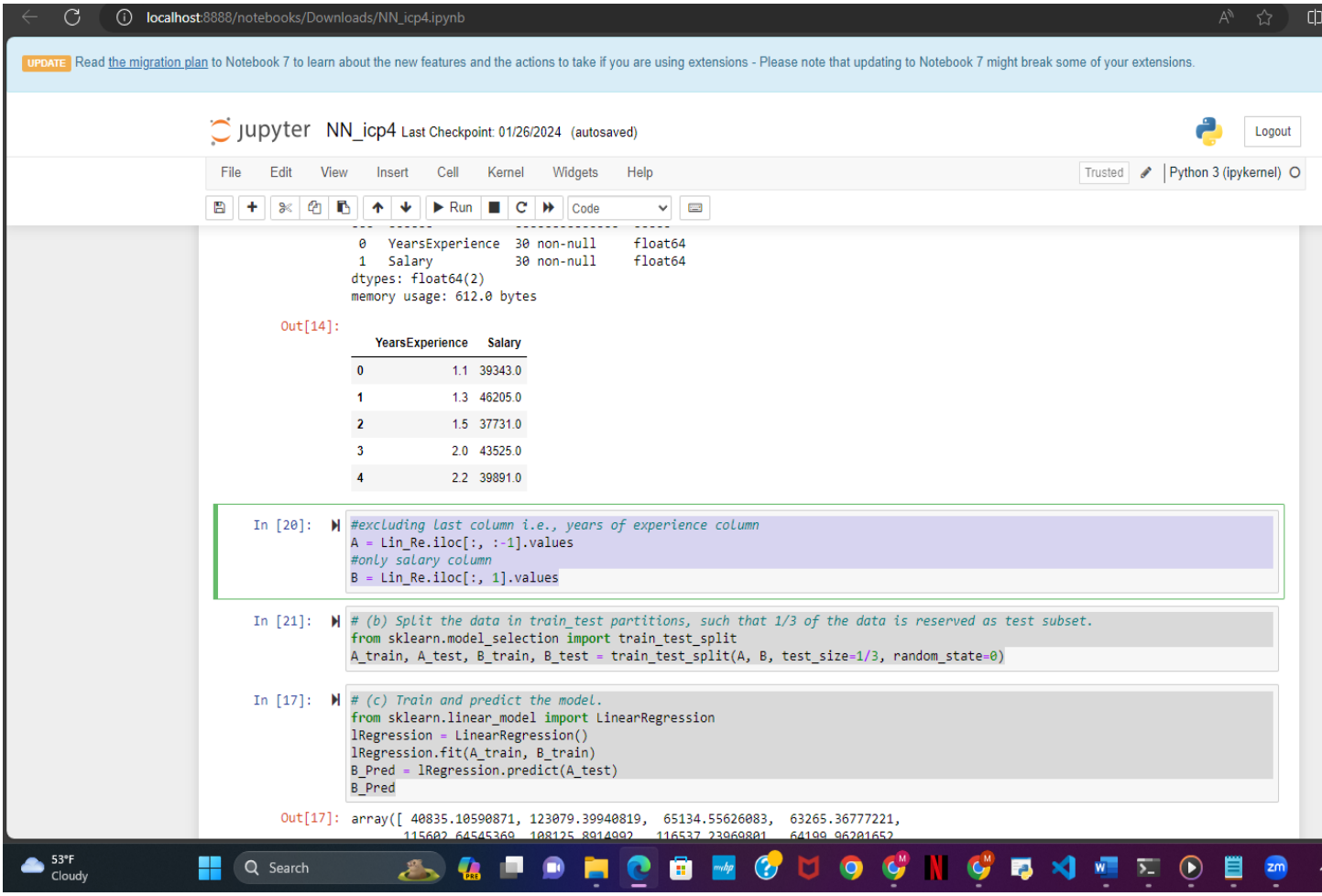
Run Code

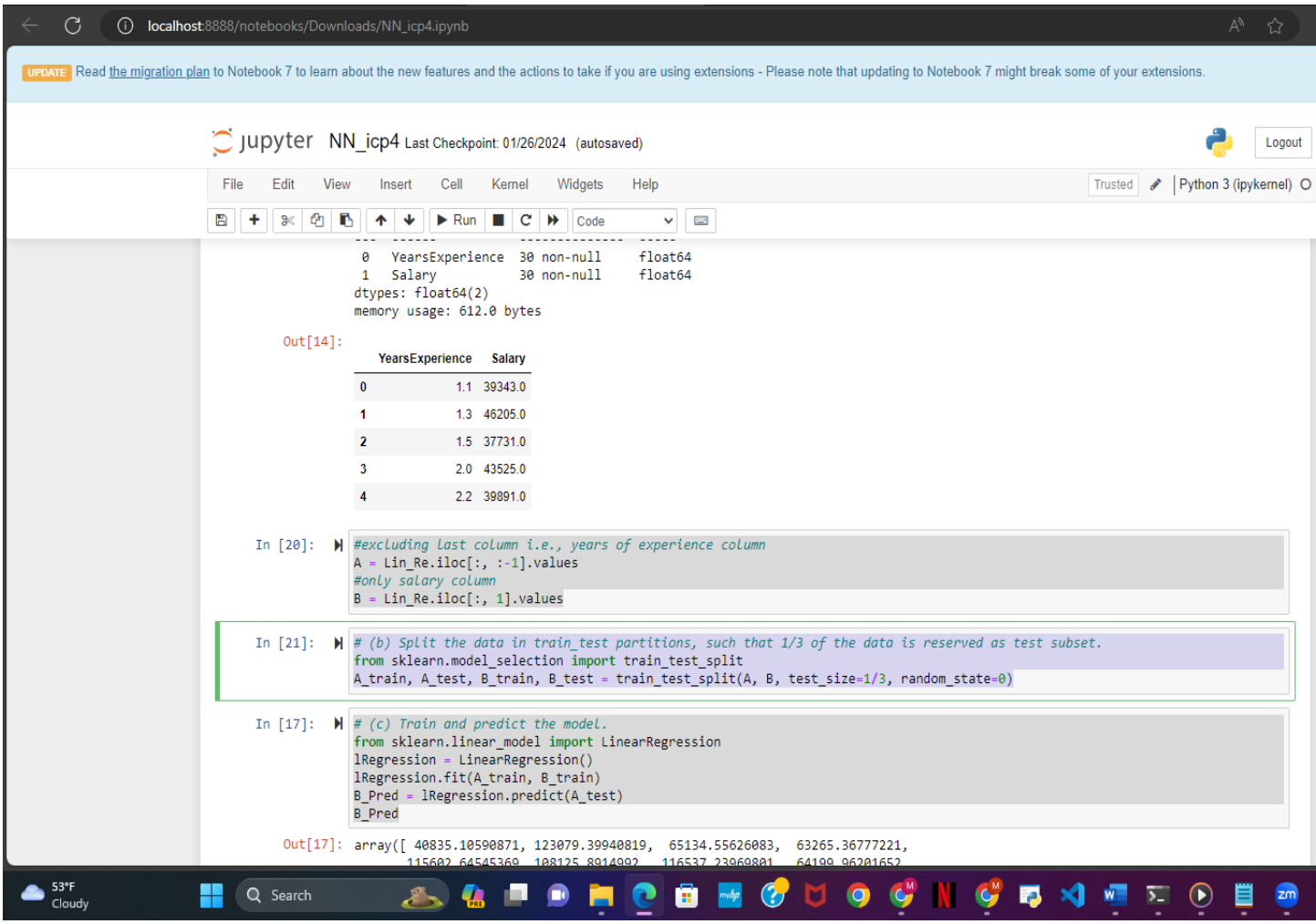
In [19]:
```
# (e) Visualize both train and test data using scatter plot.
import matplotlib.pyplot as plt
# Training Data set
plt.scatter(A_train, B_train)
plt.plot(A_train, lRegression.predict(A_train), color='red')
plt.title('Training Set')
plt.show()
```



53°F Cloudy

Search


localhost:8888/notebooks/Downloads/NN_icp4.ipynb
UPDATE Read the migration plan to Notebook 7 to learn about the new features and the actions to take if you are using extensions - Please note that updating to Notebook 7 might break some of your extensions.
jupyter NN_icp4 Last Checkpoint: 01/26/2024 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code
0 YearsExperience 30 non-null float64
1 Salary 30 non-null float64
dtypes: float64(2)
memory usage: 612.0 bytes
Out[14]:
YearsExperience Salary
0 1.1 39343.0
1 1.3 46205.0
2 1.5 37731.0
3 2.0 43525.0
4 2.2 39891.0
In [20]:
#excluding last column i.e., years of experience column
A = Lin_Re.iloc[:, :-1].values
#only salary column
B = Lin_Re.iloc[:, 1].values
In [21]:
# (b) Split the data in train_test partitions, such that 1/3 of the data is reserved as test subset.
from sklearn.model_selection import train_test_split
A_train, A_test, B_train, B_test = train_test_split(A, B, test_size=1/3, random_state=0)
In [17]:
# (c) Train and predict the model.
from sklearn.linear_model import LinearRegression
lRegression = LinearRegression()
lRegression.fit(A_train, B_train)
B_Pred = lRegression.predict(A_test)
B_Pred
Out[17]:
array([40835.10590871, 123079.39940819, 65134.55626083, 63265.36777221,
53°F
Cloudy
Search


localhost:8888/notebooks/Downloads/NN_icp4.ipynb
UPDATE Read the migration plan to Notebook 7 to learn about the new features and the actions to take if you are using extensions - Please note that updating to Notebook 7 might break some of your extensions.
jupyter NN_icp4 Last Checkpoint: 01/26/2024 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code
0 YearsExperience 30 non-null float64
1 Salary 30 non-null float64
dtypes: float64(2)
memory usage: 612.0 bytes
Out[14]:
YearsExperience Salary
0 1.1 39343.0
1 1.3 46205.0
2 1.5 37731.0
3 2.0 43525.0
4 2.2 39891.0
In [20]:
#excluding last column i.e., years of experience column
A = Lin_Re.iloc[:, :-1].values
#only salary column
B = Lin_Re.iloc[:, 1].values
In [21]:
# (b) Split the data in train_test partitions, such that 1/3 of the data is reserved as test subset.
from sklearn.model_selection import train_test_split
A_train, A_test, B_train, B_test = train_test_split(A, B, test_size=1/3, random_state=0)
In [17]:
# (c) Train and predict the model.
from sklearn.linear_model import LinearRegression
lRegression = LinearRegression()
lRegression.fit(A_train, B_train)
B_Pred = lRegression.predict(A_test)
B_Pred
Out[17]: array([40835.10590871, 123079.39940819, 65134.55626083, 63265.36777221,
53°F
Cloudy
Search

localhost:8888/notebooks/Downloads/NN_icp4.ipynb

UPDATE Read the migration plan to Notebook 7 to learn about the new features and the actions to take if you are using extensions - Please note that updating to Notebook 7 might break some of your extensions.

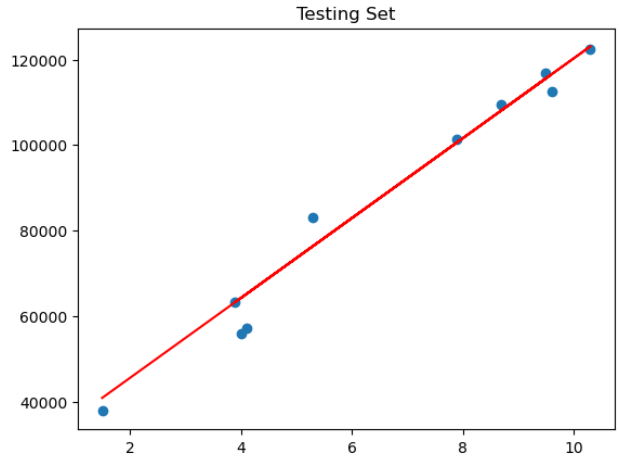
jupyter NN_icp4 Last Checkpoint: 01/26/2024 (autosaved)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

Run Code

2 4 6 8 10

In [22]:
```
# Testing Data set
plt.scatter(A_test, B_test)
plt.plot(A_test, lRegression.predict(A_test), color='red')
plt.title('Testing Set')
plt.show()
```



53°F Cloudy

Search