

ICP4

Vamsi Krishna Remala

Student id: 700744730

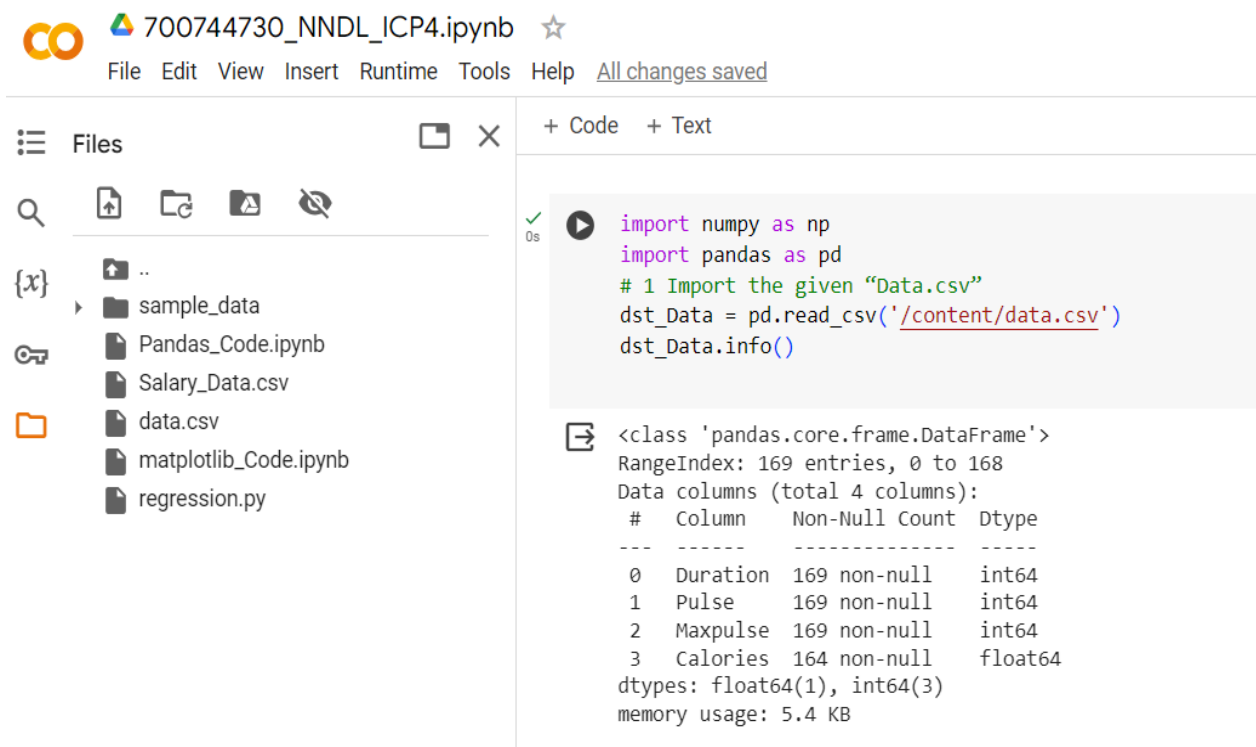
GitHub Link: https://github.com/vamsikrishnaremala/700744730_NNDL_ICP4

Video Link:

<https://drive.google.com/file/d/1OHXr3RI8r2THtC6UIHvxU7ujB-0RRF2P/view?usp=sharing>

1. Data Manipulation

- a. Read the provided CSV file 'data.csv'.



The screenshot shows a Jupyter Notebook titled "700744730_NNDL_ICP4.ipynb". The left sidebar displays a file explorer with the following files: sample_data, Pandas_Code.ipynb, Salary_Data.csv, data.csv, matplotlib_Code.ipynb, and regression.py. The main area shows a code cell with the following Python code:

```
import numpy as np
import pandas as pd
# 1 Import the given "Data.csv"
dst_Data = pd.read_csv('/content/data.csv')
dst_Data.info()
```

The output of the code cell is:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 169 entries, 0 to 168
Data columns (total 4 columns):
#   Column      Non-Null Count  Dtype
---  -
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
```

- b. Show the basic statistical description about the data.
- c. Check if the data has null values.
 - i. Replace the null values with the mean.



700744730_NNDL_ICP4.ipynb ☆

File Edit View Insert Runtime Tools Help [All changes saved](#)

Files



..

sample_data



Pandas_Code.ipynb

Salary_Data.csv



data.csv

matplotlib_Code.ipynb

regression.py

Disk 81.46 GB available

+ Code + Text

```
✓ [2] # Show the basic statistical description about the data.  
0s dst_Data.head()
```

	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

```
[ ] # Check if the data has null values.  
dst_Data.isnull().any()
```

```
Duration    False  
Pulse       False  
Maxpulse    False  
Calories    True  
dtype: bool
```

```
✓ [3] dst_Data.fillna(dst_Data.mean(), inplace=True)  
0s dst_Data.isnull().any()
```

```
Duration    False  
Pulse       False  
Maxpulse    False  
Calories    False  
dtype: bool
```



700744730_NNDL_ICP4.ipynb ☆

File Edit View Insert Runtime Tools Help [All changes saved](#)

Files



..

sample_data



Pandas_Code.ipynb

Salary_Data.csv



data.csv

matplotlib_Code.ipynb

regression.py

Disk 81.46 GB available

+ Code + Text



0s

```
[4] # Replace the null values with the mean
column_means = dst_Data.mean()
print(column_means)
dst_Data = dst_Data.fillna(column_means)
print(dst_Data.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

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



700744730_NNDL_ICP4.ipynb ☆

File Edit View Insert Runtime Tools Help [All changes saved](#)

Files



sample_data



Pandas_Code.ipynb

Salary_Data.csv



data.csv

matplotlib_Code.ipynb

regression.py

+ Code + Text



0s

```
[4]
10      60      100      120  300.000000
17      45       90      112  375.790244
18      60      103      123  323.000000
19      45       97      125  243.000000
```



0s

```
[5] # Select at least two columns and aggregate the data using: min, max, count, mean.
res = dst_Data.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

- e. Filter the dataframe to select the rows with calories values between 500 and 1000.
- f. Filter the dataframe to select the rows with calories values > 500 and pulse.

700744730_NNDL_ICP4.ipynb ☆

File Edit View Insert Runtime Tools Help All changes saved

Files

- sample_data
- Pandas_Code.ipynb
- Salary_Data.csv
- data.csv
- matplotlib_Code.ipynb
- regression.py

+ Code + Text

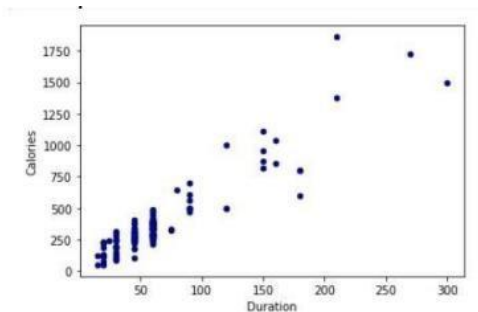
```
[6] # Filter the dataframe to select the rows with calories values between 500 and 1000
filter_dst_Data1=dst_Data[(dst_Data['Calories'] > 500) & (dst_Data['Calories'] < 1000)]
print(filter_dst_Data1)
# Filter the dataframe to select the rows with calories values > 500 and pulse < 100.
filter_dst_Data2=dst_Data[(dst_Data['Calories'] > 500) & (dst_Data['Pulse'] < 100)]
print(filter_dst_Data2)
```

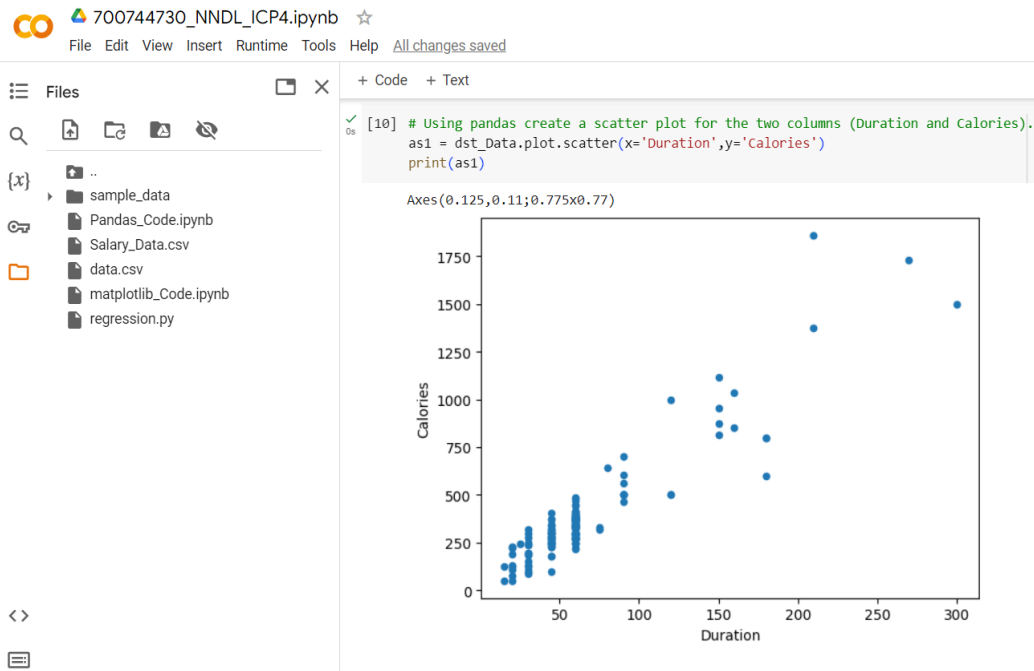
	Duration	Pulse	Maxpulse	Calories
51	80	123	146	643.1
62	160	109	135	853.0
65	180	90	130	800.4
66	150	105	135	873.4
67	150	107	130	816.0
72	90	100	127	700.0
73	150	97	127	953.2
75	90	98	125	563.2
78	120	100	130	500.4
90	180	101	127	600.1
99	90	93	124	604.1
103	90	90	100	500.4
106	180	90	120	800.3
108	90	90	120	500.3
	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

Disk 81.46 GB available

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

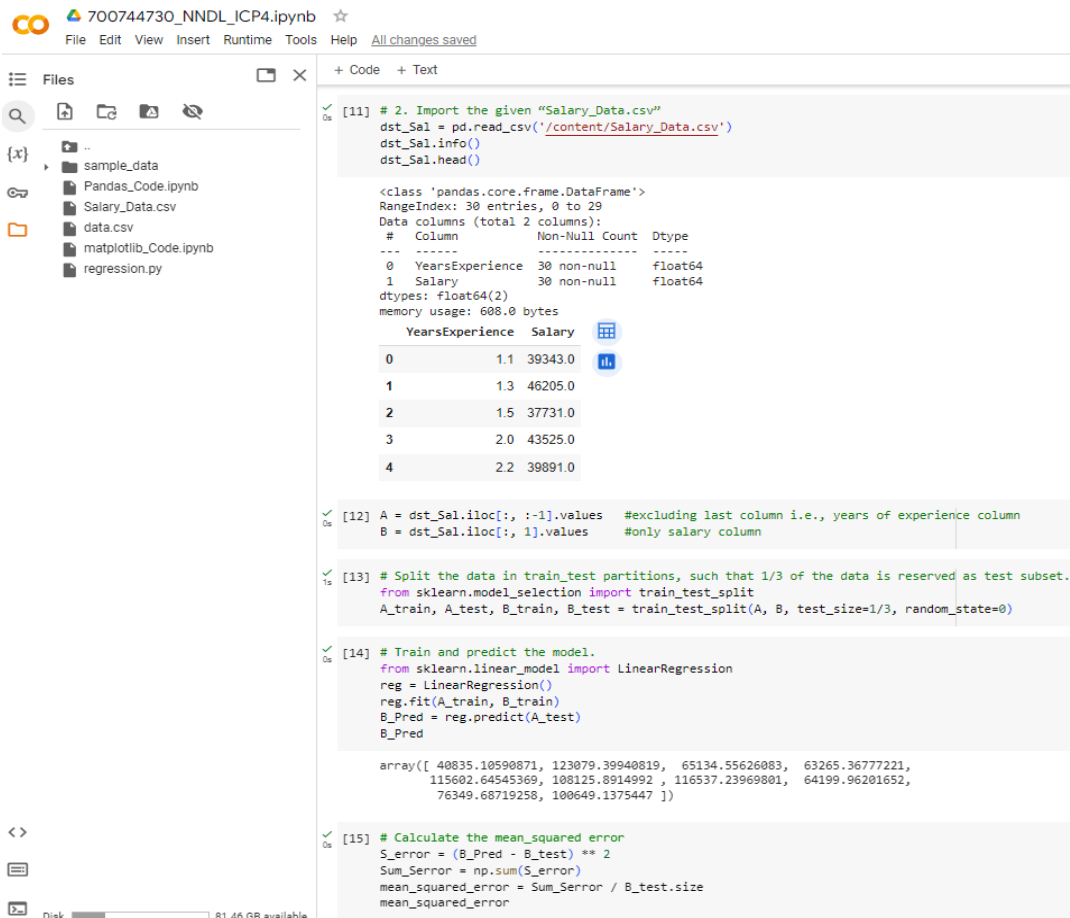
Example





2. Linear Regression

- Import the given "Salary_Data.csv"
- Split the data in train_test partitions, such that 1/3 of the data is reserved as test subset.
- Train and predict the model.
- Calculate the mean_squared error



e) Visualize both train and test data using scatter plot.



700744730_NNDL_ICP4.ipynb ☆

File Edit View Insert Runtime Tools Help

Files



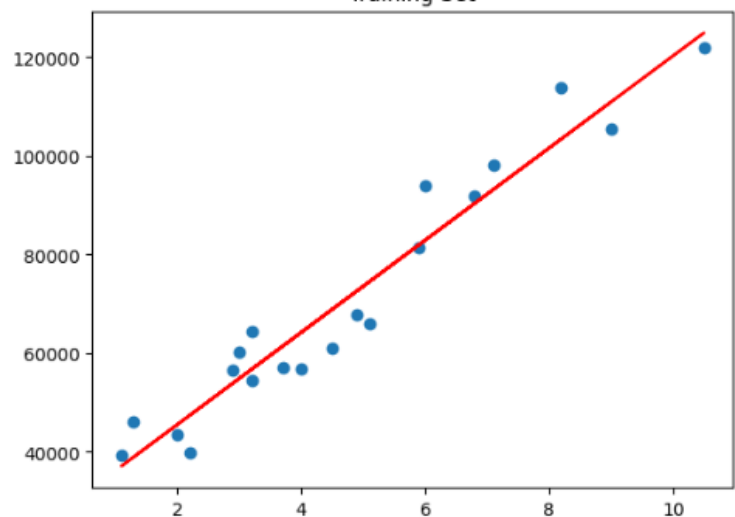
{x} ...
sample_data
Pandas_Code.ipynb
Salary_Data.csv
data.csv
matplotlib_Code.ipynb
regression.py

+ Code + Text

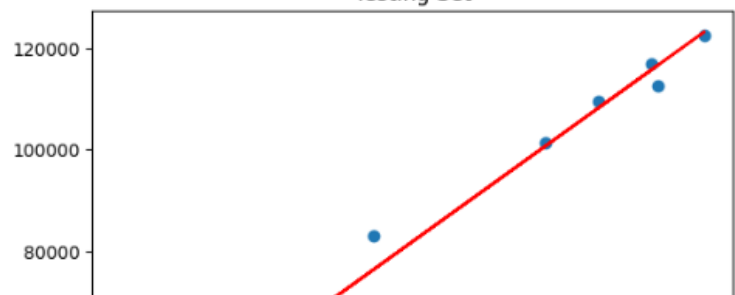
```
# Visualize both train and test data using scatter plot.
import matplotlib.pyplot as plt
# Training Data set
plt.scatter(A_train, B_train)
plt.plot(A_train, reg.predict(A_train), color='red')
plt.title('Training Set')
plt.show()

# Testing Data set
plt.scatter(A_test, B_test)
plt.plot(A_test, reg.predict(A_test), color='red')
plt.title('Testing Set')
plt.show()
```

Training Set



Testing Set



<>

≡

▶

nick 81 / 66 GB available



700744730_NNDL_ICP4.ipynb



File Edit View Insert Runtime Tools Help [All changes saved](#)

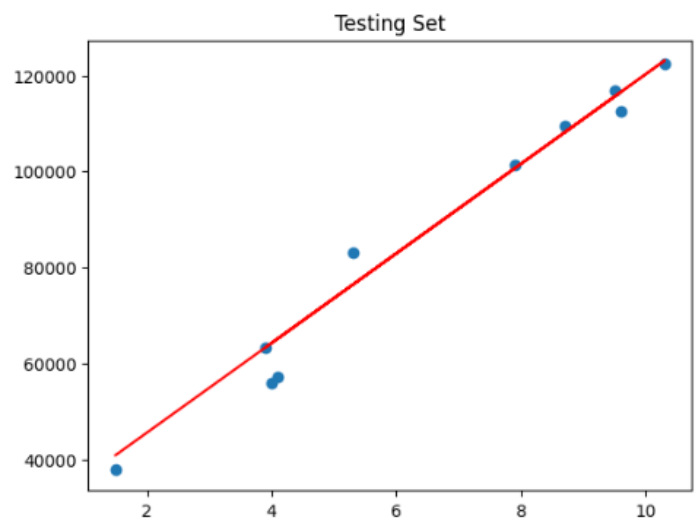
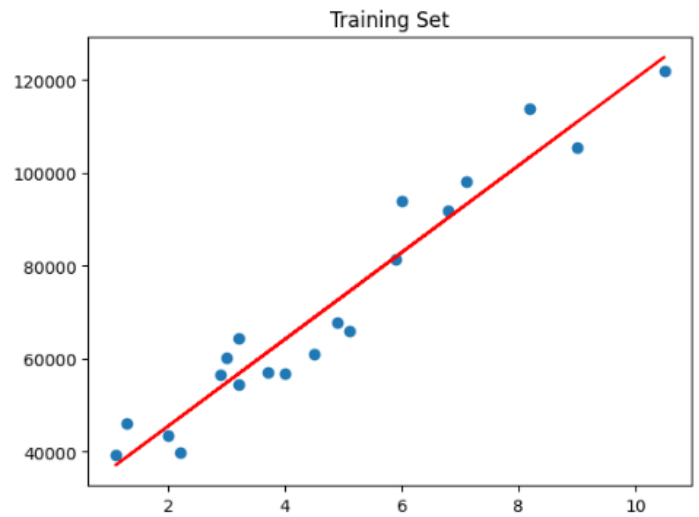
Files



{x}
 sample_data
 Pandas_Code.ipynb
 Salary_Data.csv
 data.csv
 matplotlib_Code.ipynb
 regression.py

+ Code + Text

```
plt.title('Testing Set')  
plt.show()
```



< >



Disk 81.46 GB available