DEPARTMENT OF
COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

Experiment 2.2

Student Name: Nishant Kumar Mehta **UID: 21BCS3402**
Branch: CSE **Section/Group: IOT_602-B**
Semester: 5th **Date of Performance:28/09/23**
Subject Name: AIML **Subject Code:21CSH_316**

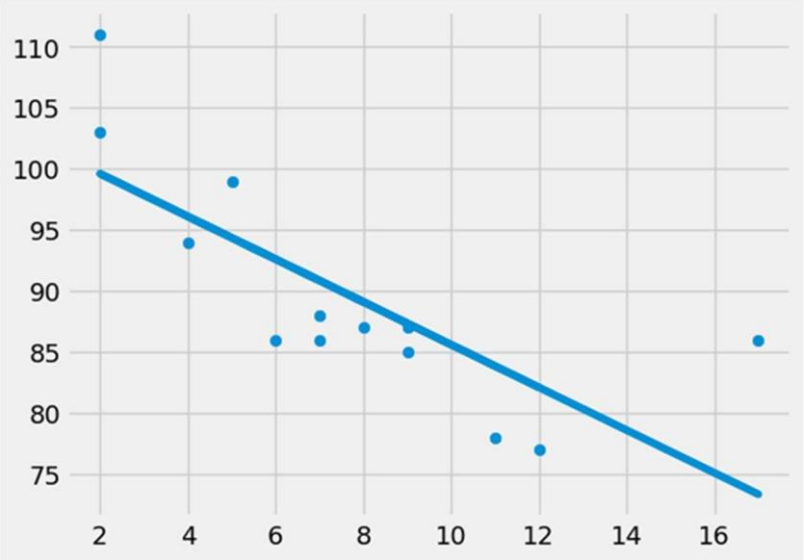
1. **Aim:** Implementing Linear Regression and Logistic Regression Models.

2. **Objective:** The objective of this experiment is to implement Linear Regression and Logistic Regression Models.

3. **Program and output:**

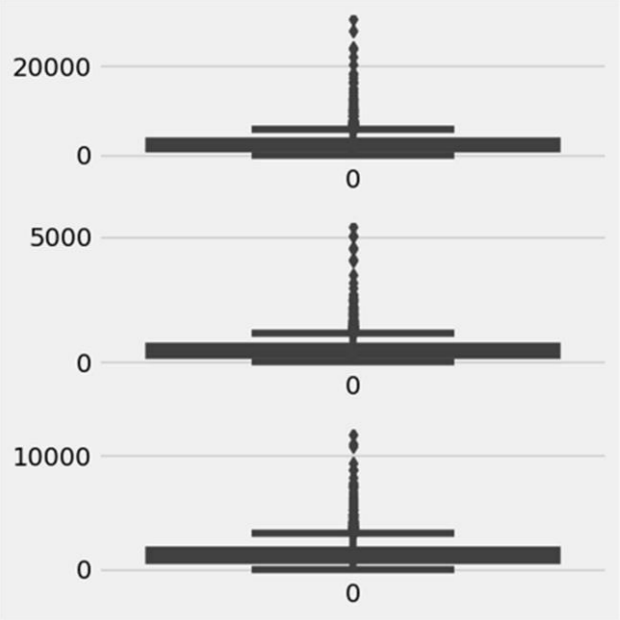
A) Linear Regression

```
import matplotlib.pyplot as plt
from scipy import stats
x = [5,7,8,7,2,17,2,9,4,11,12,9,6]
y = [99,86,87,88,111,86,103,87,94,78,77,85,86]
slope, intercept, r, p, std_err = stats.linregress(x, y)
def myfunc(x):
return slope * x + intercept
mymodel = list(map(myfunc, x))
plt.scatter(x, y)
plt.plot(x, mymodel)
plt.show()
```

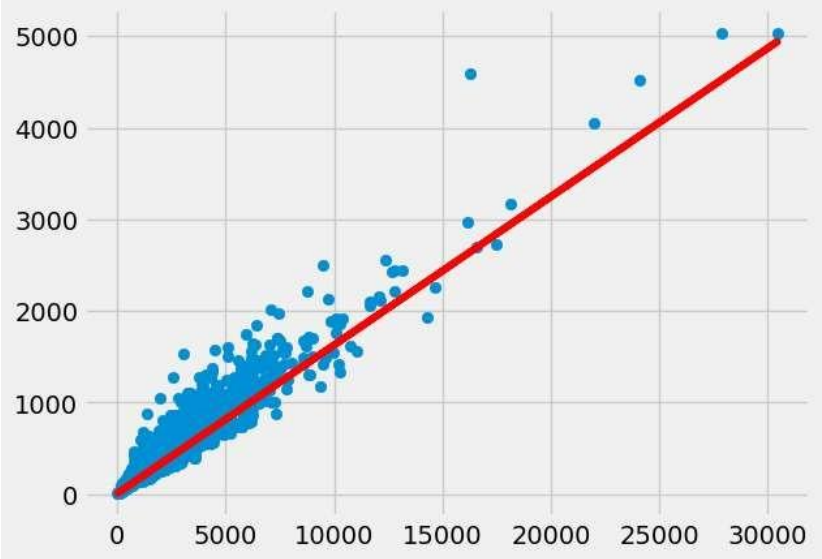


advertising.isnull().sum()*100/advertising.shape[0] longitude 0.0
latitude 0.0
housing_median_age 0.0
total_rooms 0.0
total_bedrooms 0.0
population 0.0
households 0.0
median_income 0.0
median_house_value 0.0
dtype: float64

```
fig, axs = plt.subplots(3, figsize = (5,5))
plt1 = sns.boxplot(advertising['total_rooms'], ax = axs[0])
plt2 = sns.boxplot(advertising['total_bedrooms'], ax = axs[1]) plt3 = sns.boxplot(advertising['population'], ax = axs[2])
plt.tight_layout()
```




```
plt.scatter(X_train,  y_train)
plt.plot(X_train, 6.948 + 0.162*X_train, 'r')
plt.show()
```



B) Logistic Regression

```
import numpy as np
import pandas as pd
# Data Visualisation
import matplotlib.pyplot as plt import seaborn as sns
advertising=pd.DataFrame(pd.read_csv("/content/sample_data/california_housing_test.csv"))
advertising.head()
```

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	median_house_value
0	-122.05	37.37	27.0	3885.0	661.0	1537.0	606.0	6.6085	344700.0
1	-118.30	34.26	43.0	1510.0	310.0	809.0	277.0	3.5990	176500.0
2	-117.81	33.78	27.0	3589.0	507.0	1484.0	495.0	5.7934	270500.0
3	-118.36	33.82	28.0	67.0	15.0	49.0	11.0	6.1359	330000.0
4	-119.67	36.33	19.0	1241.0	244.0	850.0	237.0	2.9375	81700.0

```
advertising.shape (3000, 9)
advertising.info()
<class 'pandas.core.frame.DataFrame'> RangeIndex: 3000 entries, 0 to 2999
Data columns (total 9 columns):
```

#	Column	Non-Null Count	Dtype
0	longitude	3000 non-null	float64
1	latitude	3000 non-null	float64
2	housing_median_age	3000 non-null	float64
3	total_rooms	3000 non-null	float64

4	total_bedrooms	3000 non-null	float64
5	population	3000 non-null	float64
6	households	3000 non-null	float64
7	median_income	3000 non-null	float64
8	median_house_value	3000 non-null	float64

dtypes: float64(9)

memory usage: 211.1 KB

advertising.describe()

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	median_hous
count	3000.000000	3000.00000	3000.000000	3000.000000	3000.000000	3000.000000	3000.00000	3000.000000	30
mean	-119.589200	35.63539	28.845333	2599.578667	529.950667	1402.798667	489.91200	3.807272	2058
std	1.994936	2.12967	12.555396	2155.593332	415.654368	1030.543012	365.42271	1.854512	1131
min	-124.180000	32.56000	1.000000	6.000000	2.000000	5.000000	2.00000	0.499900	225
25%	-121.810000	33.93000	18.000000	1401.000000	291.000000	780.000000	273.00000	2.544000	1212
50%	-118.485000	34.27000	29.000000	2106.000000	437.000000	1155.000000	409.50000	3.487150	1776
75%	-118.020000	37.69000	37.000000	3129.000000	636.000000	1742.750000	597.25000	4.656475	2639
max	-114.490000	41.92000	52.000000	30450.000000	5419.000000	11935.000000	4930.00000	15.000100	5000

Learning Outcomes:

- This experiment demonstrates us how to use a dataset or extract datasets from Kaggle.
- Perform various regression on them like Logistics and Linear Regression.
- How to implement Linear Regression on data set and make predictions.
- How to implement Logistic Regression on data set and make predictions