UNIVERSITY OF MUMBAI

DEPARTMENT OF COMPUTER SCIENCE

M.Sc. Computer Science with Spl. in Data Science – Semester III

Predictive Modeling and Analytics

JOURNAL

2022-2023

Seat No. 30283

मुंबई विद्यापीठ
University of Mumbai
Re-accredited with A++ Grade
(CGPA 3.65) by NAAC (3rd Cycle 2021)

UNIVERSITY OF MUMBAI

DEPARTMENT OF COMPUTER SCIENCE

CERTIFICATE

This is to certify that the work entered in this journal was done in the University Department of Computer Science laboratory by Mr./~~Ms.~~ **Sumon Singh** Seat No. **30283** for the course of M.Sc. Computer Science with Spl. in Data Science - Semester III (CBCS) (Revised) during the academic year 2022-2023 in a satisfactory manner.

Subject In-charge

Head of Department

External Examiner

Index

Practical - 1

Least Square estimation

Problem :

The following table relates to the tourist arrivals during 1990 to 1996 in India:

Years:	1990	1991	1992	1993	1994	1995	1996
Tourist's arrivals: (in millions)	18	20	23	25	24	28	30

Fit a straight line trend by the method of least squares and estimates the number of tourists that would arrives in the year 2000.

Libraries

In [1]:

```
import numpy as np
import pandas as pd
from sklearn.linear_model import LinearRegression
import matplotlib.pyplot as plt
```

Method : 1

In [2]:

```
years = [1990,1991,1992,1993,1994,1995,1996]
tourists = [18,20,23,25,24,28,30]
```

In [3]:

```
years=np.array(years).reshape(-1,1)
```

In [4]:

```
model = LinearRegression().fit(years,tourists)
```

In [5]:

```
model.intercept_
```

Out[5]:

```
-3748.464285714285
```

In [6]:

```
model.coef_
```

Out[6]:

```
array([1.89285714])
```

In [7]:

```
model.predict(np.array([2000]).reshape(-1,1))
```

Out[7]:

```
array([37.25])
```

Method : 2

In [8]:

```
x= [1990,1991,1992,1993,1994,1995,1996]
y= [18,20,23,25,24,28,30]
```

In [9]:

```
m,c=np.polyfit(x,y,deg= 1)
```

In [10]:

```
m
```

Out[10]:

```
1.8928571428569125
```

In [11]:

```
c
```

Out[11]:

```
-3748.464285713826
```

In [12]:

```
y_line= [m*i+c for i in x]
```

In [13]:

```
y_line
```

Out[13]:

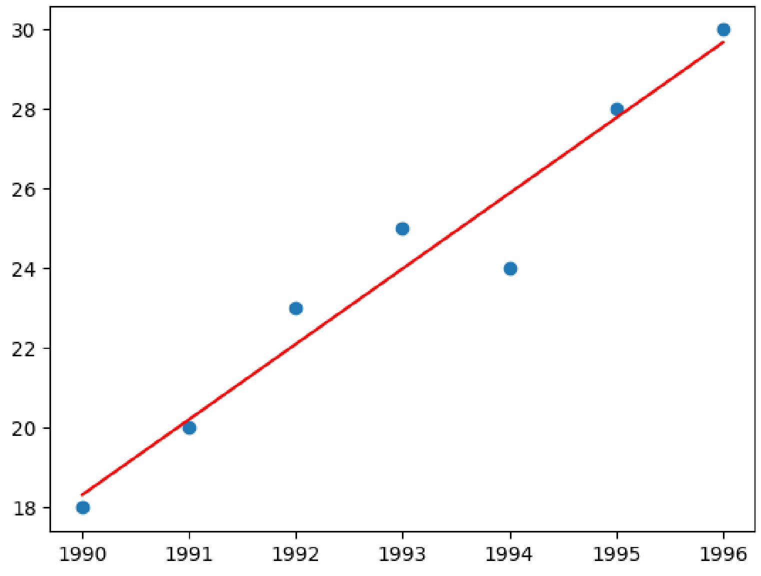
```
[18.321428571429806,
 20.21428571428669,
 22.10714285714357,
 24.000000000000455,
 25.892857142857338,
 27.785714285714675,
 29.67857142857156]
```

In [14]:

```
plt.scatter(x,y)
plt.plot(x,y_line,'r')
```

Out[14]:

```
[<matplotlib.lines.Line2D at 0x7f247a745f60>]
```



Tourists in 2000

In [15]:

```
m*2000+c
```

Out[15]:

```
37.24999999999909
```

Practical - 2

Trend Values Using Least-Squares

Problem

Below are given the figures of production (in thousand quintals) of a sugar factory.

Year	Production (thousand quintals)
1993	77
1995	88
1996	94
1997	85
1998	91
1999	98
2002	90

(i) Fit a straight line by the least squares' method and tabulate the trend values.

Libraries

In [1]:

```
import numpy as np
import matplotlib.pyplot as plt
```

Code

In [2]:

```
Years=[1993,1995,1996,1997,1998,1999,2002]
Production=[77,88,94,85,91,98,90]
```

In [3]:

```
m,c=np.polyfit(Years,Production,deg=1)
```

In [4]:

```
m,c
```

Out[4]:

```
(1.3764044943820066, -2659.8764044943505)
```

In [5]:

```
y_line=[m*i+c for i in Years]
```

Trend values

In [6]:

```
y_line
```

Out[6]:

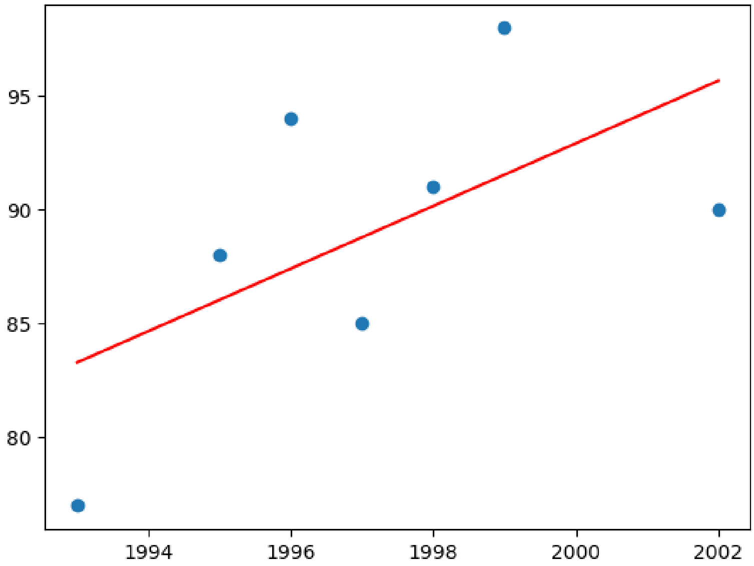
```
[83.29775280898866,
 86.05056179775283,
 87.4269662921347,
 88.80337078651655,
 90.17977528089887,
 91.55617977528073,
 95.68539325842676]
```

In [7]:

```
plt.scatter(Years,Production)
plt.plot(Years,y_line,'r')
```

Out[7]:

```
[<matplotlib.lines.Line2D at 0x7f4e78d96ce0>]
```



Practical - 3

Linear Regression Equation

Problem

The following measurements have been obtained in a study:

Nr.	1	2	3	4	5	6	7	8
y	9.29	12.67	12.42	0.32	20.77	9.52	2.38	7.46
X1	1.00	2.00	3.00	4.00	5.00	6.00	7.00	8.00
X2	4.00	12.00	16.00	8.00	32.00	24.00	20.00	28.00

The estimated linear regression equation is: $y = bo + b_1{*}x_1 + b_2{*}X_2$

Libraries

In [10]:

```
from sklearn.linear_model import LinearRegression
import pandas as pd
```

Code

In [11]:

```
x1 = [1.00,2.00,3.00,4.00,5.00,6.00,7.00,8.00]
x2 = [4.00,12.00,16.00,8.00,32.00,24.00,20.00,28.00]
y = [9.29,12.67,12.42,0.38,20.77,9.52,2.38,7.46]
```

In [12]:

```
df = pd.DataFrame({'X1':x1,'X2':x2,'Y':y})
```

In [13]:

```
df
```

Out[13]:

	X1	X2	Y
0	1.0	4.0	9.29
1	2.0	12.0	12.67
2	3.0	16.0	12.42
3	4.0	8.0	0.38
4	5.0	32.0	20.77
5	6.0	24.0	9.52
6	7.0	20.0	2.38
7	8.0	28.0	7.46

In [14]:

```
x= df[['X1','X2']]
y = df['Y']
```

In [15]:

```
lm = LinearRegression().fit(x,y)
```

In [16]:

```
lm.intercept_
```

Out[16]:

```
8.032533783783787
```

In [17]:

```
lm.coef_
```

Out[17]:

```
array([-3.57336486,  0.96715878])
```

In [18]:

```
print(f'The estimated linear regression equation will be :- y = {lm.intercept_} + {lm.coef_[0]}*x1 + {lm.coef_[1]}*x2'
```

```
The estimated linear regression equation will be :- y = 8.032533783783787 + -3.573364864864868*x1 + 0.9
671587837837844*x2
```

Practical - 4

Linear Regression Equation

Problem

The following measurements have been obtained in a study:

Nr.	1	2	3	4	5	6	7	8	9	10	11	12	13
Y	1.45	1.93	0.81	0.61	1.55	0.95	0.45	1.14	0.74	0.98	1.41	0.81	0.89
X1	0.58	0.86	0.29	0.20	0.56	0.28	0.08	0.41	0.22	0.35	0.59	0.22	0.26
X2	0.71	0.13	0.79	0.20	0.56	0.92	0.01	0.60	0.70	0.73	0.13	0.96	0.27

Nr.	14	15	16	17	18	19	20	21	22	23	24	25
Y	0.68	1.39	1.53	0.91	1.49	1.38	1.73	1.11	1.68	0.66	0.69	1.98
X1	0.12	0.65	0.70	0.30	0.70	0.39	0.72	0.45	0.81	0.04	0.20	0.95
X2	0.21	0.88	0.30	0.15	0.09	0.17	0.25	0.30	0.32	0.82	0.98	0.00

The estimated linear regression equation is: $y = bo + b_1*x_1 + b_2*x_2$

Libraries

In [19]:

```
from sklearn.linear_model import LinearRegression
import pandas as pd
```

Code

In [20]:

```
x1 = [0.58,0.86,0.29,0.20,0.56,0.28,0.08,0.41,0.22,0.35,0.59,0.22,0.26,0.12,0.65,0.70,0.30,0.70,0.39,0.72,0.45,0.81,0.
x2 = [0.71,0.13,0.79,0.20,0.56,0.92,0.01,0.60,0.70,0.73,0.13,0.96,0.27,0.21,0.88,0.30,0.15,0.09,0.17,0.25,0.30,0.32,0.
y = [1.45,1.93,0.81,0.61,1.55,0.95,0.45,1.14,0.74,0.98,1.41,0.81,0.89,0.68,1.39,1.53,0.91,1.49,1.38,1.73,1.11,1.68,0.6
```

In [21]:

```
df = pd.DataFrame({'X1':x1,'X2':x2,'Y':y})
```

In [22]:

```
df
```

Out[22]:

	X1	X2	Y
0	0.58	0.71	1.45
1	0.86	0.13	1.93
2	0.29	0.79	0.81
3	0.20	0.20	0.61
4	0.56	0.56	1.55
5	0.28	0.92	0.95
6	0.08	0.01	0.45
7	0.41	0.60	1.14
8	0.22	0.70	0.74
9	0.35	0.73	0.98
10	0.59	0.13	1.41
11	0.22	0.96	0.81
12	0.26	0.27	0.89
13	0.12	0.21	0.68
14	0.65	0.88	1.39
15	0.70	0.30	1.53
16	0.30	0.15	0.91
17	0.70	0.09	1.49
18	0.39	0.17	1.38
19	0.72	0.25	1.73
20	0.45	0.30	1.11
21	0.81	0.32	1.68
22	0.04	0.82	0.66
23	0.20	0.98	0.69
24	0.95	0.00	1.98

In [23]:

```
x= df[['X1','X2']]
y = df['Y']
```

In [24]:

```
lm = LinearRegression().fit(x,y)
```

In [25]:

```
lm.intercept_
```

Out[25]:

```
0.433547115055518506
```

In [26]:

```
lm.coef_
```

Out[26]:

```
array([1.65299345, 0.00394488])
```

In [27]:

```
print(f'The estimated linear regression equation will be :- y = {lm.intercept_} + {lm.coef_[0]}*x1 + {lm.coef_[1]}*x2'
```

```
The estimated linear regression equation will be :- y = 0.433547115055518506 + 1.6529934509657123*x1 +
0.0039448751847171865*x2
```

Practical - 5

Autocorrelation

Problem

Calculate the Lag 3 Autocorrelation for the sample dataset below. This data has 24 observations (two years of monthly sales data)

Sr No	Original Data	1-Unit Lag	2-Unit Lag	3-Unit Lag
1	9.08			
2	12.63	9.08		
3	15	12.63	9.08	
4	20.73	15	12.63	9.08
5	2.2	20.73	15	12.63
6	18	2.2	20.73	15
7	7.16	18	2.2	20.73
8	18.28	7.16	18	2.2
9	21	18.28	7.16	18
10	19.68	21	18.28	7.16
11	15.54	19.68	21	18.28
12	24	15.54	19.68	21
13	16.1	24	15.54	19.68
14	11.93	16.1	24	15.54
15	27	11.93	16.1	24
16	12.51	27	11.93	16.1
17	20.04	12.51	27	11.93
18	30	20.04	12.51	27
19	12.41	30	20.04	12.51
20	14.33	12.41	30	20.04
21	33	14.33	12.41	30
22	22.11	33	14.33	12.41
23	17.91	22.11	33	14.33
24	36	17.91	22.11	33

Libraries

In [23]:

```
import statsmodels.api as sm
from statsmodels.graphics import tsaplots
import matplotlib.pyplot as plt
import pandas as pd
```

Code

In [24]:

```
data = [9.08,12.63,15,20.73,2.2,18,7.16,18.28,21,19.68,15.54,
       24,16.1,11.93,27,12.51,20.04,30,12.41,14.33,33,22.11,
       17.91,36]
```

In [25]:

```
df = pd.DataFrame({'Original_Data':data})
```

In [26]:

```
One_Unit_Lag = df.shift(1,axis=0)
```

In [27]:

```
Second_Unit_Lag = df.shift(2,axis=0)
```

In [28]:

```
Third_Unit_Lag = df.shift(3,axis=0)
```

In [29]:

```
df['1_Unit_Lag'] = One_Unit_Lag
df['2_Unit_Lag'] = Second_Unit_Lag
df['3_Unit_Lag'] = Third_Unit_Lag
```

In [30]:

```
df
```

Out[30]:

	Original_Data	1_Unit_Lag	2_Unit_Lag	3_Unit_Lag
0	9.08	NaN	NaN	NaN
1	12.63	9.08	NaN	NaN
2	15.00	12.63	9.08	NaN
3	20.73	15.00	12.63	9.08
4	2.20	20.73	15.00	12.63
5	18.00	2.20	20.73	15.00
6	7.16	18.00	2.20	20.73
7	18.28	7.16	18.00	2.20
8	21.00	18.28	7.16	18.00
9	19.68	21.00	18.28	7.16
10	15.54	19.68	21.00	18.28
11	24.00	15.54	19.68	21.00
12	16.10	24.00	15.54	19.68
13	11.93	16.10	24.00	15.54
14	27.00	11.93	16.10	24.00
15	12.51	27.00	11.93	16.10
16	20.04	12.51	27.00	11.93
17	30.00	20.04	12.51	27.00
18	12.41	30.00	20.04	12.51
19	14.33	12.41	30.00	20.04
20	33.00	14.33	12.41	30.00
21	22.11	33.00	14.33	12.41
22	17.91	22.11	33.00	14.33
23	36.00	17.91	22.11	33.00

In [31]:

```
lag_no = 3
```

In [32]:

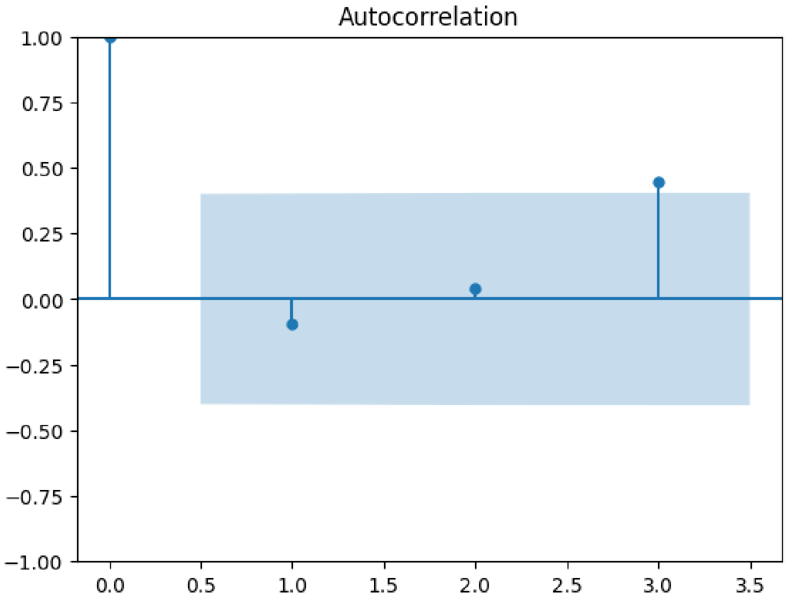
```
sm.tsa.acf(data,nlags=lag_no)
```

Out[32]:

```
array([ 1.        , -0.0928397 ,  0.03872485,  0.44511117])
```

In [33]:

```
fig = tsaplots.plot_acf(data,lags=lag_no)
plt.show()
```



Autocorrelation for 3 lags is 0.44511117

Practical - 6

T-test

Problem

Ten rats were fed with rice in 1st month and body weights of the rats were recorded. In the next month, they were fed with grams and their weights were measured again. The respective weights of ten rats in two months are as follows:

Weights in 1 month	50	60	58	52	51	62	58	55	50	65
Weights in 2 month	56	58	68	61	56	59	64	60	50	62

Prove the hypothesis
Ho: Weights of 1st and 2nd months are equal
Given: the tabulated value for 5% is 1.833

Libraries

In [1]:

```
from scipy import stats
import pingouin as pg
```

Code

In [2]:

```
alpha = 0.05
t_val = 1.833
```

In [3]:

```
weights_1st_month = [50,60,58,52,51,62,58,55,50,65]
weights_2nd_month = [56,58,68,61,56,59,64,60,50,62]
```

In [4]:

```
t_result,p_val = stats.ttest_rel(weights_1st_month,weights_2nd_month)
```

In [5]:

```
t_result,p_val
```

Out[5]:

```
(-2.129647923401715, 0.062056988380769965)
```

In [6]:

```
if p_val <= alpha:
    print('Reject Null Hypothesis')
else:
    print('Accept Null Hypothesis')
```

```
Accept Null Hypothesis
```

Alternative Way

In [7]:

```
pg.ttest(weights_1st_month,weights_2nd_month,paired=True)
```

Out[7]:

	T	dof	alternative	p-val	CI95%	cohen-d	BF10	power
T-test	-2.129648	9	two-sided	0.062057	[-6.81, 0.21]	0.643549	1.498	0.443569

Practical - 7

F-Test

Problem

From the data given below, find out whether the mean of the three samples differ significantly or not.

Sample 1	Sample 2	Sample 3
20	19	13
10	13	12
17	17	10
17	12	15
16	9	5

Given the tabulated F-score is 2.9 for 5% significance level.

Libraries

In [1]:

```
from scipy.stats import f_oneway
```

Code

In [2]:

```
Sample_1 = [20,10,17,17,16]
Sample_2 = [19,13,17,12,9]
Sample_3 = [13,12,10,15,5]
```

In [3]:

```
alpha = 0.05
```

In [4]:

```
result,p_val = f_oneway(Sample_1,Sample_2,Sample_3)
```

In [5]:

```
result,p_val
```

Out[5]:

```
(2.159090909090909, 0.15814551132051272)
```

In [6]:

```
if p_val <= alpha:
    print('Reject Null Hypothesis')
else:
    print('Accept Null Hypothesis')
```

```
Accept Null Hypothesis
```

T Test

Weights of 1st Month	Weights Of 2nd Month
50	56
60	58
58	68
52	61
51	56
62	59
58	64
55	60
50	50
65	62

Result

Paired t-test

Alpha	0.05	
Hypothesized Mean Difference	0	
	Variable 1	Variable 2
Mean	56.1	59.4
Variance	28.3222222222222	24.26666666667
Observations	10	10
Pearson Correlation	0.545041704175871	
Observed Mean Difference	-3.3	
Variance of the Differences	24.0111111111111	
df	9	
t Stat	-2.12964792340171	
P (T<=t) one-tail	0.0310284941903851	
t Critical one-tail	1.83311293265624	
P (T<=t) two-tail	0.0620569883807702	
t Critical two-tail	2.26215716279821	

Conclusion :

As P-value of two-tail t-test is higher than alpha value of this problem, we accept the Null Hypothesis

Practical-8

Anova test

Problem:

In an experiment, the men yields of three rice varieties grown with four nitrogen rates were recorded. Analyze the data using the test of analysis of variance to determine whether there is any difference in the mean yield of three varieties with nitrogen doses. The results are given in the following table

Nitrogen rate kg/ha	V1	V2	V3
0	4.50	5.01	6.11
30	4.30	6.17	6.92
60	5.60	6.37	6.37
90	5.21	6.48	6.48

Problem - 8

grown with four nitrogen rates were recorded. Analyze the data using the test of analysis of variance to determine whether there is any difference in the mean yield of three varieties with nitrogen doses. The results are given in the following table

Nitrogen rate kg/ha	V1	V2	V3
0	4.5	5.01	6.11
30	4.3	6.17	6.92
60	5.6	6.37	6.37
90	5.21	6.48	6.48

Result

Anova: Two-Factor Without Replication

SUMMARY	*Count*	*Sum*	*Average*	*Variance*
0	3	15.62	5.206666667	0.677033333
30	3	17.39	5.796666667	1.820633333
60	3	18.34	6.113333333	0.197633333
90	3	18.17	6.056666667	0.537633333
V1	4	19.61	4.9025	0.368691667
V2	4	24.03	6.0075	0.458691667
V3	4	25.88	6.47	0.114066667

ANOVA

Source of Varia	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Rows	1.5478	3	0.515933333	2.424973562	0.16386994	4.757062664
Columns	5.189316667	2	2.594658333	12.19533117	0.007695436	5.14325285
Error	1.27655	6	0.212758333			
Total	8.013666667	11				