# Software Engineering Tools Lab

## Assignment No-2

**Name** : Onkar Santosh Gavali

**Prn** : 2019BTECS00037

**Name** : Sadaf Najeem Mulla

**Prn** : 2019BTECS00038

## Q2

*Implement linear regression problem using Google colab (Perform preprocessing, training and testing)*

## Importing modules

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

## Import dataset

```
# import file and upadate the file path
URL =  "/content/Daily_Demand_Forecasting_Orders.csv"

dataset = pd.read_csv(URL, sep=';')


dataset.head(10)
```

| | Week of the month (first week, second, third, fourth or fifth week | Day of the week (Monday to Friday) | Non-urgent order | Urgent order | Order type A | Order type B | Order type C | Fiscal sector orders | Orders from the traffic controller sector | Banking orders (1) | Banking orders (2) | Banking orders (3) | Target (Total orders) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | 4 | 316.307 | 223.270 | 61.543 | 175.586 | 302.448 | 0.000 | 65556 | 44914 | 188411 | 14793 | 539.577 |
| **1** | 1 | 5 | 128.633 | 96.042 | 38.058 | 56.037 | 130.580 | 0.000 | 40419 | 21399 | 89461 | 7679 | 224.675 |
| **2** | 1 | 6 | 43.651 | 84.375 | 21.826 | 25.125 | 82.461 | 1.386 | 11992 | 3452 | 21305 | 14947 | 129.412 |
| **3** | 2 | 2 | 171.297 | 127.667 | 41.542 | 113.294 | 162.284 | 18.156 | 49971 | 33703 | 69054 | 18423 | 317.120 |
| **4** | 2 | 3 | 90.532 | 113.526 | 37.679 | 56.618 | 116.220 | 6.459 | 48534 | 19646 | 16411 | 20257 | 210.517 |
| **5** | 2 | 4 | 110.925 | 96.360 | 30.792 | 50.704 | 125.868 | 79.000 | 52042 | 8773 | 47522 | 24966 | 207.364 |
| **6** | 2 | 5 | 144.124 | 118.919 | 43.304 | 66.371 | 153.368 | 0.000 | 46573 | 33597 | 48269 | 20973 | 263.043 |
| **7** | 2 | 6 | 119.379 | 113.870 | 38.584 | 85.961 | 124.413 | 15.709 | 35033 | 26278 | 56665 | 18502 | 248.958 |
| **8** | 3 | 2 | 218.856 | 124.381 | 33.973 | 148.274 | 162.044 | 1.054 | 66612 | 19461 | 103376 | 10458 | 344.291 |
| **9** | 3 | 3 | 146.518 | 101.045 | 36.399 | 43.306 | 168.723 | 865.000 | 58224 | 7742 | 82395 | 11948 | 248.428 |

```
dataset.describe()
```

| | Week of the |
|---|---|

| | month (first week, second, third, fourth or fifth week | Day of the week (Monday to Friday) | Non-urgent order | Urgent order | Order type A | Order type B | Order type C | Fiscal sector orders | Orders from the traffic controller sector | Banking orders (1) | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **count** | 60.000000 | 60.000000 | 60.000000 | 60.000000 | 60.000000 | 60.000000 | 60.000000 | 60.000000 | 60.000000 | 60.000000 | |
| **mean** | 3.016667 | 4.033333 | 172.554933 | 118.920850 | 52.112217 | 109.229850 | 139.531250 | 77.396133 | 44504.350000 | 46640.833333 | 1 |
| **std** | 1.282102 | 1.401775 | 69.505788 | 27.170929 | 18.829911 | 50.741388 | 41.442932 | 186.502470 | 12197.905134 | 45220.736293 | 4 |
| **min** | 1.000000 | 2.000000 | 43.651000 | 77.371000 | 21.826000 | 25.125000 | 74.372000 | 0.000000 | 11992.000000 | 3452.000000 | |
| **25%** | 2.000000 | 3.000000 | 125.348000 | 100.888000 | 39.456250 | 74.916250 | 113.632250 | 1.243250 | 34994.250000 | 20130.000000 | 5 |
| **50%** | 3.000000 | 4.000000 | 151.062500 | 113.114500 | 47.166500 | 99.482000 | 127.990000 | 7.831500 | 44312.000000 | 32527.500000 | 6 |
| **75%** | 4.000000 | 5.000000 | 194.606500 | 132.108250 | 58.463750 | 132.171000 | 160.107500 | 20.360750 | 52111.750000 | 45118.750000 | 9 |
| **max** | 5.000000 | 6.000000 | 435.304000 | 223.270000 | 118.178000 | 267.342000 | 302.448000 | 865.000000 | 71772.000000 | 210508.000000 | 1 |

```
df= dataset[["Non-urgent order" , "Target (Total orders)"]]
df.head()
```

| | Non-urgent order | Target (Total orders) |
|---|---|---|
| **0** | 316.307 | 539.577 |

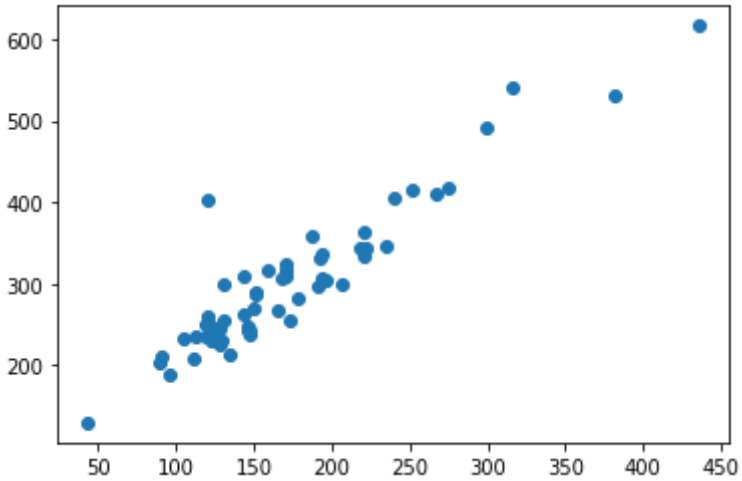
| | | |
|---|---|---|
| **1** | 128.633 | 224.675 |
| **2** | 43.651 | 129.412 |

```
X = df[["Non-urgent order"]]
```

```
Y= df[["Target (Total orders)"]]
```

```
import matplotlib.pyplot as plt
plt.scatter(X, Y)
```

```
<matplotlib.collections.PathCollection at 0x7fd67e436f10>
```




## dataset spliting

```
from sklearn.model_selection import train_test_split
X_train , X_test , Y_train , Y_test = train_test_split(X,Y , test_size = 0.25)
```

```
X_train.shape
```

```
(45, 1)
```

```
(15, 1)
```

```
X_test.shape
```

```
(15, 1)
```

```
Y_train.shape
```

```
(45, 1)
```

```
Y_test.shape
```

```
(15, 1)
```

## Linear Regression

```
from sklearn.linear_model import LinearRegression
```

```
lm = LinearRegression()
```

```
lm.fit(X_train , Y_train)
```

```
LinearRegression()
```

```
lm.coef_
```

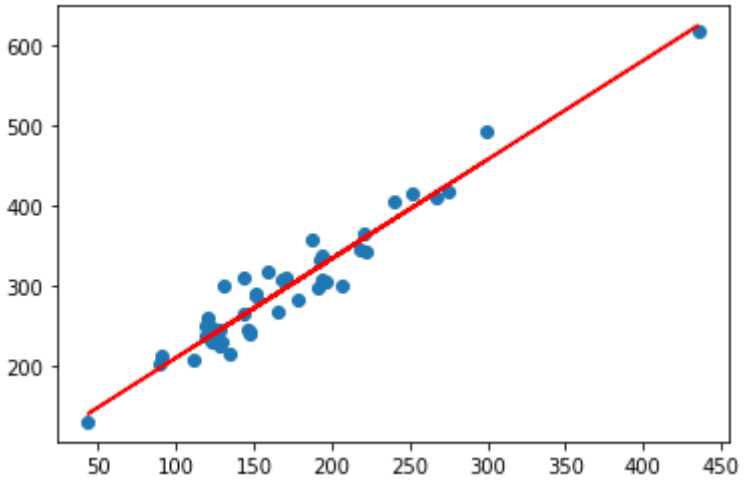
```
array([[1.23759091]])
```

```
lm.intercept_
```

```
array([85.17166465])
```

```
plt.scatter(X_train , Y_train)
plt.plot(X_train , X_train * lm.coef_ + lm.intercept_ , '-r')
```

```
[<matplotlib.lines.Line2D at 0x7fd67e154150>]
```

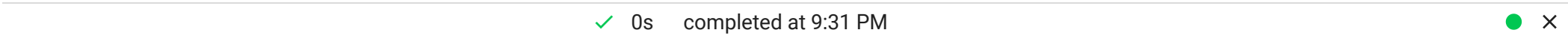



```
from sklearn.metrics import mean_squared_error, r2_score


Y_pred=lm.predict(X_test)


print("Mean squared error: %.2f"
% mean_squared_error(Y_test, Y_pred))
# Explained variance score: 1 is perfect prediction
print('Variance score: %.2f' % r2_score(Y_test, Y_pred))
```

```
Mean squared error: 2614.52
Variance score: 0.74
```


0s
completed at 9:31 PM