

MACHINE LEARNING ASSIGNMENT

Problem Statement:

Take a toy dataset or any mathematic problem which involves prediction, solve it using any machine learning algorithm of choice and show the actual value via working out the math.

Dataset Preview:

area	price					
2600	550000					
3000	565000					
3200	610000					
3600	595000					
4000	760000					
4100	810000					

This dataset contains 6 rows and 2 columns. We use this dataset to find the relationship between area and price by training with machine learning algorithm -Linear Regression.

Simple linear regression is a statistical technique used for finding the existence of an association relationship between a dependent variable (outcome variable i.e., y) and an independent variable (predictor variable or feature i.e., x). We can only establish that change in the value of the outcome variable (Y) is associated with change in the value of feature (X). When we consider an elementary algebra the equation

for a line is “ $Y=mx+c$ ” . Here, we are going to calculate linear regression and predict the equation “ $y=a+bx$ ”.

$$a = \frac{[(\sum y)(\sum x^2) - (\sum x)(\sum xy)]}{[n(\sum x^2) - (\sum x)^2]}$$

$$b = \frac{[n(\sum xy) - (\sum x)(\sum y)]}{[n(\sum x^2) - (\sum x)^2]}$$

Program:

Importing the libraries

```
import pandas as pd
```

```
import matplotlib.pyplot as plt
```

```
import numpy as np
```

```
#reading the dataset
```

```
data=pd.read_csv("C:/Users/tarun/Desktop/price.csv")
```

data.head()

	area	price
0	2600	550000
1	3000	565000
2	3200	610000
3	3600	595000
4	4000	760000

data.tail()

	area	price
1	3000	565000
2	3200	610000
3	3600	595000
4	4000	760000
5	4100	810000

data.shape

(6, 2)

data.isnull().sum()

area 0

```
price    0
```

```
dtype: int64
```

```
x=data.iloc[:, :-1].values
```

```
y=data.iloc[:, -1].values
```

```
print(x)
```

```
[[ 2600]  
 [ 3000]  
 [ 3200]  
 [ 3600]  
 [ 4000]  
 [ 4100]]
```

```
print(y)
```

```
[550000 565000 610000 595000 760000 810000]
```

```
from sklearn.model_selection import
```

```
train_test_split
```

```
x_train,x_test,y_train,y_test=train_test_split(x,y,tes  
t_size=0.2)
```

```
x_train.shape
```

(4, 1)

y_train.shape

(4,)

y_test.shape

(2,)

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

```
model=LinearRegression()
```

```
model.fit(x_train,y_train)
```

```
LinearRegression()
```

```
y_pred=model.predict(x_test)
```

```
print(y_pred)
```

```
[610019.12045889 737896.74952199]
```

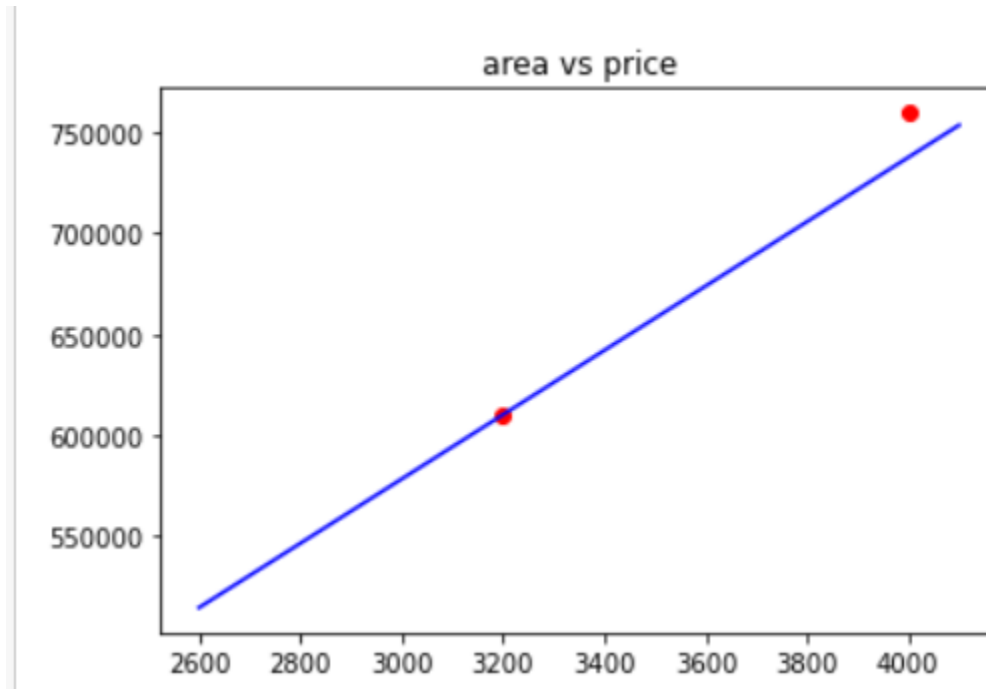
```
plt.scatter(x_test,y_test,color='red')
```

```
plt.plot(x_train,model.predict(x_train),color='blue')
```

```
plt.title('area vs price')
```

```
plt.xlabel("price")
```

```
plt.ylabel("area")
```



```
print(model.coef_)
```

```
[159.84703633]
```

```
print(model.intercept_)
```

```
98508.60420650104
```

```
y=model.predict([[20]])
```

print(y)

[101705.54493308]

mathematically solving linear Regression

area(x) price(y)

2600	550000
3000	565000
3200	610000
3600	595000
4000	760000
4100	810000

x	y	xy	x ²	y ²
2600	550000	1430000000	6760000	302500000000
3000	565000	1695000000	9000000	319225000000
3200	610000	1952000000	10240000	372100000000
3600	595000	2142000000	12960000	354025000000
4000	760000	3040000000	16000000	577600000000
4100	810000	3321000000	16810000	656100000000
20500	3890000	13510000000	71940000	2581550000000

$$a = \frac{[(\sum y)(\sum x^2) - (\sum x)(\sum xy)]}{[n(\sum x^2) - (\sum x)^2]}$$
$$a = \frac{[(3890000)(71940000) - (20500)(13510000000)]}{[6(71940000) - (20500)^2]}$$
$$= 159.84703633$$

$$b = \frac{[n(\sum xy) - (\sum x)(\sum y)]}{[n(\sum x^2) - (\sum x)^2]}$$
$$= \frac{[6(13510000000) - (20500)(3890000)]}{[6(71940000) - (20500)^2]}$$
$$= 98508.60420650104$$

$$y = 159.84703633(101705.54493308)(10)$$
$$y = 101705.54493308$$