9. Data Science – Machine Learning – Simple Linear Regression

Contents

1. Regression 2

2. A line 2

3. The goal 3

4. Linear Regression 4

5. Types of linear regression 4

6. Simple linear regression 4

7. Multiple linear regression 4

8. Simple linear regression example 5

8.1. Problem statement 5

8.2. The solution 5

9. Machine learning Terminology 6

9.1. Features 6

9.2. Label or target 6

9.3. Models 6

9.4. Prediction 6

9.5. Formula 6

10. Intercept and coefficient 17

11. Y = m * X + b (m is coefficient and b is intercept) 18

12. Best fitted line 19

13. Predicting a group of home prices 21

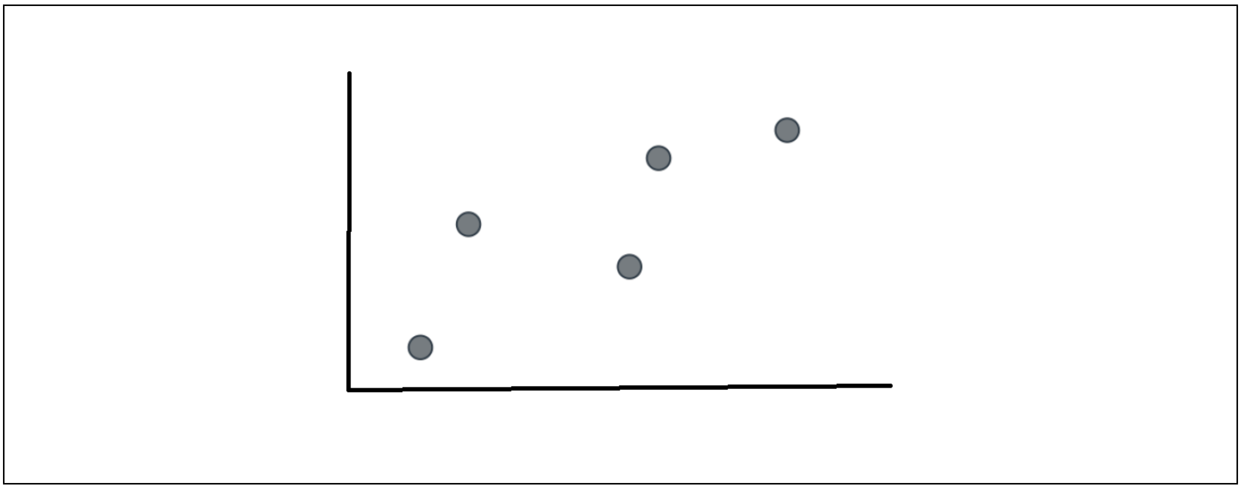
9. Data Science – Machine Learning – Simple Linear Regression

1. Regression

- ✓ Regression analysis is used to explain the relationship between a two variables.
- ✓ Also called as it’s a relationship in between dependent variable and one or more independent variables.

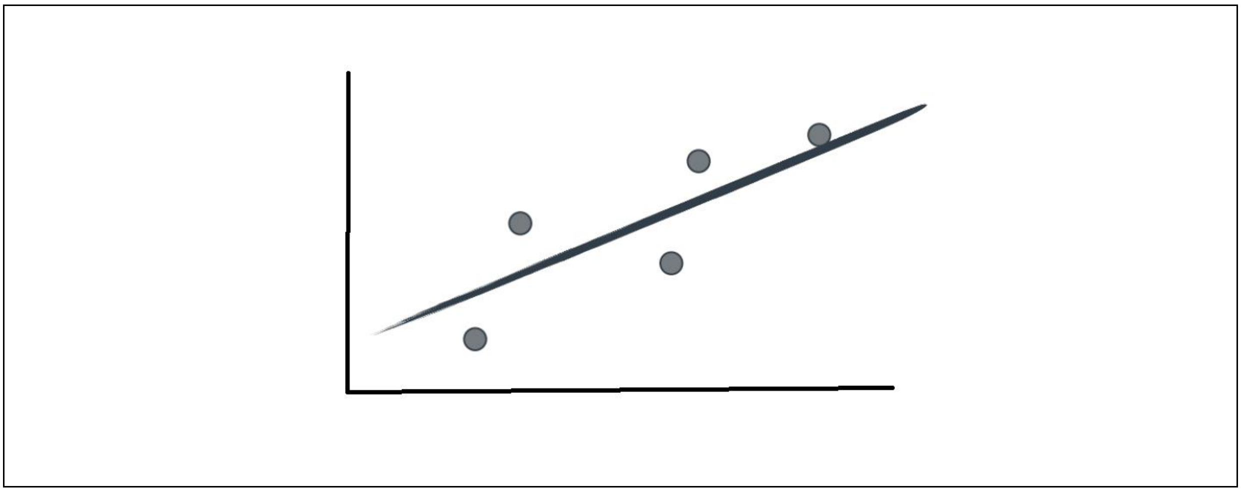
2. A line

- ✓ If two variables having relationship then if we draw this relationship in a two dimensional then we get a straight line.
- ✓ The picture of linear regression is simple.
- ✓ Let us say we have some points, a line will travel in between these points



3. The goal

- ✓ The goal of linear regression is to draw the best fitted line.
- ✓ Best fitted line means that the line which passes as close as possible to these points.



4. Linear Regression

- ✓ This is a technique and it explains the relationship between the dependent variable and independent variables

5. Types of linear regression

- ✓ There are two types of linear regression
  - Simple linear regression
  - Multiple linear regression

6. Simple linear regression

- ✓ When you have only 1 independent variable and 1 dependent variable, it is called simple linear regression.

7. Multiple linear regression

- ✓ When you have 2 or more independent variable and 1 dependent variable, it is called multiple linear regression.

8. Simple linear regression example

- ✓ When you have only 1 independent variable and 1 dependent variable, it is called simple linear regression.

8.1. Problem statement

- ✓ Assuming that we are planning to buy a new house and need to predict the price of a house

8.2. The solution

- ✓ While buying house first we need to check the area of the house

area	price
2600	550000
3000	565000
3200	610000
3600	680000
4000	725000

9. Machine learning Terminology

9.1. Features

- ✓ From the given problem the feature is area and price

9.2. Label or target

- ✓ Price of the house

9.3. Models

- ✓ A machine learning model is simply a rule, or a formula, which predicts a label from the features.
- ✓ In this case, the model is the equation we found for the price.

9.4. Prediction

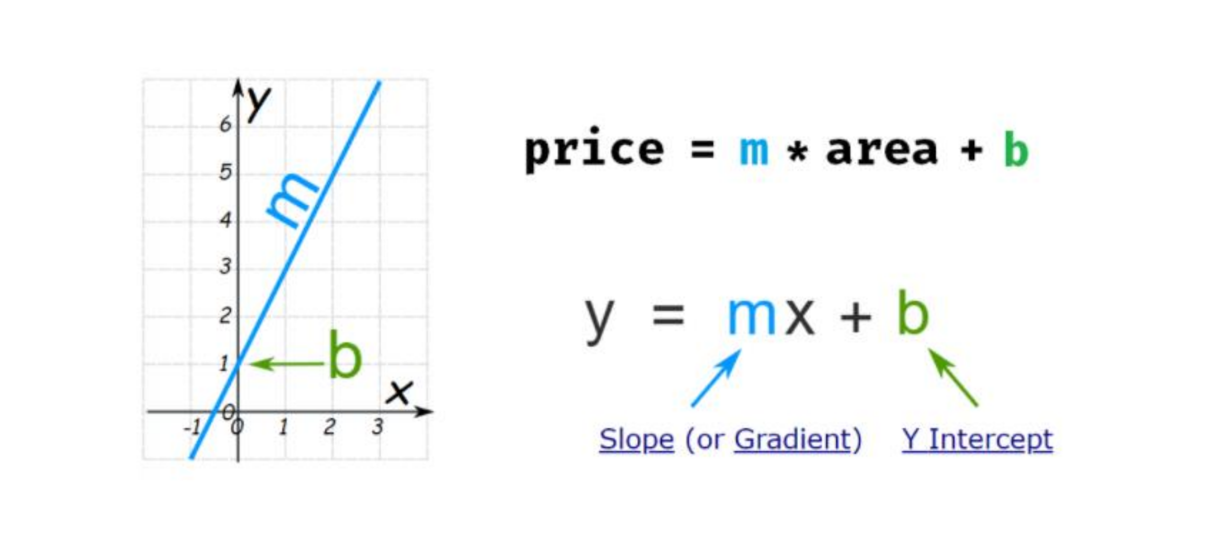
- ✓ The prediction is simply the output of the model.
- ✓ If the model gives the result as “Hey Guru I think the house with 36000 area is going to cost $300”, then the prediction is 300.

9.5. Formula

- ✓ Home price = m * (area) + b

Reminder

- ✓ Once please walk through our maths regression chapter (Chapter 7. Statistics - PART - 7 - Regression) which we have already discussed, thanks



price = m * area + b

$y = mx + b$

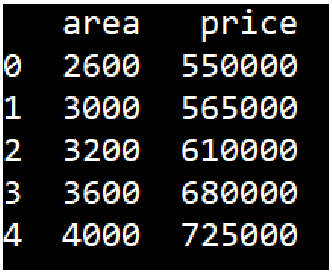
Slope (or Gradient) Y Intercept

Program Name

Loading house prices dataset
demo1.py

```
import pandas as pd

df = pd.read_csv("homeprices.csv")

print(df.head())
```

Output



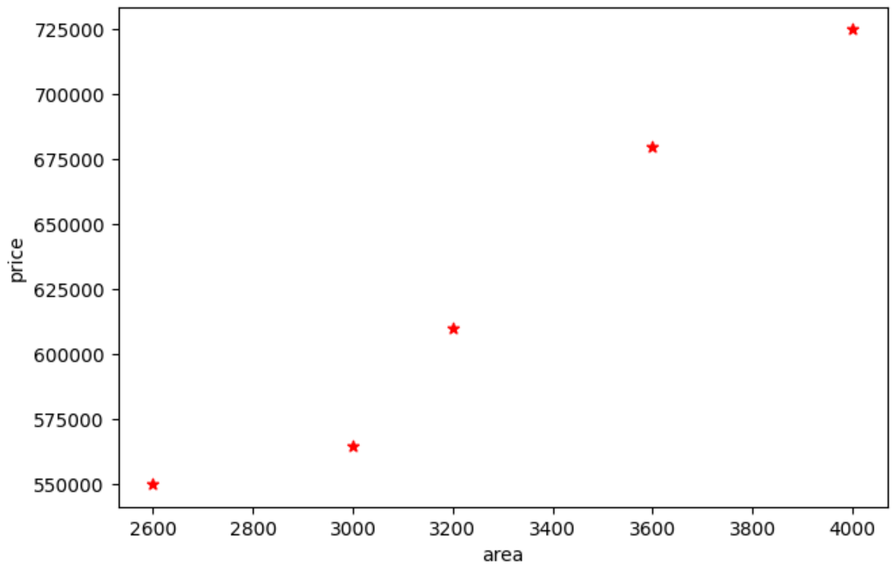
```
   area   price
0  2600  550000
1  3000  565000
2  3200  610000
3  3600  680000
4  4000  725000
```

Program Name

Creating scatter plot using matplotlib
demo2.py

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

df=pd.read_csv("homeprices.csv")

#  plotting the dataset

plt.xlabel('area')
plt.ylabel('price')

plt.scatter(df.area, df.price, color = 'red', marker = '*')

plt.show()
```

Output

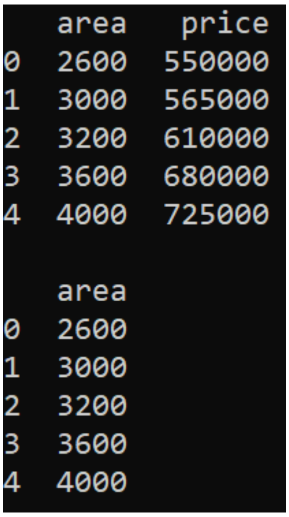


Program Name

Loading the data set
demo3a.py

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

df = pd.read_csv("homeprices.csv")
new_df = df.drop('price', axis = 'columns')

print(df)
print()
print(new_df)
```

Output



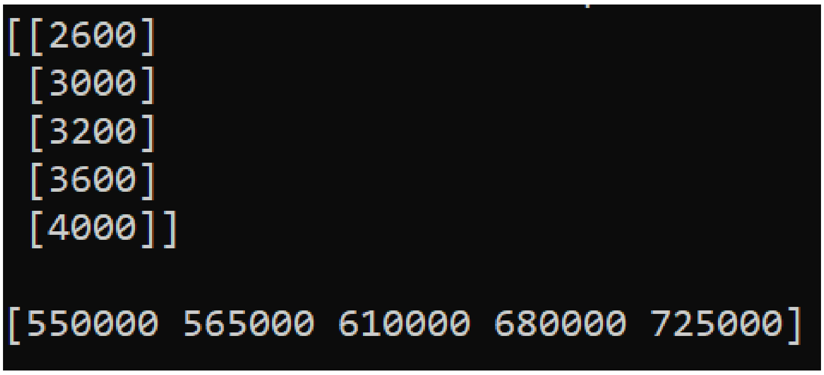
```
   area   price
0  2600  550000
1  3000  565000
2  3200  610000
3  3600  680000
4  4000  725000

   area
0  2600
1  3000
2  3200
3  3600
4  4000
```

Program Name

Loading the data set
demo3b.py

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

df = pd.read_csv("homeprices.csv")
new_df = df.drop('price', axis = 'columns')

print(new_df.values)
print()
print(df.price.values)
```

Output



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

[550000 565000 610000 680000 725000]
```

Program Name

Creating LinearRegression object
demo3.py

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

df = pd.read_csv("homeprices.csv")
new_df = df.drop('price', axis = 'columns')

# Training the Algorithm
reg = LinearRegression()
reg.fit(new_df.values, df.price.values)

print("Training the Algorithm")
```

Output

```
Training the Algorithm
```

Program Predict price of a home with area = 3300 sqr ft

Name demo4.py

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

df=pd.read_csv("homeprices.csv")
new_df = df.drop('price', axis='columns')

reg = LinearRegression()
reg.fit(new_df.values, df.price.values)

# Making Predictions
print(reg.predict([[3300]]))
```

Output

```
[628715.75342466]
```

Program Predict price of a home with area = 5000 sqr ft

Name demo5.py

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

df=pd.read_csv("homeprices.csv")
new_df = df.drop('price', axis='columns')

reg = LinearRegression()
reg.fit(new_df.values, df.price.values)

# Making Predictions
print(reg.predict([[5000]]))
```

Output

```
[859554.79452055]
```

Program Name

Capture the coefficient from regression
demo6.py

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

df=pd.read_csv("homeprices.csv")
new_df = df.drop('price', axis='columns')

reg = LinearRegression()
reg.fit(new_df.values, df.price.values)

print(reg.coef_)
```

Output

```
[135.78767123]
```

Program Name

Capture the intercept from regression
demo7.py

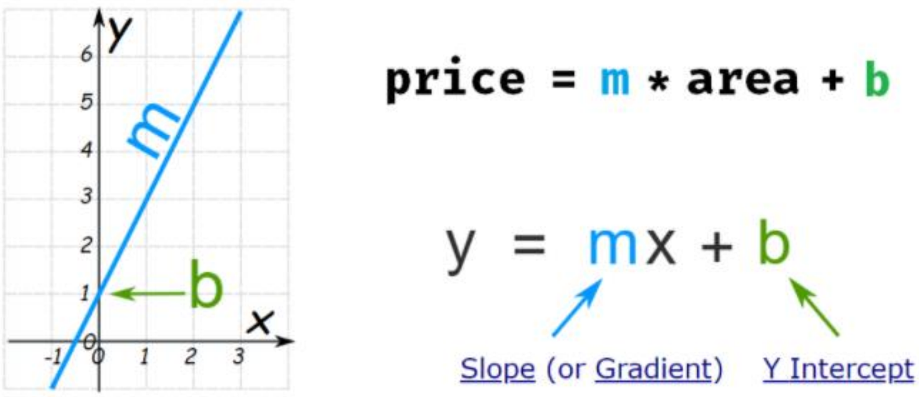
```
import pandas as pd
from sklearn.linear_model import LinearRegression

df=pd.read_csv("homeprices.csv")
new_df = df.drop('price', axis='columns')

reg = LinearRegression()
reg.fit(new_df.values, df.price.values)

print(reg.intercept_)
```

Output

```
180616.43835616432
```

10. Intercept and coefficient

✓	Intercept	=	180616.43835616432
✓	Coefficient	=	135.78767123



11. Y = m * X + b (m is coefficient and b is intercept)

- ✓ Let’s calculate the above formula.
- ✓ In the given formula m is coefficient and b is intercept
- ✓ Y = m * X + b
- ✓ Y = 135.78767123 * 3300 + 180616.43835616432
- ✓ Y = 628715.75342466
- ✓ Awesome....!!!!

12. Best fitted line

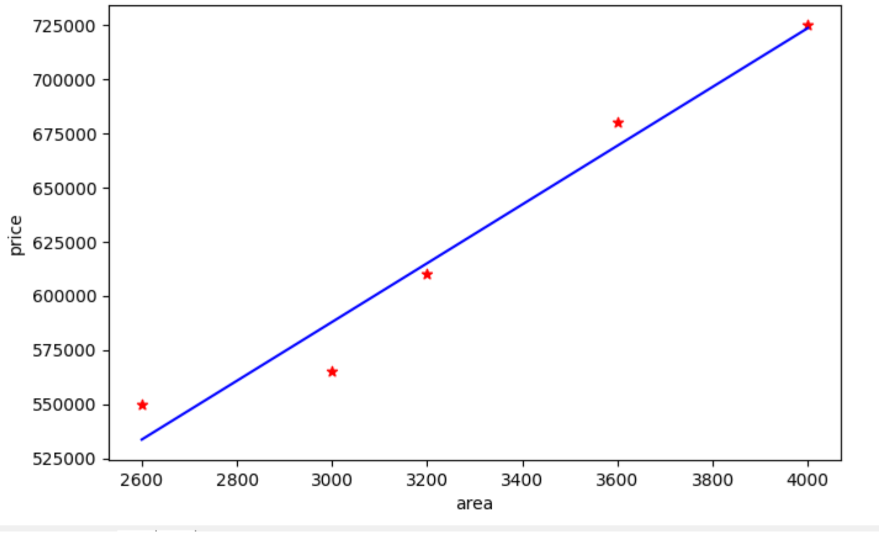
- ✓ Let’s calculate the above formula.
- ✓ We can draw a line

Program Name

Drawing a best fitted line
demo8.py

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

df=pd.read_csv("homeprices.csv")

new_df = df.drop('price', axis='columns')

reg = LinearRegression()
reg.fit(new_df.values, df.price.values)

plt.xlabel('area')
plt.ylabel('price')

plt.scatter(df.area.values, df.price.values, color = 'red', marker = '*')

plt.plot(df.area.values, reg.predict(df[['area']].values), color = 'blue')

plt.show()
```

Output


725000
700000
675000
650000
625000
600000
575000
550000
525000
price
2600
2800
3000
3200
3400
3600
3800
4000
area

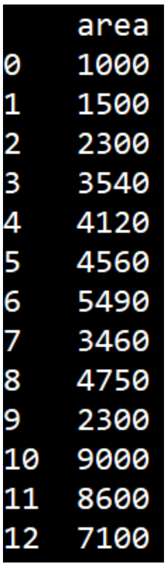
13. Predicting a group of home prices

- ✓ By using above model we can predict the group of home prices as well

Program Name
Loading a group of house areas
demo9.py

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

df = pd.read_csv("homeprices.csv")
new_df = df.drop('price', axis='columns')

reg = LinearRegression()
reg.fit(new_df.values, df.price.values)

area_df = pd.read_csv("areas.csv")
print(area_df)
```

Output



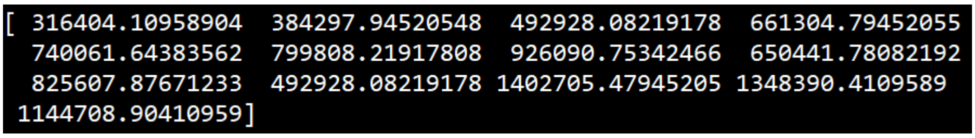
```
    area
0   1000
1   1500
2   2300
3   3540
4   4120
5   4560
6   5490
7   3460
8   4750
9   2300
10  9000
11  8600
12  7100
```

Program Name

Predicting a group of home prices
demo10.py

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

df = pd.read_csv("homeprices.csv")
new_df = df.drop('price', axis='columns')

reg = LinearRegression()
reg.fit(new_df.values, df.price.values)

area_df = pd.read_csv("areas.csv")

prices = reg.predict(area_df.values)
print(prices)
```

Output



```
[ 316404.10958904  384297.94520548  492928.08219178  661304.79452055
  740061.64383562  799808.21917808  926090.75342466  650441.78082192
  825607.87671233  492928.08219178 1402705.47945205 1348390.4109589
 1144708.90410959]
```

Program Name

Create a csv file with predictions
demo11.py

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

df = pd.read_csv("homeprices.csv")
new_df = df.drop('price', axis='columns')

reg = LinearRegression()
reg.fit(new_df.values, df.price.values)

area_df = pd.read_csv("areas.csv")
p = reg.predict(area_df.values)

area_df['prices'] = p
area_df.to_csv('output.csv')
print("Please check in current directory for output.csv")
```

Output

```
Please check in current directory for output.csv
```