LINEAR REGRESSION

SIMPLE LINEAR REGRESSION

$$y = b_0 + b_1 * x$$

Dependent Variable(DV)

Constant

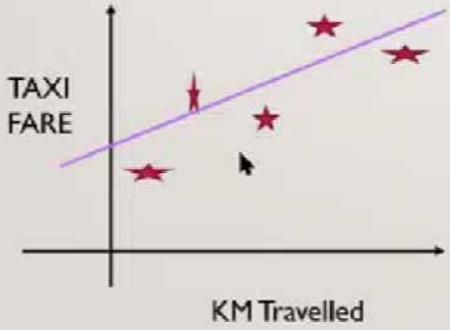
Coefficient

Independent Variable (IV)

SIMPLE LINEAR REGRESSION PROBLEM

1. Depending on the size of the house, predict the price of the house
2. Depending on no of hours studied, predict the marks obtained
3. Depending on the exp (in years) , predict the salary of a person

GRAPHICAL REPRESENTATION OF LINEAR REGRESSION

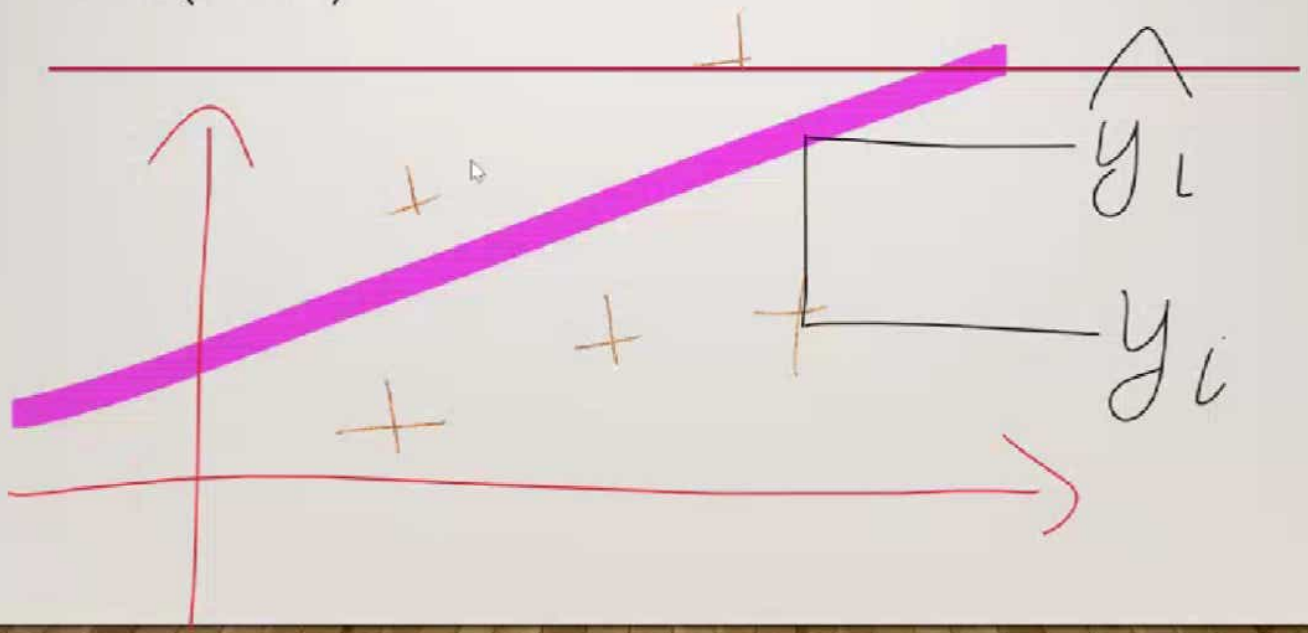


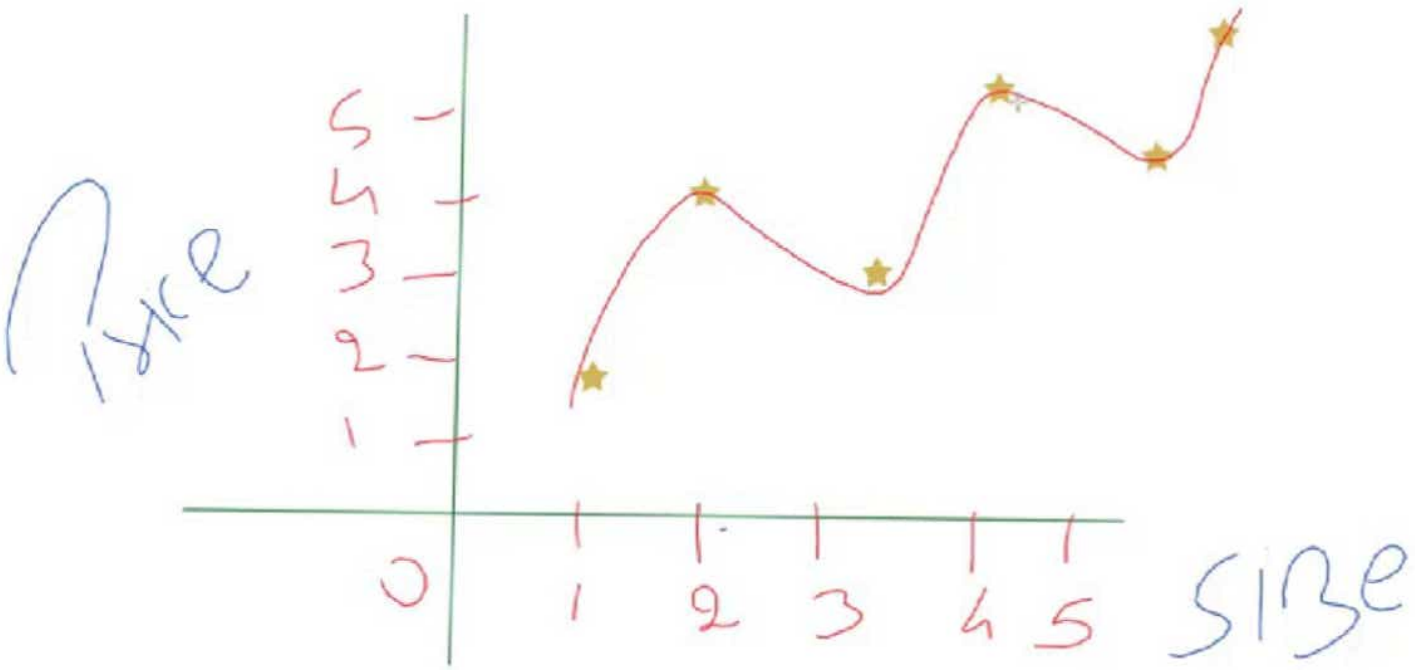
$y = b_0 + b_1 x$

Taxi Fare = b0 +b1 *km travelled

Purple line represents best fit line

SUM (Y – Y^) ^{2}




Price
5
4
3
2
1
0
1
2
3
4
5
size

SPLITTING THE DATASET INTO TRAINING AND TEST DATASET

TRAINING Dataset **X_train**	**Labels for Training DataSet** **y_train**
Test Dataset X_test	Labels for Test Dataset y_test

STEPS

- 1. Importing the Libraries
- 2. Importing the dataset
- 3. Splitting the dataset into Training set and Test set
- 4. Training Simple Linear Regression model on the training set
- 5. Predicting the Test set results
- 6. Visualizing the Training set results
- 7. Visualizing the Test set result

Linear Regression/ | simple_linear_regression - Jupyt

localhost:8890/notebooks/Linear%20Regression/simple_linear_regression.ipynb#

File Edit View Insert Cell Kernel Widgets Help Python 3

PROJECT TITLE : Predict the price based on the size of the house

Simple Linear Regression

Importing the libraries

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

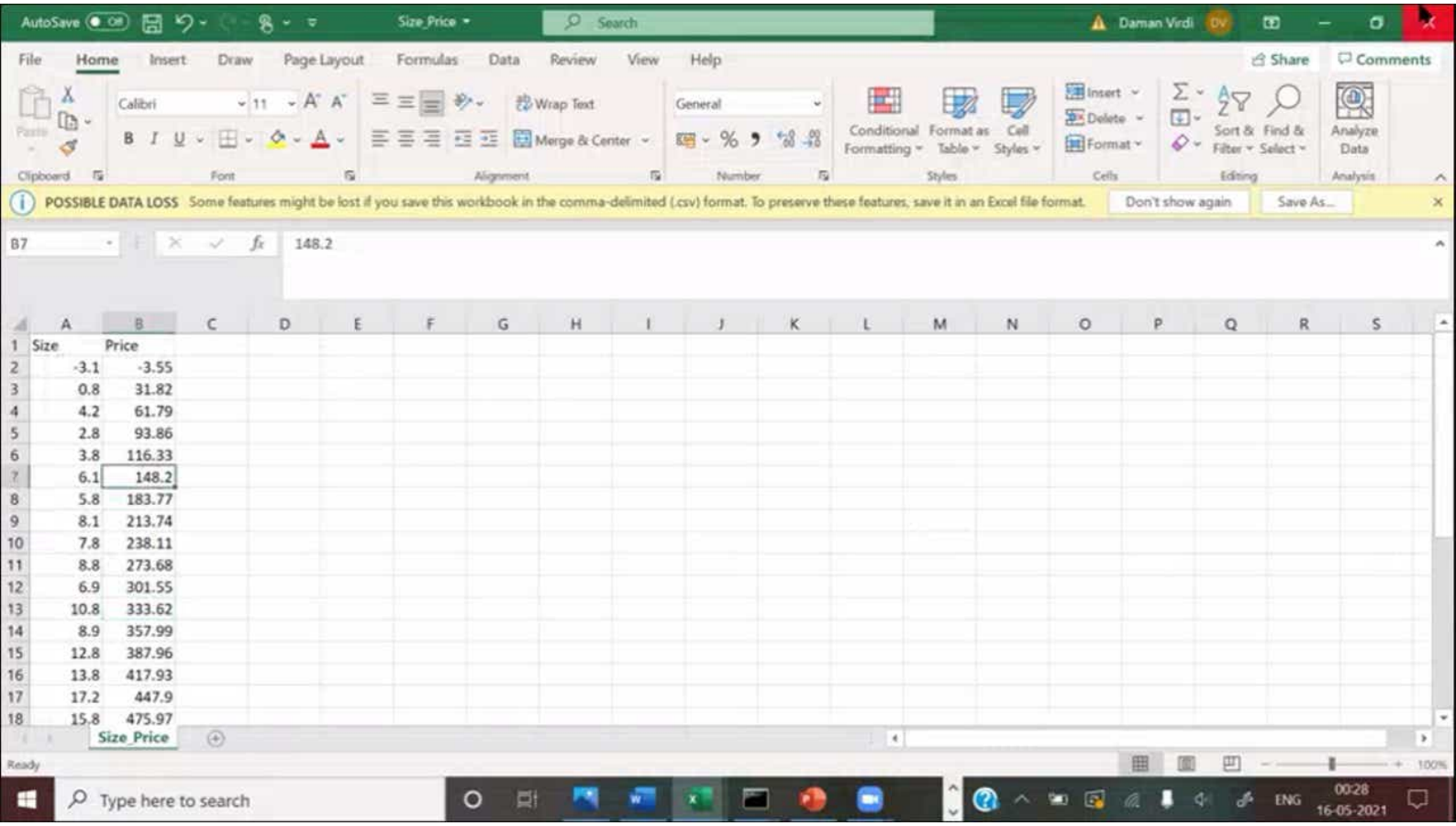
Creating Dataset

```
from random import choice

size = []
price = []
for i in range(30):
    size.append(i + choice([1.1 , 2.2, -3.1 , -0.2]))
    price.append(29.97*i - 0.35 +choice([2.2 ,-3.2 , 4.3,-1.3]))
```

Type here to search

ENG 00:26 16-05-2021



Size	Price
-3.1	-3.55
0.8	31.82
4.2	61.79
2.8	93.86
3.8	116.33
6.1	148.2
5.8	183.77
8.1	213.74
7.8	238.11
8.8	273.68
6.9	301.55
10.8	333.62
8.9	357.99
12.8	387.96
13.8	417.93
17.2	447.9
15.8	475.97

Linear Regression/ | Size_Price.csv - Jupyter Text Edi | simple_linear_regression - Jupy

localhost:8890/notebooks/Linear%20Regression/simple_linear_regression.ipynb#

File Edit View Insert Cell Kernel Widgets Help Python 3

Creating Dataset

```
from random import choice

size = []
price = []
for i in range(14 , 44):
    size.append(i + choice([1.1 , 2.2, -3.1 , -0.2]))
    price.append(29.97*i - 0.35 +choice([2.2 ,-3.2 , 4.3,-1.3]))

df = pd.DataFrame(list(zip(size, price)), columns = "Size Price".split())

df.to_csv('Size_Price.csv',index=False)
```

Importing the dataset

```
dataset = pd.read_csv('Size_Price.csv')
X = dataset.iloc[:, :-1].values
y = dataset.iloc[:, -1].values

print(X)
print(y)
```

Type here to search ENG 00:33 16-05-2021

Linear Regression/ | Size_Price.csv - Jupyter Text Editi | simple_linear_regression - Jupyte

localhost:8890/notebooks/Linear%20Regression/simple_linear_regression.ipynb#

File Edit View Insert Cell Kernel Widgets Help Python 3

```
for i in range(14 , 44):
    size.append(i + choice([1.1 , 2.2, -3.1 , -0.2]))
    price.append(29.97*i - 0.35 +choice([2.2 ,-3.2 , 4.3,-1.3]))

df = pd.DataFrame(list(zip(size, price)), columns = "Size Price".split())

df.to_csv('Size_Price.csv',index=False)
```

Importing the dataset

```
dataset = pd.read_csv('Size_Price.csv')
X = dataset.iloc[:, :-1].values
y = dataset.iloc[:, -1].values

print(type(X))
print(y)
```

```
<class 'numpy.ndarray'>
[ 421.43  447.9   481.37  513.44  535.91  567.78  601.25  627.72  657.09
  685.76  717.63  753.2   777.57  807.54  835.61  873.08  896.55  925.52
  960.89  985.46 1022.93 1045.4  1077.27 1112.84 1137.21 1170.68 1200.65
 1227.12 1250.59 1287.06]
```

Type here to search ENG 00:36 16-05-2021

Linear Regression/ | Size_Price.csv - Jupyter Text Edit | simple_linear_regression - Jupyt

localhost:8890/notebooks/Linear%20Regression/simple_linear_regression.ipynb#

File Edit View Insert Cell Kernel Widgets Help Python 3

```
df = pd.DataFrame(list(zip(size, price)), columns = "Size Price".split())

df.to_csv('Size_Price.csv',index=False)
```

Importing the dataset

```
dataset = pd.read_csv('Size_Price.csv')
X = dataset.iloc[:, :-1].values
y = dataset.iloc[:, -1].values

print(type(X))
print(type(y))
print(X)
print(Y)
```

```
<class 'numpy.ndarray'>
<class 'numpy.ndarray'>
[[18.9]
 [17.2]
 [18.2]
 [16.8]
 [14.9]
```

Type here to search

ENG 00:36 16-05-2021

Linear Regression/ | Size_Price.csv - Jupyter Text Edit | simple_linear_regression - Jupyt

localhost:8890/notebooks/Linear%20Regression/simple_linear_regression.ipynb#

File Edit View Insert Cell Kernel Widgets Help Python 3

```
 [33.9]
 [40.2]
 [41.2]
 [56.9]
 [42.1]
 [41.8]
 [39.9]]
[ 421.43  487.9   481.37  513.44  535.91  567.78  601.25  627.72  657.69
  685.76  717.63  751.2   777.57  807.54  835.61  873.08  895.55  925.52
  960.89  985.46 1022.93 1045.4  1077.27 1112.84 1137.21 1170.68 1200.65
 1237.12 1260.59 1287.06]
```

Splitting the dataset into the Training set and Test set

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 1/4, random_state = 0)
print(X_test)
print(y_test)
```

```
[[ 3.1]
 [30.2]
 [14.1]
 [ 6.9]
 [22.9]
 [20.9]
 [26.8]
 [12.1]]
[ 61.29 641.01 391.46 296.15 775.67 717.63 813.14 331.62]
```

Type here to search ENG 00:38 16-05-2021

Linear Regression/ | Size_Price.csv - Jupyter Text Edit | simple_linear_regression - Jupyt

localhost:8890/notebooks/Linear%20Regression/simple_linear_regression.ipynb#

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3

```
array([[18.2],
       [41.0],
       [28.3],
       [26.2],
       [36.9],
       [40.2],
       [42.3],
       [24.8]])
array([481.37, 1260.59,  807.54,  717.61, 1200.65, 1137.21, 1227.12,  753.2 ])
```

Training the Simple Linear Regression model on the Training set

In [16]:

```
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(X_train, y_train)
```

```
LinearRegression()
```

Predicting the Test set results

In [17]:

```
X_test
```

```
array([[ 3.1],
```

Type here to search

ENG 00:39 16-05-2021

Linear Regression/ | Size_Price.csv - Jupyter Text Editi | simple_linear_regression - Jupyt

localhost:8890/notebooks/Linear%20Regression/simple_linear_regression.ipynb#

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

```
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(X_train, y_train)
```

```
LinearRegression()
```

Predicting the Test set results

```
X_test
```

```
array([[ 1.1],
       [30.2],
       [14.1],
       [ 6.9],
       [22.9],
       [20.0],
       [26.8],
       [17.1]])
```

```
y_pred = regressor.predict(X_test)
y_pred
```

```
array([127.11724667, 882.69950831, 433.81115849, 283.06605072,
       629.16627911, 623.40375056, 787.90320978, 378.40862494])
```

Type here to search

ENG 00:39 16-05-2021

Linear Regression/ | Size_Price.csv - Jupyter Text Edit | simple_linear_regression - Jupyt

localhost:8890/notebooks/Linear%20Regression/simple_linear_regression.ipynb#

File Edit View Insert Cell Kernel Widgets Help

Python 3

```
y_pred = regressor.predict(X_test)
y_pred
```

```
array([ 530.85268083, 1209.42543255,  936.75596686,  785.56883329,
       1073.8017667 , 1162.7221957 , 1251.5972886,  787.45354181])
```

```
y_test
```

```
array([ 481.37, 1260.59,  807.54,  717.63, 1200.65, 1152.21, 1227.14,
        753.2 ])
```

```
print(X_test[7])

y_pred = regressor.predict(X_test)
print(type(y_pred))
print(y_pred[7])
print("\n")
print(y_test[7])
```

```
[12.1]
<class 'numpy.ndarray'>
```

Type here to search

ENG 00:41 16-05-2021

Linear Regression/ | Size_Price.csv - Jupyter Text Editi | simple_linear_regression - Jupyt

localhost:8890/notebooks/Linear%20Regression/simple_linear_regression.ipynb#

File Edit View Insert Cell Kernel Widgets Help Python 3

```
array([ 570.05268903, 1205.82543255,  836.75396703,  785.56882328,
       1073.8217697 , 1162.7221957 , 1213.90728946,  747.85354103])
```

[6]:
```
y_test
```

```
array([ 481.37, 1260.59,  807.54,  717.61, 1200.65, 1137.21, 1227.12,
        753.2 ])
```

[7]:

[8]:
```
print(X_test[7])

print(type(y_pred))
print(y_pred[7])
print("\n")
print(y_test[7])
```

```
[24.8]
<class 'numpy.ndarray'>
747.85354103177


753.1999999999998
```

Type here to search ENG 00:41 16-05-2021

Linear Regression/ | Size_Price.csv - Jupyter Text Editi | simple_linear_regression - Jupyte

localhost:8890/notebooks/Linear%20Regression/simple_linear_regression.ipynb#

File Edit View Insert Cell Kernel Widgets Help Python 3

```
print(y_test[7])
```

```
[24.8]
<class 'numpy.ndarray'>
747.8535410331775


751.1999999999998
```

```
y_train
```

```
array([ 935.52, 1077.27,  567.78,  895.55,  657.66,  835.61, 1112.84,
       1012.93,  447.9 , 1287.06,  601.25,  535.91,  960.09,  985.46,
        885.76,  627.72, 1170.68,  513.44,  421.43, 1045.4 ,  871.08,
        772.57])
```

Visualising the Training set results

```
plt.scatter(X_train, y_train, color = 'red')
plt.plot(X_train, regressor.predict(X_train), color = 'blue')
plt.title('Size vs Price (Training set)')
plt.xlabel('Size ')
plt.ylabel('Price')
plt.show()
```

Type here to search ENG 00:41 16-05-2021

Linear Regression/ | Size_Price.csv - Jupyter Text Editi | simple_linear_regression - Jupyt

localhost:8890/notebooks/Linear%20Regression/simple_linear_regression.ipynb#

File Edit View Insert Cell Kernel Widgets Help Python 3

Visualising the Training set results

In [40]:
```
plt.scatter(X_train, y_train, color = 'red')
#plt.plot(X_train, regressor.predict(X_train), color = 'blue')
#plt.title('Size vs Price (Training set)')
#plt.xlabel('Size')
#plt.ylabel('Price')
#plt.show()
```

```
<matplotlib.collections.PathCollection at 0x18270f51a00>
```

Type here to search

ENG 00:42 16-05-2021

Linear Regression/ | Size_Price.csv - Jupyter Text Edit | simple_linear_regression - Jupyt

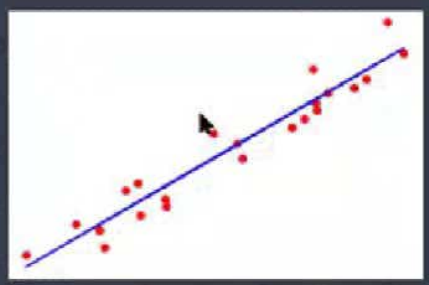
localhost:8890/notebooks/Linear%20Regression/simple_linear_regression.ipynb#

File Edit View Insert Cell Kernel Widgets Help Python 3

Visualising the Training set results

```
plt.scatter(X_train, y_train, color = 'red')
plt.plot(X_train, regressor.predict(X_train), color = 'blue')
#plt.title('Size vs Price (Training set)')
#plt.xlabel('Size')
#plt.ylabel('Price')
#plt.show()
```

```
[<matplotlib.lines.Line2D at 0x18270fa99d0>]
```



Type here to search

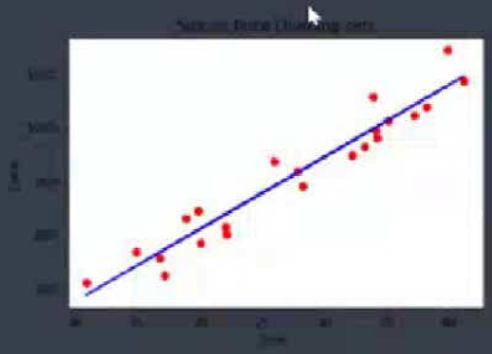
ENG 00:43 16-05-2021

Linear Regression/ | Size_Price.csv - Jupyter Text Editi | simple_linear_regression - Jupyt

localhost:8890/notebooks/Linear%20Regression/simple_linear_regression.ipynb#

File Edit View Insert Cell Kernel Widgets Help Python 3

```
plt.title('Size vs Price (Training set)')
plt.xlabel('Size ')
plt.ylabel('Price')
plt.show()
```



Visualising the Test set results

```
plt.scatter(X_test, y_test, color = 'red')
plt.plot(X_train, regressor.predict(X_train), color = 'blue')
```

Type here to search

ENG 00:43 16-05-2021

Linear Regression/ | Size_Price.csv - Jupyter Text Editi | simple_linear_regression - Jupyt

localhost:8890/notebooks/Linear%20Regression/simple_linear_regression.ipynb#

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Visualising the Test set results

```
plt.scatter(X_test, y_test, color = 'red')
plt.plot(X_train, regressor.predict(X_train), color = 'blue')
plt.title('House Price (Test set)')
plt.xlabel('Size of House')
plt.ylabel('Price')
plt.show()
```

Making a single prediction (price of house for the size 4)

```
print(regressor.predict([[4]]))
```

Getting the final linear regression equation with the values of the coefficients

```
print(regressor.coef_)
print(regressor.intercept_)
```

Type here to search ENG 00:43 16-05-2021

Linear Regression/ | Size_Price.csv - Jupyter Text Editor | simple_linear_regression - Jupyter

localhost:8890/notebooks/Linear%20Regression/simple_linear_regression.ipynb#

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3

Making a single prediction (price of house for the size 4)

```
print(regressor.predict([[4]]))
```

Getting the final linear regression equation with the values of the coefficients

```
print(regressor.coef_)
print(regressor.intercept_)
```

PROJECT EXTENTION : Predict the price of the house based on multiple factors

```
#price = __ + __ *size
```

```
regressor.predict([[0]])
```

Type here to search

ENG 00:44 16-05-2021

Linear Regression/ | Size_Price.csv - Jupyter Text Editor | simple_linear_regression - Jupyter

localhost:8890/notebooks/Linear%20Regression/simple_linear_regression.ipynb#

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3

Making a single prediction (price of house for the size 4)

```
print(regressor.predict([[4]]))
```

```
[494839.78349689]
```

Getting the final linear regression equation with the values of the coefficients

```
print(regressor.coef_)
print(regressor.intercept_)
```

```
[26.93952303]
79.75336988277957
```

PROJECT EXTENTION : Predict the price of the house based on multiple factors

```
#price = 79.7533 + 26.9395*size
```

```
regressor.predict([[0]])
```

```
array([79.75336988])
```

Type here to search

ENG 00:45 16-05-2021

Linear Regression	Logistic Regression
Linear regression is used to predict the continuous dependent variable using a given set of independent variables.	Logistic Regression is used to predict the categorical dependent variable using a given set of independent variables.
Linear Regression is used for solving Regression problem.	Logistic regression is used for solving Classification problems.
In Linear regression, we predict the value of continuous variables.	In logistic Regression, we predict the values of categorical variables.
In linear regression, we find the best fit line, by which we can easily predict the output.	In Logistic Regression, we find the S-curve by which we can classify the samples.
Least square estimation method is used for estimation of accuracy.	Maximum likelihood estimation method is used for estimation of accuracy.
The output for Linear Regression must be a continuous value, such as price, age, etc.	The output of Logistic Regression must be a Categorical value such as 0 or 1, Yes or No, etc.
In Linear regression, it is required that relationship between dependent variable and independent variable must be linear.	In Logistic regression, it is not required to have the linear relationship between the dependent and independent variable.
In linear regression, there may be collinearity between the independent variables.	In logistic regression, there should not be collinearity between the independent variable.