# Machine Learning

## Simple Linear Regression

> You have to install Scikit-learn via pip install scikit-learn

### How Scikit-learn Works!!

1) Data ---> Model ---> Prediction 2) Give Data to the model and divide it into test and train data 3) *Test* data is use to test he mode when its made and *Train* data is use to train the model 4) MOdel is first created and then Model Learns and then it will predict

> Data --> Split_data--> Create_model--> learn the model(fiting)-->Test model---> **Prediction**

### Explaination and Task in Simple Linear Regression

1) We have have two continuous variables one is dependent and other is independent

> y=a+bx ( where y is dependent and x is independent ,a is the y-intercept) \ b = slop

### How to Find out Slop ----> (Task)

a = (y2-y1)/(x2-x1) where xi and yi are the points on x and y a = slope (x1, y1) = coordinates of first point in the line (x2, y2) = coordinates of second point in the line

## Step 1: Import the DataSet

In [216…

```
import pandas as pd
df = pd.read_csv("ml_data_salary.csv")
df.head()
```

Out[216…

| | age | distance | YearsExperience | Salary |
|---|---|---|---|---|
| **0** | 31.1 | 77.75 | 1.1 | 39343 |
| **1** | 31.3 | 78.25 | 1.3 | 46205 |
| **2** | 31.5 | 78.75 | 1.5 | 37731 |
| **3** | 32.0 | 80.00 | 2.0 | 43525 |
| **4** | 32.2 | 80.50 | 2.2 | 39891 |

### Check and Remove the null values

In [217…

```
df.isnull().sum()
```

Out[217…

```
age               0
distance          0
YearsExperience   0
Salary            0
dtype: int64
```

# Step 2: Splitting Dataset into Training Data and testing Data

In [218…

```
x= df [["YearsExperience"]]
y= df ["Salary"]
```

In [ ]:

```

```

In [219…

```
# import library
from sklearn.linear_model import  LinearRegression
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2 ,random_state=0)
```

# Step-3 Fit the Model

In [220…

```
from sklearn.linear_model import  LinearRegression
model=LinearRegression().fit(x_train,y_train)
model
```

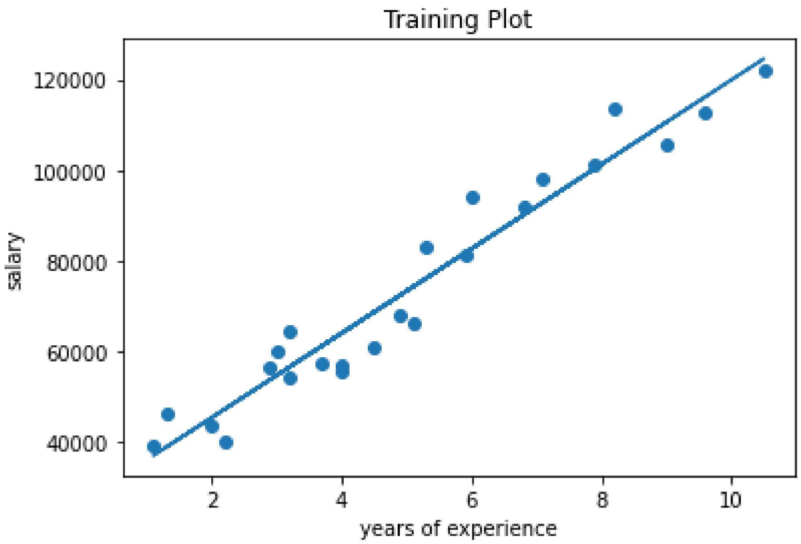
Out[220…

```
LinearRegression()
```

# Step-4 Ploting

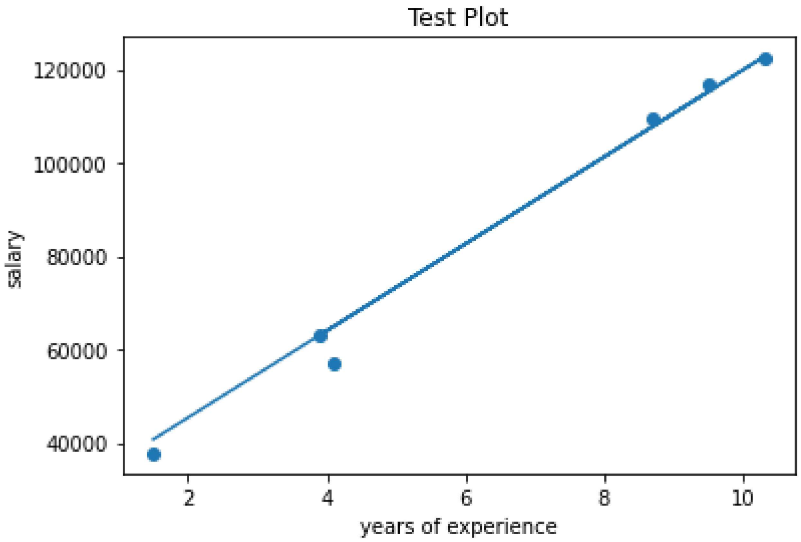
In [221…

```
import matplotlib.pyplot as pt
pt.scatter(x_train,y_train)
# make plot with prediction
pt.plot(x_train,model.predict(x_train))
# Labeling the plot
pt.xlabel("years of experience")
pt.ylabel("salary")
pt.title("Training Plot")
pt.show()
```



In [222…

```
pt.scatter(x_test,y_test)
# make plot with prediction
pt.plot(x_test,model.predict(x_test))
# Labeling the plot
pt.xlabel("years of experience")
pt.ylabel("salary")
pt.title("Test Plot")
pt.show()
```



# Step-5 Evaluating

In [223…

```
print ("Score of train model",model.score(x_test,y_test))
print ("Score of test model",model.score(x_train,y_train))
```

```
Score of train model 0.988169515729126
Score of test model 0.9411949620562126
```

In [208…

```
# x=([3],[5],[7])
# model.predict(x)
```

# Multiple Linear Regression

> More than two variables like age and year of experience as independent variable and dependent is one i.e Salary

In [224…

```
# import libararies we have already imported them
df = pd.read_csv("ml_data_salary.csv")
df.head()
```

Out[224…

| | age | distance | YearsExperience | Salary |
|---|---|---|---|---|
| **0** | 31.1 | 77.75 | 1.1 | 39343 |
| **1** | 31.3 | 78.25 | 1.3 | 46205 |
| **2** | 31.5 | 78.75 | 1.5 | 37731 |
| **3** | 32.0 | 80.00 | 2.0 | 43525 |
| **4** | 32.2 | 80.50 | 2.2 | 39891 |

In [225…

```
x=df[["age","distance","YearsExperience"]]
y=df["Salary"]
```

In [226…

```
# creating the model
model = LinearRegression().fit(x,y)
model
```

Out[226…

```
LinearRegression()
```

In [227…

```
# to see the coffiencts
model.coef_
```

Out[227…

```
array([-2.79782201e+15,  1.10953700e+15,  2.39795093e+13])
```

In [228…

```
# to see the intercept
model.intercept_
```

Out[228…

```
719385278130753.2
```

In [229…

```
model.predict([[31.1,77.75,1.1]])
```

```
C:\Users\Asfandyar\AppData\Roaming\Python\Python310\site-packages\sklearn\base.py:450: U
serWarning: X does not have valid feature names, but LinearRegression was fitted with fe
ature names
  warnings.warn(
array([36205.125])
```

Out[229…

# Efficeny with out the split and with split

## Assignment

> How to plot multiple linear reg model? \ How test the effiecncy of model?

In [238…

```
import pandas as pd
df1 = pd.read_csv("ml_data_salary.csv")
df1.head()
```

Out[238…

| | age | distance | YearsExperience | Salary |
|---|---|---|---|---|
| **0** | 31.1 | 77.75 | 1.1 | 39343 |
| **1** | 31.3 | 78.25 | 1.3 | 46205 |
| **2** | 31.5 | 78.75 | 1.5 | 37731 |
| **3** | 32.0 | 80.00 | 2.0 | 43525 |
| **4** | 32.2 | 80.50 | 2.2 | 39891 |

In [239…

```
ages= df1[['age']]
dist= df1[['distance']]
yearss = df1[['YearsExperience']]
sals= df1['Salary']
```

In [240…

```
# Lets train the model and test it
from sklearn.model_selection import train_test_split
x_train ,x_test ,y_train, y_test =train_test_split(ages, sals, test_size=0.2, random_st
model= LinearRegression().fit(x_train, y_train)
model
```

Out[240…

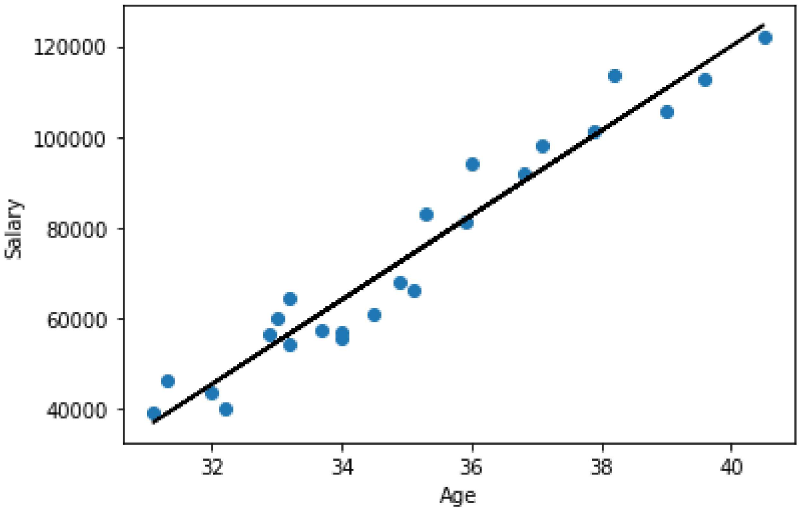
```
LinearRegression()
```

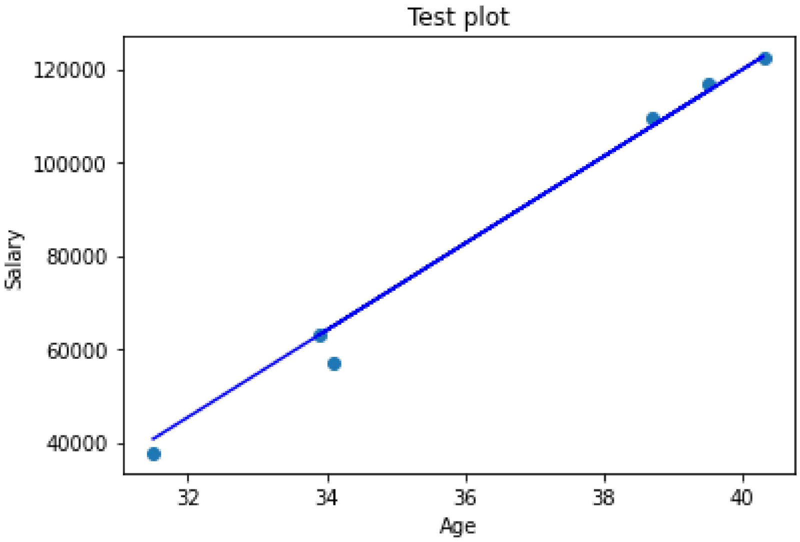
In [241…

```
## NOw ploting
pt.scatter(x_train, y_train)
pt.plot(x_train, model.predict(x_train), color= "black")
pt.xlabel("Age")
pt.ylabel("Salary")
pt.title("Train plot",color="white")
pt.show()


pt.scatter(x_test, y_test)
pt.plot(x_test, model.predict(x_test), color= "blue")
pt.xlabel("Age")
pt.ylabel("Salary")
pt.title("Test plot")
pt.show()


# pt.xlabel("years of experience",color="white")
# pt.ylabel("salary",color="white")
# pt.title("Test Plot",color="white")
# pt.show()
```




Test plot

# Efficiency and Accuracy !!

In [242…

```
print ("Accuracy of the age and salary mode full model",model.score(ages, sals))
print ("Train model score: ",model.score(x_train, y_train))
print ("Test model score: ",model.score(x_test, y_test))
```

```
Accuracy of the age and salary mode full model 0.9566628347576891
Train model score:  0.9411949620562127
Test model score:  0.988169515729126
```

# Year and Salary

In [243…

```
x1_train, x1_test, y1_train, y1_test= train_test_split(yearss, sals, test_size=0.2, ran
model1= LinearRegression().fit(x1_train, y1_train)
model1


# ages= df[['age']]
# dist= df[['distance']]
# yearss = df[['YearsExperience']]
# sals= df['Salary']
```

Out[243…

```
LinearRegression()
```

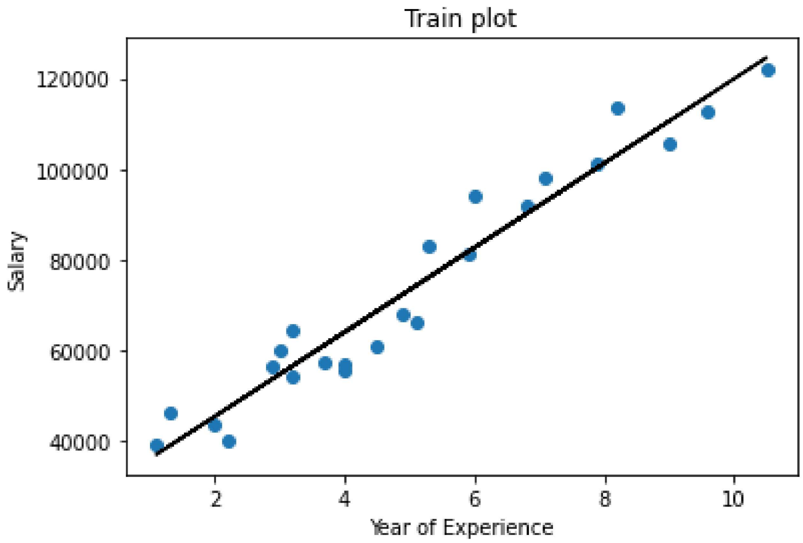
## Ploting the Graphs

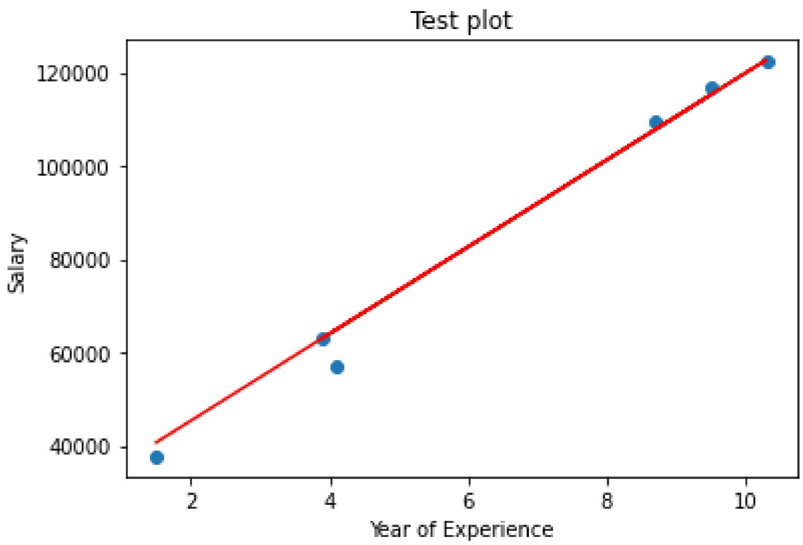
In [244…

```
pt.scatter(x1_train, y1_train)
pt.plot(x1_train, model1.predict(x1_train), color= "black")
pt.xlabel("Year of Experience")
pt.ylabel("Salary")
pt.title("Train plot")
pt.show()



pt.scatter(x1_test, y1_test)
pt.plot(x1_test, model1.predict(x1_test), color= "red")
pt.xlabel("Year of Experience")
pt.ylabel("Salary")
pt.title("Test plot")
pt.show()
```





# Efficiency and Accuracy of the Year Experience and Salary !!

In [245…

```
print ("Accuracy of model with out split",model1.score(yearss, sals))
print ("Train model score: ",model1.score(x1_train, y1_train))
print ("Test model score: ",model1.score(x1_test, y1_test))
```

```
Accuracy of model with out split 0.9566628347576891
Train model score:  0.9411949620562126
Test model score:  0.988169515729126
```

# Distance and Salary

In [246…

```
disx_train, disx_test, y11_train, y11_test= train_test_split(dist, sals, test_size=0.2,
model2= LinearRegression().fit(disx_train, y11_train)
model2
```

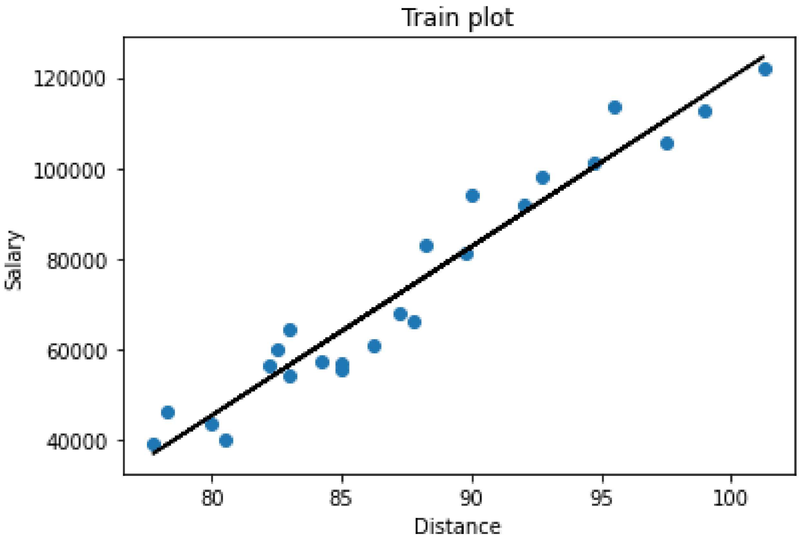
Out[246…

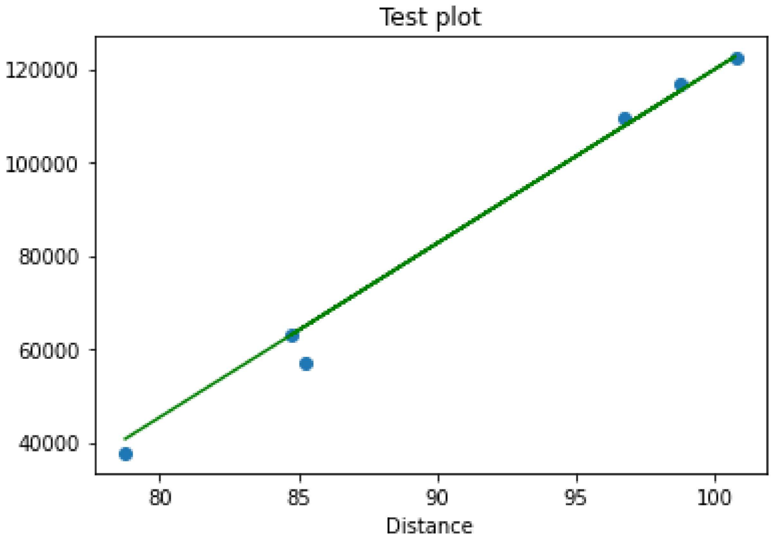
```
LinearRegression()
```

In [247…

```
pt.scatter(disx_train, y11_train)
pt.plot(disx_train, model2.predict(disx_train), color= "black")
pt.xlabel("Distance")
pt.ylabel("Salary")
pt.title("Train plot")
pt.show()


pt.scatter(disx_test, y11_test)
pt.plot(disx_test, model2.predict(disx_test), color= "green")
pt.xlabel("Distance")
pt.title("Test plot")
pt.show()
```





# Efficiency Distance and Salary !!

In [248…

```
# ages= df[['age']]
# dist= df[['distance']]
# yearss = df[['YearsExperience']]
# sals= df['Salary']
print ("Accuracy of the Distance and salary model without split",model2.score(dist, sal
print ("Train model score: ",model2.score(disx_train, y11_train))
print ("Test model score: ",model2.score(disx_test, y11_test))
```

```
Accuracy of the Distance and salary model without split 0.9566628347576892
Train model score:  0.9411949620562127
Test model score:  0.988169515729126
```