Types of machine Learning

- supervised learning
  - Regression
    - depends on continous data
      1. +ve Linearity(Linear Regression)
         - Linear regression with one feature/one input
         - Linear regression with multiple features/multiple inputs
      2. -ve Linearity(Polynomial Regression)
         - Polynomial with one feature
         - Polynomial with multiple features
  - Classification
    - depends on catagorical(0,1,yes,no,true,False,goog,bad)
      - KNN Algorithm - K-Nearest Neighbours
      - Logistic Regression
      - SVM - Support Vector Machine Regressor/Classifier
      - Decision Tree Regressor/Classifier
      - Random Forest Regressor/Classifier

- unsupervised learning
  - Clustering
    - K-Means Algorithm
  - Dimensionlity Reduction PCA - Pricipal Component Analysis
- reinforcement Learning

Linear Regression

- Linear model is sum weighted predict data to target values
- linear regression with one feature formula
  - Y= mx+c
  - y is a Target value or output value
  - m is a Slope
  - x is a input variable
  - c is a Coefficient/Intercept

Slope Formula

(x-xmean)*(y-ymean)/(x-xmean)^2

Coefficient/Intercept Formula

c = ymean-(m*xmean)

Machine Learning Steps

1.get or load the data 2.preprocessing the data 3.Define input and output data 4.applying model or algorithm 5.train the data 6.calculate the score

In [1]:
```
## getting the libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

In [4]:
```
# 1. getting the data
df = pd.read_csv("https://raw.githubusercontent.com/AP-State-Skill-Developme
```

In [5]:
```
df
```

...

In [6]:
```
df = pd.read_csv('Salary_Data.csv')
```

In [7]:
```
1 df
```

Out[7]:

	YearsExperience	Salary
0	1.1	39343.0
1	1.3	46205.0
2	1.5	37731.0
3	2.0	43525.0
4	2.2	39891.0
5	2.9	56642.0
6	3.0	60150.0
7	3.2	54445.0
8	3.2	64445.0
9	3.7	57189.0
10	3.9	63218.0
11	4.0	55794.0
12	4.0	56957.0
13	4.1	57081.0
14	4.5	61111.0
15	4.9	67938.0
16	5.1	66029.0
17	5.3	83088.0
18	5.9	81363.0
19	6.0	93940.0
20	6.8	91738.0
21	7.1	98273.0
22	7.9	101302.0
23	8.2	113812.0
24	8.7	109431.0
25	9.0	105582.0
26	9.5	116969.0
27	9.6	112635.0
28	10.3	122391.0
29	10.5	121872.0

In [8]:
```
# Data preprocessing
df.shape
```

Out[8]:
```
(30, 2)
```

In [9]:
```
df.isnull().sum()
```

Out[9]:
```
YearsExperience    0
Salary             0
dtype: int64
```

In [10]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30 entries, 0 to 29
Data columns (total 2 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   YearsExperience  30 non-null     float64
 1   Salary           30 non-null     float64
dtypes: float64(2)
memory usage: 608.0 bytes
```

In [11]:
```
df.describe()
```

Out[11]:

	YearsExperience	Salary
count	30.000000	30.000000
mean	5.313333	76003.000000
std	2.837888	27414.429785
min	1.100000	37731.000000
25%	3.200000	56720.750000
50%	4.700000	65237.000000
75%	7.700000	100544.750000
max	10.500000	122391.000000

In [13]:
```
# Define input and output data
X = df[['YearsExperience']]# input data
y = df['Salary']# output data
```

In [14]:
```
# applying the model
from sklearn.linear_model import LinearRegression

```

In [15]:
```
# create object
model = LinearRegression()
```

In [17]:
```
# fit the data
model.fit(X,y)
```

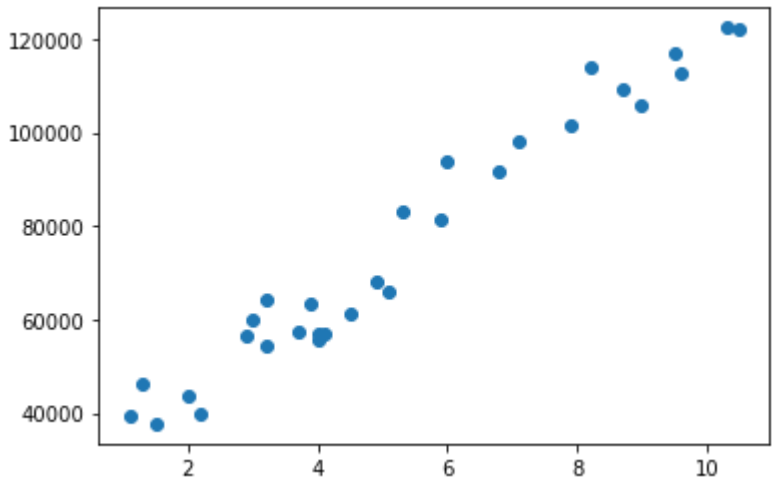
Out[17]:
```
LinearRegression()
```

In [18]:
```
y_pred = model.predict(X)
```

In [19]:
```
y_pred
```

Out[19]:
```
array([ 36187.15875227,  38077.15121656,  39967.14368085,  44692.12484158,
        46582.11730587,  53197.09093089,  54142.08716303,  56032.07962732,
        56032.07962732,  60757.06078805,  62647.05325234,  63592.04948449,
        63592.04948449,  64537.04571663,  68317.03064522,  72097.0155738 ,
        73987.00803809,  75877.00050238,  81546.97789525,  82491.9741274 ,
        90051.94398456,  92886.932681  , 100446.90253816, 103281.8912346 ,
       108006.87239533, 110841.86109176, 115566.84225249, 116511.83848464,
       123126.81210966, 125016.80457395])
```

In [20]:
```
# VIsualize the data to relationship of input and output
plt.scatter(df['YearsExperience'],df['Salary'])
plt.show()
```



In [22]:
```
# identify the score
model.score(X,y)*100
```

Out[22]:
```
95.69566641435085
```

In [23]:
```
plt.scatter(df['YearsExperience'],df['Salary'],c='g',label='Actual values')
plt.plot(X,y_pred,c='r',label = 'predicted values')
plt.title('yearsExperience vs Salary')
plt.xlabel('yearsExperience')
plt.ylabel('Salary')
plt.legend()
plt.show()
```

Predict the data jio subscribers

In [24]:
```
df = pd.read_csv("https://raw.githubusercontent.com/AP-State-Skill-Developme
```

In [25]:
```
1 df
```

Out[25]:

	Revenue	JioSubscribers
0	8136	160.1
1	8421	186.6
2	10023	215.3
3	11416	252.3
4	12893	280.1
5	14328	306.7
6	15741	331.3
7	16534	355.2
8	17555	370.0
9	18632	387.5

In [26]:
```
1 df.isnull().sum()
```

Out[26]:
```
Revenue           0
JioSubscribers    0
dtype: int64
```

In [27]:
```
1 df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 2 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   Revenue         10 non-null     int64  
 1   JioSubscribers  10 non-null     float64
dtypes: float64(1), int64(1)
memory usage: 288.0 bytes
```

In [28]:
```
1 # define input and ouput
2 X = df[['Revenue']]
3
```

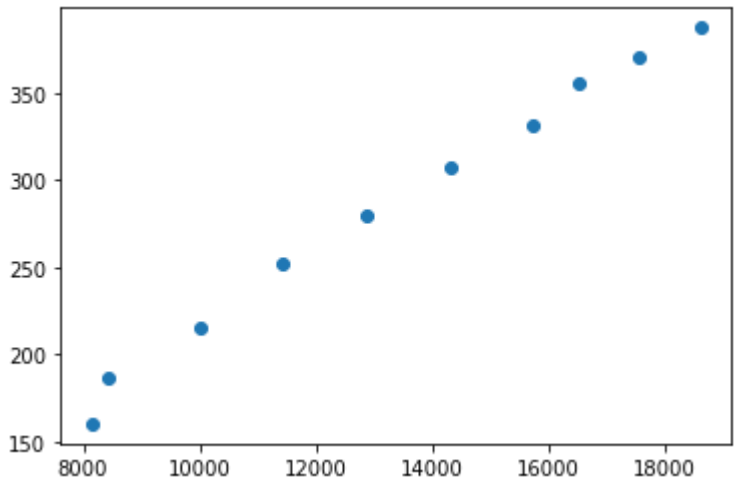
In [29]:
```
1 y= df['JioSubscribers']
```

In [30]:
```
1 from sklearn.linear_model import LinearRegression
2 model = LinearRegression()
```

In [31]:
```
model.fit(X,y)
```

Out[31]:
```
LinearRegression()
```

In [32]:
```
plt.scatter(df['Revenue'],df['JioSubscribers'])
plt.show()
```



In [33]:
```
model.score(X,y)
```

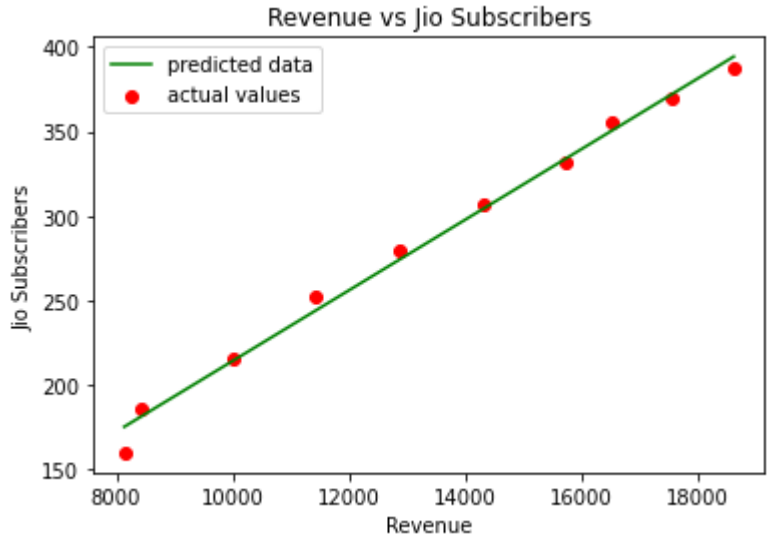
Out[33]:
```
0.9920788182785107
```

In [34]:
```
y_pred =model.predict(X)
```

In [35]:
```
y_pred
```

Out[35]:
```
array([175.39851552, 181.34220196, 214.75197625, 243.80304716,
       274.60594144, 304.53292403, 334.00109574, 350.53914257,
       371.83213855, 394.29301678])
```

In [36]:
```
plt.scatter(df['Revenue'],df['JioSubscribers'],c='r',label = 'actual values'
plt.plot(X,y_pred,c='g',label = 'predicted data')
plt.title('Revenue vs Jio Subscribers')
plt.xlabel('Revenue')
plt.ylabel('Jio Subscribers')
plt.legend()
plt.show()
```



Linear Regression with Multiple variables

- input = more than one feature
- output = single target
- y = ax^2+bx+c (degree = 2)
- y =ax^3+bx^2+cx+1(degree = 3)

In [37]:
```
# 1.read the data and get the data
df = pd.read_csv("https://raw.githubusercontent.com/AP-State-Skill-Developme
```

In [38]:
```
1 df
```

Out[38]:

	Gender	Age Range	Head Size(cm^3)	Brain Weight(grams)
0	1	1	4512	1530
1	1	1	3738	1297
2	1	1	4261	1335
3	1	1	3777	1282
4	1	1	4177	1590
...	...	...	...	...
232	2	2	3214	1110
233	2	2	3394	1215
234	2	2	3233	1104
235	2	2	3352	1170
236	2	2	3391	1120

237 rows × 4 columns

In [39]:
```
1 df.head()
```

Out[39]:

	Gender	Age Range	Head Size(cm^3)	Brain Weight(grams)
0	1	1	4512	1530
1	1	1	3738	1297
2	1	1	4261	1335
3	1	1	3777	1282
4	1	1	4177	1590

In [40]:
```
1 df.columns
```

Out[40]:
```
Index(['Gender', 'Age Range', 'Head Size(cm^3)', 'Brain Weight(grams)'], dtype
='object')
```

In [41]:
```
1 df.Gender.unique()
```

Out[41]:
```
array([1, 2], dtype=int64)
```

In [42]:
```
1 df['Gender'].value_counts()
```

Out[42]:
```
1    134
2    103
Name: Gender, dtype: int64
```

In [43]:
```
1 df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 237 entries, 0 to 236
Data columns (total 4 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   Gender               237 non-null    int64
 1   Age Range            237 non-null    int64
 2   Head Size(cm^3)      237 non-null    int64
 3   Brain Weight(grams)  237 non-null    int64
dtypes: int64(4)
memory usage: 7.5 KB
```

In [44]:
```
1 df.describe()
```

Out[44]:

	Gender	Age Range	Head Size(cm^3)	Brain Weight(grams)
count	237.000000	237.000000	237.000000	237.000000
mean	1.434599	1.535865	3633.991561	1282.873418
std	0.496753	0.499768	365.261422	120.340446
min	1.000000	1.000000	2720.000000	955.000000
25%	1.000000	1.000000	3389.000000	1207.000000
50%	1.000000	2.000000	3614.000000	1280.000000
75%	2.000000	2.000000	3876.000000	1350.000000
max	2.000000	2.000000	4747.000000	1635.000000

In [45]:
```
1 df.isnull().sum()
```

Out[45]:
```
Gender                0
Age Range             0
Head Size(cm^3)       0
Brain Weight(grams)   0
dtype: int64
```

In [79]:
```
1 # separate input and ouput
2 inputdata = df[['Head Size(cm^3)']]
3 outputdata = df['Brain Weight(grams)']
4 len(inputdata)
5 len(outputdata)
```

Out[79]:
```
237
```

In [80]:
```
1 inputdata.columns
```

Out[80]:
```
Index(['Head Size(cm^3)'], dtype='object')
```

In [81]:
```
1 # applying the model
2 from sklearn.linear_model import LinearRegression
```

In [82]:
```
1 linear = LinearRegression()
```

In [83]:
```
1 linear.fit(inputdata,outputdata)
```

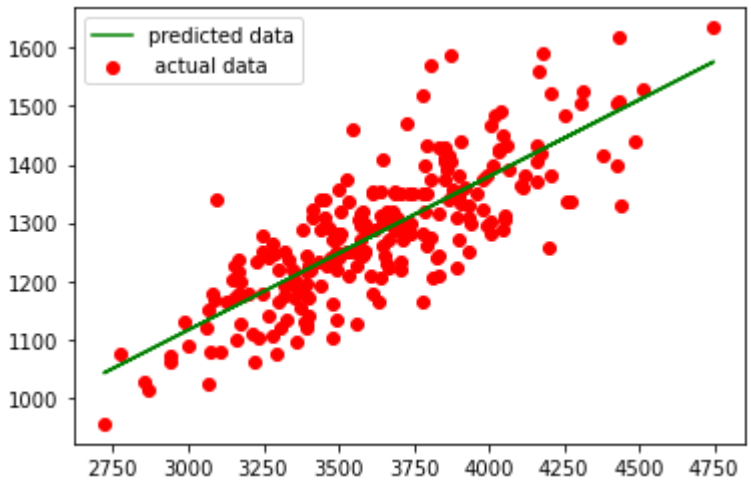
Out[83]:
```
LinearRegression()
```

In [84]:
```
1 pred = linear.predict(inputdata)
```

In [85]:
```
1 linear.score(inputdata,outputdata)
```

Out[85]:
```
0.639311719957
```

In [87]:
```
1 plt.scatter(inputdata,outputdata,label= ' actual data',c='red')
2 plt.plot(inputdata,pred,c='g',label = 'predicted data')
3 plt.legend()
4 plt.show()
```



In [ ]:
```
1
```