Project on Linear Regression

- Linear regression:
- It is also a type of machine-learning algorithmSupervised machine-learning algorithm.
- Learns from the labelled datasets and maps the data points to the most optimized linear function.
- These points can be used for prediction on new dataset
- **Dependent,Independent variable one parameter depends on another parameter - dependent variabledoesnot not depend on other variablesindependent variable
- 1.Find mean for both dependent and independent variables x',y'
- 2.Find difference between x point and x-mean(x-x')
- Find difference between y point and y-mean(y-y')
- 3.sum of squares(x-x')^2
- 4.sum of Product of (x-x') and (y-y')
- y=a+bx
- b=s(xy)/s(xx)=sum(x(i)-x')(y(i)-y')/sum(x(i)-x')^2
- =sum(products)/sum(squares)
- a=y'-b'x

CODE:

#Importing libraries:

```
import pandas as pd

from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error import
matplotlib.pyplot as plt import seaborn as sns

# Load the health insurance cost dataset CSV file into a DataFrame

data = pd.read_csv('/content/drive/MyDrive/Colab
Notebooks/Health_insurance.csv') print(data)
Output:
```

```
    age     sex     bmi  children smoker     region      charges  0
19  female  27.900         0    yes  southwest  16884.92400
```

1	18	male	33.770	1	no	southeast	1725.55230
2	28	male	33.000	3	no	southeast	4449.46200
3	33	male	22.705	0	no	northwest	21984.47061
4	32	male	28.880	0	no	northwest	3866.85520
5	31	female	25.740	0	no	southeast	3756.62160
6	46	female	33.440	1	no	southeast	8240.58960
7	37	female	27.740	3	no	northwest	7281.50560
8	37	male	29.830	2	no	northeast	6406.41070
9	60	female	25.840	0	no	northwest	28923.13692
10	25	male	26.220	0	no	northeast	2721.32080
11	62	female	26.290	0	yes	southeast	27808.72510
12	23	male	34.400	0	no	southwest	1826.84300
13	56	female	39.820	0	no	southeast	11090.71780
14	27	male	42.130	0	yes	southeast	39611.75770
15	19	male	24.600	1	no	southwest	1837.23700
16	52	female	30.780	1	no	northeast	10797.33620
17	23	male	23.845	0	no	northeast	2395.17155
18	56	male	40.300	0	no	southwest	10602.38500
19	30	male	35.300	0	yes	southwest	36837.46700
20	60	female	36.005	0	no	northeast	13228.84695
21	30	female	32.400	1	no	southwest	4149.73600
22	18	male	34.100	0	no	southeast	1137.01100
23	34	female	31.920	1	yes	northeast	37701.87680
24	37	male	28.025	2	no	northwest	6203.90175
25	59	female	27.720	3	no	southeast	14001.13380
26	63	female	23.085	0	no	northeast	14451.83515
27	55	female	32.775	2	no	northwest	12268.63225
28	23	male	17.385	1	no	northwest	2775.19215
29	31	male	36.300	2	yes	southwest	38711.00000
30	22	male	35.600	0	yes	southwest	35585.57600
31	18	female	26.315	0	no	northeast	2198.18985
32	19	female	28.600	5	no	southwest	4687.79700
33	63	male	28.310	0	no	northwest	13770.09790
34	28	male	36.400	1	yes	southwest	51194.55914
35	19	male	20.425	0	no	northwest	1625.43375
36	62	female	32.965	3	no	northwest	15612.19335
37	26	male	20.800	0	no	southwest	2302.30000
38	35	male	36.670	1	yes	northeast	39774.27630
39	60	male	39.900	0	yes	southwest	48173.36100
40	24	female	26.600	0	no	northeast	3046.06200
41	31	female	36.630	2	no	southeast	4949.75870
42	41	male	21.780	1	no	southeast	6272.47720
43	37	female	30.800	2	no	southeast	6313.75900
44	38	male	37.050	1	no	northeast	6079.67150
45	55	male	37.300	0	no	southwest	20630.28351

```
46   18  female  38.665         2     no  northeast   3393.35635
     47 28  female  34.770         0     no  northwest   3556.92230
48   60  female  24.530         0     no  southeast  12629.89670

#Printing first 5 rows
data.head() OUTPUT:
```

age sex bmi children smoker region charges

0 19 female 27.900 0 yes southwest 16884.92400
1 18 male 33.770 1 no southeast 1725.55230

2 28 male 33.000 3 no southeast 4449.46200 **age sex**

bmi children smoker region charges

3 33 male 22.705 0 no northwest 21984.47061

4 32 male 28.880 0 no northwest 3866.85520

```
#Printing last 10 rows data.tail(10)
Output:
```

age sex bmi children smoker region charges

39 60 male 39.900 0 yes southwest 48173.36100

40 24 female 26.600 0 no northeast 3046.06200

41 31 female 36.630 2 no southeast 4949.75870

42 41 male 21.780 1 no southeast 6272.47720

43 37 female 30.800 2 no southeast 6313.75900

44 38 male 37.050 1 no northeast 6079.67150

45 55 male 37.300 0 no southwest 20630.28351

46 18 female 38.665 2 no northeast 3393.35635

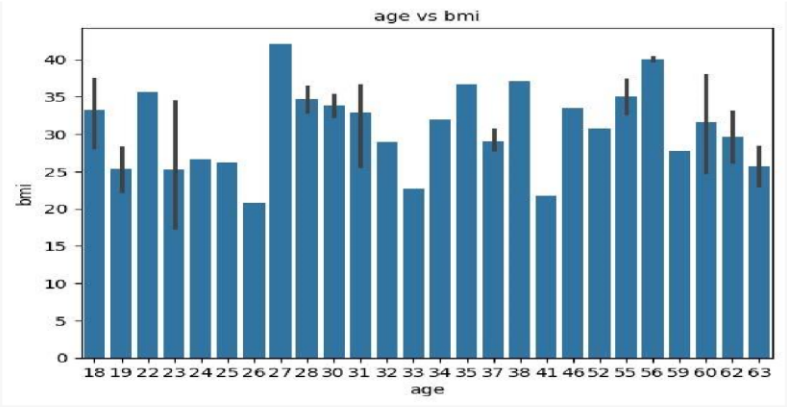
47 28 female 34.770 0 no northwest 3556.92230
48 60 female 24.530 0 no southeast 12629.89670

```
#Printing number  of rows and columns
data.shape
Output:
```

(49, 7)

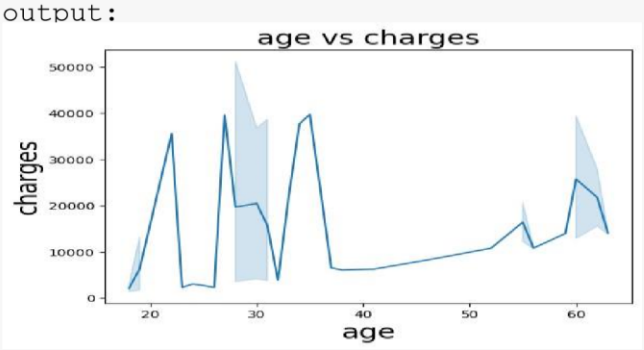
```
#Printing age and charge with no of rows and columns
X = data[['age']]
Y = data['charges']
print(X.shape)
print(Y.shape)
```

```
Output:
(49 ,  1)
(49,)
#Plotting the graph using barplot:
sns.barplot(x="age",y="bmi",data=data)
plt.title("age vs bmi",size=10)
plt.xlabel("age",size=10) plt.ylabel("bmi",size=10)
plt.show()
Output:
```



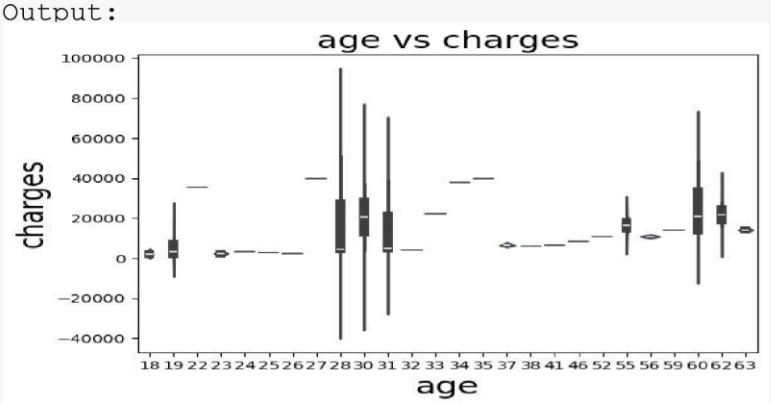

```
#Printing the graph by using Lineplot:
b=sns.lineplot(x='age',y='charges',data=data)
plt.title("age vs charges",size=20)
plt.xlabel("age",size=20)
plt.ylabel("charges",size=20)
plt.show()
```

output:




```
#printing the graph by using violinplot:
b=sns.violinplot(x='age',y='charges',data=data)
plt.title("age vs charges",size=20)
plt.xlabel("age",size=20)
plt.ylabel("charges",size=20)
plt.show()
```

Output:




```
# Split the data into training and testing sets
X_train, X_test, Y_train, Y_test = train_test_split(X, Y,
test_size=0.2, random_state=0)
print(X_test.shape)
print(X_train.shape)
```

```
Output:
(10, 1)
(39, 1)
            te a linear regression
            model =
            LinearRegression()  n the
            model fit(X_train,
Output:     n)
```

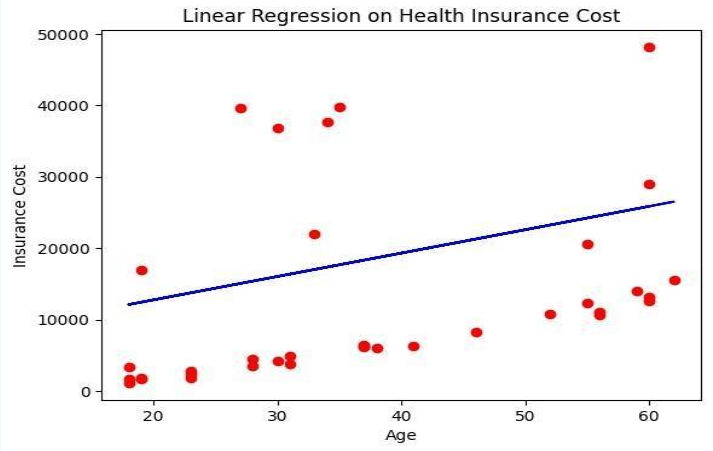
LinearRegression

```
LinearRegression()
#predict
Y_pred=model.predict(X_test)
print(Y_pred) print(Y_pred.shape)
```

```
Output:
[16366.48441495 16693.72656182 26838.23311464 13421.30509316
12439.57865257 14730.27368062 15384.75797436 14075.78938689
18329.93729614 14403.03153376 26510.99096778 12112.3365057
 26838.23311464 24220.29593972 25856.50667405 15384.75797436
 12112.3365057  24547.53808659 12439.57865257 13748.54724003
 12112.3365057  23238.56949912 16366.48441495 25856.50667405
 19638.90588361 18329.93729614 24547.53808659 25529.26452718
 16366.48441495 13748.54724003 12439.57865257 15057.51582749
 17675.45300241 12112.3365057  13748.54724003 18329.93729614]
(36,)
```

```
# Plot the regression line for one feature (e.g., age)
plt.scatter(X_train, Y_train, color = 'red')
plt.plot(X_train,model.predict(X_train), color =
'blue',label='Regression Line') plt.xlabel('Age')
plt.ylabel('Insurance Cost') plt.title('Linear Regression on
Health Insurance Cost') plt.show()
```

Output:




```
plt.scatter(X_test, Y_test, color = 'red')
plt.plot(X_train,model.predict(X_train), color =
'blue',label='Regression Line')
```

```
plt.xlabel('Age')
plt.ylabel('Insurance Cost')
plt.title('Linear Regression on Health Insurance Cost')
plt.show()
```

Output: