

Mohd Zain Multivariate Regression

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
from google.colab import files
al=files.upload()
```



Choose files insurance.csv

- **insurance.csv**(text/csv) - 55628 bytes, last modified: 18/08/2022 - 100% done
Saving insurance.csv to insurance.csv

```
import io
df=pd.read_csv(io.BytesIO(al['insurance.csv']))
```

```
df.head(7)
```

	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	


```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score
import warnings
warnings.filterwarnings("ignore")
```

```
shape=df.shape
print("Dataset contains {} rows and {} columns".format(shape[0],shape[1]))
```

Dataset contains 1338 rows and 7 columns

```
x=df.iloc[:, :6].values
y=df.iloc[:, 6].values
```

```
#transforming categorical input features to numerical form
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
label = le.fit_transform(df['region'])
label2=le.fit_transform(df['smoker'])
label3=le.fit_transform(df['sex'])
df.drop('smoker',axis=1,inplace=True)
df.drop("region", axis=1, inplace=True)
df.drop('sex',axis=1,inplace=True)
df["region"] = label
df['smoker']=label2
df['sex']=label3
df
```

	age	bmi	children	charges	region	smoker	sex	
0	19	27.900	0	16884.92400	3	1	0	
1	18	33.770	1	1725.55230	2	0	1	
2	28	33.000	3	4449.46200	2	0	1	
3	33	22.705	0	21984.47061	1	0	1	
4	32	28.880	0	3866.85520	1	0	1	
...	
1333	50	30.970	3	10600.54830	1	0	1	
1334	18	31.920	0	2205.98080	0	0	0	
1335	18	36.850	0	1629.83350	2	0	0	
1336	21	25.800	0	2007.94500	3	0	0	
1337	61	29.070	0	29141.36030	1	1	0	

1338 rows × 7 columns

```
from sklearn.model_selection import train_test_split
# splitting the data
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.3, r
```

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

LinearRegression()

```
y_pred = LR.predict(x_test)
```

```
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
```

```
mean_absolute_error(y_test, y_pred)
```

```
7.081389095893695e-12
```

```
mean_squared_error(y_test, y_pred)
```


```
8.530660158015776e-23
```

```
acc=r2_score(y_test, y_pred)
```

```
acc
```

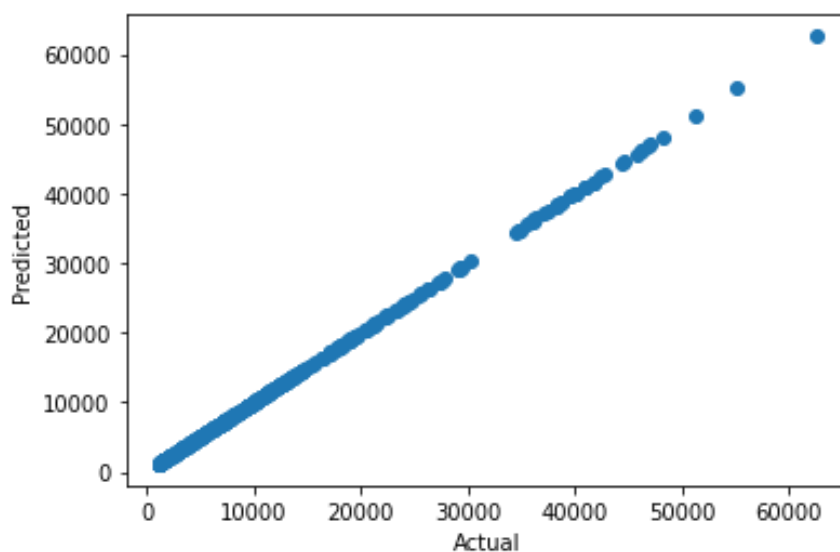
```
1.0
```

```
pred_df=pd.DataFrame({'Actual Value':y_test,'Predicted Value':y_pred,'Difference':y_test-y_pred})
pred_df
```

	Actual Value	Predicted Value	Difference	
0	7281.50560	7281.50560	8.185452e-12	
1	5267.81815	5267.81815	7.275958e-12	
2	12347.17200	12347.17200	1.818989e-12	
3	24513.09126	24513.09126	3.637979e-12	
4	3736.46470	3736.46470	8.185452e-12	
...	
397	24106.91255	24106.91255	3.637979e-12	
398	17878.90068	17878.90068	-1.091394e-11	
399	22462.04375	22462.04375	-3.637979e-12	
400	1391.52870	1391.52870	2.046363e-12	
401	8240.58960	8240.58960	3.637979e-12	

402 rows × 3 columns

```
plt.scatter(y_test, y_pred);
plt.xlabel('Actual');
plt.ylabel('Predicted');
```



```
sns.regplot(x=y_test,y=y_pred,ci=None,color='blue');
```

