

```
# Name:Sheikpareeth
# Multiple Linear regression
# Machine Learning,Supervised,regression,by used
SVM method
```

In [1]:

```
#Import the Libraries and put nicknames
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

In [2]:

```
#Reading the dataset
dataset=pd.read_csv("insurance_pre.csv")
```

In [3]:

dataset

Out[3]:

	age	sex	bmi	children	smoker	charges
0	19	female	27.900	0	yes	16884.92400
1	18	male	33.770	1	no	1725.55230
2	28	male	33.000	3	no	4449.46200
3	33	male	22.705	0	no	21984.47061
4	32	male	28.880	0	no	3866.85520
...
1333	50	male	30.970	3	no	10600.54830
1334	18	female	31.920	0	no	2205.98080
1335	18	female	36.850	0	no	1629.83350
1336	21	female	25.800	0	no	2007.94500
1337	61	female	29.070	0	yes	29141.36030

1338 rows × 6 columns

In [4]:

```
#Above the value is categorical value so i have used "one hot encoding method" (we cannot a
#I have removed the duplcates or dummy value
datasets=pd.get_dummies(dataset,drop_first=True)
```

In [5]:

```
#Show the datasets value is converted into the numbers
datasets
```

Out[5]:

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

1338 rows × 6 columns

In [6]:

```
#provide input columns name for the easily without seeing the table
dataset.columns
```

Out[6]:

```
Index(['age', 'sex', 'bmi', 'children', 'smoker', 'charges'], dtype='object')
t')
```

In [7]:

```
#Put the input and aouput column name
indep=datasets[['age', 'bmi', 'children', 'sex_male', 'smoker_yes']]
dep=datasets[["charges"]]
```

In []:

```
#output values (dependent)
dep
```

In []:

```
#Input values (independent)
indep
```

In [8]:

```
#split into training set and test set  
from sklearn.model_selection import train_test_split  
X_train,X_test,y_train,y_test=train_test_split(indep,dep,test_size=0.3,random_state=0)
```

In [9]:

```
#X_training value with their columns and rows numbers  
X_train.shape
```

Out[9]:

(936, 5)

In [10]:

```
#y_training value with their columns and rows numbers  
y_train.shape
```

Out[10]:

(936, 1)

In [11]:

```
#X_test value with their columns and rows numbers  
X_test.shape
```

Out[11]:

(402, 5)

In [12]:

```
#y_test value with their columns and rows numbers  
y_test.shape
```

Out[12]:

(402, 1)

In [13]:

```
#Create the model for multiple linear regression  
#Assign the values of linear regression into the regressor(LHS=RHS)  
#This is only formula so we will substitute the value using .fit  
from sklearn.linear_model import LinearRegression  
regressor=LinearRegression()  
regressor.fit(X_train,y_train)#y=w*x1+b0 for this equation we got value for b1 and b0
```

Out[13]:

LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)

In [14]:

```
#viewing the b1 and b0 value  
#viewing the weight and bias value  
weight=regressor.coef_  
print("weight of the model={}".format(weight))  
bias=regressor.intercept_  
print("intercept of the model={}".format(bias))
```

```
weight of the model=[[ 257.8006705    321.06004271   469.58113407   -41.748  
25718  
23418.6671912  ]]  
intercept of the model=[-12057.244846]
```

In [163]:

```
from sklearn.svm import SVR  
regressor=SVR(kernel="linear",C=1000000)  
regressor=regressor.fit(X_train,y_train)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py:724:  
DataConversionWarning: A column-vector y was passed when a 1d array was expected.  
Please change the shape of y to (n_samples, ), for example using ravel().  
(  
y = column_or_1d(y, warn=True)
```

In [164]:

```
#Evaluation metrics to use test set  
#y_test output of predicted value  
y_pred=regressor.predict(X_test)
```

In [165]:

```
#Find the r2 value  
from sklearn.metrics import r2_score  
r_score=r2_score(y_test,y_pred)
```

In [166]:

```
r_score
```

Out[166]:

```
0.7853460209312159
```

In [167]:

```
#Save the model use pickle  
import pickle
```

In [168]:

```
#Give the file name and write the model  
filename="mulline_final.sav"  
pickle.dump(regressor,open(filename,"wb"))
```

In [169]:

```
#Load the model and read it  
model=pickle.load(open(filename,"rb"))
```

In [170]:

```
#Improve r2 value use trial and error method  
age_input=float(input("Age:"))  
bmi_input=float(input("BMI:"))  
children_input=float(input("Children:"))  
sex_male_input=float(input("Sex Male 0 or 1:"))  
smoker_yes_input=float(input("Smoker Yes 0 or 1:"))
```

Age:19
BMI:27.900
Children:0
Sex Male 0 or 1:0
Smoker Yes 0 or 1:1

In [171]:

```
#Substitute values of input in result variable and finally got the result  
result=regressor.predict([age_input,bmi_input,children_input,sex_male_input,smoker_yes_inp  
print("result={}".format(result))
```

result=[25736.10691945]