**INTRODUCTION :**

We are using linear regression techniques.

**DATASET :**

We are using the**Medical Cost Personal Dataset** Insurance dataset containing training data.

Columns

age: age of primary beneficiary

sex: insurance contractor gender, female, male

bmi: Body mass index, providing an understanding of body, weights that are relatively high or low relative to height,

objective index of body weight (kg / m ^ 2) using the ratio of height to weight, ideally 18.5 to 24.9

children: Number of children covered by health insurance / Number of dependents

smoker: Smoking

region: the beneficiary's residential area in the US, northeast, southeast, southwest, northwest.

charges: Individual medical costs billed by health insurance

**NOTE :**

**Just upload class3.ipynb in the google colab and just upload the dataset insurance after that your code is ready to run.**

**CODE :**

***# -\*- coding: utf-8 -\*-***

***"""class3\_insurance.ipynb***

***Automatically generated by Colaboratory.***

***Original file is located at***

***https://colab.research.google.com/drive/1rwKhmOjxahaopxMvIwHCgMqy7eMd77sb***

***"""***

***import pandas as pd***

***import numpy as np***

***import seaborn as sns***

***train=pd.read\_csv('insurance.csv')***

***train.head()***

***sns.heatmap(train.isnull(),yticklabels=False,cbar=False)***

***"""this data is clean we dirrectly go to model"""***

***cat\_features=[i for i in train.columns if train.dtypes[i]=='object']***

***cat\_features***

***def category\_onehot\_multcols(multcolumns):***

***df\_final=train***

***i=0***

***for fields in multcolumns:***

***print(fields)***

***df1=pd.get\_dummies(train[fields],drop\_first=True)***

***train.drop([fields],axis=1,inplace=True)***

***if i==0:***

***df\_final=df1.copy()***

***else:***

***df\_final=pd.concat([df\_final,df1],axis=1)***

***i=i+1***

***df\_final=pd.concat([train,df\_final],axis=1)***

***return df\_final***

***train1=category\_onehot\_multcols(cat\_features)***

***train1***

***from sklearn.linear\_model import LinearRegression***

***from sklearn.model\_selection import train\_test\_split***

***x = train1.drop(['charges'], axis = 1)***

***y = train1.charges***

***x\_train,x\_test,y\_train,y\_test = train\_test\_split(x,y, random\_state = 0)***

***lr = LinearRegression().fit(x\_train,y\_train)***

***y\_train\_pred = lr.predict(x\_train)***

***y\_test\_pred = lr.predict(x\_test)***

***print(lr.score(x\_test,y\_test))***

**RESULT :**

**Accuracy of linear regression is 79.58%**