## Problem Statement

In [1]:
```
#Medical insurance is the coverage that provides for the payments of benefit as a result of sickness or injury.
#It includes insurance for losses from accident, medical expense, disability, or accidental death and dismemberment.
#It is not ideal for people to live without any medical insurance as medical emergencies such as illnesses
#and accidents can arise out of nowhere.
#according to research, those without insurance die younger as a result of leading sick lives than those with insurance.
#This is because they may not seek medical care due to high costs and avoid regular health screenings.
#These will then lead to increased rates of communicable diseases, and higher insurance premiums.
#It is very important this days to have medical insurance because of the rising
#costs of healthcare as well as the evident need for adequate healthcare.
#when it comes to critical and emergency illnesses,
#it becomes very difficult for a family to quickly arrange for huge
#amounts of money required for treatment as most of the savings of a family are in the form of fixed assets,
#which cannot be liquidated quickly.
#The advantages of having medical insurance include:
#Cashless hospitalization where the insurance company will pay your medical bills to the hospital.
#Thus, you will not be required to bear the high treatment costs from your pockets.
#Peace of Mind as you will not have to worry about healthcare expenses,
#and it allows you to choose the best medical care for yourself and your family.
#Best care can also give good recovery, allowing you to get back to your healthy life.
#factors affecting one's health include includes
#Age
#Sex
#Body mass index
#Habits (such as non-smoker  or not)
#There are diseases that affects specific age range, sex,
#people with certain weight and people with certain habits (e.g., Smokers are more prone to TB than non-smokers).
#factors affecting the cost of medical insurance include
#Region
#Age
#Body mass index
```

## Downloading, Installing and initiating Python SciPy

In [ ]:
```
!pip install SciPy
!pip install numpy
!pip install matplotlib
!pip install pandas
!pip install sklearn
```

## Importing the Dependencies

In [3]:
```
import numpy as numpy
import pandas as panda
import matplotlib.pyplot as plt
import seaborn as sns


from sklearn.model_selection import cross_val_score
from sklearn.model_selection import StratifiedKFold
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC

from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn import metrics
from matplotlib import pyplot
from pandas.plotting import scatter_matrix
```

## Checking library Versions

In [4]:
```
# Python version
import sys
print('Python: {}'.format(sys.version))
# scipy
import scipy
print('scipy: {}'.format(scipy.__version__))
# numpy
import numpy
print('numpy: {}'.format(numpy.__version__))
# matplotlib
import matplotlib
print('matplotlib: {}'.format(matplotlib.__version__))
# pandas
import pandas
print('pandas: {}'.format(pandas.__version__))
# scikit-learn
import sklearn
print('sklearn: {}'.format(sklearn.__version__))
```

```
Python: 3.10.0 (tags/v3.10.0:b494f59, Oct  4 2021, 19:00:18) [MSC v.1929 64 bit (AMD64)]
scipy: 1.7.3
numpy: 1.21.4
matplotlib: 3.5.0
pandas: 1.3.4
sklearn: 1.0.2
```

## Loading The Data (Data Collection)

In [5]:
```
dataset = panda.read_excel('SurveyDatasets.xls')
```

In [6]:
```
#print out first five rows of dataset.
dataset.head()
```

Out[6]:

| | 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 |

## Summarizing the Dataset

In [7]:
```
#the shape of dataset
print(dataset.shape)
```

```
(1338, 7)
```

In [8]:
```
# Statistical Summary
print(dataset.describe())
```

```
                age         bmi     children       charges
count  1338.000000  1338.000000  1338.000000   1338.000000
mean     39.207025    30.663397     1.094918  13270.422265
std      14.049960     6.098187     1.205493  12110.011237
min      18.000000    15.960000     0.000000   1121.873900
25%      27.000000    26.296250     0.000000   4740.287150
50%      39.000000    30.400000     1.000000   9382.033000
75%      51.000000    34.693750     2.000000  16639.912515
max      64.000000    53.130000     5.000000  63770.428010
```

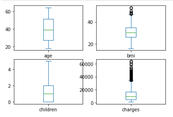
In [9]:
```
#Class Distribution
print(dataset.groupby('sex').size())
```

```
sex
female    662
male      676
dtype: int64
```

## Data Visualization

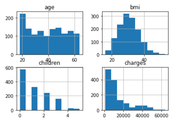
In [10]:
```
# Univariate Plots

# box and whisker plots
dataset.plot(kind='box', subplots=True, layout=(2,2), sharex=False, sharey=False)
pyplot.show()
```



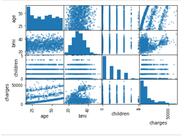

In [11]:
```
#for more visualization

# histograms
dataset.hist()
pyplot.show()
```




In [12]:
```
# Now we can look at the interactions between the variables.

# scatter plot matrix
scatter_matrix(dataset)
pyplot.show()
```




In [35]:
```
# Data Pre-Processing
dataset.replace({'sex':{'male':0,'female':1}},inplace =True)

dataset.replace({'smoker':{'yes':0,'no':1}},inplace =True)

dataset.replace({'region':{'southeast':0,'southwest':1,'northwest':3, 'northeast':2}},inplace =True)
dataset.head()
```

Out[35]:

| | age | sex | bmi | children | smoker | region | charges |
|---|---|---|---|---|---|---|---|
| 0 | 19 | 1 | 27.900 | 0 | 0 | 1 | 16884.92400 |
| 1 | 18 | 0 | 33.770 | 1 | 1 | 0 | 1725.55230 |
| 2 | 28 | 0 | 33.000 | 3 | 1 | 0 | 4449.46200 |
| 3 | 33 | 0 | 22.705 | 0 | 1 | 3 | 21984.47061 |
| 4 | 32 | 0 | 28.880 | 0 | 1 | 3 | 3866.85520 |

## Splitting dataset

In [14]:
```
X = dataset.drop(columns = 'charges', axis =1)
Y = dataset['charges']
```

## Training data

In [16]:
```
X_train, X_validation, Y_train, Y_validation = train_test_split(X, Y, test_size=0.2, random_state=2)
print(X_train.shape,X_validation.shape)
```

```
(1338, 6) (1070, 6) (268, 6)
```

In [17]:
```
#Model Training
```

In [28]:
```
instance_of_reg = LinearRegression()
```

In [29]:
```
instance_of_reg.fit(X_train,Y_train)
LinearRegression(copy_X = True, fit_intercept =True, n_jobs = None,normalize  = False)
```

Out[29]:
```
LinearRegression(normalize=False)
```

In [20]:
```
# Evaluation of the Model

#predictions
prediction_data = instance_of_reg.predict(X_train)
```

In [22]:
```
#R squared value
r_sqrd_train = metrics.r2_score(Y_train,prediction_data)
print('this is R squared value',r_sqrd_train )
```

```
this is R squared value 0.7518713667681967
```

In [ ]:
```
# we can see that LinearRegression has a high estimated accuracy score of about 0.75 or 75%.
```

## Build a Predictive system

In [36]:
```
#this should return 3637.97163732
data_to_test = (31,1,25.74,0,1,0)

data_as_numpy = numpy.asarray(data_to_test)
data_reshape = data_as_numpy.reshape(1,-1)
prediction = instance_of_reg.predict(data_reshape)

print(prediction)
```

```
[3637.97163732]
C:\Users\User\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\base.py:450: UserWarning: X does not have valid feature names, but LinearRegression was fitted with feature names
  warnings.warn(
```

In [ ]:
```
```