

```
In [1]: import pandas as pd
```

```
In [41]: df=pd.read_csv("C://Users//ALWAYS RAMESH//Downloads//insurance.csv")
```

```
In [42]: df
```

```
Out[42]:
```

	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
...	...	...	...	...	...	...	...
1333	50	male	30.970	3	no	northwest	10600.54830
1334	18	female	31.920	0	no	northeast	2205.98080
1335	18	female	36.850	0	no	southeast	1629.83350
1336	21	female	25.800	0	no	southwest	2007.94500
1337	61	female	29.070	0	yes	northwest	29141.36030

1338 rows × 7 columns

```
In [43]: from sklearn.preprocessing import LabelEncoder
```

```
In [50]: model=LabelEncoder()
df['sex']=model.fit_transform(x['sex'])
```

```
In [ ]: x=df[['age','sex','bmi','children','charges']]
```

```
In [61]: y=df['smoker']
```

```
In [62]: y
```

```
Out[62]: 0      yes
         1      no
         2      no
         3      no
         4      no
         ...
        1333    no
        1334    no
        1335    no
        1336    no
        1337    yes
        Name: smoker, Length: 1338, dtype: object
```

```
In [66]: x
```

```
Out[66]:
```

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

1338 rows × 5 columns

```
In [67]: from sklearn.model_selection import train_test_split
```

```
In [68]: x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3,random_state=4)
```

```
In [69]: from sklearn.linear_model import LogisticRegression
```

```
In [70]: model=LogisticRegression()
         model.fit(x_train,y_train)
```

```
C:\ProgramData\anaconda3\Lib\site-packages\sklearn\linear_model\_logistic.py:469: ConvergenceWarning: lbfgs failed to converge (status=1):  
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.
```

Increase the number of iterations (max\_iter) or scale the data as shown in:

<https://scikit-learn.org/stable/modules/preprocessing.html>

Please also refer to the documentation for alternative solver options:

[https://scikit-learn.org/stable/modules/linear\\_model.html#logistic-regression](https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression)

```
n_iter_i = _check_optimize_result(
```

Out[70]:

▼ LogisticRegression ⓘ ?

LogisticRegression()

In [71]: `y_pred=model.predict(x_test)`

In [72]: `y_pred`

```
Out[72]: array(['no', 'no', 'no', 'no', 'no', 'no', 'no', 'no', 'yes', 'yes', 'no',
               'no', 'no', 'no', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no',
               'yes', 'no', 'no', 'no', 'no', 'no', 'no', 'yes', 'no', 'no', 'no',
               'yes', 'no', 'no', 'no', 'no', 'no', 'no', 'no', 'no', 'yes', 'no',
               'yes', 'no', 'no', 'yes', 'no', 'no', 'no', 'no', 'no', 'no', 'no',
               'no', 'no', 'no', 'yes', 'no', 'no', 'no', 'no', 'no', 'no', 'yes',
               'no', 'no', 'no', 'no', 'no', 'no', 'no', 'no', 'no', 'yes', 'no',
               'no', 'no', 'yes', 'no', 'no', 'no', 'yes', 'yes', 'no', 'no',
               'no', 'no', 'no', 'yes', 'no', 'no', 'no', 'no', 'no', 'no', 'no',
               'yes', 'no', 'no', 'no', 'no', 'no', 'no', 'no', 'no', 'no', 'yes',
               'yes', 'no', 'no', 'no', 'no', 'no', 'yes', 'no', 'no', 'no', 'no',
               'no', 'no', 'no', 'no', 'no', 'no', 'no', 'no', 'no', 'no', 'no',
               'no', 'no', 'no', 'no', 'no', 'no', 'no', 'no', 'no', 'no', 'no',
               'no', 'yes', 'no', 'no', 'no', 'no', 'yes', 'no', 'no', 'no', 'no',
               'no', 'no', 'no', 'no', 'no', 'no', 'no', 'no', 'no', 'no', 'no',
               'no', 'yes', 'no', 'no', 'no', 'no', 'yes', 'no', 'no', 'no', 'no',
               'no', 'no', 'no', 'no', 'no', 'no', 'no', 'no', 'no', 'no', 'no',
               'no', 'no', 'no', 'no', 'no', 'no', 'no', 'no', 'no', 'no', 'no',
               'yes', 'no', 'no', 'no', 'no', 'no', 'yes', 'yes', 'no', 'no',
               'yes', 'no', 'no', 'no', 'no', 'yes', 'no', 'no', 'no', 'no',
               'yes', 'no', 'no', 'no', 'no', 'no', 'yes', 'no', 'no', 'no',
               'yes', 'no', 'no', 'no', 'no', 'no', 'yes', 'no', 'no', 'no',
               'no', 'no', 'no', 'yes', 'no', 'no', 'no', 'no', 'no', 'yes', 'no',
               'no', 'no', 'no', 'no', 'no', 'no', 'no', 'no', 'no', 'no', 'yes',
               'no', 'no', 'no', 'no', 'no', 'no', 'no', 'no', 'no', 'no', 'yes',
               'no', 'no', 'no', 'no', 'no', 'no', 'yes', 'no', 'no', 'no', 'no',
               'no', 'no', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes', 'no',
               'no', 'no', 'no', 'no', 'no', 'no', 'no'], dtype=object)
```

```
In [74]: from sklearn.metrics import confusion_matrix, classification_report
```

```
In [84]: graph=confusion_matrix(y_test,y_pred)
          print(graph)
```

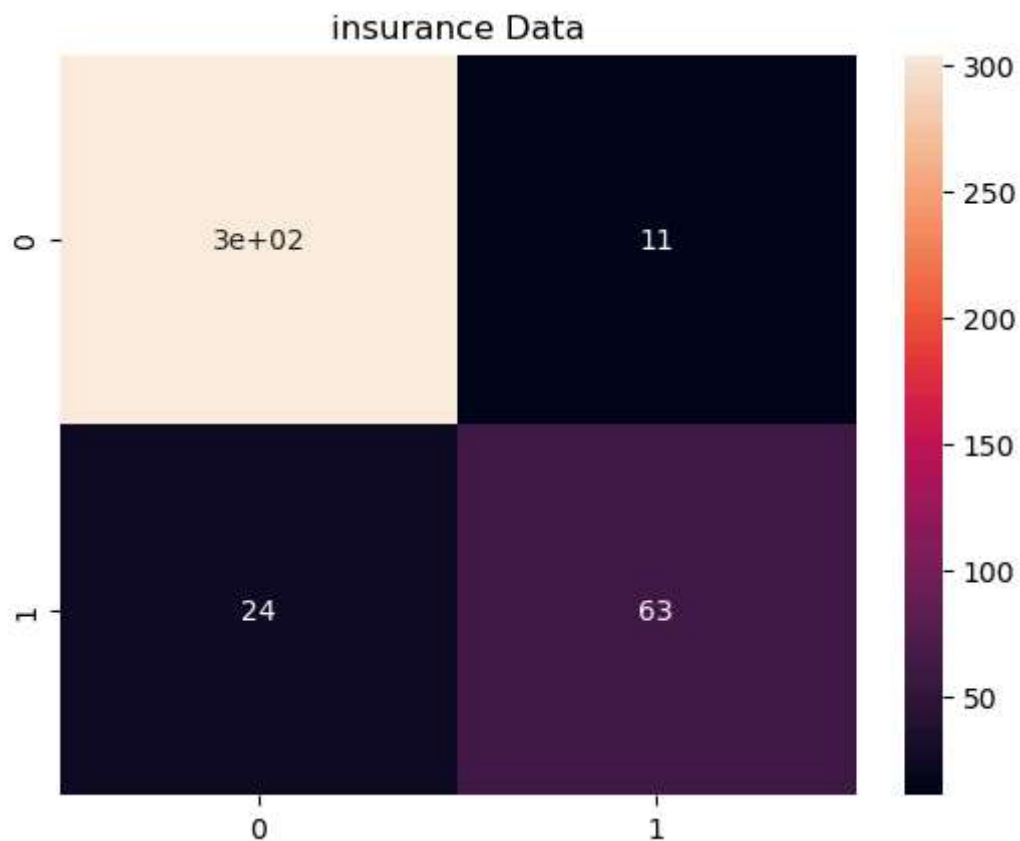
```
[[304  11]
 [ 24  63]]
```

```
In [79]: print(classification_report(y_test,y_pred))
```

	precision	recall	f1-score	support
no	0.93	0.97	0.95	315
yes	0.85	0.72	0.78	87
accuracy			0.91	402
macro avg	0.89	0.84	0.86	402
weighted avg	0.91	0.91	0.91	402

```
In [91]: import seaborn as sns
import matplotlib.pyplot as plt
```

```
In [92]: sns.heatmap(graph,annot=True)
plt.title('insurance Data')
plt.show()
```



```
In [101... from sklearn.neighbors import KNeighborsClassifier
```

```
In [105... model = KNeighborsClassifier(n_neighbors=3)
model.fit(x_train,y_train)
```

```
Out[105... KNeighborsClassifier
KNeighborsClassifier(n_neighbors=3)
```

```
In [ ]:
```

```
In [107... prediction = model.predict(x_test)
print("Predicted Class:", prediction)
```

```
Predicted Class: ['no' 'no' 'no' 'no' 'no' 'no' 'no' 'yes' 'yes' 'no' 'no' 'no' 'no'
'no'
'no' 'yes' 'no' 'no' 'no' 'no' 'no' 'yes' 'no' 'no' 'no' 'no' 'no' 'no'
'yes' 'no' 'no' 'no' 'yes' 'no' 'no' 'no' 'no' 'no' 'no' 'yes' 'no' 'yes'
'no' 'yes' 'no' 'no' 'yes' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no'
'no' 'yes' 'no' 'no' 'no' 'no' 'no' 'no' 'yes' 'no' 'no' 'no' 'no' 'no'
'no' 'no' 'no' 'no' 'yes' 'no' 'no' 'no' 'yes' 'no' 'no' 'no' 'yes' 'yes'
'no' 'no' 'no' 'yes' 'no' 'yes' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'yes'
'yes' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'yes' 'yes' 'no' 'no' 'no'
'no' 'no' 'yes' 'no' 'no' 'no' 'no' 'no' 'yes' 'no' 'no' 'no' 'yes' 'no'
'no' 'no' 'no' 'yes' 'no' 'yes' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no'
'no' 'no' 'no' 'no' 'no' 'no' 'yes' 'no' 'no' 'no' 'no' 'no' 'no' 'no'
'no' 'no' 'yes' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'yes' 'no'
'no' 'yes' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'yes' 'no' 'no'
'no' 'no' 'no' 'no' 'no' 'no' 'yes' 'no' 'no' 'yes' 'no' 'yes' 'no' 'no'
'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no'
'no' 'no' 'yes' 'yes' 'yes' 'no' 'no' 'yes' 'no' 'no' 'no' 'no' 'no' 'no'
'no' 'yes' 'no' 'yes' 'no' 'no' 'yes' 'yes' 'no' 'yes' 'no' 'no' 'no'
'no' 'no' 'yes' 'yes' 'no' 'yes' 'no' 'no' 'no' 'no' 'no' 'yes' 'no' 'no'
'yes' 'no' 'yes' 'no' 'no' 'yes' 'no' 'no' 'yes' 'no' 'no' 'no' 'no' 'no'
'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no'
'yes' 'yes' 'no' 'yes' 'no' 'yes' 'no' 'no' 'yes' 'no' 'no' 'no' 'yes'
'no' 'no' 'no' 'yes' 'no' 'no' 'no' 'no' 'no' 'yes' 'no' 'no' 'no' 'no'
'no' 'no' 'yes' 'yes' 'no' 'yes' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no'
'yes' 'no' 'no' 'no' 'yes' 'no' 'no' 'no' 'no' 'no' 'yes' 'no' 'no' 'no'
'yes' 'no' 'no' 'no' 'no' 'no' 'yes' 'no' 'no' 'yes' 'yes' 'no' 'no' 'no'
'yes' 'no' 'yes' 'no' 'yes' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no'
'no' 'yes' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no'
'no' 'no' 'no' 'no' 'no' 'yes' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no'
'yes' 'yes' 'no' 'yes' 'yes' 'no' 'no' 'no' 'no' 'no' 'no' 'yes' 'no']
```

```
In [ ]:
```