Importing the dependencies

In [20]:
```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
import warnings
warnings.filterwarnings("ignore", message="X does not have valid feature names")
```

Data collection and data processing

In [21]:
```
sonar_data = pd.read_csv("sonar.mine.csv")
sonar_data.head()
```

Out[21]:

	Freq_1	Freq_2	Freq_3	Freq_4	Freq_5	Freq_6	Freq_7	Freq_8	Freq_9	Freq_10	...	Freq_52
0	0.0200	0.0371	0.0428	0.0207	0.0954	0.0986	0.1539	0.1601	0.3109	0.2111	...	0.0027
1	0.0453	0.0523	0.0843	0.0689	0.1183	0.2583	0.2156	0.3481	0.3337	0.2872	...	0.0084
2	0.0262	0.0582	0.1099	0.1083	0.0974	0.2280	0.2431	0.3771	0.5598	0.6194	...	0.0232
3	0.0100	0.0171	0.0623	0.0205	0.0205	0.0368	0.1098	0.1276	0.0598	0.1264	...	0.0121
4	0.0762	0.0666	0.0481	0.0394	0.0590	0.0649	0.1209	0.2467	0.3564	0.4459	...	0.0031

5 rows × 61 columns

Number of rows and Columns

In [22]:
```
sonar_data.shape
```

Out[22]:
```
(208, 61)
```

In [23]:
```
sonar_data.describe()
```

Out[23]:

	Freq_1	Freq_2	Freq_3	Freq_4	Freq_5	Freq_6	Freq_7	Fre
count	208.000000	208.000000	208.000000	208.000000	208.000000	208.000000	208.000000	208.000
mean	0.029164	0.038437	0.043832	0.053892	0.075202	0.104570	0.121747	0.134
std	0.022991	0.032960	0.038428	0.046528	0.055552	0.059105	0.061788	0.085
min	0.001500	0.000600	0.001500	0.005800	0.006700	0.010200	0.003300	0.005
25%	0.013350	0.016450	0.018950	0.024375	0.038050	0.067025	0.080900	0.080
50%	0.022800	0.030800	0.034300	0.044050	0.062500	0.092150	0.106950	0.112
75%	0.035550	0.047950	0.057950	0.064500	0.100275	0.134125	0.154000	0.169
max	0.137100	0.233900	0.305900	0.426400	0.401000	0.382300	0.372900	0.459

8 rows × 60 columns

In [24]:
```
sonar_data["Label"].value_counts()
```

Out[24]:
```
Label
M    111
R     97
Name: count, dtype: int64
```

M ----> Mine

R ----> Rock

In [25]:
```
sonar_data.groupby("Label").mean()
```

Out[25]:

Label	Freq_1	Freq_2	Freq_3	Freq_4	Freq_5	Freq_6	Freq_7	Freq_8	Freq_9	Fr
M	0.034989	0.045544	0.050720	0.064768	0.086715	0.111864	0.128359	0.149832	0.213492	0.2
R	0.022498	0.030303	0.035951	0.041447	0.062028	0.096224	0.114180	0.117596	0.137392	0.1

2 rows × 60 columns

In [26]:
```
# separating data and Labels
x =sonar_data.drop(columns="Label", axis =1)
y = sonar_data["Label"]
```

Training and test data

In [27]:
```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size= 0.1, stratify=
```

In [28]:
```
print(x.shape, x_train.shape, x_test.shape)
```

```
(208, 60) (187, 60) (21, 60)
```

Mode Training --> Logistic Regression

In [29]:
```
model= LogisticRegression()
```

In [30]:
```
#training the logistic regression model with training data
model.fit(x_train, y_train)
```

Out[30]:
```
▾ LogisticRegression

LogisticRegression()
```

Model Evaluation

In [31]:
```
# accuracy on training data
x_train_pred = model.predict(x_train)
training_data_accu = accuracy_score(x_train_pred, y_train)
```

In [32]:
```
print('accuracy on training data :',training_data_accu)
```

```
accuracy on training data : 0.8342245989304813
```

In [33]:
```
x_test_pred = model.predict(x_test)
test_data_accu = accuracy_score(x_test_pred, y_test)
```

In [34]:
```
print('accuracy on training data :',test_data_accu)
```

```
accuracy on training data : 0.7619047619047619
```

Making predictive system

In [35]:
```
input_data =(0.0286,0.0453, 0.0277,0.0174,0.0384,0.099,0.1201,0.1833,0.2105,0.3039,
)

#changing the input data to a numpy array
input_data_as_numpy_array = np.asarray(input_data)


#reshape the np array as we are predecting for one instance

input_data_reshaped = input_data_as_numpy_array.reshape(1,-1)
prediction = model.predict(input_data_reshaped)
if(prediction[0]=='R'):
    print('the object is Rock')
else:
    print('the object is a Mine')
```

```
the object is Rock
```

In [ ]:
```
```