importing dependencies

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score

#loading dataset into pandas data frame
sonar_data = pd.read_csv('mention path/name.csv')

#first 5 rows of data set
sonar_data.head()

#finding number of rows and columns
sonar_data.shape

#gives parameters related to statistical measures of data
sonar_data.describe()

#To know how many unique examples are there
sonar_data[60].value_counts()

#mean value for all columns rock and mine separately
sonar_data.groupby(60).mean()

#separating data and Labels
X = sonar_data.drop(columns=60, axis=1)
Y = sonar_data[60]

print(X)
print(Y)

#Training and Test data
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.1, stratify=Y, random_state=1)

#To show difference between original, training, testing data
print(X.shape, X_train.shape, X_test.shape)
```

Model training using Logistic regression

```
model = LogisticRegression()

#training logistic regression model with training model
model.fit(X_train,Y_train)
```

MODEL EVALUATION

```
#accuracy on training data
X_train_prediction = model.predit(X_train)
```

```
accuracy_data_accuracy = accuracy_score(X_train_prediction, Y_train)

#accuracy on test data
X_test_prediction = model.predict(X_test)
test_data_accuracy = accuracy_score(X_test_presiction, Y_test)

print('Accuracy on test data: ', test_data_accuracy)
```

MAKING A PREDICTIVE SYSTEM

```
input_data = ()
# changing the input_data to a numpy array

input_data_as_numpy_array = np.asarray(input_data)

#reshape the np array as we are predicting for one instance
input_data_reshaped = input_data_as_numpy_array.reshape(1,-1)

prediction = model.predict(input_data_reshaped)
print(prediction)

if (prediction[0]=='R'):
print('The object is a Rock')
else:
print('The object is a mine')
```