**PROBLEM STATEMENT** -

Rock vs Mine prediction using SONAR data. Considering a submarine during war between two countries. As it moves along the water it must try to identify through which it is passing through if its either a Mine or a Rock. Our goal is to make a system which can predict whether the object beneath the submarine is either a Rock or a Mine.

The sound signals from SONAR are transmitted and received back, these signals are then processed to identify the object beneath.

**WORKFLOW** -

**1.Collecting of sonar data** - An experiment is done in a laboratory using sonar system to send & receive (bounce back) signals from a metal cylinder (considered as mine) also from hard rocks.

**2.Data preprocessing** - We cannot use the data directly obtained from the sonar we need to preprocess it.

**3.Splitting of data** – After preprocessing the data we split it into TRAIN & TEST data. The purpose of splitting the data is, suppose there are 100 instances of data. We will train out ML model with 90 instances and we will test (evaluate) our ML model with 10 other instances.

**4.ML Model** – We are then going to feed our data to a **LOGISTIC REGRESSION MODEL**. The use of this model is, it works well under a binary classification problem as our goal is to identify the object, we consider this model.

**5.Training our ML model** - We then train our model with training data and get a trained logistic regression model. This model can now differentiate between a metal cylinder and a rock, and it can recognize them based on the sonar data.

**USED ALGORITHM -**

**SUPERVISED LEARNING**

Under this the machines are trained using well "labelled" training data, and based on that data, machines predict the output. The labelled data means some input data is already tagged with the correct output; the training data provided to the machines works as the supervisor that teaches the machines to predict the output correctly. It applies the same concept as a student learns under the supervision of the teacher.

The aim of a supervised learning algorithm is to **find a mapping function to map the input variable(x) with the output variable(y)**.

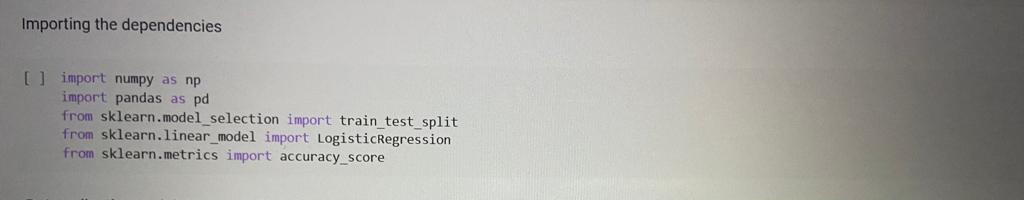
**WORKING-**

Models are trained using labelled dataset, where the model learns about each type of data. Once the training process is completed, the model is tested based on test data (a subset of the training set), and then it predicts the output.

**CODE FOLLOW UP -**

**Importing Required Libraries** -

* Use of NumPy is for the arrays and use of pandas here is for loading our data into tables called as data frames.
* For splitting our data, we use train\_test\_split function from sklearn.model\_selection. It is used to split arrays or matrices into random trains & test subsets.
* Importing our logistic regression model as LogisticRegression function from sklearn.linear\_model. This model is widely used for any binary classification problems. It is used to predict the probability of an event occurring based on input features.
* Then to find the accuracy of our model we use accuracy\_score function from sklearn.metrics.

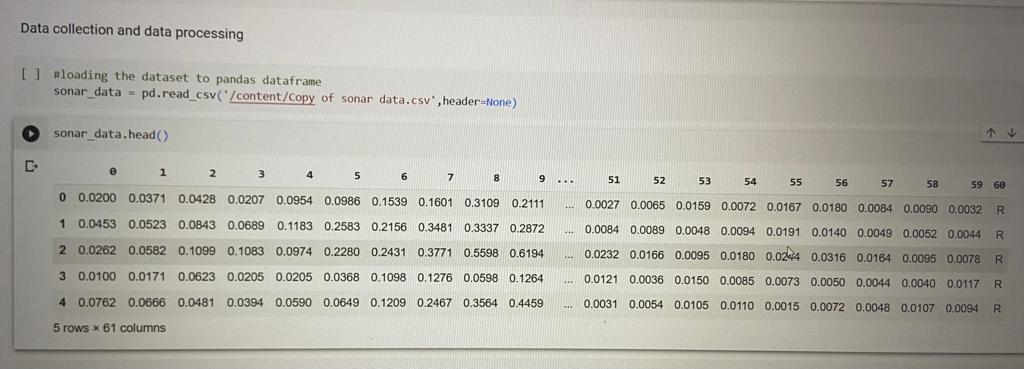


**Data Collection & Data Processing** -

**1.Loading the dataset to pandas Data frame** -

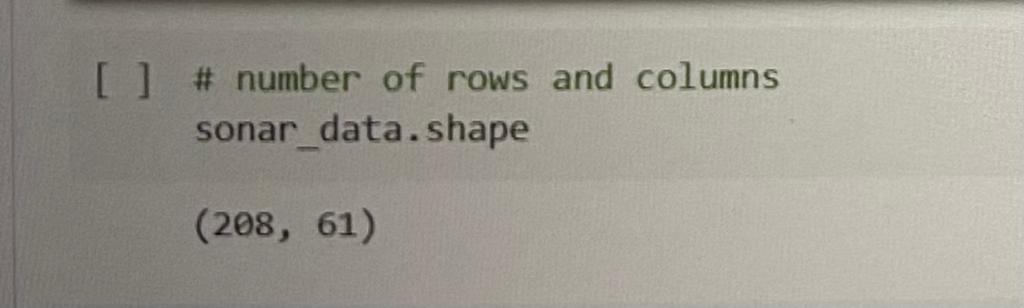
* Create a variable (sonar\_data) to load the data. As we have the data as csv file, we need to mention the name and location of the file.
* We don’t have any name for columns, so our header is None.

**2.Display 5 rows of our data set** -



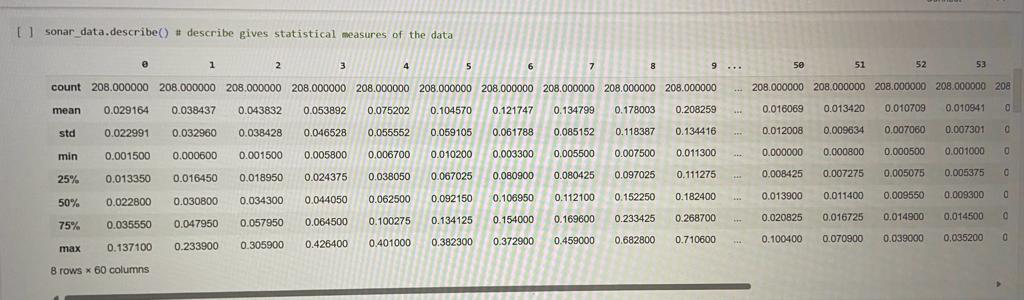
**3.Total number of rows & columns** -

* 208 rows represent the instances of data, 61 columns represent the features.



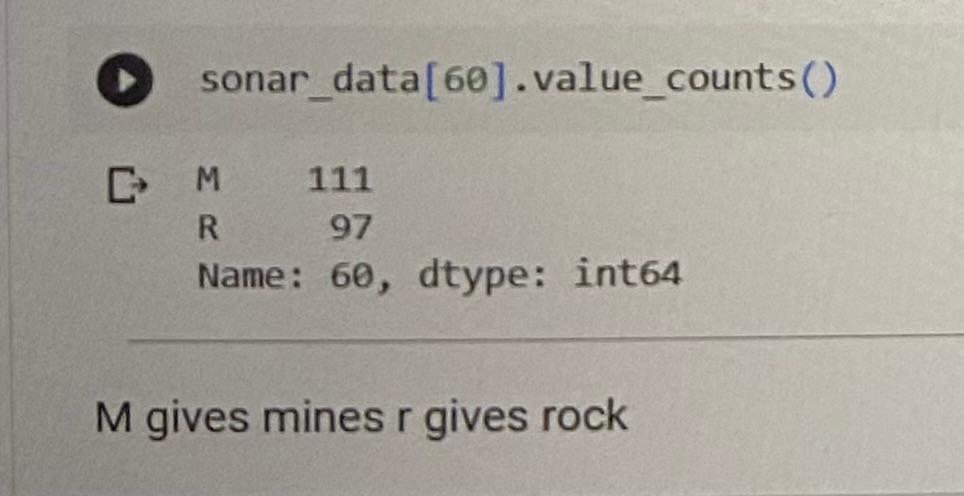
**4.Statistical measures** -

* Including mean, standard deviation, count and some other features which are important for some use cases.



**5.Finding number of rocks and mines in our dataset** -

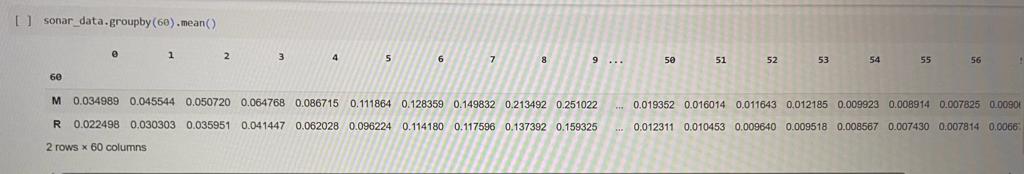
* [60] represents the column index in which rock and mine are specified.



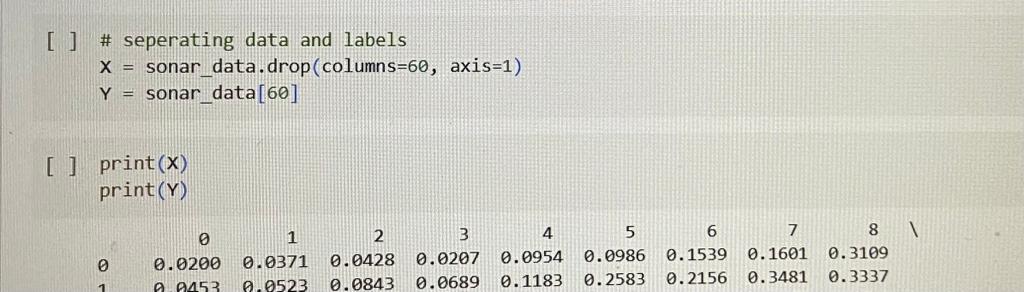
\*\*\*More the data more accurate is the model\*\*\*

**6.Grouping data** -

* We get the mean value for all column of mine and rock. The difference between them identifies the object.

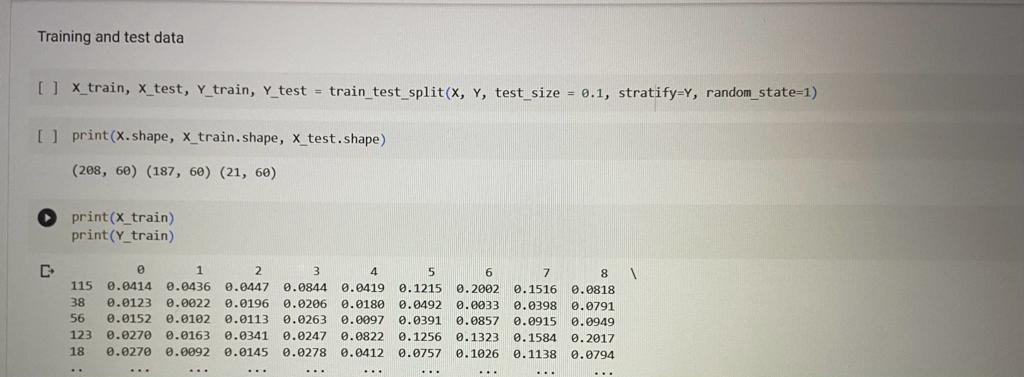


**7.Seperating data & labels** -



**Training And Test Data** -

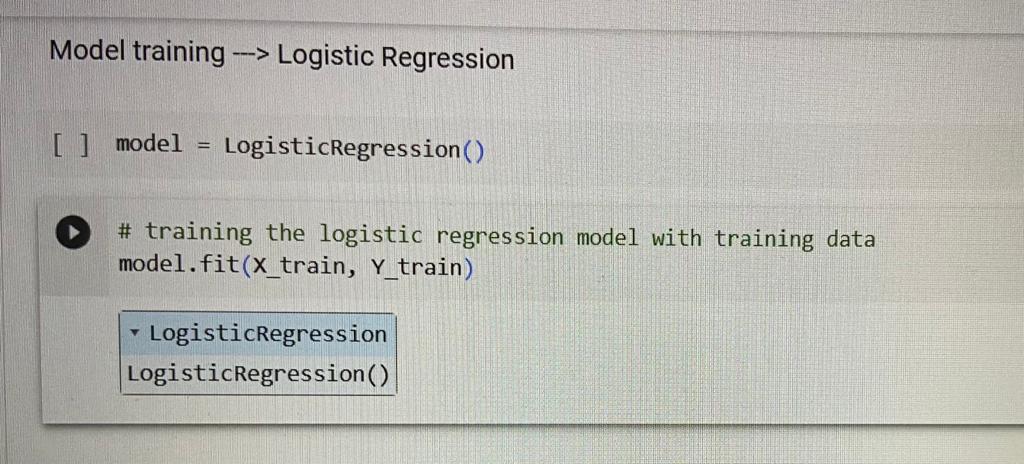
* First comes the train data then test data (represented by X) next the labels of train data and labels of test data (represented by Y). The **test\_size** defines for suppose say 0.1 we need 10 % of the data to be test data.
* While splitting the data for having almost equal rock and mine data we used **Stratify**.
* **Random\_state** is to split the data in a particular order, producing the same result for an integer value.
* Printing the total number of instances, train data and label data -



**Model Training using Logistic Regression** -

**1.Training Logistic Regression model with training data** -

* By including training data and training label



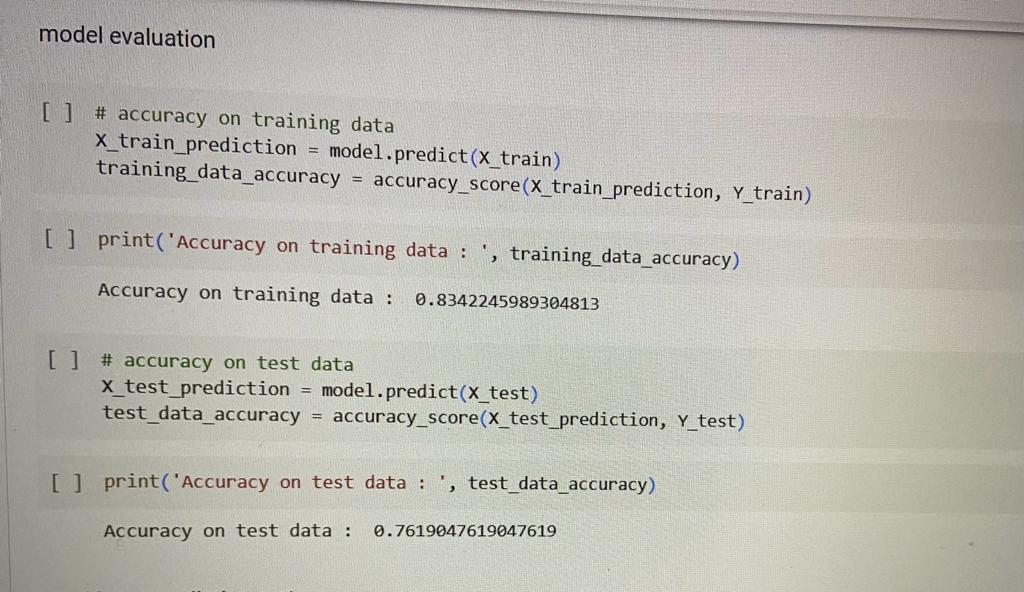
**Model Evaluation** -

**1.Accuracy on Training Data** -

* Acc > 70 % (good model).
* Printing training accuracy by comparing the prediction of our model with the original training label data.

**2.Accuracy on Test Data** -

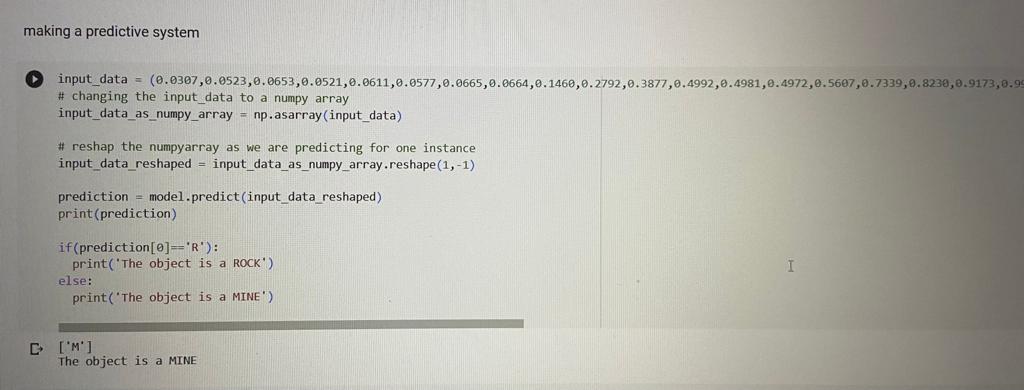
* Printing test accuracy by comparing the prediction of our model with the original test label data.



**A Predictive System** -

* We create the system by taking a few examples from the sonar data to identify if the object is either rock or mine, then we need to reshape the data as we are considering only one instance.

**Printing our prediction** -



**CODE -**

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 the dataset to pandas dataframe

sonar\_data = pd.read\_csv('\*/Path of the file/\*', header=None)

sonar\_data.head()

# number of rows and columns

sonar\_data.shape

sonar\_data.describe() # describe gives statistical measures of the data

sonar\_data[60].value\_counts()

sonar\_data.groupby(60).mean()

# seperating data and labels

X = sonar\_data.drop(columns=60, axis=1)

Y = sonar\_data[60]

print(X)

print(Y)

X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(X, Y, test\_size = 0.1, stratify=Y, random\_state=1)

print(X.shape, X\_train.shape, X\_test.shape)

print(X\_train)

print(Y\_train)

model = LogisticRegression()

# training the logistic regression model with training data

model.fit(X\_train, Y\_train)

# accuracy on training data

X\_train\_prediction = model.predict(X\_train)

training\_data\_accuracy = accuracy\_score(X\_train\_prediction, Y\_train)

print ('Accuracy on training data: ', training\_data\_accuracy)

# accuracy on test data

X\_test\_prediction = model.predict(X\_test)

test\_data\_accuracy = accuracy\_score(X\_test\_prediction, Y\_test)

print ('Accuracy on test data: ', test\_data\_accuracy)

input\_data = (0.0307,0.0523,0.0653,0.0521,0.0611,0.0577,0.0665,0.0664,0.1460,0.2792,0.3877,0.4992,0.4981,0.4972,0.5607,0.7339,0.8230,0.9173,0.9975,0.9911,0.8240,0.6498,0.5980,0.4862,0.3150,0.1543,0.0989,0.0284,0.1008,0.2636,0.2694,0.2930,0.2925,0.3998,0.3660,0.3172,0.4609,0.4374,0.1820,0.3376,0.6202,0.4448,0.1863,0.1420,0.0589,0.0576,0.0672,0.0269,0.0245,0.0190,0.0063,0.0321,0.0189,0.0137,0.0277,0.0152,0.0052,0.0121,0.0124,0.0055)

# changing the input\_data to a numpy array

input\_data\_as\_numpy\_array = np.asarray(input\_data)

# reshap the numpyarray 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')

**CONCLUSION -** The created system is designed to identify the object beneath a submarine if it’s either a Rock or a Mine.