REPORT

**1. Introduction**

Sonar technology is widely used for underwater target detection, and in this project, we are focusing on predicting whether the detected object is a rock (R) or a mine (M) based on sonar data. The dataset used for this analysis contains various features related to sonar signals.

**2. Data Preprocessing**

**2.1 Loading and Exploring the Dataset**

The first step involves loading the dataset and exploring its characteristics:

import pandas as pd

import numpy as np

# Importing the dataset

sonar\_data = pd.read\_csv("/content/Copy of sonar data.csv", header=None)

# Display the first 5 rows of the dataset

sonar\_data.head()

# Check the number of rows and columns in the dataset

sonar\_data.shape

# Display statistical information about the dataset

sonar\_data.describe()

# Count of different classes (R and M)

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

# Calculate the mean for each class

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

**2.2 Splitting the Dataset**

Split the dataset into features (X) and labels (y), and further split it into training and testing sets:

from sklearn.model\_selection import train\_test\_split

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

y = sonar\_data[60]

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, stratify=y, random\_state=0)

# Display the shapes of the datasets

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

**3. Model Training and Evaluation**

**3.1 Logistic Regression Model**

from sklearn.linear\_model import LogisticRegression

from sklearn.metrics import accuracy\_score

# Initialize the Logistic Regression model

model = LogisticRegression()

# Train the model on the training set

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

# Predictions on the training set

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)

# Predictions on the testing set

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

testing\_data\_accuracy = accuracy\_score(X\_test\_prediction, y\_test)

print("Accuracy on testing data:", testing\_data\_accuracy)

**4. Prediction for New Data**

# New input data for prediction

input\_data = (0.0307, 0.0523, ..., 0.0055)

# Convert input data to a numpy array

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

# Reshape the array to match the model's expected input shape

input\_data\_reshaped = input\_data\_as\_numpy\_array.reshape(1, -1)

# Make a prediction using the trained model

prediction = model.predict(input\_data\_reshaped)

print(prediction)

# Interpret the prediction

if prediction[0] == "R":

print("The object is a rock.")

else:

print("The object is a mine.")

**5. Conclusion**

In this report, we implemented a logistic regression model to predict whether a sonar signal corresponds to a rock or a mine. The model demonstrated accuracy on both training and testing datasets. The provided code also includes an example of using the model for predicting the class of new data. Further model evaluation metrics and exploration of alternative models could be considered for a more comprehensive analysis.