A PROJECT REPORT

Comparative Analysis of Machine Learning Methods for Sonar-Based Mine vs Rock Identification

SUBMITTED BY

NAZIREEN SANIA

SAP ID: 75252100103

Under the guidance of

Dr. Shweta Dixit Kadam

In partial fulfilment for the award of the degree of

Bachelor of Science

In

APPLIED STATISTICS AND ANALYTICS

APRIL 2024

SVKM'S NMIMS Deemed to be UNIVERSITY | SCHOOL OF MATHEMATICS, APPLIED STATISTICS & ANALYTICS

DECLARATION BY THE STUDENT

This is to declare that the project report entitled "Comparative Analysis of Machine Learning Methods for Sonar-Based Mine vs Rock Identification" submitted for the award of the degree of Bachelor of Science is a bonafide record of project work carried out by me. The contents of the report have not been submitted to any other Institute or University for the award of any other degree or diploma.

Student Name: Nazireen Sania

SAP ID: 75252100103

Department: School of Mathematics, Applied Statistics and Analytics

Student's Signature:

Date: April 2024

SVKM'S NMIMS Deemed to be UNIVERSITY

SCHOOL OF MATHEMATICS, APPLIED STATISTICS & ANALYTICS

CERTIFICATE FOR CHECK AGAINST PLAGIARISM

This is to certify that the project report titled "Comparative Analysis of Machine Learning Methods for Sonar-Based Mine vs Rock Identification" submitted by Nazireen Sania for the award of Bachelor of Science in Applied Statistics and Analytics is a bonafide record of the project work done by me. The contents of the project have been verified for originality through the plagiarism check software "TURNITIN" and no unacceptable similarity was found through the software check as per the norms.

Signature:

SVKM'S NMIMS Deemed to be UNIVERSITY

SCHOOL OF MATHEMATICS, APPLIED STATISTICS & ANALYTICS

CERTIFICATE FOR EXAMINATION

SAP ID: 75252100103

Name: NAZIREEN SANIA

Title of Project: Comparative Analysis of Machine Learning Methods for Sonar-Based Mine vs Rock Identification

We the below signed, after checking the project mentioned above of the student, hereby state our approval of the project submitted in partial fulfilment of the requirements of the degree of Bachelor of Sciences (Applied Statistics and Analytics) in Narsee Monjee Institute of Management Studies, Hyderabad. We are satisfied with the volume, quality, correctness, and originality of the work.

……..	……..
Faculty Mentor	Program Chair
Dr. Shweta Dixit Kadam	Dr. Ashish Kumar Biswas

……..

External Examiner

ACKNOWLEDGEMENT

I would like to sincerely express my gratitude to my program chair Dr. Ashish Kumar Biswas for letting me be a part of the project.

I would also like to thank my faculty guide Dr. Shweta Dixit Kadam for guiding me through the project. I am deeply thankful to our director Dr. Siddhartha Gosh for allowing me to present my project to my mentors which will help me to shape my academic and personal growth.

I am also deeply thankful to SVKM's Narsee Monjee Institute of Management Studies, Hyderabad, for providing me with the resources and facilities necessary for the successful completion of this project. The library, laboratories, and IT services were essential in facilitating my research and analysis.

Furthermore, I would like to acknowledge my professors, peers and family for their camaraderie, motivation, and inspiration. Their diverse perspectives, constructive feedback, and engaging discussions were instrumental in shaping my ideas and refining my arguments. This project was a great opportunity that allowed me to develop myself academically, professionally, and socially, allowing me to justifiably put my theoretical knowledge into practical instances.

ABSTRACT

Sonar technology has emerged as a crucial tool for underwater object detection and classification, with applications ranging from marine navigation to military operations. In this study, we present a comprehensive analysis of machine learning methods for identifying mines and rocks using sonar signals captured at various angles. The project uses machine learning (AIML) classification algorithms are used in this research to suggest a sonar-based system for the identification of rocks or mines. The dataset utilized in this study comprises patterns observed from bouncing sonar signals off metal cylinders (representing mines) and rocks, collected under different conditions and angles. The data set includes signals acquired at a wide range of aspect angles, ranging from 180 degrees for the rock to 90 degrees for the cylinder. The device uses sonar technology to gather acoustic signals from the underwater environment and then uses signal processing techniques to extract sound characteristics.

The primary objective of this research is to evaluate and compare the performance of four prominent machine learning algorithms: Logistic Regression, k-Nearest Neighbours (KNN), Decision Tree, and Support Vector Machine (SVM) in classifying mines and rocks based on sonar signals. Unlike traditional accuracy-based evaluations, this study places a particular emphasis on precision, recall, and F1 score metrics to provide a more nuanced understanding of the models' performance. The experimental methodology involves dividing the dataset into training and testing sets, followed by training each machine learning model on the training data and evaluating their performance on the testing data. Precision, recall, and F1 scores are computed for each model to assess its ability to correctly identify positive instances (mines) while minimizing false positives and false negatives. By analyzing these metrics, we aim to identify the method that is most accurate in distinguishing between mines and rocks based on the sonar signals.

The results of the analysis reveal nuanced differences in the performance of the machine learning models. While accuracy remains an essential metric, the emphasis on precision, recall, and F1 score offers deeper insights into the models' capabilities in handling imbalanced datasets and avoiding misclassification errors. Furthermore, the study provides recommendations on the selection of the most suitable machine learning algorithm for sonar-based mine vs rock identification applications based on the specific requirements of the task. Additionally, it will contribute to the selection of the most suitable method for real-world applications involving mine detection and underwater object identification. Overall, this research contributes to advancing the understanding of machine learning methods in underwater object classification and provides valuable insights for the development of robust and reliable sonar-based detection systems in various domains, including maritime security, ocean mining, and underwater exploration.

TABLE OF CONTENTS

LIST OF FIGURES

1. INTRODUCTION

1.1 Overview:

Sonar technology has revolutionized the field of underwater exploration and surveillance by providing an efficient means of detecting and classifying submerged objects. In maritime operations, the ability to differentiate between mines and natural rock formations is of utmost importance for ensuring navigational safety, protecting marine assets, Military, and defence. and safeguarding against potential threats. Traditional methods of manual inspection and sonar interpretation are often time-consuming and prone to human error, due to an uncountable number of values and patterns, highlighting the need for classification systems powered by machine learning algorithms.

In this study, we delve into the realm of sonar-based mine vs rock identification, focusing on the utilization of machine learning techniques to analyze sonar signals captured from different angles. The dataset under investigation comprises a diverse collection of patterns derived from bouncing sonar signals off metal cylinders (mimicking mines) and rock, acquired under similar conditions. Additionally, the project involves the development of a system that uses sonar sensors to collect data about the underwater environment. This data is then processed using AI and ML algorithms to identify and classify underwater objects as either rocks or mines. The system is trained on a large dataset of sonar values to improve its accuracy in detecting and classifying objects

1.2 Problem Statement:

To develop an artificial intelligence and machine learning-based predictive model that can compare the performance of different machine learning models based on various metrics for the detection of underwater mines and non-mines using numerical datasets obtained from sonar signals. Furthermore, to compare the accuracy of each of these models using statistical tools.

1.3 Project Gap

There are several existing systems and platforms that provide SONAR Mine or Rock detection but all of those are done using traditional methods. The predicted values are not of a high accuracy. The data rows are imbalanced, that is inconsistent in length.

However, these existing systems have limitations, including a lack of comprehensive data analysis and prediction models that accurately detect the kind of object with provided SONAR data. Unlike many existing systems that focus on a single machine learning algorithm or a limited set of techniques, this project conducts a comprehensive comparison of multiple algorithms, including Logistic Regression, K-Nearest Neighbours (KNN), Decision Tree, and Support Vector Machine (SVM). By exploring a diverse range of methods, the project seeks to identify the most effective approach for sonar-based mine vs rock identification. Therefore, this project aims to develop a comparison between comprehensive and accurate predictive models using machine learning techniques to provide valuable insights into the potential of machine learning algorithms for the detection of underwater mines, which has significant implications for maritime security and defence.

1.4 Objectives

The main objectives of this project are to:

1. Visualise the data to see meaningful correlations.
2. Develop a predictive model by using different Machine Learning algorithms.
3. Using this predictive model and data to find the accuracy which will help in the detection of underwater mines.
4. To provide valuable insights into the potential of machine learning algorithms for the detection of underwater mines, which has significant implications for maritime security and defence.

1.5 About the Data

The dataset was taken from an information science and machine learning stage ‘Kaggle’. The dataset contains 111 patterns obtained by bouncing sonar signals off a metal barrel at different points and under different conditions and 97 designs obtained from rocks under comparable conditions.

The information set contains signals obtained from an assortment of diverse angle points, traversing 90 degrees for the barrel and 180 degrees for the shake. Each pattern is a set of 60 numbers in the range of 0.0 to 1.0. Each number represents the energy inside a specific frequency band and coordinates over a certain period of time. The integration gap for higher frequencies happens later in time since these frequencies are transmitted later amid the chirp. The label “R” is associated with Rock and “M” is associated with Mine. The patterns in each label are in increasing order of angle.

2. DATA PREPARATION and EXPLORATION

2.1 Data Cleaning

The purpose of data cleaning is to prepare the dataset for analysis and modelling by identifying and correcting errors, inconsistencies, and inaccuracies. The data cleaning process involves several steps, including but not limited to:

1. Handling missing or null values: Null or missing values can be due to various reasons such as data entry errors, system failure, or non-response. Null values can adversely affect the analysis and may result in incorrect conclusions. Therefore, it is important to remove them from the dataset or impute them with reasonable values. We use the isnull() function under the pandas library in Python. On applying, we infer that there are no null or missing values in the dataset.

2. Removing duplicate values: Duplicate values in the dataset can distort the results and lead to incorrect conclusions. Therefore, it is important to remove duplicate values before analysis. We use the duplicated() function under the pandas library in Python. On applying, we infer that there are no duplicated values in the dataset.

Since there are no duplicate or null values in the dataset, it is considered to be clean and hence can be proceeded for data visualisation.

2.2 Data Augmentation

The number of data rows associated with each classifier, that is with R (Rock) are 97 while the data rows associated with M (Mine) are 111.

Although there is only a difference of only 14 samples, in comparison to the total number of data samples available, this difference is significant and needs to be balanced. In order to balance the dataset, there are two options:

1. Upsampling: Resample the values of the label with a lower count to make their count equal to the class label with the higher count (here, 111). Here, we will be upsampling. First, we divide the whole dataset into 2, one for each label.

2. Radome sampling - Pick ‘n’ samples from the class label with the least count (here, 97) to form new rows to match the number of rows in the class label with a higher count (111). The replace() function

of the Pandas library in Python is used to resample and obtain samples. After sampling The append() function of the data frame is used to combine the rows in both the datasets.

2.3 Data Visulaization

The comparison of the graph before and after resampling the number of data rows associated with each classifier:

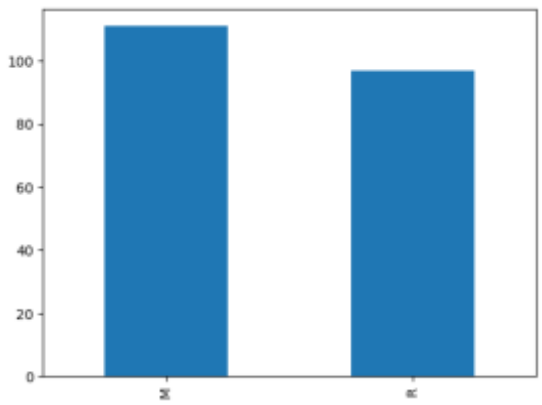


Figure 1(i)Number of R vs Number of M

Graph(a) Before

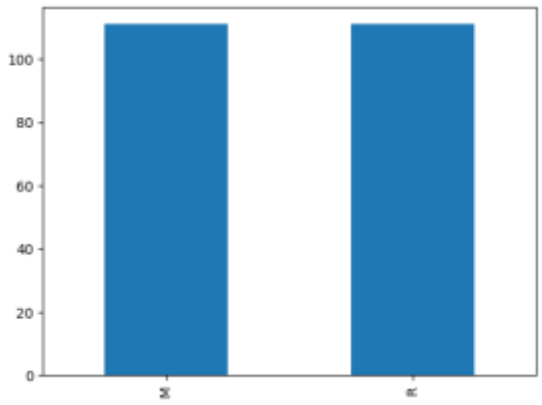


Figure 1(ii) Number of R vs Number of M

Graph(b) After

```
plt.figure(figsize=(8,5))
plt.plot(df[df[60] == 'R'].values[0][:-1], label='Rock', color='black')
plt.plot(df[df[60] == 'M'].values[0][:-1], label='Mine', color='gray', linestyle='--')
plt.legend()
plt.title('Example of both classes')
plt.xlabel('Frequency bin')
plt.ylabel('Power spectral density (normalized)')
plt.tight_layout()
plt.show()
```

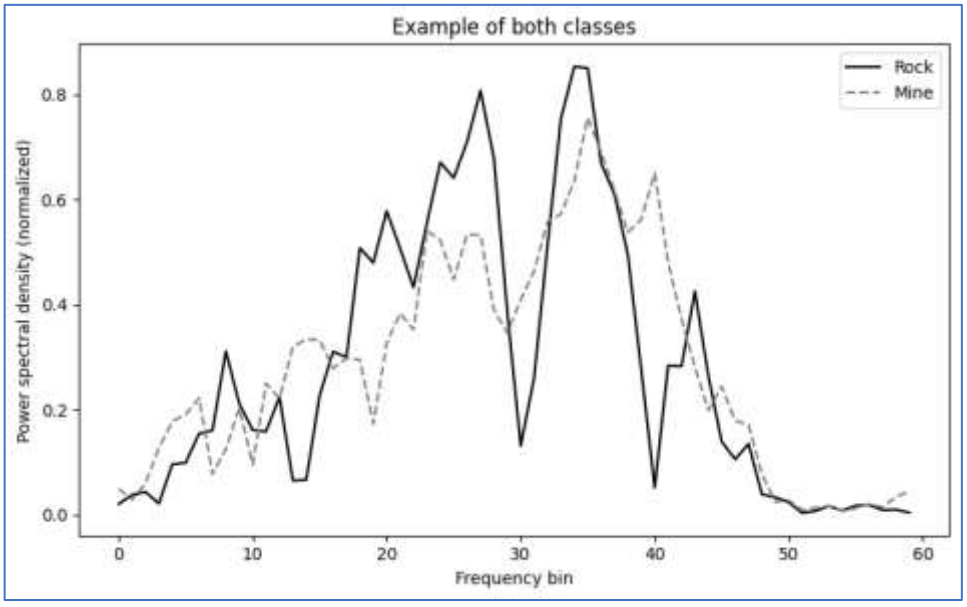


Figure 2 Example spectral envelope

The Spectral Envelope shows the overall content and behaviour of the labels ‘R’ and ‘M’

We then construct a histogram for all features, i.e. columns.

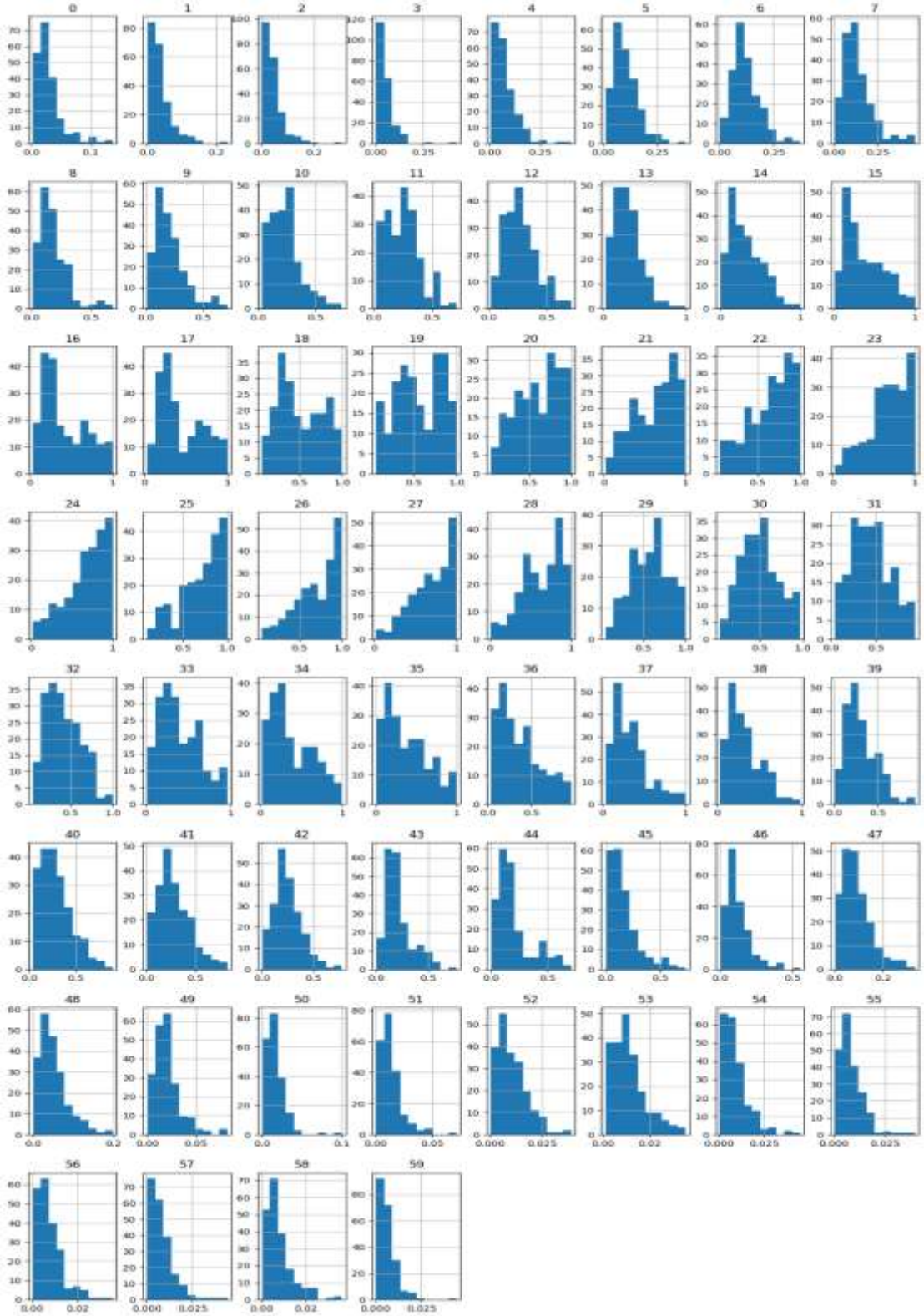


Figure 3 Histograms of all features

3. METHODOLOGY

Figure 4 Methodology

3.1 Splitting the Data

After data preprocessing and visualisation the dependent and independent data values must be split so that they can be fit in a machine learning model. In this case, we split the first 60 features or attributes into the independent variable ‘X’ and the classifier attribute (labels) in the dependent variable ‘Y’.

3.2 Training and Testing the Data

We train the data using the train_test_split module from the sklearn library. It splits the data into training and testing data variables which will later be used in fitting into models.

3.2.1 Training Set

Training Set: The training set is a subset of the original dataset used to train the machine learning model. It contains labelled examples (input features and corresponding target labels) that the model uses to learn patterns and relationships between the input features and the target variable. The training set typically comprises a majority of around 70-80% of the dataset. We set 80% of our data as the training set. The smaller the dataset, the smaller should be the training set because the algorithm does not require big training sets to learn the patterns of a small dataset

3.2.2 Testing Set

The testing set, also known as the validation set or holdout set, is a separate subset of the original dataset that is not used during the training phase. It serves as an independent dataset for evaluating the performance of the trained model. The testing set contains examples with known labels that the model has not seen during training, allowing for an unbiased assessment of the model's generalization ability.We set 10% of our dataset as testing set.

3.3 Using Appropriate Machine Learning Model

To get optimal precision and accuracy, we train the training data to different machine learning models. These models will train the data and produce an accuracy score using the testing data. The models to be used are:

1. Logistic Regression
2. K Nearest Neighbour
3. Decision Tree

4. Support Vector Machine

By comparing the accuracy score, confusion matrix and classification report we can deduce as to which model generates optimal accuracy.

3.3.1 Logistic Regression

Logistic Regression is a Machine Learning method that is used to solve classification issues. The classification algorithm Logistic Regression is used to predict the likelihood of a categorical dependent variable. The dependent variable in logistic regression is a binary variable with data coded as 1 (yes, True, normal, success, etc.) or 0 (no, False, abnormal, failure, etc.). The logistic function in linear regression is a type of sigmoid, a class of functions with the same specific properties.

$$f(x) = \frac{1}{1 + e^{-x}}$$

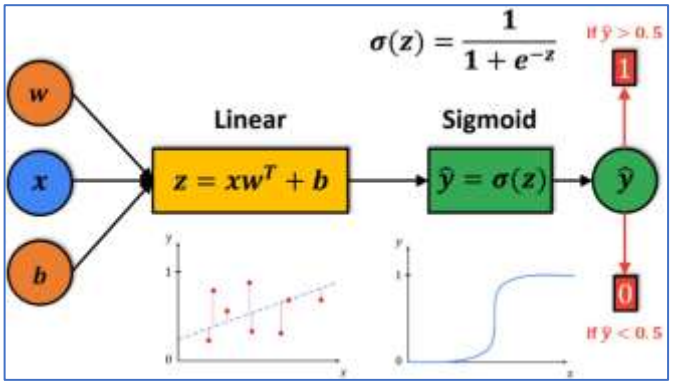


Figure 5 Logistic Regression

3.3.2 K-Nearest Neighbour

K-nearest neighbours (KNN) is a type of supervised learning algorithm used for both regression and classification. KNN tries to predict the correct class for the test data by calculating the distance between the test data and all the training points. Then select the K number of points which is closest to the test data.The KNN algorithm calculates the probability of the test data belonging to the classes of 'K' training data and which class holds the highest probability will be selected. In the case of regression, the value is the mean of the 'K' selected training points.

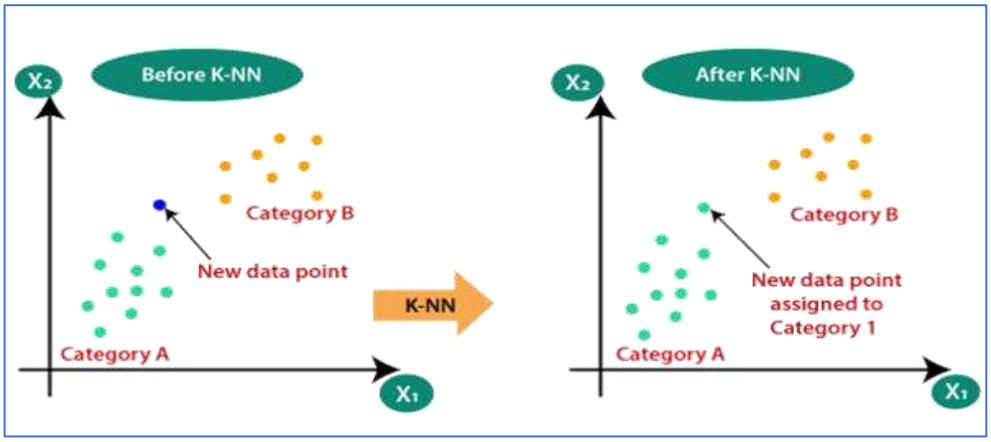


Figure 6 K-Nearest Neighbours

3.3.3 Decision Tree

A decision tree algorithm is a type of machine learning algorithm used for classification and regression tasks. It works by creating a tree-like model of decisions and their possible consequences.

In a decision tree, each node represents a decision based on a feature or attribute of the data, and each branch represents a possible outcome of that decision. The leaf nodes of the tree represent the final classification or prediction.

Decision trees are relatively easy to interpret and visualize, and they can handle both categorical and numerical data. However, they can be sensitive to small variations in the data and prone to overfitting if the tree is too complex or the training data is noisy. Ensemble methods like random forests and gradient boosting can help to mitigate these issues.

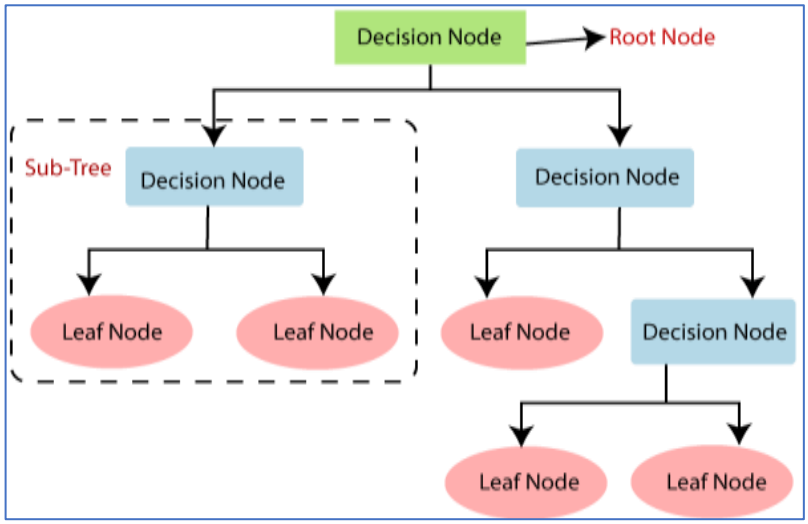


Figure 7 Decision Tree

3.3.4 Support Vector Machine

Support Vector Machines (SVM) is a popular machine learning algorithm used for classification and regression analysis. It is based on the concept of finding a hyperplane that maximally separates the data points into different classes.

In an SVM model, the data is represented as points in a high-dimensional space. The algorithm then tries to find the hyperplane that best separates the data into different classes by maximising the margin between the hyperplane and the nearest data points. The data points that lie closest to the hyperplane are called support vectors.

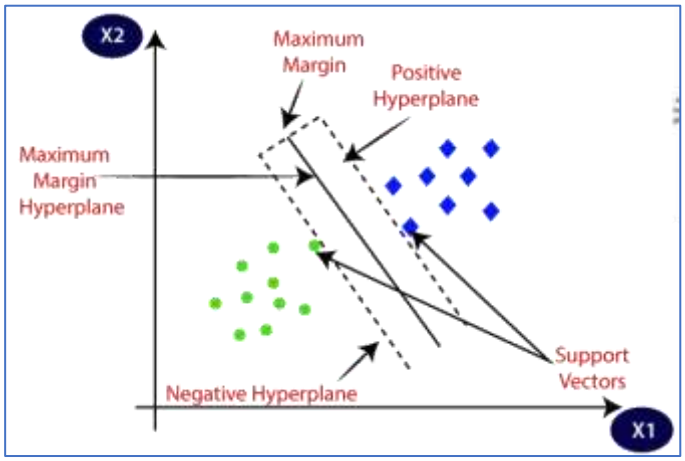


Figure 8 Support Vector Machine

3.4 Important Metrics

When using classification models in machine learning, there are three common metrics that we use to assess the quality of the model:

Confusion Matrix:

A matrix showing true positives, true negatives, false positives, false negatives. It helps us in analysing the model's performance by showing how many cases the model predicted correctly and which ones the model mispredicted.

Precision (also called Positive Predictive Value): Precision measures the proportion of true positive predictions among all positive predictions made by the model. It indicates the accuracy of the positive predictions. A high precision value indicates that the model is making a few false positive predictions. Recall is calculated as:

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

Recall (also called Sensitivity or True Positive Rate): Recall measures the proportion of true positive predictions that are correctly identified by the model out of all actual positive instances in the dataset. It indicates the ability of the model to correctly detect positive instances. A high recall value indicates that the model is effectively capturing most of the positive instances in the dataset. It is calculated as:

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

F1-score: The F1-score is the harmonic mean of precision and recall, providing a single metric that balances both precision and recall. The F1-score ranges from 0 to 1, where a higher value indicates better performance in terms of both precision and recall. It is calculated as:

$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Support: Support refers to the number of actual occurrences of each class in the dataset. It represents the number of samples in the dataset that belong to each class.

4. MODEL ARCHITECTURE

- **Data collection:** We have obtained SONAR data from reliable sources, such as the KAGGLE. This dataset was used by Gorman, R. P., and Sejnowski, T. J. (1988). "Analysis of Hidden Units in a Layered Network Trained to Classify Sonar Targets" in Neural Networks, Vol. 1, pp. 75–89.

- **Data preprocessing:** The data comes with no null values or duplicate values so there was no need to clean it. However, it had an imbalance in the number of data values of the classifier attribute which was later fixed by sampling.

- **Train-test split:** Split the data into training and testing sets using the train_test_split function from the sklearn library. We have chosen 90% of the data as train data and the remaining 10% for the test data.

- **Model selection and training:** We have chosen different machine learning models and trained them. Then we obtained the accuracy scores and confusion matrices of those models to compare.

- **Model evaluation:** We have evaluated the model's performance on the testing set by calculating various metrics.

- **Prediction on new data:** We have tested the model on new data by making predictions using the predict method. We have created a new dataset of predicted variables and used the model to predict new cases.

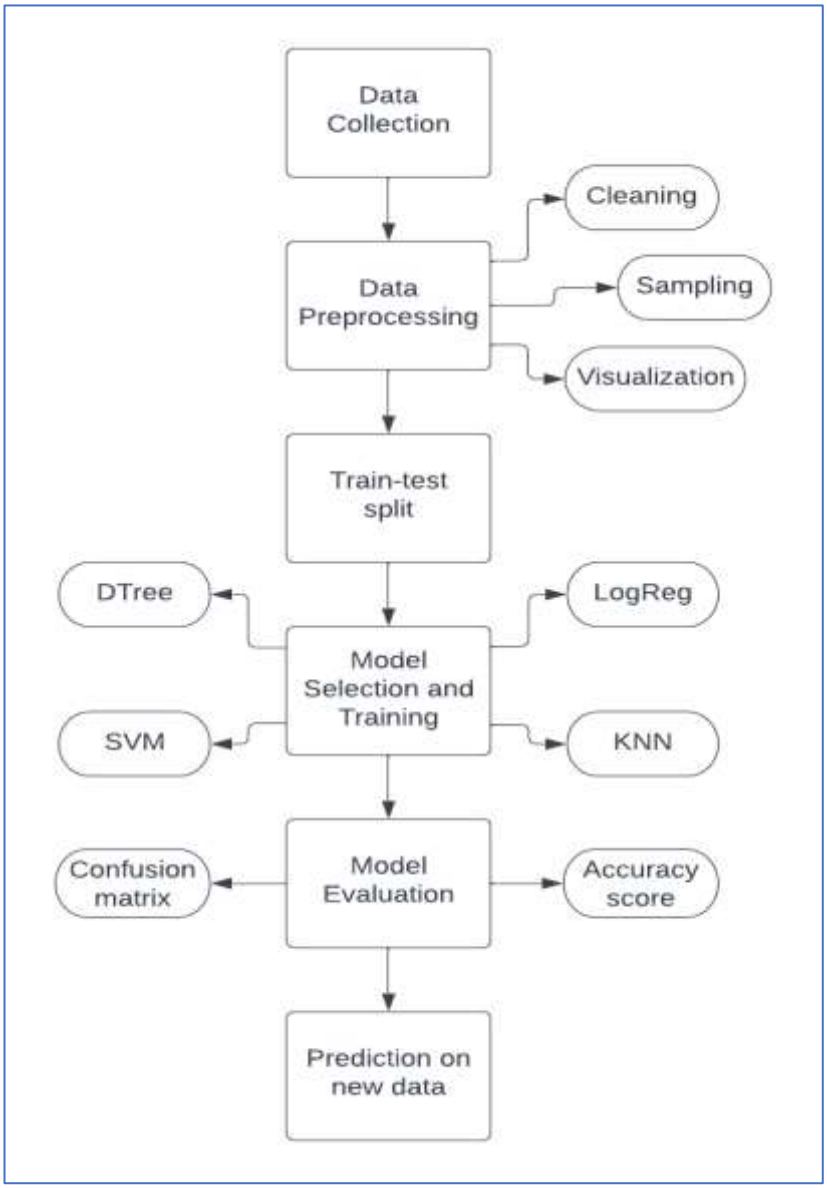


Figure 9 Model Architecture

5. SOFTWARE DESCRIPTION

5.1 Jupyter

This project is developed using Jupyter Notebook, which is a popular web-based interactive development environment for creating and sharing data science projects. It supports various programming languages, including Python, R, Julia, and Scala, but is primarily used with Python. The key features of Jupyter include: Interactive Computing: Jupyter provides an interactive computing environment where you can write and execute code in a step-by-step manner. It supports the execution of code cells individually, allowing you to test and debug code snippets interactively.

- **Notebook Interface:** Jupyter notebooks are documents that contain a mixture of code, text, images, and mathematical equations. Notebooks are organized into cells, which can contain either code or text. This flexible format allows you to create documents that combine code, visualizations, and explanations in a single place.
- **Kernel Support:** Jupyter notebooks are associated with kernels, which are separate processes that execute the code contained in the notebook. Each kernel is responsible for executing code in a specific programming language, such as Python or R. This allows you to work with multiple programming languages within the same notebook.
- **Rich Output:** Jupyter notebooks support rich output formats, including interactive visualizations, tables, and multimedia content. This allows you to create dynamic and engaging presentations, reports, and tutorials directly within the notebook.
- **Collaboration and Sharing:** Jupyter Notebooks can be easily shared with others via email, GitHub, or hosted services like JupyterHub and Google Colab. This makes it easy to collaborate on projects, share findings, and reproduce analyses.

Overall, Jupyter provides a powerful and flexible environment for interactive computing, data analysis, scientific computing, and machine learning. It is widely used by data scientists, researchers, educators, and professionals across various domains for prototyping, experimentation, and communication of results.

5.2 Python

The code is written in Python programming language and uses several Python libraries, including Pandas, NumPy, Scikit-learn, Matplotlib, and Seaborn.

Pandas

Pandas is a library that is used for data manipulation and analysis. It provides a set of data structures and functions to work with structured data, such as data frames and series.

Numpy

NumPy is a library that is used for numerical computing with Python. It provides a powerful array processing capability and mathematical functions to work with large, multi-dimensional arrays and matrices. In this project, NumPy is used to perform numerical operations, such as calculating mean and standard deviation of the data.

sklearn

sklearn, short for scikit-learn, is a popular open-source Machine Learning library for Python. It provides a wide range of tools and algorithms for machine learning tasks, including classification, regression, clustering, dimensionality reduction, and model selection.

Matplotlib

Matplotlib is a library that is used for creating visualisations and plots. It provides a set of functions to create various types of plots, such as line, bar, scatter, and histogram. In this project, Matplotlib is used to create scatter plots and regression lines to visualize the relationship between the input variables.

Seaborn

Seaborn is a library that is used for creating more advanced visualisations and plots. It provides a high-level interface to create complex visualisations, such as heatmaps, pair plots, and violin plots. In this project, Seaborn is used to create a scatter plot with a regression line and a residual plot to evaluate the performance of the linear regression model.

6. RESULTS and ANALYSIS

6.1 Data Preparation and Exploration

After data preparation and exploration, we now know that our data has no missing values and no duplicated values. This is important to decipher since missing values lead to:

Bias in Descriptive Statistics: The presence of missing values in the dataset can lead to biases in descriptive statistics. Missing values can distort summary statistics such as mean, median, and standard deviation, leading to biased estimates of central tendency and variability.

Incomplete Insights: They can lead to incomplete insights and analysis, as certain observations or variables may be excluded from the analysis due to missing data.

Reduced Sample Size: Missing values reduce the effective sample size available for analysis, potentially reducing the statistical power of analyses and increasing uncertainty in results.

Model Performance: Missing values can affect the performance of predictive models, especially if the missingness is related to the outcome variable or other predictors. Our project uses four different machine learning algorithms all of which are affected in different ways due to missing values. Logistic Regression can incur errors during model training and KNN gives biased predictions due to missing values.

The number of patterns for Rock (R) is 97 and the number of patterns for Mine (M) is 111.

We sample the rows of the data frame containing ‘R’ using the sample() function and use append() function to join the newly formed data frame for ‘R’ with the data frame for ‘M’. The sampling is done with replacement, meaning that the same row can be selected multiple times. This is useful for balancing the sizes of two datasets. When you sample from a DataFrame using the sample() function in pandas with replacement (i.e., replace=True), the sampled rows are not added to the original DataFrame. Instead, the sampled rows are used to create a new DataFrame.

Sampling, in this context, refers to randomly selecting a subset of data from a larger dataset. In this specific case, sampling rows is done to match the length of both the dataframes. The purpose of sampling here is to balance the classes represented by 'R' and 'M' in the two If the original dataset is imbalanced, meaning one class has significantly more samples than the other, it might skew the analysis or model training process. By sampling rows from the smaller class (in this case, 'R’) to match the number of rows in the larger class ('M'), we create a more balanced dataset.

However, it's important to note that sampling with replacement (replace=True) means that some rows may be selected multiple times, while others may not be selected at all. This could introduce some level of randomness into the resulting dataset.

Whether or not this affects the output depends on the specific analysis or model training process you're performing. In some cases, having a balanced dataset can lead to more accurate models or more meaningful analysis results, particularly in tasks like classification where class imbalance can be problematic. However, in other cases or with certain algorithms, the impact of class balance may be less significant.

We visualize the data to check if both dataframes have an equal number of rows. Histograms are plotted for each feature (column) for visualization purposes.

6.2 Splitting, Training and Testing

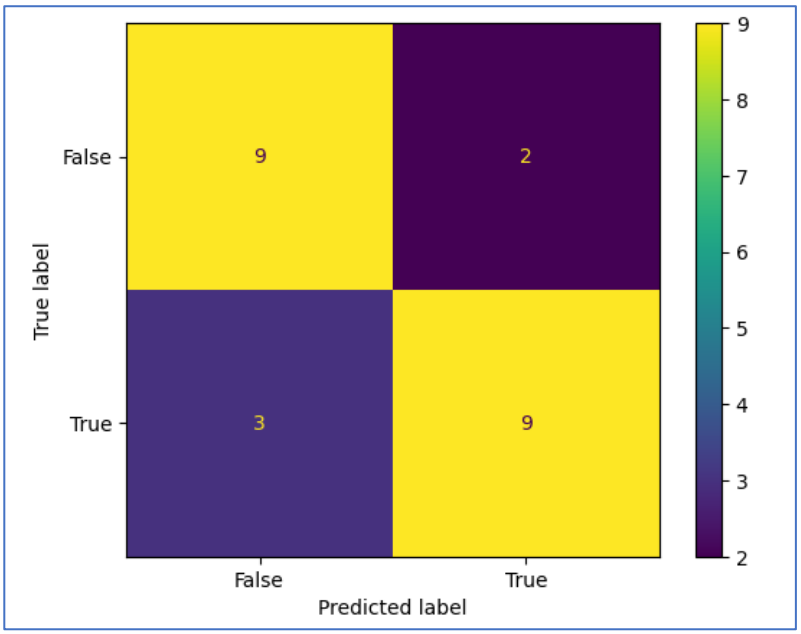
The dataframe is split into two each containing two components. The dataframe is split based on features (which we name as ‘X’) and labels (which we name as ‘Y’). The features are the patterns recorded when SONAR was bounced off of mines (metal cylinders) and rocks at different angles. The labels are Rock, labelled as ‘R’ and Mine labelled as ‘M’.

An important part of training and testing is splitting the data. Splitting data into training and testing sets is a fundamental step in machine learning model development. It involves partitioning the available dataset into two distinct subsets: one for training the model and the other for evaluating its performance.

We split our data in the size ratio of 0.2 when 90% of the data is split into training data and 20% of the data is split into testing data.

6.3 Logistic Regression

We perform logistic regression on the trained data. To implement the logistic regression model, we first import the LogisticRegression module from sklearn. Logistic regression is performed on the testing data and compared with the test labels to check the accuracy pf the model. We obtain an accuracy of 0.826 . The confusion matrix and the classification report obtained for this is as follows:



	precision	recall	f1-score	support
M	0.75	0.82	0.78	11
R	0.82	0.75	0.78	12
accuracy			0.78	23
macro avg	0.78	0.78	0.78	23
weighted avg	0.79	0.78	0.78	23

INTERPRETATION –

- Accuracy - The proportion of correctly classified instances out of all instances in the dataset is 78%.
- Precision – 75% of the patterns predicted to be of Mine, ‘M’ are actually ‘M’.

  82% of the patterns predicted to be of Rock ‘R’ are actually ‘R’.

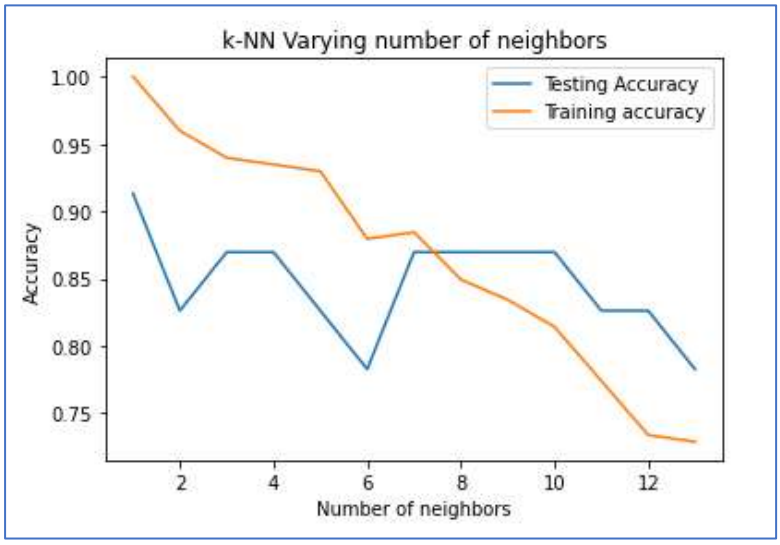
- Recall – 82% of the actual ‘M’ cases were correctly identified by the model.

  75% of the actual ‘R’ cases were correctly identified by the model.

- F1 Score – The harmonic mean of precision and recall for ‘M’ and ‘R’ is 0.78.
- Support – There are 11 samples in ‘M’ and 12 samples in ‘R’.

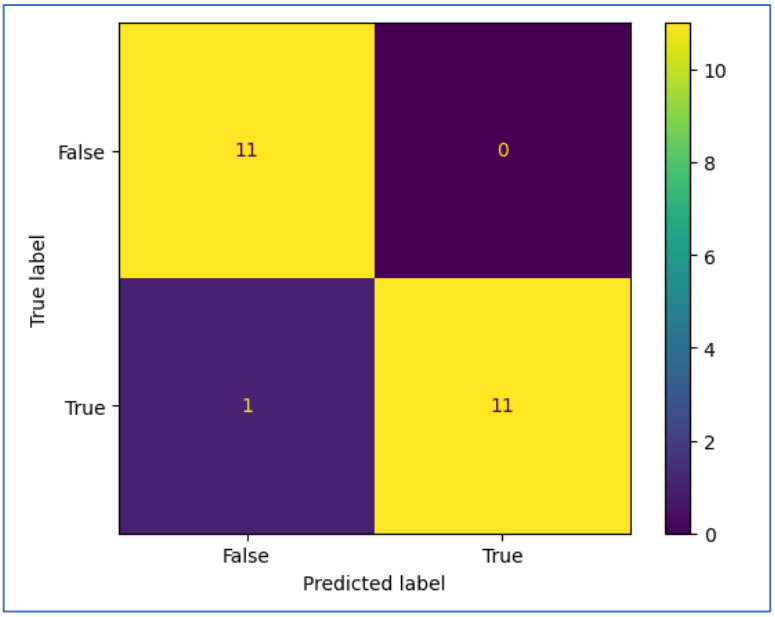
6.4 K-Nearest Neighbourss

We perform K-Nearest Neighbourss on the trained data. To implement the K-Nearest Neighbourss model, we first import the KNeighbourss module from sklearn. KNN is performed on the testing data and compared with the test labels to check the accuracy of the model.On plotting a graph against the number of neighbours against accuracy to select the most suited number of neighboursss, we see that the it can be seen that accuracy for both the training as well as the testing data decreases with the increasing number of neighbours, so k=2 would be a safe number to assume.



However, on increasing the number of neighbours to 5, the accuracy of the model significantly changes, hence we consider k=5. We obtain an accuracy of 0.956. The confusion matrix and the classification report obtained for this is as follows:

	precision	recall	f1-score	support
M	0.92	1.00	0.96	11
R	1.00	0.92	0.96	12
accuracy			0.96	23
macro avg	0.96	0.96	0.96	23
weighted avg	0.96	0.96	0.96	23



INTERPRETATION –

- Accuracy - The proportion of correctly classified instances out of all instances in the dataset is 96%.
- Precision – 92% of the patterns predicted to be of Mine, ‘M’ are actually ‘M’.

  100% of the patterns predicted to be of Rock ‘R’ are actually ‘R’.

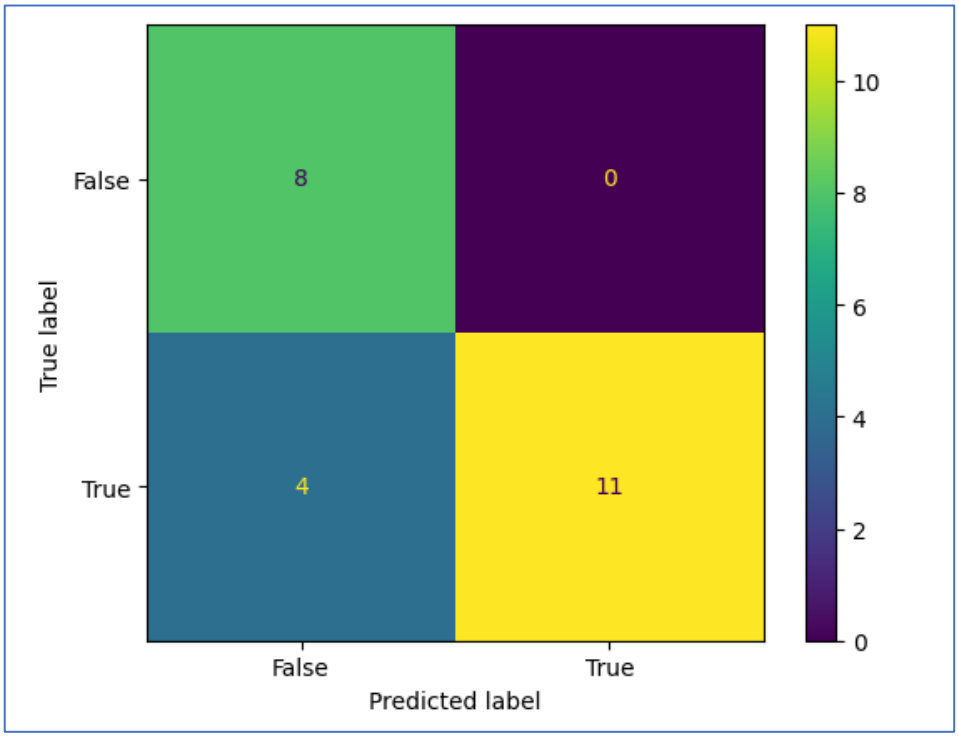
- Recall – 100% of the actual ‘M’ cases were correctly identified by the model.

  92% of the actual ‘R’ cases were correctly identified by the model.

- F1 Score – The harmonic mean of precision and recall for ‘M’ and ‘R’ is 0.96.
- Support – There are 11 samples in ‘M’ and 12 samples in ‘R’.

6.5 Decision Tree

We perform the Decision Tree method on the trained data. To implement the Decision Tree model, we first import the DecisionTreeClassifier module from sklearn. Decision Tree is performed on the testing data and compared with the test labels to check the accuracy of the model. We obtain an accuracy of 0.826. The confusion matrix and the classification report obtained for this is as follows:



	precision	recall	f1-score	support
M	0.67	1.00	0.80	8
R	1.00	0.73	0.85	15
accuracy			0.83	23
macro avg	0.83	0.87	0.82	23
weighted avg	0.88	0.83	0.83	23

INTERPRETATION –

- Accuracy - The proportion of correctly classified instances out of all instances in the dataset is 83%.
- Precision – 67% of the patterns predicted to be of Mine, ‘M’ are actually ‘M’.

  100% of the patterns predicted to be of Rock ‘R’ are actually ‘R’.

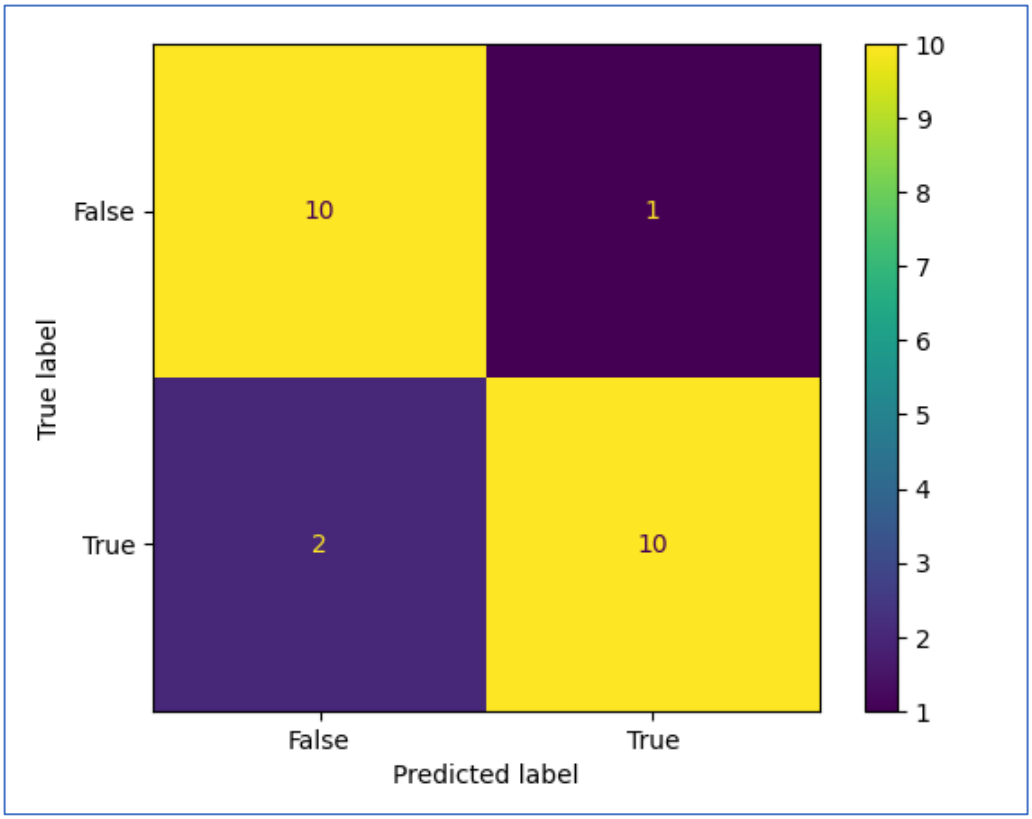
- Recall – 100% of the actual ‘M’ cases were correctly identified by the model.

  73% of the actual ‘R’ cases were correctly identified by the model.

- F1 Score – The harmonic mean of precision and recall for ‘M’ is 0.8 and for ‘R’ is 0.85.
- Support – There are 8 samples in ‘M’ and 15 samples in ‘R’.

6.6 Support Vector Machine

We perform a Support Vector Machine algorithm on the trained data. To implement the Support Vector Machine, we first import the SVC module from sklearn. Support Vector Machine is performed on the testing data and compared with the test labels to check the accuracy of the model. We obtain an accuracy of 0.8695. The confusion matrix and the classification report obtained for this is as follows:



	precision	recall	f1-score	support
M	0.83	0.91	0.87	11
R	0.91	0.83	0.87	12
accuracy			0.87	23
macro avg	0.87	0.87	0.87	23
weighted avg	0.87	0.87	0.87	23

INTERPRETATION –

- Accuracy - The proportion of correctly classified instances out of all instances in the dataset is 87%.
- Precision – 83% of the patterns predicted to be of Mine, ‘M’ are actually ‘M’.

  91% of the patterns predicted to be of Rock ‘R’ are actually ‘R’.

- Recall – 91% of the actual ‘M’ cases were correctly identified by the model.

  83% of the actual ‘R’ cases were correctly identified by the model.

- F1 Score – The harmonic mean of precision and recall for ‘M’ and ‘R’ is 0.87.
- Support – There are 11 samples in ‘M’ and 12 samples in ‘R’

6.7 Comapring Accuracy

To compare the accuracies of the different Machine learning classification methods that we used, we construct a bar graph with the accuracy of each of the methods.

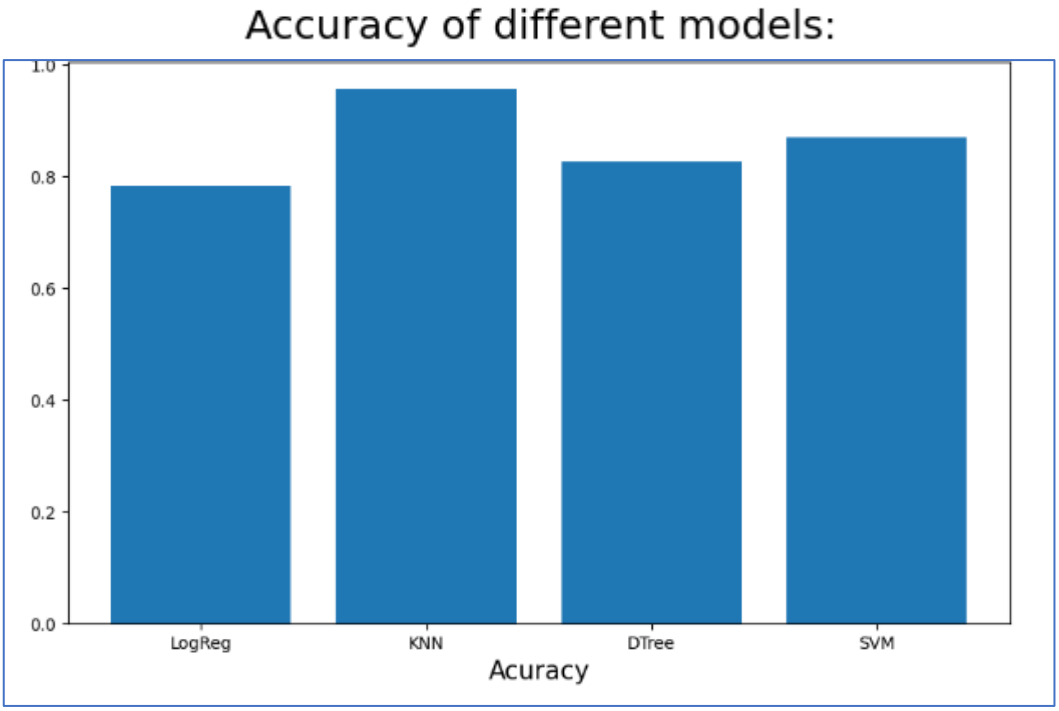


Figure 10 Cumulative Accuracy of Different Models

From the above results, it is clear that KNN generated the highest amount of accuracy while Logistic Regression generated lower accuracy compared to other models.

7. LIMITATIONS

- Out of the Machine Learning algorithms we used, Logistic Regression and K-Nearest Neighbourss are affected by missing values, while Decision Tree and Support vector Machine are more robust to missing values. In case of missing values, the accuracy of the algorithms can significantly change.

- In case of more features, the algorithms may need more parameters to consider in case of analysis.
- The conclusions of our project might not be sufficient to build complex algorithms like Artificial Neural networks, Fuzzy Logic, etc.

8. CONCLUSION and FUTURE SCOPE

Based on the metrics obtained after fitting the training data and comparing the testing data, the accuracy scores of each model have been obtained. Out of which KNN generated the highest accuracy. Hence, KNN can be used for accurate prediction.

Even Though the accuracy generated by the KNN model is reasonable enough, it can further be developed by reducing the number of features with feature reduction techniques like Principal Component Analysis, Backward Elimination, Forward Selection, Score comparison, Missing Value Ratio etc.

Furthermore, this project can be expanded by using more powerful machine learning models such as Random Forest, Gradient Boosting, or Neural Networks to improve prediction accuracy.

Overall, This model will have a significant impact on maritime security and defence. This model can be further used to detect any kind of mineral if their frequency data is provided.

8. REFERENCES

1 . *Find open datasets and machine learning projects | Kaggle*. (n.d.).

https://www.kaggle.com/datasets.

2 *pandas documentation — pandas 2.2.2 documentation*. (n.d.-c). https://pandas.pydata.org/docs/

3 *NumPy documentation — NumPy v1.26 Manual*. (n.d.). https://numpy.org/doc/stable/

3. *seaborn: statistical data visualization — seaborn 0.13.2 documentation*. (n.d.).

https://seaborn.pydata.org/

4. *scikit-learn: machine learning in Python*. (n.d.). https://scikit-learn.org/stable/documentation.html

5. Matplotlib documentation. (2021).
Retrieved from https://matplotlib.org/stable/contents.html

6. Zach. (2022, May 9). *How to Interpret the Classification Report in sklearn (With Example)*.

Statology. https://www.statology.org/sklearn-classification-report/