

PROFESSIONAL TRAINING REPORT
at
Sathyabama Institute of Science and Technology
(Deemed to be University)

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Technology Degree in
Information Technology

By

Name: JENILA J
(REG.NO:37120035)



DEPARTMENT OF INFORMATION TECHNOLOGY
SCHOOL OF COMPUTING
SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY
JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI – 600119. TAMIL NADU

AUGUST 2020



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with "A" Grade by NAAC

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai - 600 119.

Phone: 044 - 2450 3150 / 51 / 52 / 54 / 55 Fax: 044 - 2450 2344

www.sathyabama.ac.in



SCHOOL OF COMPUTING

BONAFIDE CERTIFICATE

This is to certify that this Professional Training Report is the bonafide work of JENILA J (Reg. No. 37120035) who underwent the professional training in "PREDICTION OF UNDERWATER SURFACE OF ROCKS AND MINES THROUGH SONAR" under our supervision from July 2020 to August 2020.

Internal Guide

DR.S. GOWRI, M.E., Ph.D.,

Head of the Department

Dr. R.SUBHASHINI, M.E.,Ph.D.,

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I, **JENILA J (Reg no:37120035)** hereby declare that the Professional Training Report on “**PREDICTION OF UNDERWATER SURFACE OF ROCKS AND MINES THROUGH SONAR**”done by me under the guidance of guide **Dr.S.GOWRI** , Department of Information Technology at Sathyabama Institute of Science and technology is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Information Technology.

DATE:

PLACE:

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

The satisfaction and elation that accompany the successful completion of any task would be incomplete without the mention of the people who have made it possible. It is our great privilege to express our gratitude and respect to all those who have guided me during the course of my Professional Training.

First and foremost, we would express our sincere gratitude to our beloved Founder Chancellor **Col. Dr. JEPPIAAR, M.A., B.L., Ph.D.**, and Chancellor **Thirumathi. REMIBAI JEPPIAAR**. I extend my sincere thanks to our Chancellor **Dr. Mariazeena Johnson, B.E., M.B.A., M.Phil., Ph.D.**, and the President **Dr. Marie Johnson, B.E., M.B.A., M.Phil., Ph.D.**, and for providing me the necessary facilities for the completion of the professional training. I also acknowledge our Vice Chancellor **Dr. S.Sundar Manoharan Ph.D.**, and the Pro Vice Chancellor **Dr. T.Sasipraba, M.E., Ph.D.**, for their constant support and endorsement.

I would like to express our gratitude to our Registrar **Dr. S. S. Rau, Ph.D.** and Controller of Examinations **Dr. Igni Sabasti Prabu M.E., Ph.D.**, for their valuable support offered to complete my professional training successfully.

I like to express our gratitude to **Dr.T.Sasikala, M.E., Ph.D.**, Dean of School of Computing and **R. Subhashini, M.E., Ph.D.**, Head of the Department of Information Technology, Sathyabama University for having been a constant source of support and encouragement for the completion of the professional training. I would also like to express our sincere thanks to our internal guide **Dr.S. Gowri, M.E., Ph.D.**, who guided me in the preparation of the report.

Also, I thank the **Almighty** and my **Parents** for supporting me in the completion of the professional Training.





ABSTRACT

PREDICTION OF UNDERWATER SURFACE OF ROCKS AND MINES THROUGH SONAR

SONAR, Sound Navigation and ranging used to detect object or obstacles underwater and it is also used to determine the depth of water. In Modern days the discovery of rocks and mines have been very difficult in the past development of SONAR technique, which depends on the parameters to detect the obstacle on the underwater surface is a rock or mine.

SONAR dataset involves the prediction of whether or not an object is a mine or a rock given the strength of sonar returns different angle. Through Machine learning the predictive analysis out of the target whether rock or mine or the object which will not cause any damage. attempts are done highly on SONAR dataset and achieve accuracy of approximately 88%. There are 208 observation with 60 variables and 1 output variable.

Machine Learning is performing major role in improving the quality of detection of underwater resource will tend to be far better in future.

TABLE OF CONTENTS

CH NO	TITLE NAME	PAGE NO
	ABSTRACT	7
	LIST OF FIGURES	9
	LIST OF ABBREVIATIONS	10
1	INTRODUCTION	11
	1.1 OBJECTIVE	12
	1.2 PROPOSED APPROACH	12
2	LITERATURE SURVEY	13
	2.1 PROCESSING METHODS	14
	2.1.1 DATA SET	14
	2.1.2 EXPERIMENTAL SETTING	14
	2.1.3 MACHINE LEARNING CLASSIFIERS	14
	2.2 PROPOSED FRAMEWORK	15
	2.2.1 PROPOSED FRAMEWORK METHODS	15
3	SYSTEM REQUIREMENTS	17
	3.1 SOFTWARE AND HARDWARE REQUIREMENTS	17
	3.2 MATPLOTLIB	17
	3.3 PANDAS	19
	3.4 SKLEARN	19
4	IMPLEMENTATION	20
	4.1 EXPERIMENTAL RESULTS AND DISCUSSIONS	20
	4.2 IMPLEMENTATION CODE	23
5	SCREENSHOTS	25
6	CONCLUSION	28
	REFERENCE	30
	APPENDIX	31
	A. SOURCE CODE	31

LIST OF FIGURES

FIG.NO	FIGURE NAME	PAGE NO
2.1	PREDICTION FRAMEWORK	16
3.1	MATPLOTLIB PLOT	18
3.2	SKLEARN	19
4.1	DATA CORRELATION REPRESENTATION	20
4.2	DIFFERENT DIMENSIONS OF FREQUENCY	20
4.3	DATA DISTRIBUTION PROCESS IN DENSITY PLT REPRESENTATION	21
4.4	ALGORITHMS COMPARISON	21
4.5	SCALED ALGORITHM ACCURACY VALUES	22
4.6	SCALED ALGORITHM COMPARISON	22
4.7	BOSTING ALGORITHMS ACCURACY VALUES	23
5.1	OUTPUT 1	25
5.2	OUTPUT 2	25
5.3	DATA CORRELATION REPRESENTATION	26
5.4	ALGORITHM COMPARISON	26
5.5	ALGORITHM COMPARISON	27

LIST OF ABBREVIATIONS

LR	Logistic Regression
LDA	Linear Discriminant Analysis
KNN	K-nearest neighbors
CART	Classification and Regression Tree
NB	Naïve Bayes Classifier
SVM	Support Sector Machine
SLR	Scaled Logistic Regression
SLDA	Scaled Linear Discriminant Analysis
SKNN	Scaled K-nearest neighbors
SCART	Scaled Classification and Regression
SNB	Scaled Naïve Bayes Classifier
SSVM	Scaled Support Sector Machine

1.INTRODUCTION

Sonar is a modern technology which helps us to track submarines, fish, ship wrecks, map the seabed and for other navigational purposes. The for main factors that affect the performance of a sonar system are a high-power transmitter, an efficient transducer, a sensitive receiver and an acoustic communication system. There is a lot to explore under the deep waters (seas and oceans), rocks and mines are two of those crucial natural resources, and this would have been very difficult to find these resources past development of the SONAR technique, which is an acronym for Sound Navigation And Ranging, and is used to measure the depth of the sea or the ocean or the distance in the water. In the similar way these sound in the probe, after the pre-processing of the input, different machines learning classifiers are trained to check the achievement of classification. The conduct for the finest classifier included comparison with some standard up-to-date classifiers like Random Forest, SVM, C4.5, Adabag etc. Although the present day sonar systems have only a limited number of features such as a 2-dimensional screen and a view of less than 360 degrees of the surrounding, we are most likely to see better sonar systems with a 3-dimensional screen, attached to new computer systems along with 360 degree view of the surrounding in the near future.

Sonar Navigation and Ranging A sensor that is used to detect objects through the use of high or low frequency sound waves. Although some animals (dolphins and bats) have used sound for communication and object detection for millions of years, use by humans in the water is initially recorded. The use of sound of 'echo locate' underwater in the same way as bats use sound for aerial navigation seems to have been prompted. Advantageous results are achieved, when we compare the performance of the classifiers in the framework like standard classifiers like SVM, random forest, adabag, neural networks, etc., using various evaluating metrics like accuracy, area under curve, sensitivity, specificity etc. waves can be used to make predictions for the underwater surfaces, mines and rocks. After the pre-processing of the input, different machine learning classifiers are trained to check the achievement of classification. The conduct for the finest classifier included comparison with some standard up-to-date classifiers like Random forest, etc.,

1.1 OBJECTIVE

The task is to train a network to discriminate between sonar signals bounced off a metal cylinder and those bounced off a roughly cylindrical rock. The data set was contributed to the benchmark collection by terry Sejnowski, now at the Salk Institute and the University of California at San Deigo.

1. To compare between Logistic Regression and K-nearest neighbours performance
2. To assess the models by a binary classification metric, i.e. confusion matrix.
3. To interpretate the model (only for Logistic Regression)
4. To test the model using dataset test and discuss the results

1.2 PROPOSED APPROACH

The Sonar Dataset involves the prediction of whether or not an object is a mine or a rock given the strength of sonar returns at different angles. It is a binary (2-class) classification problem. The number of observations for each class is not balanced. There are 208 observations with 60 input variables and 1 output variable. The variable names are as follows:

1. Sonar returns at different angles
2. ...
3. Class (M for mine and R for rock)

The baseline performance of predicting the most prevalent class is a classification accuracy of approximately 53%. Top results achieve a classification accuracy of approximately 88%.

This is a dataset that describes sonar chirp returns bouncing off different services. The 60 input variables are the strength of the returns at different angles. It is a binary classification problem that requires a model to differentiate rocks from metal cylinders. you may find the csv file in the repository of the project. The main concern of analysis in the field of machine learning is being to form a scheduled computational machine for the categorizing the forecast of the objects, based on the attainable information. The outcome of proposed framework helps to predict the triggered sound waves reflect back from what surface: Rock or a Mine.

2.LITERATURE SURVEY

Sonar (sound navigation ranging) is a technique that uses sound propagation (usually underwater, as in submarine navigation) to navigate, communicate with or detect objects on or under the surface of the water, such as other vessels. This data set can be used in a number of different ways to test learning speed, quality of ultimate learning, ability to generalize, or combinations of these factors. Gorman and Sejnowski report two series of experiments: an “aspect-angle independent” series, in which the whole data set is used without controlling for aspect angle, and an “aspect-angle dependent” series in which the training and testing sets were carefully controlled to ensure that each set contained cases from each aspect angle in appropriate proportions.

For the aspect-angle independent experiments the combined set of 208 cases is divided randomly into 13 disjoint sets with 16 cases in each. For each experiment, 12 of these sets are used as training data, while the 13th is reserved for testing. The experiment is repeated 13 times so that every case appears once as part of a test set. The reported performance is an average over the entire set of 13 different test sets, each run 10 times.

It was observed that this random division of the sample set led to rather uneven performance. A few of the splits gave poor results, presumably because the test set contains some samples from aspect angles that are under-represented in the corresponding training set. This motivated Gorman and Sejnowski to devise a different set of experiments in which an attempt was made to balance the training and test sets so that each would have a representative number of samples from all aspect angles. Since detailed aspect angle information was not present in the data base of samples, the 208 samples were first divided into clusters, using a 60-dimensional Euclidian metric; each of these clusters was then divided between the 104-member training set and the 104-member test set.

The actual training and testing samples used for the “aspect angle dependent” experiments are marked in the data files. The reported performance is an average over 10 runs with this single division of the data set.

2.1 PROCESSING METHODS

The process and methods used for proposing the prediction model is discussed in this section.

2.1.1 DATASET: In our proposed method dataset has been collected from UCI Repository. It has come across 61 features which define and differentiate Rocks and Mines and comprises of 209 samples.

2.1.2 EXPERIMENTAL SETTING: In building of this system we are using WEKA tool for the purpose of implementing the varied feature selection and model systems. The main motive is to measure the predicting efficiency of the classifier when it is functional and operating and then classifying new samples outside the benefit of perceiving the bona fide class of the samples. The comparators have been designed to implement a 10-fold cross validation trial. The dataset is split into 10 equally distributed subsets. The most exact machine learning classifier is chosen as a base classifier to instruct the nine-subset layer and examine it on the last subset layer. To measure the durability of crafted groundwork, the step is repeated. To appraise the performance of the considered framework, seven different specifications listed as, F measure, accuracy, MCC, error rate, True and False Positive rates, and area under curve (AUC) are used.

2.1.3 MACHINE LEARNING CLASSIFIERS: In below Section explains Machine Learning classifiers with different algorithms.

a). Neural Network: An unreal neural network linked group of nodes, known as perceptron's, and is like a colossal network of neurons in a human brain. In this, the perceptron algorithm has been used to train the machine. It is for a managed learning of two-fold classifiers that can decide if an input belongs to some unique category or not.

b). Support Vector Machine (SVM) networks, are super visional learning algorithms that figure out the data used for classification and backsliding analysis. SVM model is a depiction of the examples as points in space, charted to create separate categories, divided by a clear chasm. New samples are then mapped into that same

space and then concluded to belong to a category based on the side of the chasm they fall.

c). Random Forest: Random Forest comes under the category of tree type classifiers, in this the dataset values are inspected separately and by the same distribution of all the trees in the forest. Internal valuation monitors strength, errors and the correlations which are implemented to display the response to the growing number of features that have been used in splitting.

d). Adaboost is capable of being used in a partnership with many other types of classifiers to boost performance. Adaboost is often said to be best the out-of-the-box classifier. Information collected at each stage of the Adaboost algorithm of each training sample is stuffed into the tree viable algorithm in such a way that later trees favor to focus on harder to classify examples.

e). Logistic Regression-Bayesian networks are aimed acyclic graphs whose nodes show variables in the Bayesian sense. Each node is correlated with a probability function that takes a specific set of values, as input, for the node's parent variables, and gives the probability distribution of the variable represented by the node.

2.2 PROPOSED FRAMEWORK

The main concern of analysis in the field of machine learning is being to form a scheduled computational machine for the categorizing the forecast of the objects, based on the attainable information. The outcome of proposed framework helps to predict the triggered sound waves reflect from surface Rock or a Mine.

2.2.1 PROPOSED FRAMEWORK METHODS:

Broadly in physical world or realistic issues, there is no curb over the types of data. Some dire pre-processing like removal of missing values, feature selection, etc. are always required. Machine learning focuses on taking up contemporary techniques to process huge amount of complex data with lower expense. The abstract view of proposed framework has been represented in Figure 1. Figure 1 describes the framework of the prediction model created to determine the surface to be a rock or a mine based on about 61 factors or features, processed by 10 different

classifier models, which give outputs with an acceptable accuracy and precision percentage.

i. Preprocessing: Missing values are removed by replacing them by mean value imputation.

ii. Feature Selection: Mean Gini index is used to rank the important features. The top 50 features ranked by mean gini index is selected and fed to the prediction model.

iii. Prediction Model: Different ML classifiers are explored and implemented to find the best possible solution. Random forest, being an ensemble model has shown the highest performance with 83.17% of accuracy. The results are further optimized by applying feature selection technique to feed the prediction model with the best features and accuracy reached at 90.20% after optimization. The outcome of this proposed framework helps to predict the targeted surface to be a Rock or a Mine.

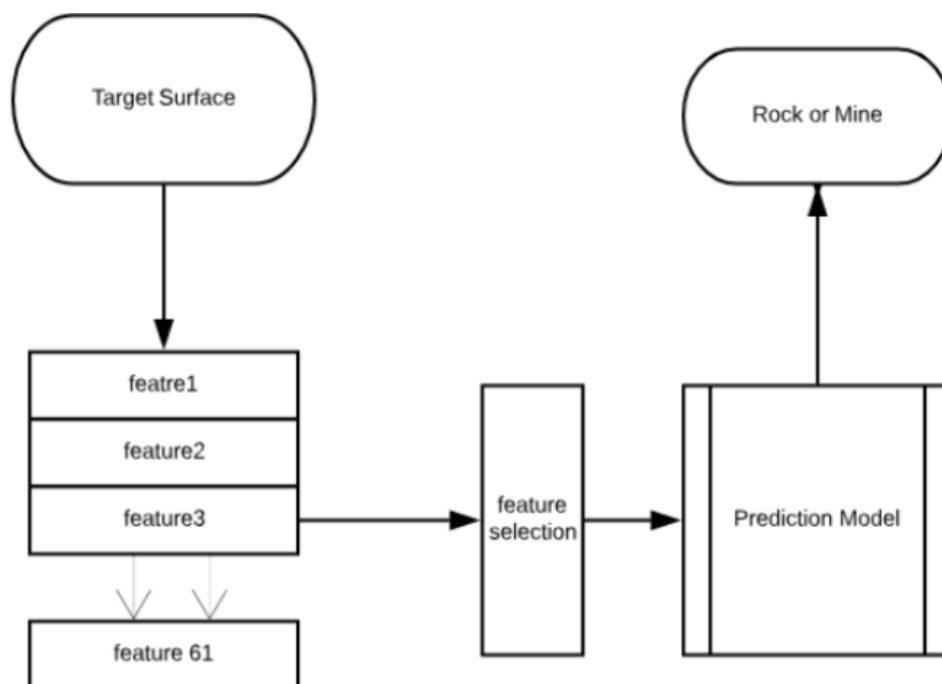


Figure 2.1. Prediction Framework

3.SYSTEM REQUIREMENTS

3.1 SOFTWARE AND HARDWARE REQUIREMENTS

Software Requirements

- Operating System: Windows10 and other versions.
- Language: Python 3.8.5

Hardware requirements

- Ram: 8 GB and more
- Processor: Any Intel Processor
- Hard Disk: 6 GB and more
- Speed: 1 GHZ and more

3.2 MATPLOTLIB

Humans are very visual creatures: we understand things better when we see things visualized. However, the step to presenting analyses, results or insights can be a bottleneck: you might not even know where to start or you might have already a right format in mind, but then questions like “Is this the right way to visualize the insights that I want to bring to my audience?” will have definitely come across your mind.

When you’re working with the Python plotting library Matplotlib, the first step to answering the above questions is by building up knowledge on topics like:

- The **Anatomy of Matplotlib plot**: what is a subplot? What are the Axes? What exactly is a figure?
- **Plot Creation**: which could raise questions about what module you exactly need to import (pylab or pyplot?), how you exactly should go about initializing the figure and the Axes of your plot, how to use matplotlib in Jupyter notebooks, etc.
- **Plotting Routines**: from simple ways to plot your data to more advanced ways of visualizing your data.

- **Basic Plot customization**, with a focus on plot legends and text, titles, axes labels and plot layout.
- **Saving, Showing and Clearing your plots**: show the plot, save one or more figures to, for example, pdf files, clear the axes, clear the figure or close the plot, etc.

(To practice matplotlib interactively, try the free Matplotlib chapter at the start of this Intermediate Python course or see DataCamp's Viewing 3D Volumetric Data With Matplotlib tutorial to learn how to work with matplotlib's event handler API.)

What does the Matplotlib looks like?

At first sight, it will seem that there are quite some components to consider when you start plotting with this Python data visualization library. You'll probably agree with me that it's confusing and sometimes even discouraging seeing the amount of code that is necessary for some plots, not knowing where to start yourself and which components you should use. Luckily, this library is very flexible and has a lot of handy, built-in defaults that will help you out tremendously. As such, you don't need much to get started: you need to make the necessary imports, prepare some data, and you can start plotting with the help of the `plot()` function! When you're ready, don't forget to show your plot using the `show()` function.

```
script.py
1  # Import the necessary packages and modules
2  import matplotlib.pyplot as plt
3  import numpy as np
4
5  # Prepare the data
6  x = np.linspace(0, 10, 100)
7
8  # Plot the data
9  plt.plot(x, x, label='linear')
10
11 # Add a legend
12 plt.legend()
13
14 # Show the plot
15 plt.show()
```

Run

Figure 3.1. matplotlib plot

3.3 PANDAS

pandas is a Python package that provides fast, flexible, and expressive data structures designed to make working with structured (tabular, multidimensional, potentially heterogeneous) and time series data both easy and intuitive. It aims to be the fundamental high-level building block for doing practical, **real world** data analysis in Python. Additionally, it has the broader goal of becoming **the most powerful and flexible open source data analysis / manipulation tool available in any language**. It is already well on its way toward this goal. Pandas is a popular Python package for data science, and with good reason: it offers powerful, expressive and flexible data structures that make data manipulation and analysis easy, among many other things. The DataFrame is one of these structures.

3.4 SKLEARN

Simple and efficient tools for predictive data analysis. Accessible to everybody, and reusable in various contexts. Built on NumPy, SciPy, and matplotlib. Open source, commercially usable - BSD license. Used in: Classification, Regression, Clustering, Dimensionality reduction, Model selection, pre-processing

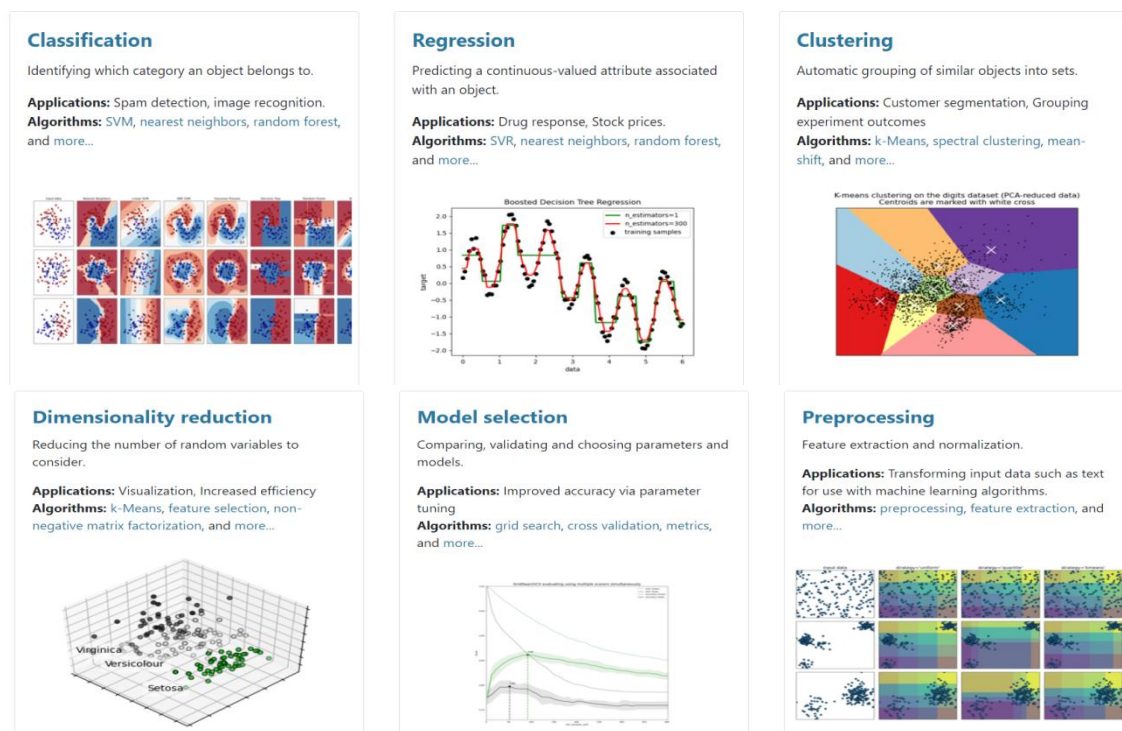


Figure 3.2. sklearn

4.IMPLEMENTATION

4.1 EXPERIMENTAL RESULTS AND DISCUSSION:

This section discusses parameter evaluation metrics to measure the performance of various machine learning algorithms. The results of 10-fold cross validation method are presented graphically and discussed much in detail.

A) DATA CORRELATION REPRESENTATION

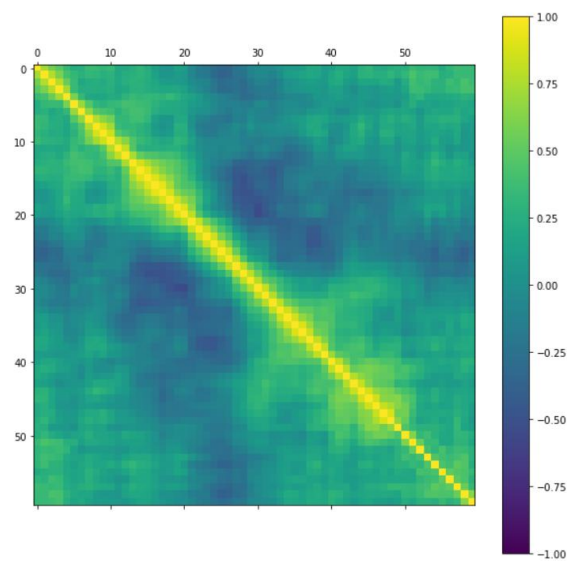


Figure 4.1. Data correlation representation

B) DIFFERENT DIMENSIONS OF FREQUENCY IN VERTICAL AXIS AND HORIZONTAL AXIS

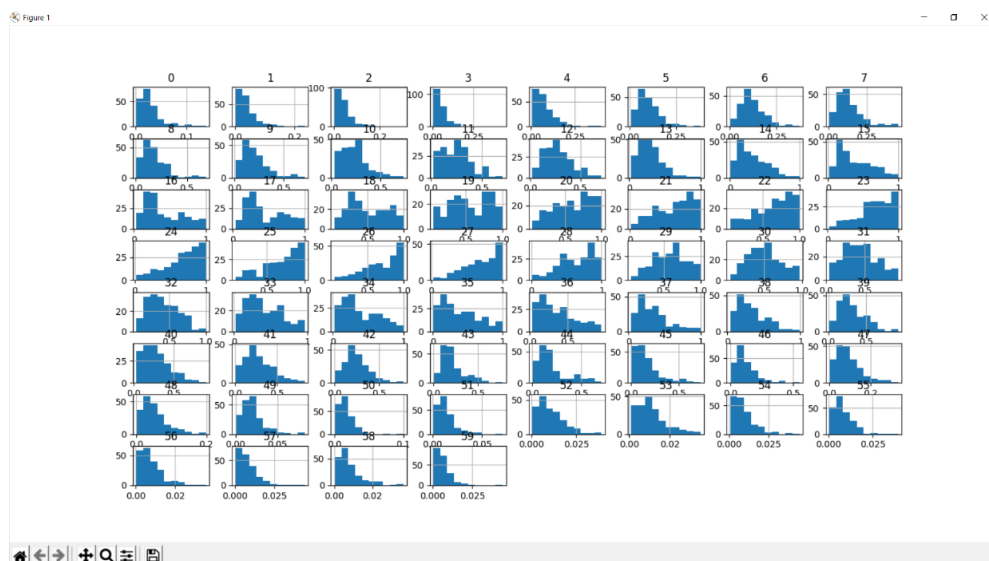


Figure 4.2. Different dimensions of frequency

C) DATA DISTRIBUTION PROCESS IN DENSITY PLOTS REPRESENTATION:

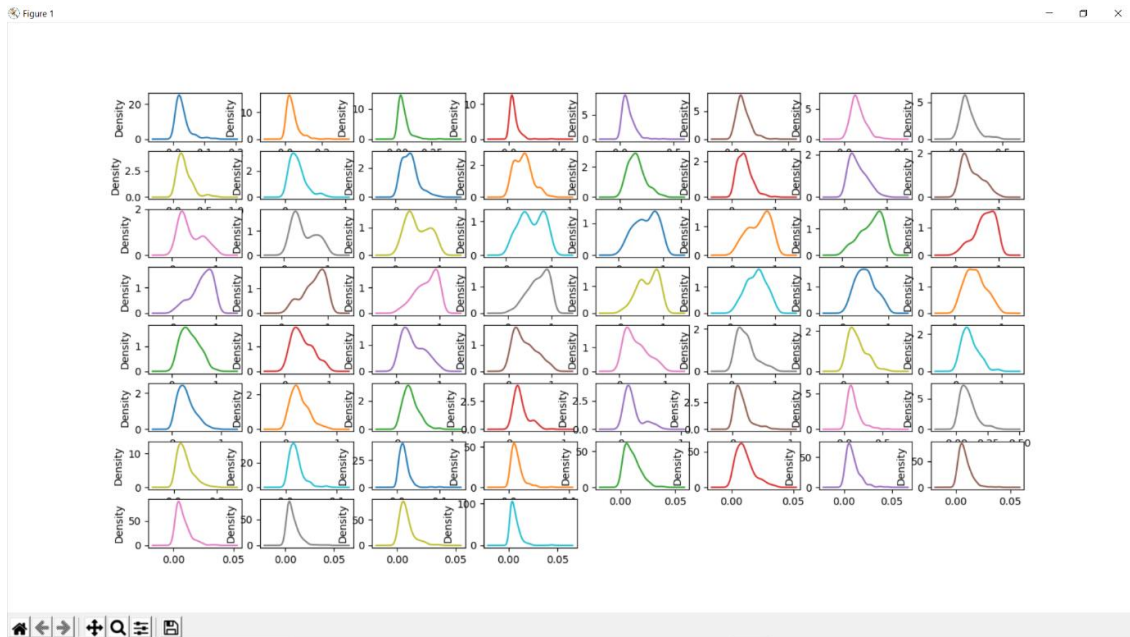


Figure 4.3. Data distribution process in density plot representation

D) ALGORITHMS COMPARISON:

In this module, we compare Numerical Data based on Quartile Values with different algorithms.

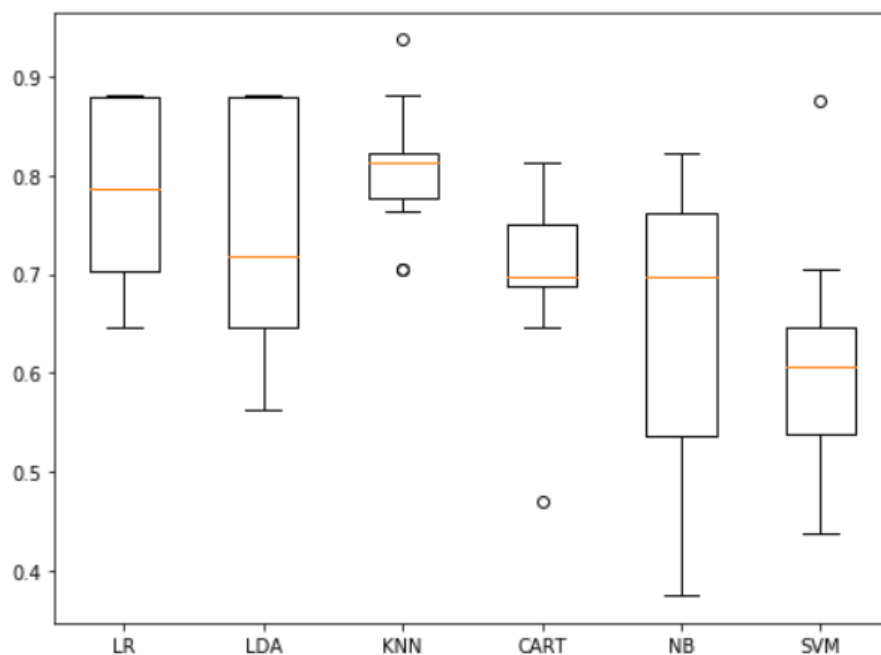


Figure 4.4. Algorithms comparison

- **LR:** Logistic Regression.
- **LDA:** Linear discriminant analysis.
- **KNN:** k-nearest neighbors.
- **CART:** Classification and Regression Trees.
- **NB:** Naive Bayes classifier.
- **SVM:** Support Vector Machine.

E) SCALED ALGORITHMS ACCURACY VALUES

```
ScaledLR: 0.734191 (0.095885)
ScaledLDA: 0.746324 (0.117854)
ScaledKNN: 0.825735 (0.054511)
ScaledCART: 0.705147 (0.095543)
ScaledNB: 0.648897 (0.141868)
ScaledSVM: 0.836397 (0.088697)
```

Figure 4.5. Scaled Algorithm accuracy values

F) SCALED ALGORITHM COMPARISON

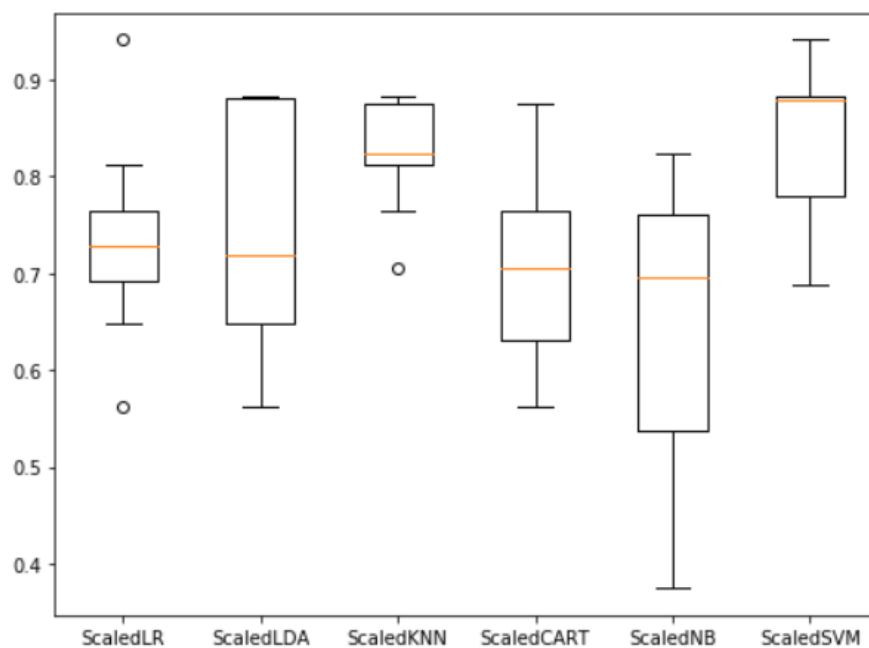
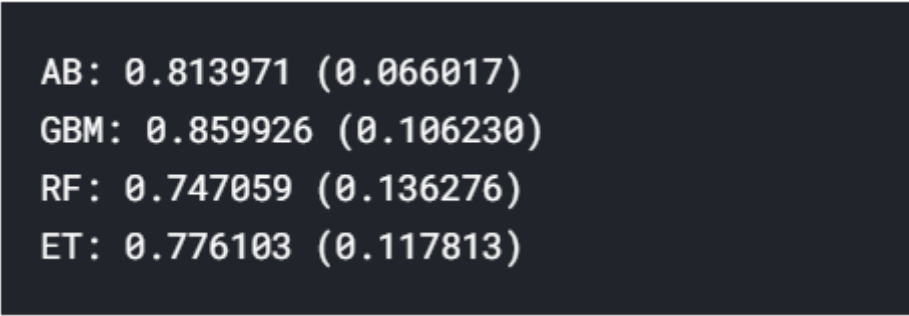


Figure 4.6. Scaled Algorithm comparison

- **ScaledLR:** Scaled Logistic Regression.
- **ScaledLDA:** Scaled Linear discriminant analysis.
- **ScaledKNN:** Scaled k-nearest neighbors.
- **ScaledCART:** Scaled Classification and Regression Trees.
- **ScaledNB:** Scaled Naive Bayes classifier.
- **ScaledSVM:** Scaled Support Vector Machine.

G) BOSTING ALGORITHMS ACCURACY VALUES



```

AB: 0.813971 (0.066017)
GBM: 0.859926 (0.106230)
RF: 0.747059 (0.136276)
ET: 0.776103 (0.117813)
  
```

Figure 4.7. Bosting algorithms accuracy values

4.2 IMPLEMENTATION CODE

```

Packages or module to be imported
from matplotlib import pyplot
from pandas import read_csv
from pandas import set_option
from pandas.plotting import scatter_matrix
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.model_selection import KFold
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.pipeline import Pipeline
from sklearn.linear_model import LogisticRegression
  
```

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC
from sklearn.ensemble import AdaBoostClassifier
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import ExtraTreesClassifier
```

For final model to derive output

```
# Finalize Model
```

```
# prepare the model
scaler = StandardScaler().fit(X_train)
rescaledX = scaler.transform(X_train)
model = SVC(C=1.5)
model.fit(rescaledX, Y_train)
# estimate accuracy on validation dataset
rescaledValidationX = scaler.transform(X_validation)
predictions = model.predict(rescaledValidationX)
print(accuracy_score(Y_validation, predictions))
print(confusion_matrix(Y_validation, predictions))
print(classification_report(Y_validation, predictions))
```

5.SCREENSHOTS

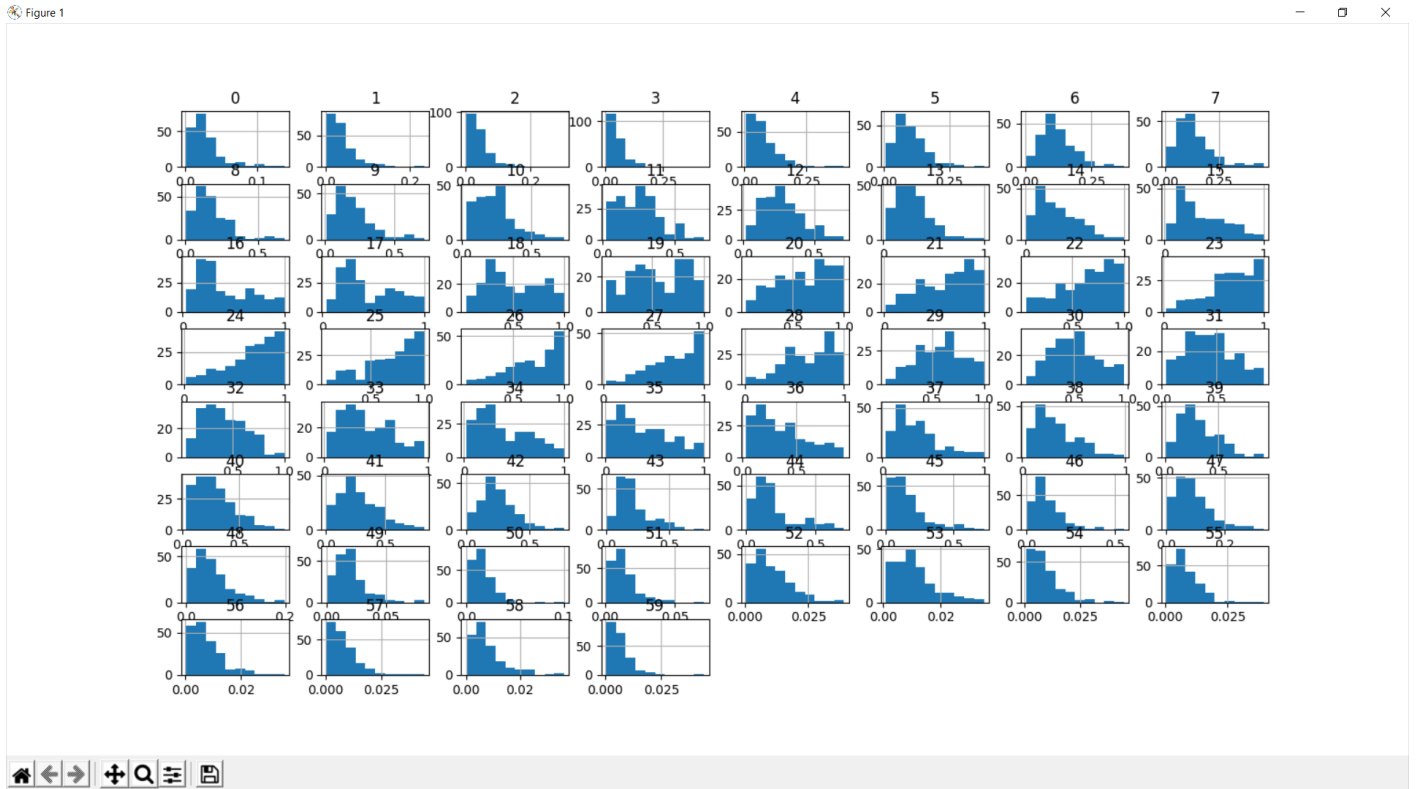


Figure 5.1 Output 1

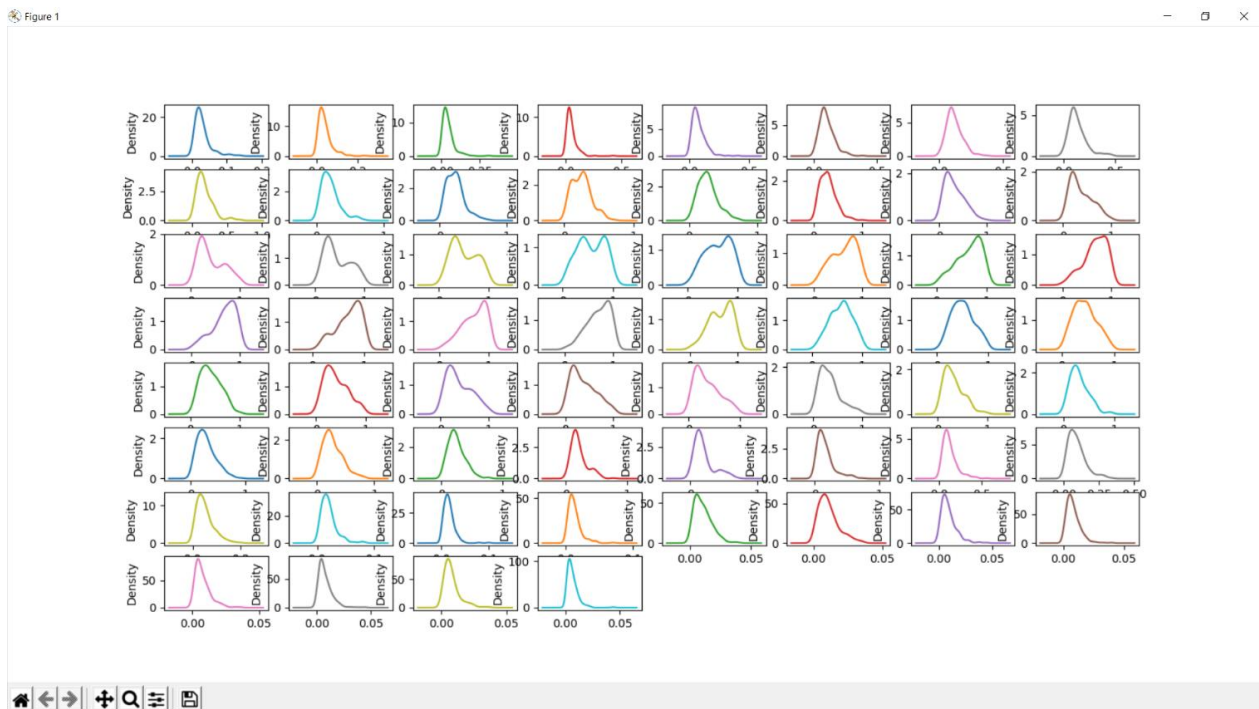


Figure 5.2. Output 2

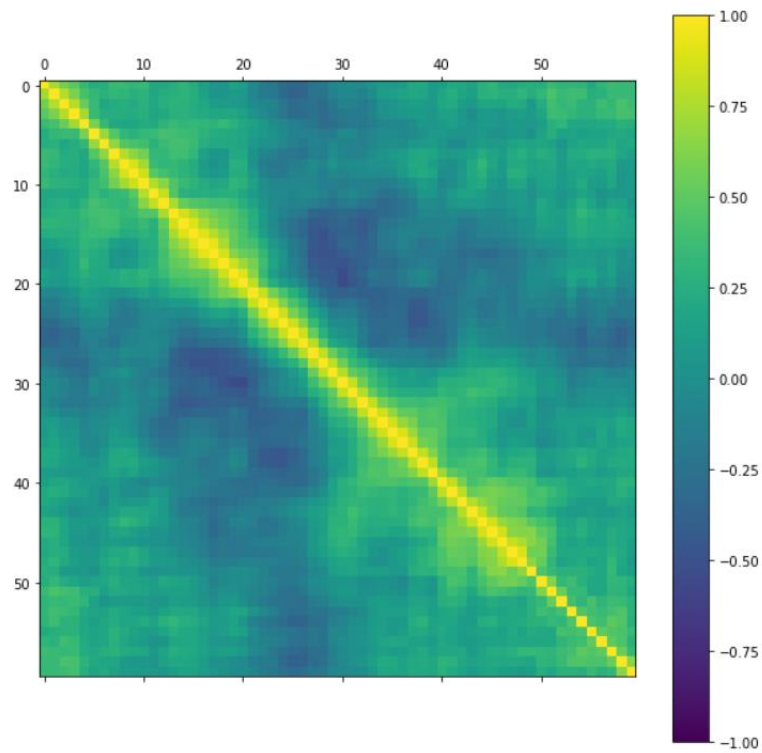


Figure 5.3. Data correlation representation

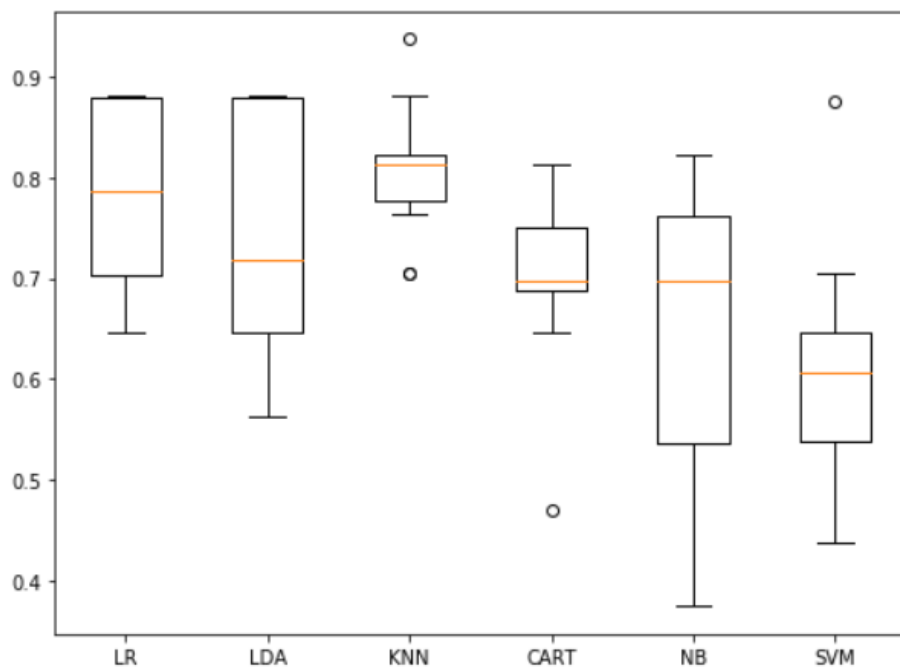


Figure 5.4 Algorithm Comparison

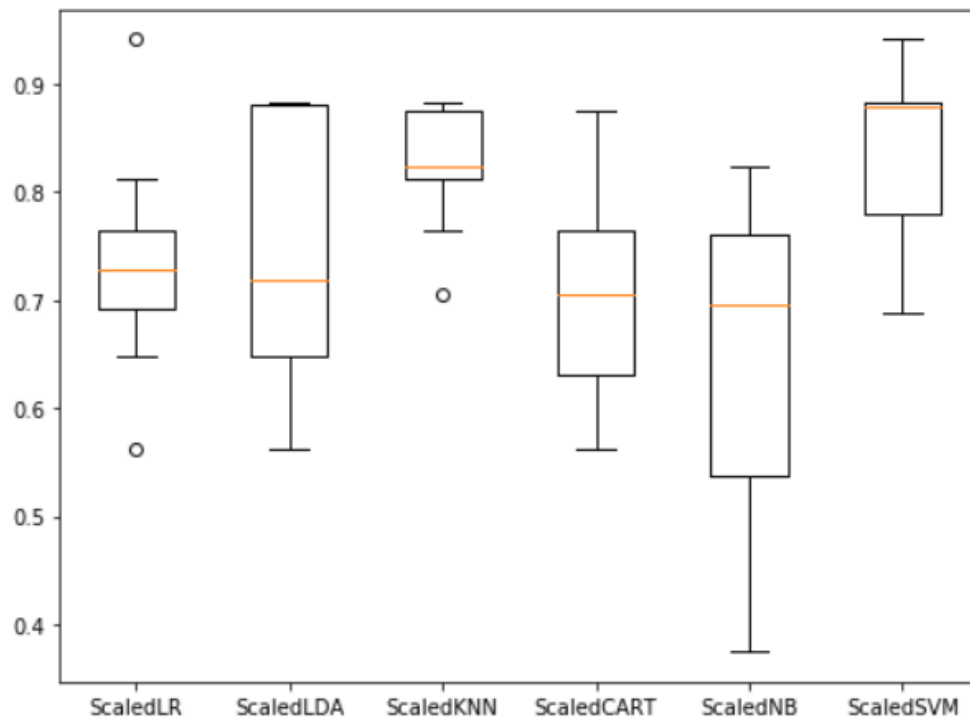


Figure 5.5 Algorithm Comparison

6.CONCULSION

An adequate prediction miniature, united with the machine learning classifying features, is proposed which can conclude if the target of the sound wave is either a rock or a mine or any other organism or any kind of other body. Research is carried out for predicting the best possible result for the target to be a rock or a mine, which is found to be best through the random forest model, which is an ensemble tree-based classifier in machine learning with the highest accuracy rate of 83.17% and giving the best ROCAUC rate 0.93, with least error for better elaboration of this prediction model. For future work more, complex data will be handled using big data Hadoop framework. With random forest algorithm, the results are further optimized by feature selection to get the accuracy of 91.15%.

SONAR technique, which relays on certain parameters to be able to detect the obstacle or the surface is a rock or a mine. Machine learning has drawn the attention of maximum part of the technology related and based industries, by showing advancements in the predictive analytics. The main aim is to emanate a capable prediction representative, united by the machine learning algorithmic characteristics, which can figure out if the target of the sound wave is either a rock or a mine or any other organism or any kind of other body. This attempt is a clear-cut case study which comes up with a machine learning plan for the grading of rocks and minerals, executed on a huge, highly spatial and complex SONAR dataset.

REFERENCE

- 1) . Ho, Tin Kam. Random Decision Forests (PDF). Proceedings of the 3rd International Conference on Document Analysis and Recognition, Montreal, QC, 14–16 August 1995. pp. 278–282. (1995).
- 2). Dura, Esther, et al. "[Active learning for detection of mine-like objects in side-scan sonar imagery](#)." IEEE Journal of Oceanic Engineering 30.2: 360-371 (2005).
- 3). Erkmen, Burcu, and TülayYıldırım. "[Improving classification performance of sonar targets by applying general regression neural network with PCA](#)." Expert Systems with Applications 35.1-2: 472-475. (2008).
- 4)Zhi-Hua Zhou and Yuan Jiang. NeC4.5: Neural Ensemble Based C4.5. IEEE Trans.Knowl.Data Ang, 16. 2004.[View Context].
- 5)Jianbin Tan and David L. Dowe. MML Inference of Oblique Decision Trees. Australian Conference on Artificial Intelligence. 2004. [View Context].
- 6)Jeremy Kubica and Andrew Moore. Probabilistic Noise Identification and Data Cleaning. ICDM. 2003. [View Context].
- 7)Dennis DeCoste. Anytime Query-Tuned Kernel Machines via Cholesky Factorization. SDM. 2003. [View Context].
- 8)Dennis DeCoste. Anytime Interval-Valued Outputs for Kernel Machines: Fast Support Vector Machine Classification via Distance Geometry. ICML. 2002. [View Context].
- 9)Ayhan Demiriz and Kristin P. Bennett and Mark J. Embrechts. A Genetic Algorithm Approach for Semi-Supervised Clustering. E-Business Department, Verizon Inc.. 2002. [View Context].
- 10)Michail Vlachos and Carlotta Domeniconi and Dimitrios Gunopulos and George Kollios and Nick Koudas. Non-linear dimensionality reduction techniques for classification and visualization. KDD. 2002. [View Context].
- 11)Xavier Llor and David E. Goldberg and Ivan Traus and Ester Bernad i Mansilla. Accuracy, Parsimony, and Generality in Evolutionary Learning Systems via Multiobjective Selection. IWLCS. 2002. [View Context].

APPENDIX

A. SOURCE CODE

```
# Classification Project: Sonar rocks or mines

# Load libraries
from matplotlib import pyplot
from pandas import read_csv
from pandas import set_option
from pandas.plotting import scatter_matrix
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.model_selection import KFold
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.pipeline import Pipeline
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC
from sklearn.ensemble import AdaBoostClassifier
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import ExtraTreesClassifier
# Load dataset
url='https://archive.ics.uci.edu/ml/machine-learning-
databases/undocumented/connectionist-bench/sonar/sonar.all-data'
```

```

dataset = read_csv(url, header=None)
# Summarize Data
# Descriptive statistics
# shape
print(dataset.shape)
# types
set_option('display.max_rows', 500)
print(dataset.dtypes)
# head
set_option('display.width', 100)
print(dataset.head(20))
# descriptions, change precision to 3 places
set_option('precision', 3)
print(dataset.describe())
# class distribution
print(dataset.groupby(60).size())

# Data visualizations
# histograms
dataset.hist()
pyplot.show()
# density
dataset.plot(kind='density',      subplots=True,      layout=(8,8),      sharex=False,
legend=False)
pyplot.show()
# box and whisker plots
# dataset.plot(kind='box', subplots=True, layout=(w,8), sharex=False, sharey=False)
# pyplot.show()

# scatter plot matrix
scatter_matrix(dataset)

```

```
pyplot.show()
```

```
# correlation matrix
```

```
fig = pyplot.figure()
```

```
ax = fig.add_subplot(111)
```

```
cax = ax.matshow(dataset.corr(), vmin=-1, vmax=1, interpolation='none')
```

```
fig.colorbar(cax)
```

```
pyplot.show()
```

```
# Prepare Data
```

```
# Split-out validation dataset
```

```
array = dataset.values
```

```
X = array[:,0:60].astype(float)
```

```
Y = array[:,60]
```

```
validation_size = 0.20
```

```
seed = 7
```

```
X_train, X_validation, Y_train, Y_validation = train_test_split(X, Y,  
test_size=validation_size, random_state=seed)
```

```
# Evaluate Algorithms
```

```
# Test options and evaluation metric
```

```
num_folds = 10
```

```
seed = 7
```

```
scoring = 'accuracy'
```

```
# Spot Check Algorithms
```

```
models = []
```

```
models.append(('LR', LogisticRegression()))
```

```
models.append(('LDA', LinearDiscriminantAnalysis()))
```

```
models.append(('KNN', KNeighborsClassifier()))
```

```
models.append(('CART', DecisionTreeClassifier()))
```

```

models.append(('NB', GaussianNB()))
models.append(('SVM', SVC()))
results = []
names = []
for name, model in models:
    kfold = KFold(n_splits=num_folds, random_state=seed)
    cv_results = cross_val_score(model, X_train, Y_train, cv=kfold,
scoring=scoring)
    results.append(cv_results)
    names.append(name)
    msg = "%s: %f (%f)" % (name, cv_results.mean(), cv_results.std())
    print(msg)

```

```

# Compare Algorithms
fig = pyplot.figure()
fig.suptitle('Algorithm Comparison')
ax = fig.add_subplot(111)
pyplot.boxplot(results)
ax.set_xticklabels(names)
pyplot.show()

```

```

# Standardize the dataset
pipelines = []
pipelines.append(('ScaledLR', Pipeline([('Scaler', StandardScaler()), ('LR',
LogisticRegression())])))
pipelines.append(('ScaledLDA', Pipeline([('Scaler', StandardScaler()), ('LDA',
LinearDiscriminantAnalysis())])))
pipelines.append(('ScaledKNN', Pipeline([('Scaler', StandardScaler()), ('KNN',
KNeighborsClassifier())])))
pipelines.append(('ScaledCART', Pipeline([('Scaler', StandardScaler()), ('CART',
DecisionTreeClassifier())])))
pipelines.append(('ScaledNB', Pipeline([('Scaler', StandardScaler()), ('NB',
GaussianNB())])))

```

```

pipelines.append(('ScaledSVM', Pipeline([('Scaler', StandardScaler()),('SVM',
SVC())])))
results = []
names = []
for name, model in pipelines:
    kfold = KFold(n_splits=num_folds, random_state=seed)
    cv_results = cross_val_score(model, X_train, Y_train, cv=kfold,
scoring=scoring)
    results.append(cv_results)
    names.append(name)
    msg = "%s: %f (%f)" % (name, cv_results.mean(), cv_results.std())
    print(msg)

```

Compare Algorithms

```

fig = pyplot.figure()
fig.suptitle('Scaled Algorithm Comparison')
ax = fig.add_subplot(111)
pyplot.boxplot(results)
ax.set_xticklabels(names)
pyplot.show()

```

Tune scaled KNN

```

scaler = StandardScaler().fit(X_train)
rescaledX = scaler.transform(X_train)
neighbors = [1,3,5,7,9,11,13,15,17,19,21]
param_grid = dict(n_neighbors=neighbors)
model = KNeighborsClassifier()
kfold = KFold(n_splits=num_folds, random_state=seed)
grid = GridSearchCV(estimator=model, param_grid=param_grid, scoring=scoring,
cv=kfold)
grid_result = grid.fit(rescaledX, Y_train)
print("Best: %f using %s" % (grid_result.best_score_, grid_result.best_params_))
means = grid_result.cv_results_['mean_test_score']

```

```

stds = grid_result.cv_results_['std_test_score']
params = grid_result.cv_results_['params']
for mean, stdev, param in zip(means, stds, params):
    print("%f (%f) with: %r" % (mean, stdev, param))

# Tune scaled SVM
scaler = StandardScaler().fit(X_train)
rescaledX = scaler.transform(X_train)
c_values = [0.1, 0.3, 0.5, 0.7, 0.9, 1.0, 1.3, 1.5, 1.7, 2.0]
kernel_values = ['linear', 'poly', 'rbf', 'sigmoid']
param_grid = dict(C=c_values, kernel=kernel_values)
model = SVC()
kfold = KFold(n_splits=num_folds, random_state=seed)
grid = GridSearchCV(estimator=model, param_grid=param_grid, scoring=scoring,
cv=kfold)
grid_result = grid.fit(rescaledX, Y_train)
print("Best: %f using %s" % (grid_result.best_score_, grid_result.best_params_))
means = grid_result.cv_results_['mean_test_score']
stds = grid_result.cv_results_['std_test_score']
params = grid_result.cv_results_['params']
for mean, stdev, param in zip(means, stds, params):
    print("%f (%f) with: %r" % (mean, stdev, param))

# ensembles
ensembles = []
ensembles.append(('AB', AdaBoostClassifier()))
ensembles.append(('GBM', GradientBoostingClassifier()))
ensembles.append(('RF', RandomForestClassifier()))
ensembles.append(('ET', ExtraTreesClassifier()))
results = []
names = []
for name, model in ensembles:

```

```

        kfold = KFold(n_splits=num_folds, random_state=seed)
        cv_results = cross_val_score(model, X_train, Y_train, cv=kfold,
scoring=scoring)
        results.append(cv_results)
        names.append(name)
        msg = "%s: %f (%f)" % (name, cv_results.mean(), cv_results.std())
        print(msg)

```

```

# Compare Algorithms
fig = pyplot.figure()
fig.suptitle('Ensemble Algorithm Comparison')
ax = fig.add_subplot(111)
pyplot.boxplot(results)
ax.set_xticklabels(names)
pyplot.show()

```

```

# Finalize Model

```

```

# prepare the model
scaler = StandardScaler().fit(X_train)
rescaledX = scaler.transform(X_train)
model = SVC(C=1.5)
model.fit(rescaledX, Y_train)
# estimate accuracy on validation dataset
rescaledValidationX = scaler.transform(X_validation)
predictions = model.predict(rescaledValidationX)
print(accuracy_score(Y_validation, predictions))
print(confusion_matrix(Y_validation, predictions))
print(classification_report(Y_validation, predictions))

```