DETECTION OF PHISHISING WEBSITES USING MACHINE LEARNING

A report submitted in partial fulfilment of the requirements for the award of the degree of

Bachelor of Technology
in
Department of Computer Science and Engineering – (Data Science) by

(Gajjala Samba Siva - 21691A3294)
(Kanneluri Sravani - 21691A32A6)
(Thrushitha Reddy Konkala - 21691A32C2)

DEPARTMENT OF
COMPUTER SCIENCE AND ENGINEERING
NATIONAL INSTITUTE OF TECHNOLOGY PUDUCHERRY
KARAIKAL – 609 609

DR. Narendran Rajagopalan	**Dr. Narendran Rajagopalan**
Associate Professor	Head of the Department
Project Guide	

Abstract

Phishing attacks pose a significant threat to cybersecurity, targeting unsuspecting users to steal sensitive information such as login credentials, credit card details, and personal data. The purpose of this project is to develop an effective method to detect phishing websites using a Random Forest Classifier. This machine learning approach leverages various features extracted from websites to distinguish between legitimate and phishing sites accurately. Our study demonstrates the efficacy of the Random Forest Classifier in identifying phishing websites, offering a robust solution to mitigate this pervasive cyber threat.

Introduction

Background

Phishing websites are designed to deceive users into believing they are visiting legitimate websites, thereby tricking them into disclosing sensitive information. These malicious websites are a growing concern, with sophisticated techniques employed by cybercriminals to evade detection. Traditional detection methods, such as blacklisting, are often insufficient due to the dynamic nature of phishing sites.

Objective

The primary objective of this project is to develop a machine learning model, specifically a Random Forest Classifier, to detect phishing websites. By analyzing various features of websites, the model aims to distinguish between legitimate and phishing sites with high accuracy.

Methodology

Data Collection

The dataset used for this project comprises labeled examples of phishing and legitimate websites. The data was sourced from publicly available phishing site databases and legitimate website lists. The dataset includes various features, such as URL-based features, HTML and JavaScript-based features, and domain-based features. It contains 11,054 records with 32 features, which include:

- Index
- UsingIP
- LongURL
- ShortURL
- Symbol@
- Redirecting//
- PrefixSuffix-
- SubDomains
- HTTPS
- DomainRegLen
- Favicon
- NonStdPort
- HTTPSDomainURL
- RequestURL
- AnchorURL
- LinksInScriptTags
- ServerFormHandler
- InfoEmail
- AbnormalURL
- WebsiteForwarding
- StatusBarCust
- DisableRightClick
- UsingPopupWindow
- IframeRedirection
- AgeofDomain
- DNSRecording

- WebsiteTraffic

- PageRank

- GoogleIndex

- LinksPointingToPage

- StatsReport

- class

Model Selection

A Random Forest Classifier was chosen for this project due to its ability to handle large datasets with high dimensionality and its robustness to overfitting. The classifier operates by constructing multiple decision trees during training and outputting the class that is the mode of the classes (classification) of the individual trees.

Model Training and Evaluation

The dataset was split into training and testing sets using an 80-20 ratio. The Random Forest Classifier was trained on the training set, and its performance was evaluated on the testing set using metrics such as accuracy, precision, recall, and F1-score.

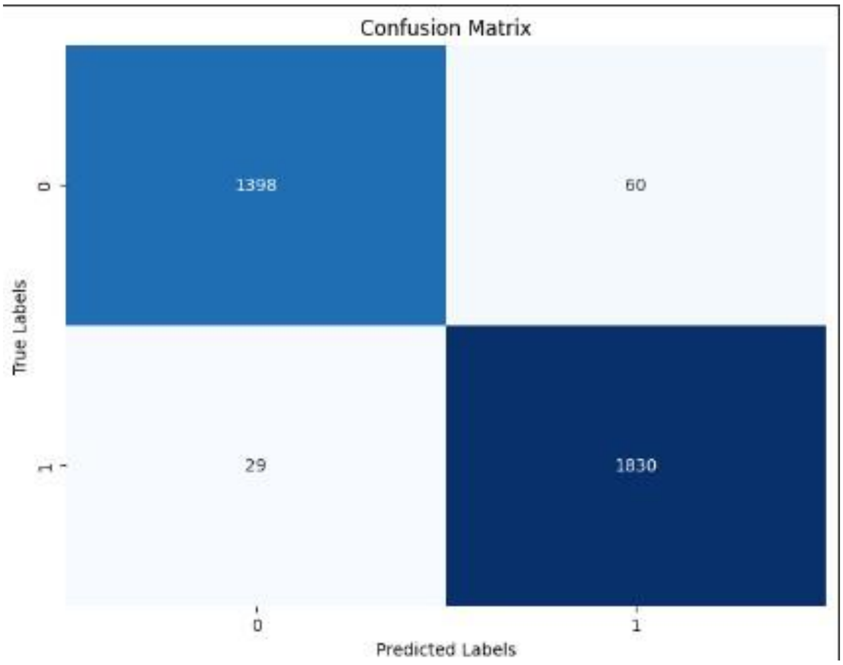
RESULTS Classifiers and their Performance Metrics

Random Forest

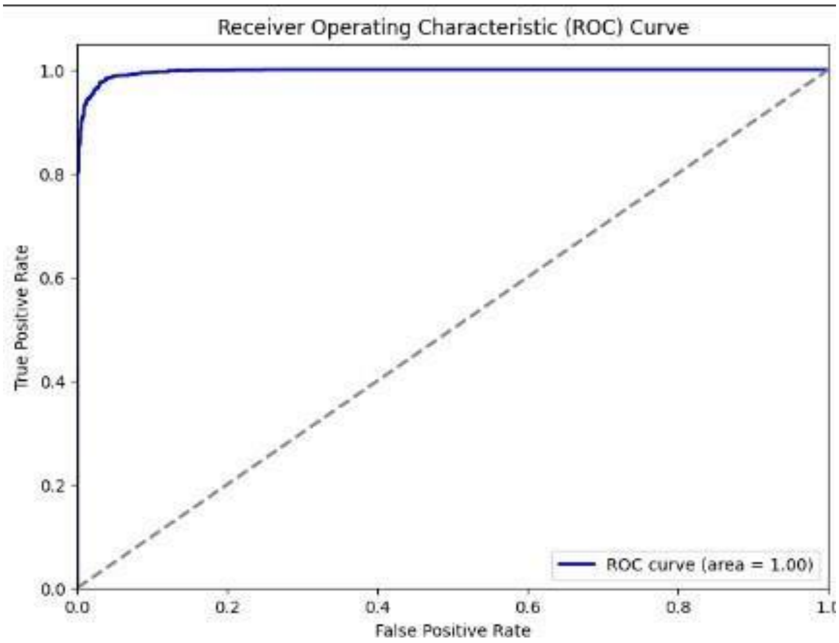
The Random Forest or Random Decision Forest is a supervised machine learning algorithm, used for both classification and regression. It is an ensemble learning method, works by creating a number of Decision Trees during the training phase. It generally provides high accuracy and robust performance.

- Accuracy : 97.32
- Precision : 98
- Recall : 96
- F1-Score : 97

Confusion Matrix



ROC Curve



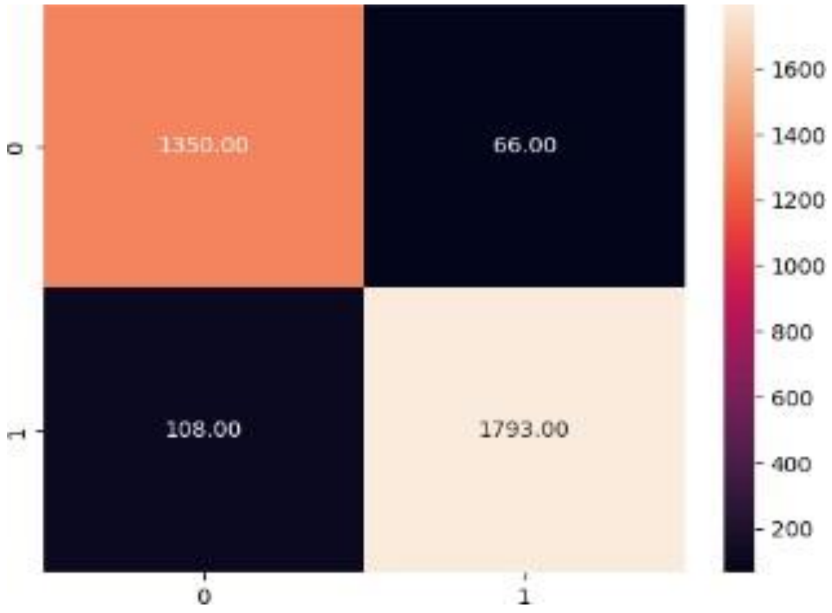
Decision Tree

Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving

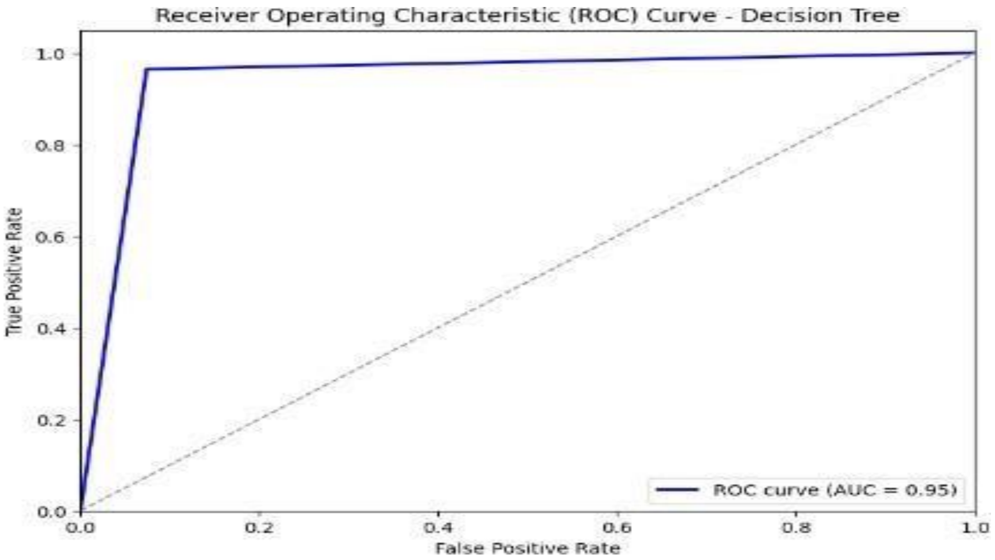
Classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome. Decision Tree consists of two nodes, the decision node and the leaf node. The decision node is responsible for making decisions, which consists of multiple branches. Leaf nodes are the outputs of those decisions and do not contain any branches.

- Accuracy : 95
- Precision : 93
- Recall : 95
- F1-Score : 94

Confusion Matrix



ROC Curve

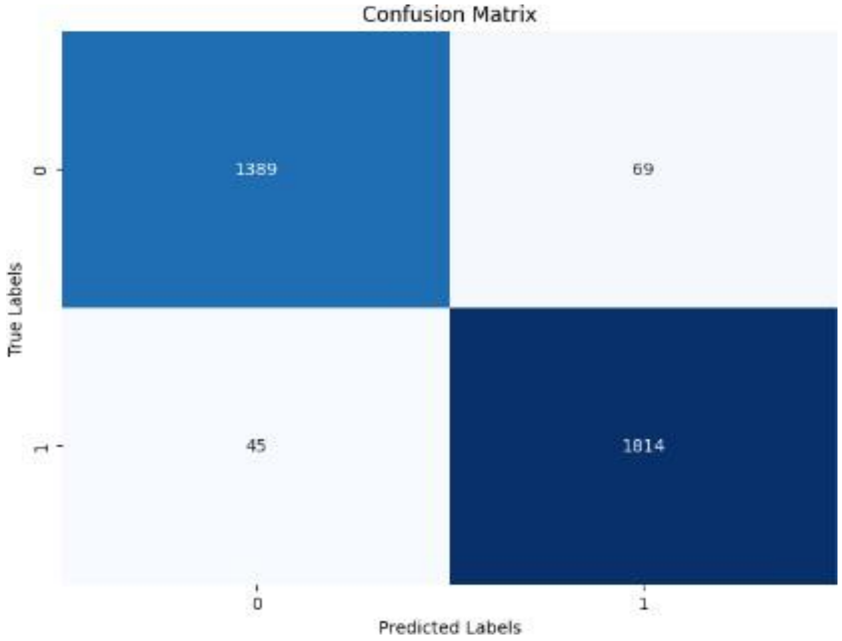


XGBoost

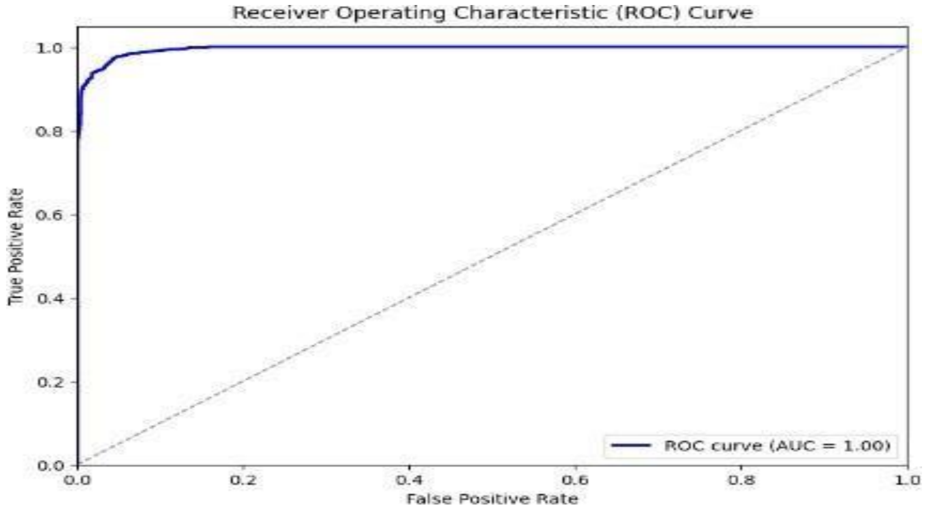
XGBoost i.e., eXtreme Gradient Boosting is a powerful and scalable machine learning system for tree boosting. It is known for its efficiency, speed and accuracy. It belongs to the family of boosting algorithms, which are ensemble learning techniques that combine the predictions of multiple weak learners.

- Accuracy : 97
- Precision : 97
- Recall : 95
- F1-Score : 96

Confusion Matrix



ROC Curve

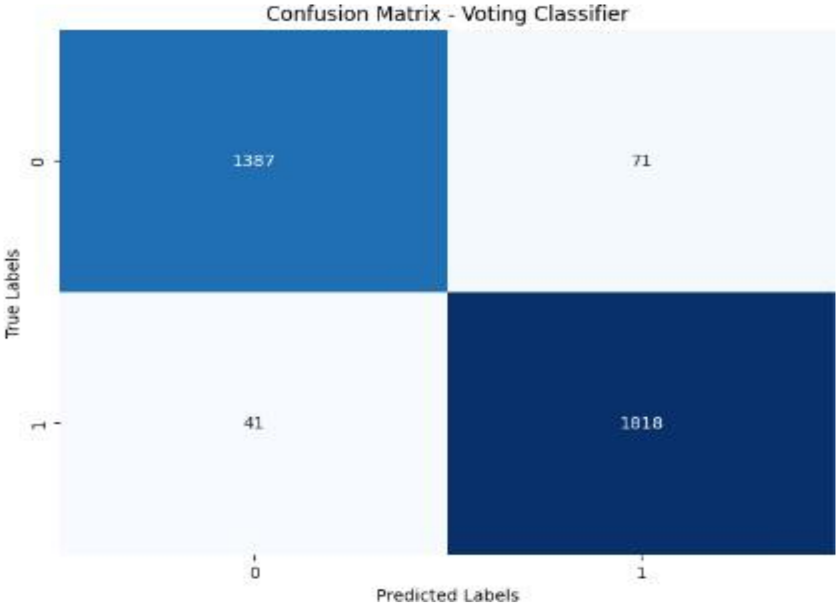


Voting Classifier

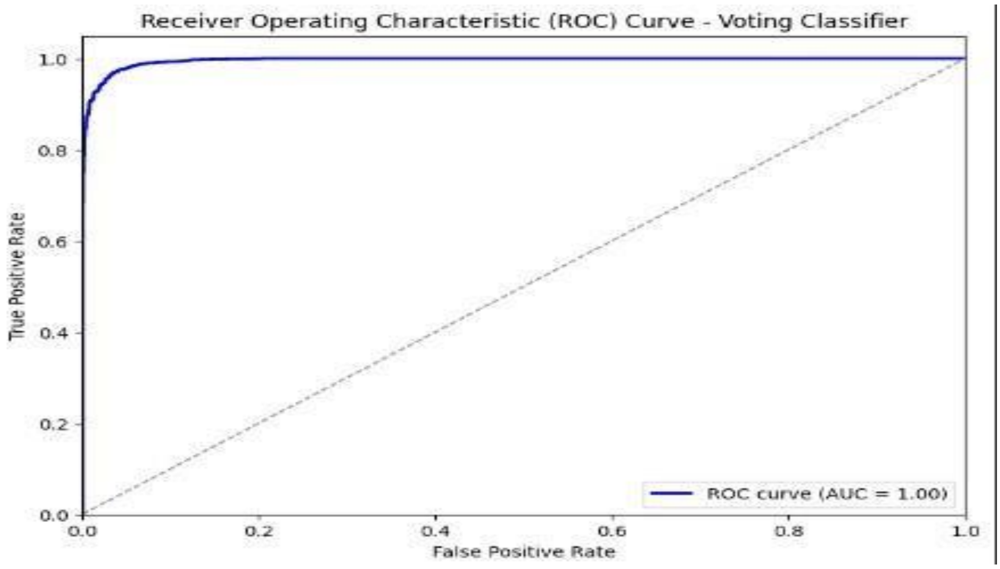
The Voting Classifier combines multiple machine learning models and predicts the output class based on the highest majority of voting. It helps in leveraging the strengths of multiple models to achieve better performance.

- Accuracy : 97
- Precision : 97
- Recall : 95
- F1-Score : 96

Confusion Matrix



ROC Curve

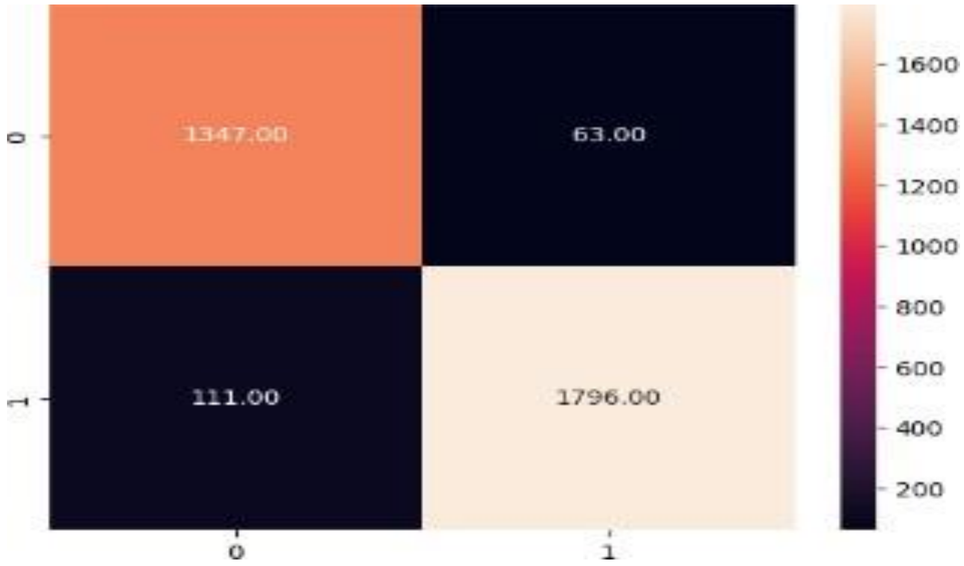


Gradient Boosting

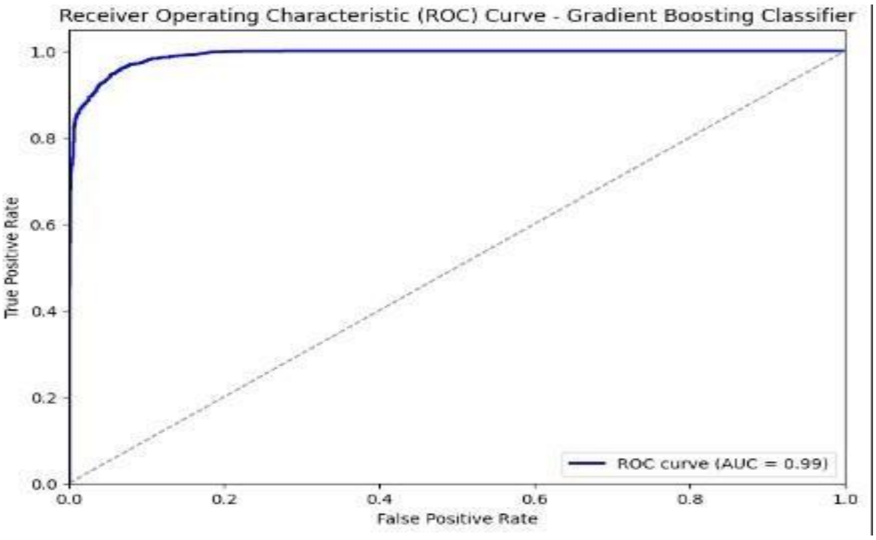
Gradient Boosting is a popular machine learning algorithm for classification and regression tasks. It is another powerful ensemble learning method that builds models sequentially, with each new model attempting to correct the errors made by the previous models. It often provides high predictive accuracy and robustness.

- Accuracy : 95
- Precision : 92
- Recall : 96
- F1-Score : 94

Confusion Matrix



ROC Curve

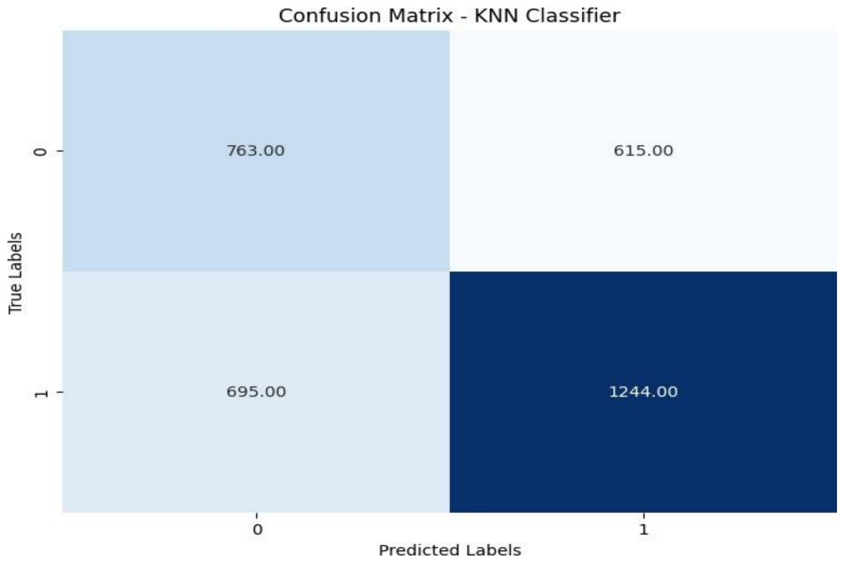


K-Nearest Neighbors (KNN)

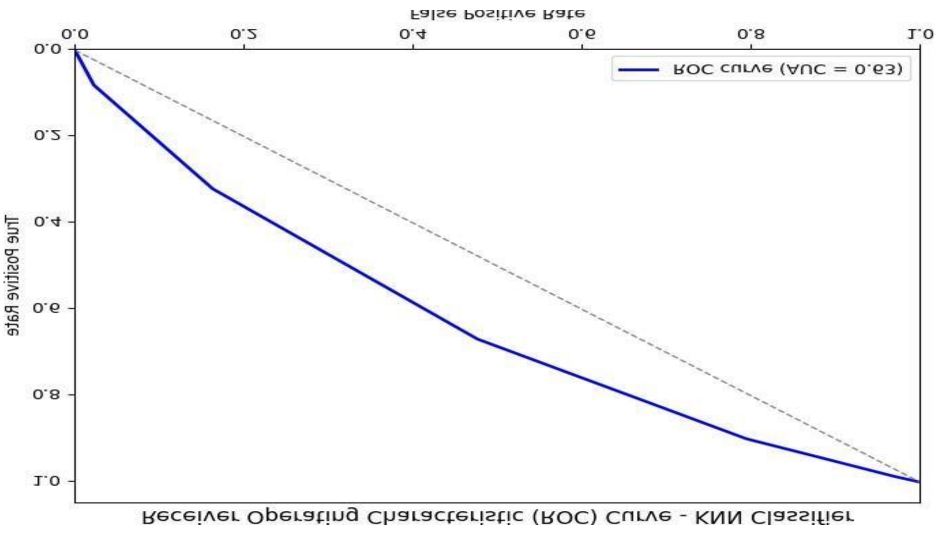
K-Nearest Neighbors (KNN) is a simple, non-parametric algorithm that classifies data points based on their nearest neighbors. It uses a distance metric (e.g., Euclidean distance) to find the K closest neighbors and assigns the most common class among them. KNN is easy to implement and works well for small datasets but can be computationally expensive for large or high-dimensional data.

- Accuracy : 61
- Precision : 67
- Recall : 64
- F1-Score : 66

Confusion Matrix



ROC Curve

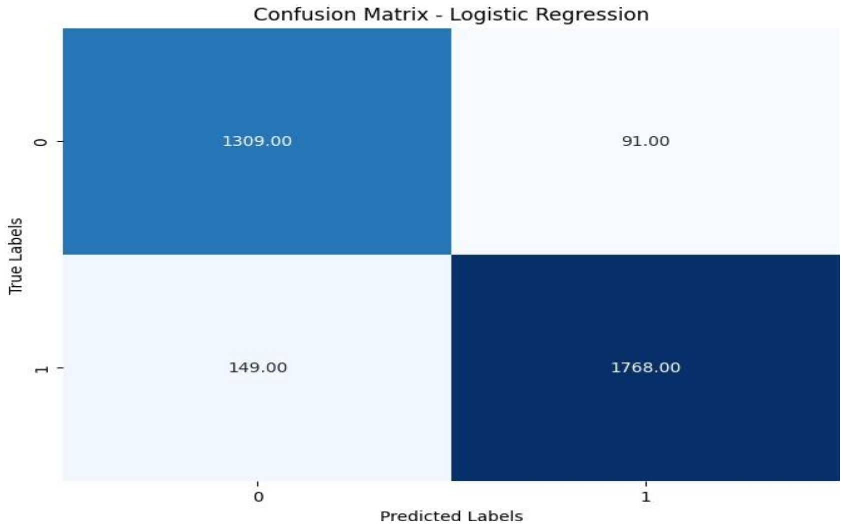


Logistic Regression

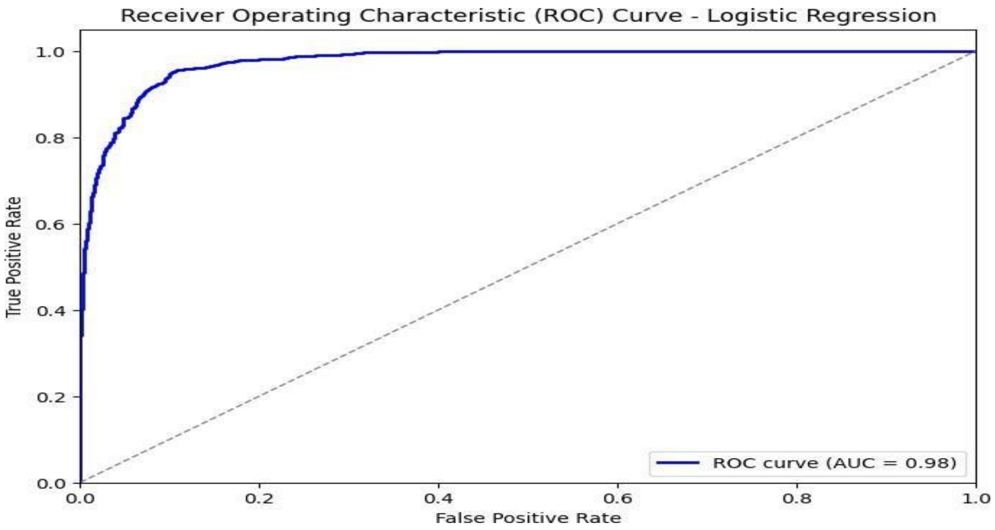
Logistic Regression is a method for binary classification that models the probability of an outcome based on predictor variables. It uses the logistic function to convert the linear combination of input features into a probability between 0 and 1. It is simple, interpretable, and effective with linearly separable data but may struggle with non-linear relationships unless combined with other techniques.

- Accuracy : 93
- Precision : 95
- Recall : 94
- F1-Score : 94

Confusion Matrix



ROC Curve

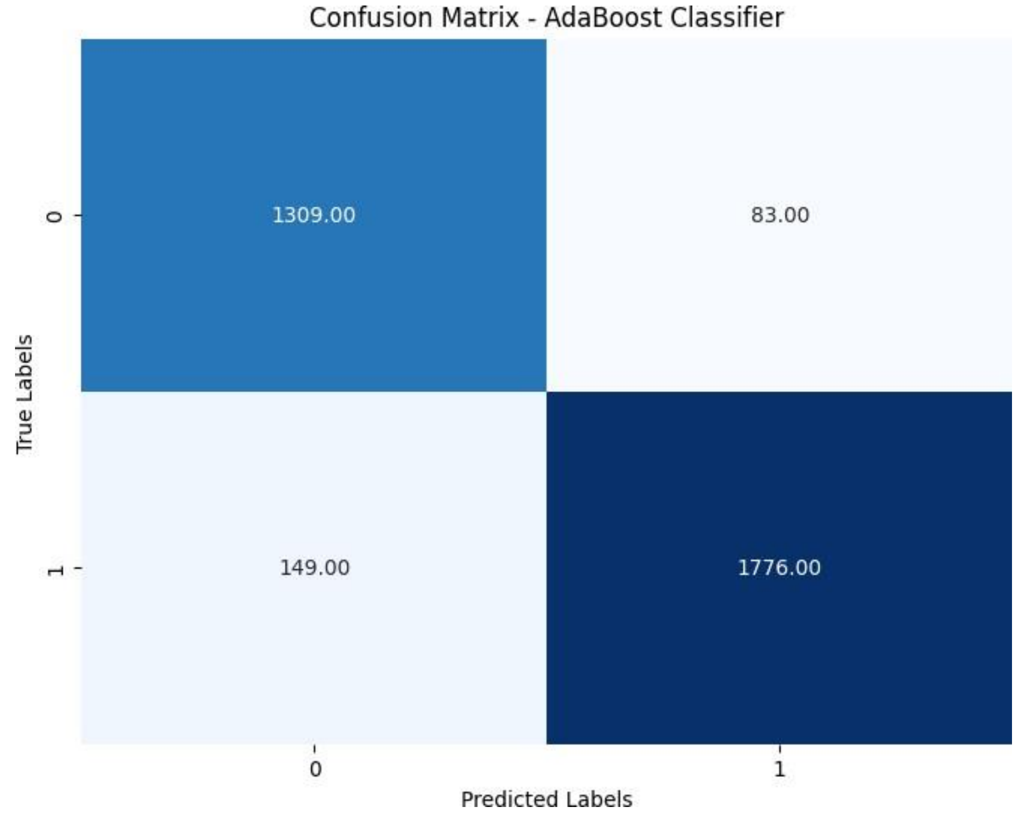


AdaBoost

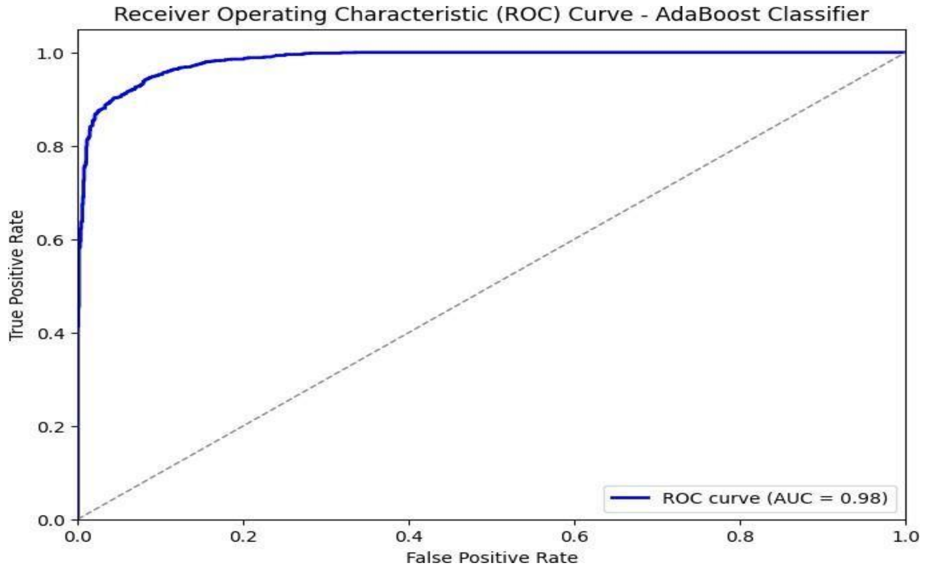
AdaBoost (Adaptive Boosting) is an ensemble learning technique that combines multiple weak classifiers to form a strong classifier. It works by sequentially training weak classifiers, usually decision trees with a single split (stumps), and adjusting the weights of misclassified instances to focus more on difficult cases. Each subsequent classifier is trained to correct the errors of the previous ones. AdaBoost is effective in improving the accuracy of weak classifiers and is less prone to overfitting than other ensemble methods. However, it can be sensitive to noisy data and outliers.

- Accuracy : 93
- Precision : 96
- Recall : 94
- F1-Score : 94

Confusion Matrix



ROC Curve



Acknowledgments

We would like to express our sincere gratitude to Dr. Narendran Rajagopalan, Associate Professor, Department of CSE, National Institute of Technology - Puducherry, for their invaluable guidance, support, and encouragement throughout the duration of this project. I also extend my appreciation to my colleagues and the National Institute of Technology Puducherry for providing the resources and encouragement needed to complete this work. Finally, I acknowledge the contributions of the open-source community for providing the tools and libraries that made this research possible.