Predict and Prevent DDOS Attacks Using Machine Learning and Statistical Algorithms

A report submitted in partial fulfillment of the requirements for

the award of the degree of

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

By

(Sunkayyagari Venkatesh – 21691A32C9)
(Kangati Sravya –21691A32A9)
(Sompalli Likhitha - 21691A3247)

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

Dr. Narendran Rajagopalan
Associate Professor
Project Guide

Dr. Narendran Rajagopalan
Head of the Department

Abstract:

Distributed Denial of Service (DDoS) attacks pose a significant threat to the security and availability of online services. This project aims to develop a comprehensive framework for predicting and preventing DDoS attacks using advanced machine learning and statistical algorithms. By leveraging the CICDDoS2019 dataset, we explore the efficacy of various algorithms including KNN, Logistic Regression, SVM, Naive Bayes , Decision Tree , Gradient Boost and ANN. Our objective is to identify the most accurate model for detecting DDoS attacks by comparing their performance metrics. The project involves extensive data preprocessing, feature selection, model training, and validation phases to ensure robust and reliable predictions. The outcomes of this research will provide valuable insights into the strengths and weaknesses of each algorithm, contributing to the development of more effective DDoS mitigation strategies.

Introduction:

Distributed Denial of Service (DDoS) attacks have become a prevalent and severe threat to the stability and security of online services and networks. Theseattacks overwhelm a target system with a flood of malicious traffic, causing service disruptions and significant financial losses. With the increasing sophistication and frequency of DDoS attacks, traditional defense mechanisms are often inadequate. As a result, there is a growing need for advanced predictive and preventive strategies.

Machine learning (ML) and statistical algorithms offer promising solutions to the challenges of DDoS detection and prevention. These technologies can analyze vast amounts of network traffic data to identify patterns indicative of potential attacks. By leveraging their predictive capabilities, organizations can preemptively respond to threats, reducing the likelihood of successful DDoS incidents.

Types of DDOS attacks:

1-UDP Flood

2- ICMP(Ping) Flood

3- SYN Flood

4- Ping Of Death

5- HTTP Flood.

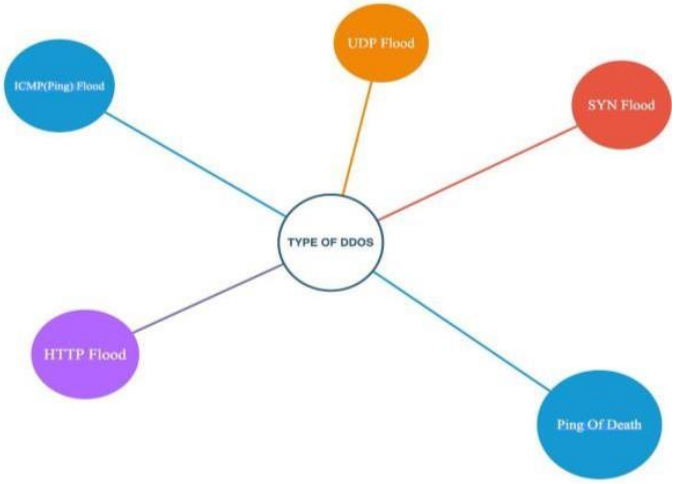


Fig 1: Types of DDOS attacks

- **UDP Flood**: In this attack, the attacker sets up random ports on the target network by sending IP packets containing UDP datagrams. The victim system repeatedly attempts to prevent the UDP packet from responding in an attempt to match each datagram technique to a program, but it is unable to do so and will eventually wear out and fail.
- **HTTP Flood**: This attack uses several, seemingly genuine HTTP GET or POST requests to target an application or web server. These inquiries are frequently made to help criminals avoid being discovered by learning crucial details about their intended victims before an attack.
- **Ping Flood and Ping of Death**: Another typical flood attack exploits many ICMP echo queries. The target system attempts to react to numerous requests, eventually restricting its network bandwidth because each ping given requires a cross-response that constitutes the same number of packets to be returned. Another variation of this attack known as "ping of death" causes the operating system to crash by having the victim send ping packets with the incorrect format and shape.
- **SYN Flood**: Three-way communication between two systems is necessary for every TCP session. Using an SYN flood, the attacker rapidly overwhelms the victim with connection requests, so numerous that it can no longer handle them, causing network saturation. This occurs when the host sends a large number of TCP/SYN packets with a forged sender address. Each of these packets functions as a connection request, causing the server to maintain several open half connections.

Objective:

Ultimately, the goal of this project is to enhance the prediction and prevention of DDoS attacks, contributing to more resilient and secure network infrastructures. Through this work, we aim to provide a foundation for future advancements in the development of sophisticated DDoS mitigation strategies .

Dataset:

The CICDDoS2019 dataset is a comprehensive collection of network traffic data specifically designed for the analysis and detection of Distributed Denial ofService (DDoS) attacks. Created by the Canadian Institute for Cybersecurity, this dataset serves as a valuable resource for researchers and practitioners in the field of cybersecurity.

Dataset Link: https://www.kaggle.com/datasets/aymenabb/ddos-evaluation-dataset-cic-ddos2019

Methodology:

The methodology for the project "Predict and Prevent DDoS Attacks Using Machine Learning and Statistical Algorithms" consists of several key steps, including data collection, preprocessing, feature selection, model training, evaluation, and implementation. The following outlines the approach in detail:

- Data collection
- Data preprocessing
- Feature selection
- Model development
- Model evaluation
- Comparison and Analysis
- Implementation
- Testing and Validation
- Documentation and Reporting

Machine Learning Data Process:

Predicting and preventing DDoS (Distributed Denial of Service) attacks using machine learning (ML) and statistical algorithms involves a structured data processing pipeline. This process starts with data collection, where network traffic data, logs, and historical attack data are gathered from various sources like routers, firewalls, and servers.

Next, in data preprocessing, the data is cleaned, normalized, and any missing values are handled to ensure completeness. Feature extraction follows, where key metrics such as traffic volume, patterns, anomalies, and statistical features like mean, variance are calculated. For model training, both supervised and unsupervised learning techniques are employed, including Support Vector Machines (SVM), Neural Networks, K-Means Clustering, and Autoencoders, to detect and predict DDoS attacks. The model is then evaluated using cross-validation and performance metrics like accuracy, precision, recall, F1-score, and ROC-AUC.

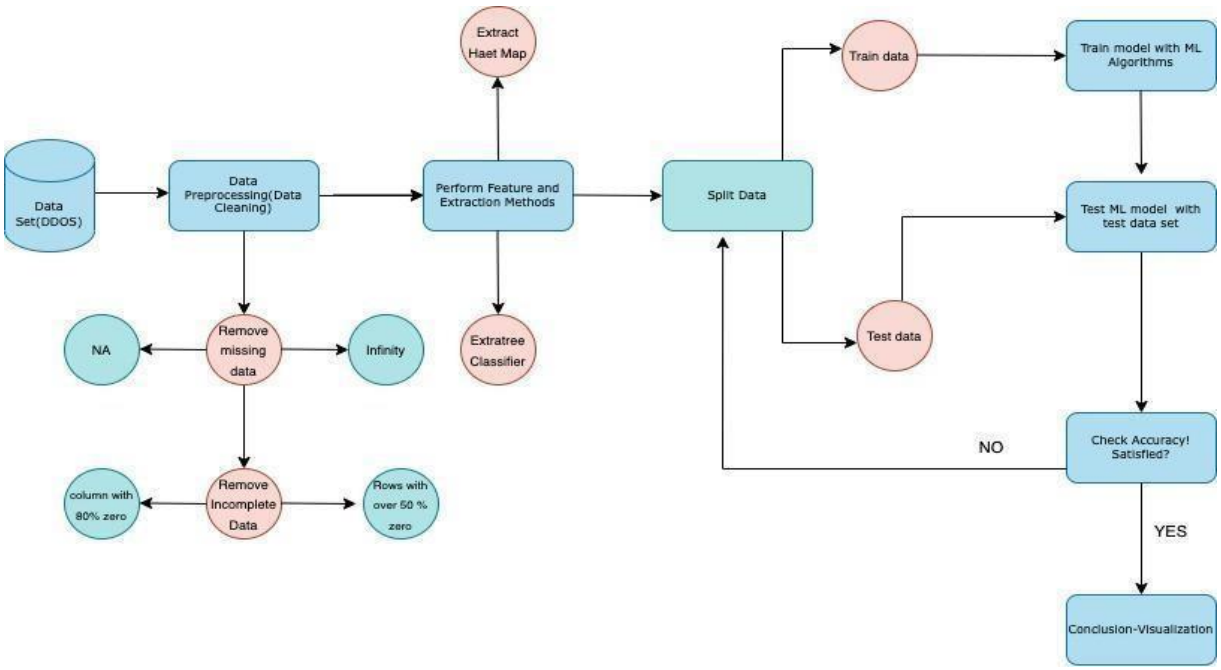


Fig 2: ML Data Process

Upon successful training and evaluation, the model is deployed for real-time monitoring, setting up alert systems to notify administrators of potential attacks, and integrating automated mitigation measures such as rate limiting and traffic filtering.

Visualizations:

Visualization is the process of representing data or information graphically, allowing for better understanding and interpretation of complex datasets. It involves creating visual elements such as charts, graphs, maps, and diagrams to communicate data insights clearly and effectively. Visualization is a crucial part of data analysis, as it helps to identify patterns, trends, and anomalies that might not be immediately apparent in raw data.

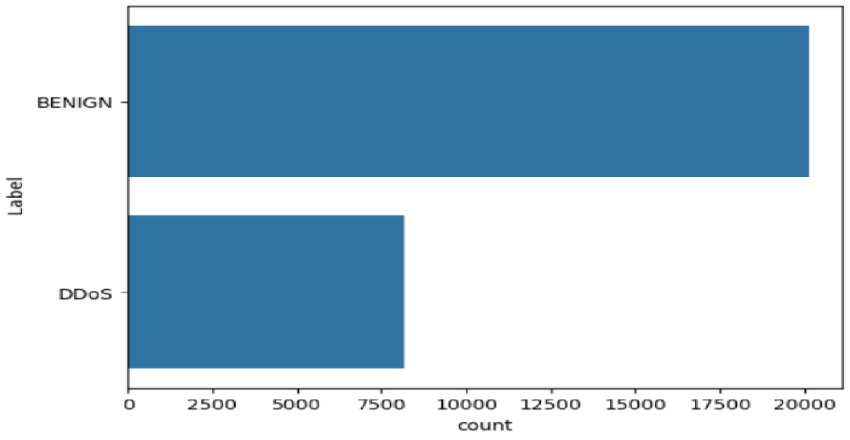
Python Libraries

- **Matplotlib:** A basic plotting library for creating static, interactive, and animated visualizations.
- **Seaborn:** Built on Matplotlib, it provides a high-level interface for drawing attractive statistical graphics.

Count of Benign vs. DDoS Traffic Instances

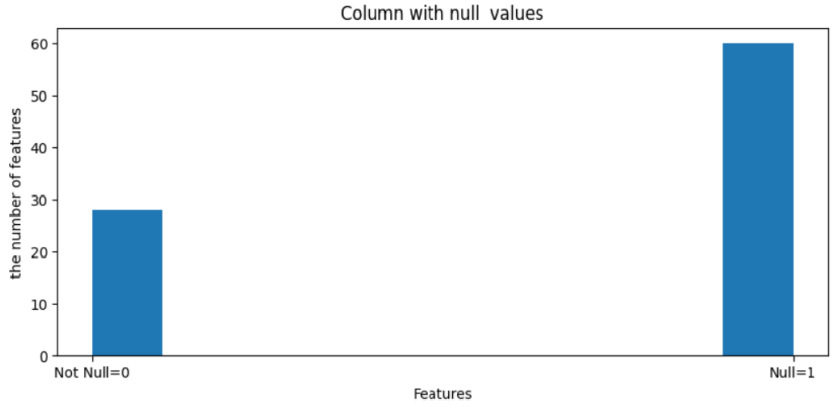
X-axis representing the unique values in the ' Label' column ('BENIGN', 'DDoS') and the y-axis representing the count of each unique value. Each bar's height indicates the number of occurrences of each label.

The plot shows the distribution of different types of network traffic (benign traffic versus DDoS attacks) in the dataset.



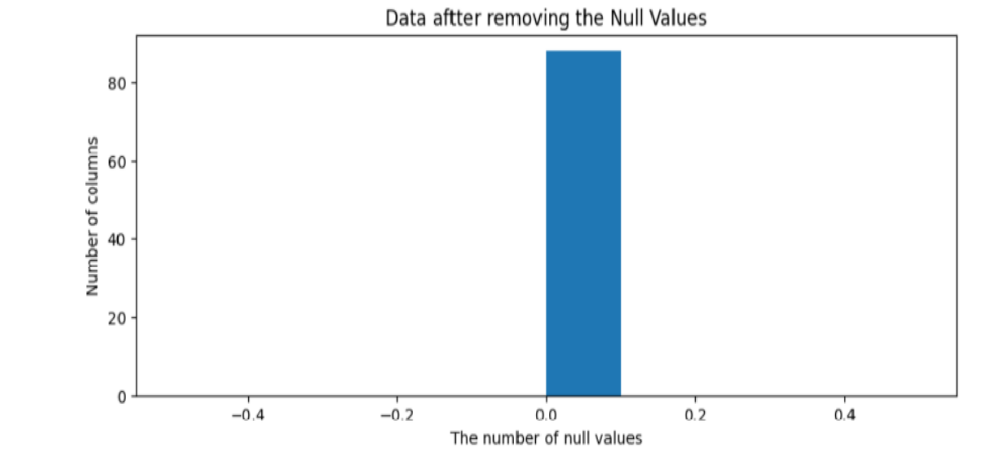
Distribution of Null Values Across Dataset Columns

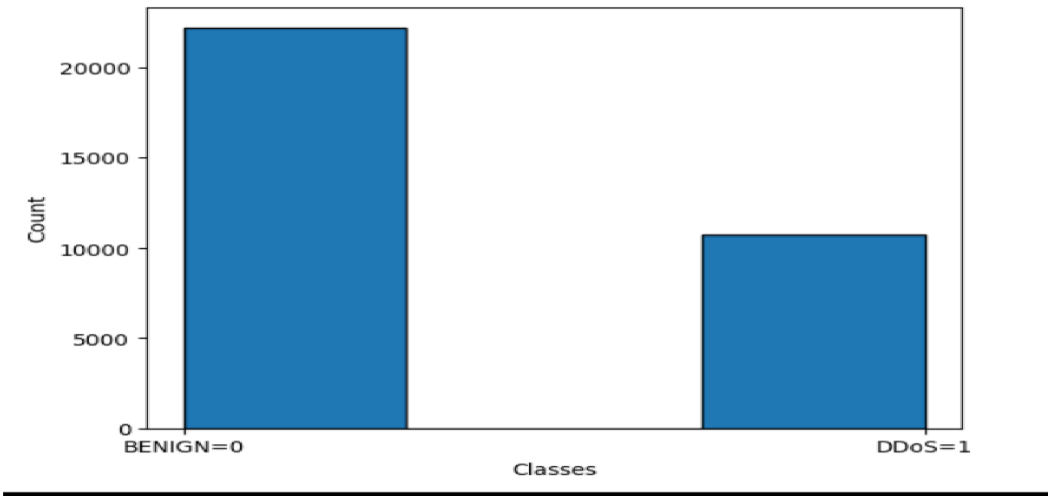
Histogram showing the distribution of the count of null values across all columns in the DataFrame. The x-axis represents the number of null values in the columns, and the y-axis represents the number of columns with that count of null values.



Distribution of Null Values in Dataset Columns After Cleaning

Histogram showing the distribution of null values across all columns in the DataFrame data_fafter handling null values (e.g., removing or imputing them). The x-axis represents the numberof null values in the columns, and the y-axis represents the number of columns with that countof null values.





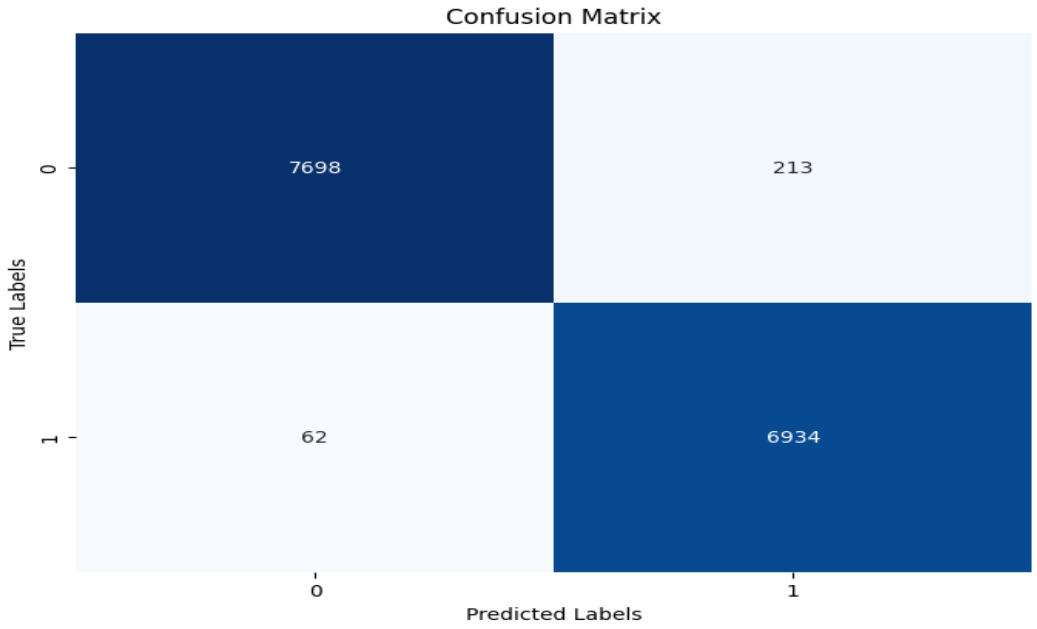
Confusion matrix:

A confusion matrix table is used to describe the performance of a classification system. The effectiveness of the classification was represented and summarized using a confusion matrix.

- **TP:** You predicted positive, and it's true
- **FP(Type 1 Error):** You predicted positive, and it's false
- **TN:** You predicted negative, and it's true
- **FN(Type 2 Error):** You predicted negative, and it's false

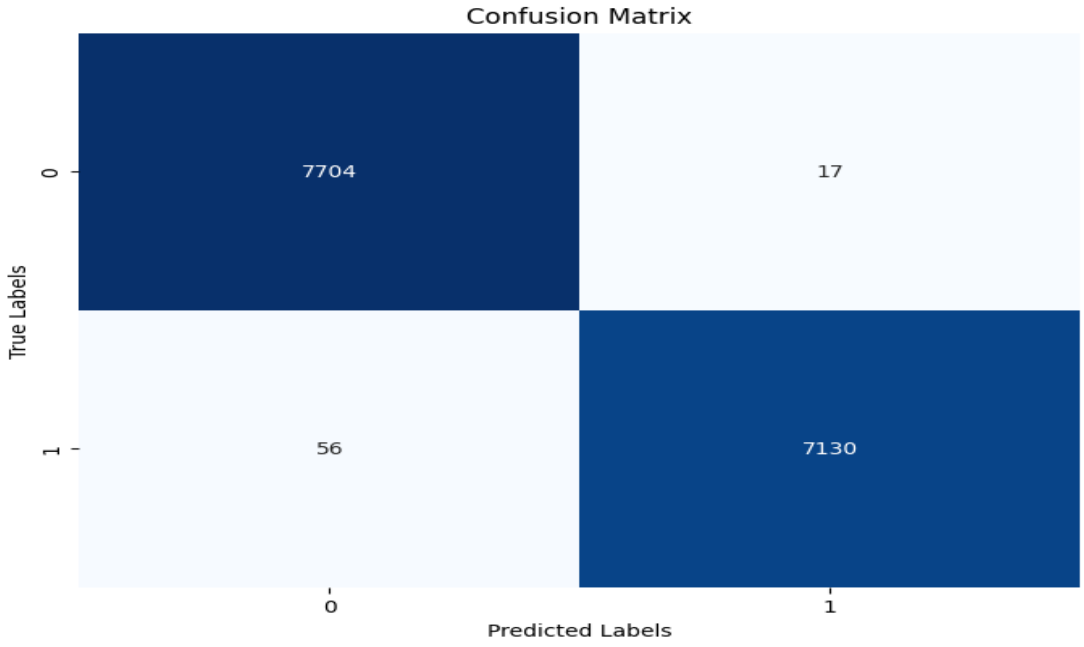
Confusion Matrix Heatmap of Logistic Regression classifier:

Heatmap shows the performance of the classification model by comparing the true and predicted labels.



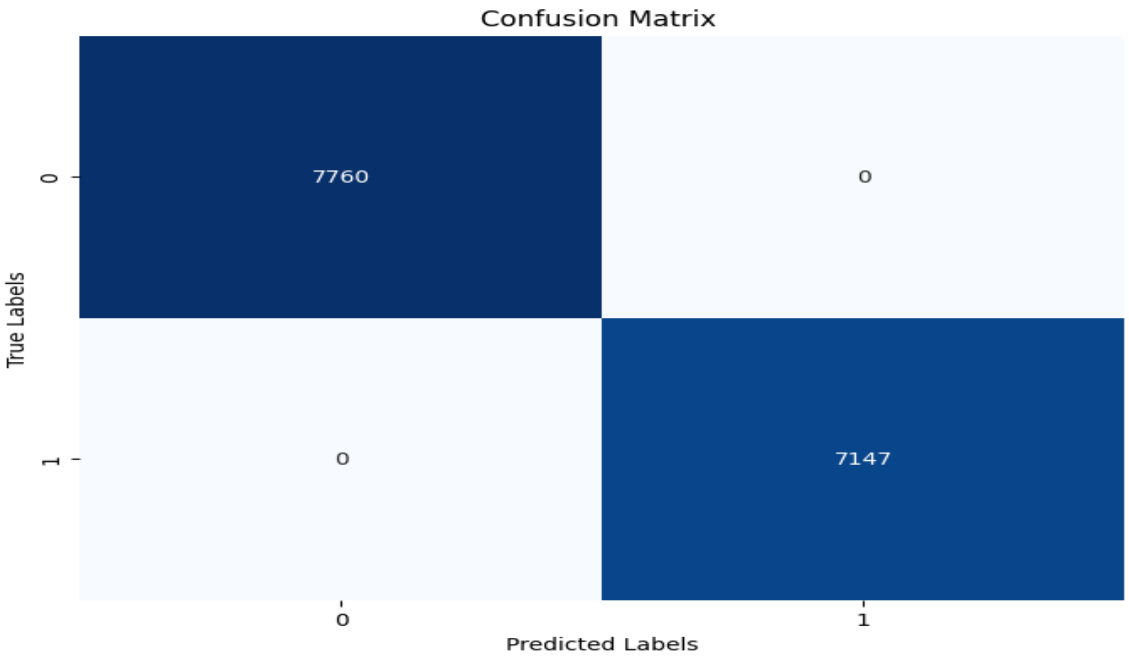
Confusion Matrix Heatmap of KNeighbors Classifier:

Heatmap shows the performance of the classification model by comparing the true and predicted labels.



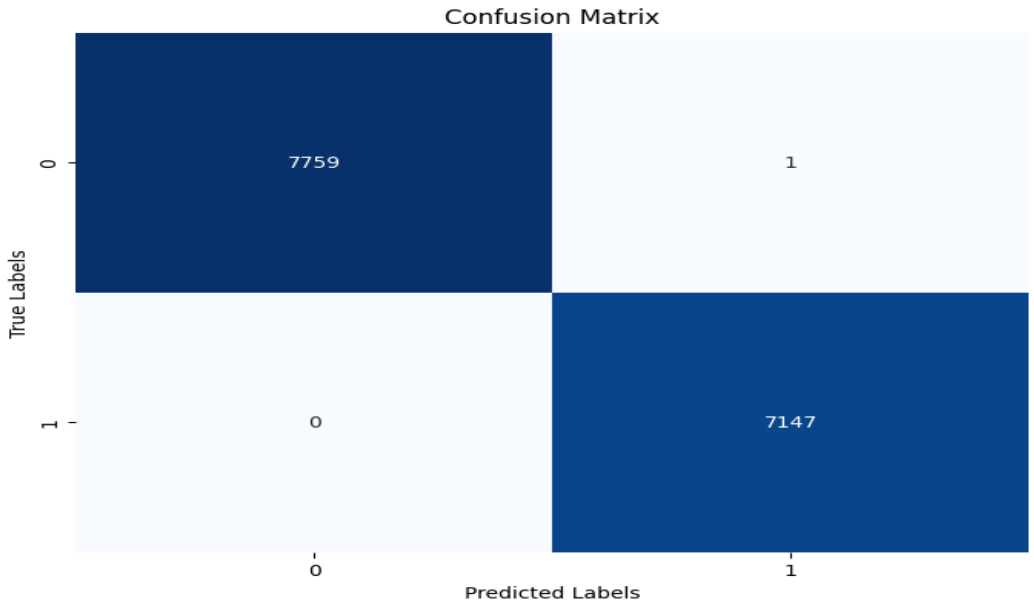
Confusion Matrix Heatmap of SVM Classifier:

Heatmap shows the performance of the classification model by comparing the true and predicted labels.



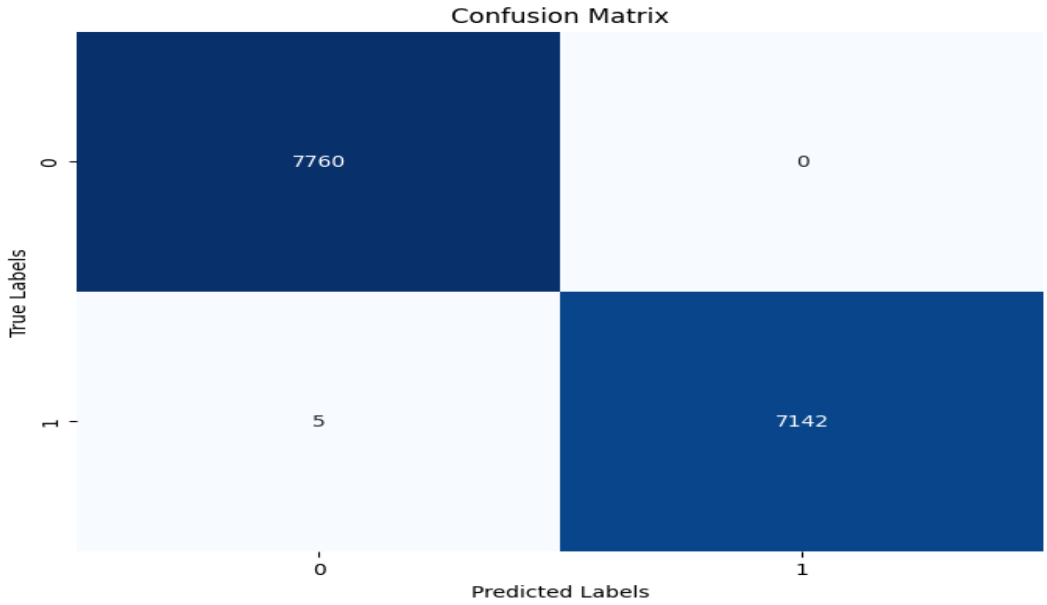
Confusion Matrix Heatmap of Decision Tree Classifier:

Heatmap shows the performance of the classification model by comparing thetrue and predicted labels.



Confusion Matrix Heatmap of Naïve Bayes Classifier:

Heatmap shows the performance of the classification model by comparing thetrue and predicted labels.



Model Evaluation:

Accuracy: Evaluation of a model's performance in a dataset to find relationships and patterns based on input data, also known as training data.

Recall: Recall is determined as the proportion of positive samples correctly identified as positive for all positive samples.

$$Recall = \frac{TP}{TP + FN}$$

F1-Score: The harmonic mean of recall and precision is known as the F1-score. The following formula combines the recall and precision into a single formula:

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

Precision: Metrics such as precision and recall allow us to assess how well a classification model predicts outcomes for a given class of interest, or "positive class.". While recall measures the degree of the error caused by false negatives (FNs), precision measures the degree of the error caused by False Positives (FPs).

$$Precision = \frac{TP}{TP + FP}$$

Results:

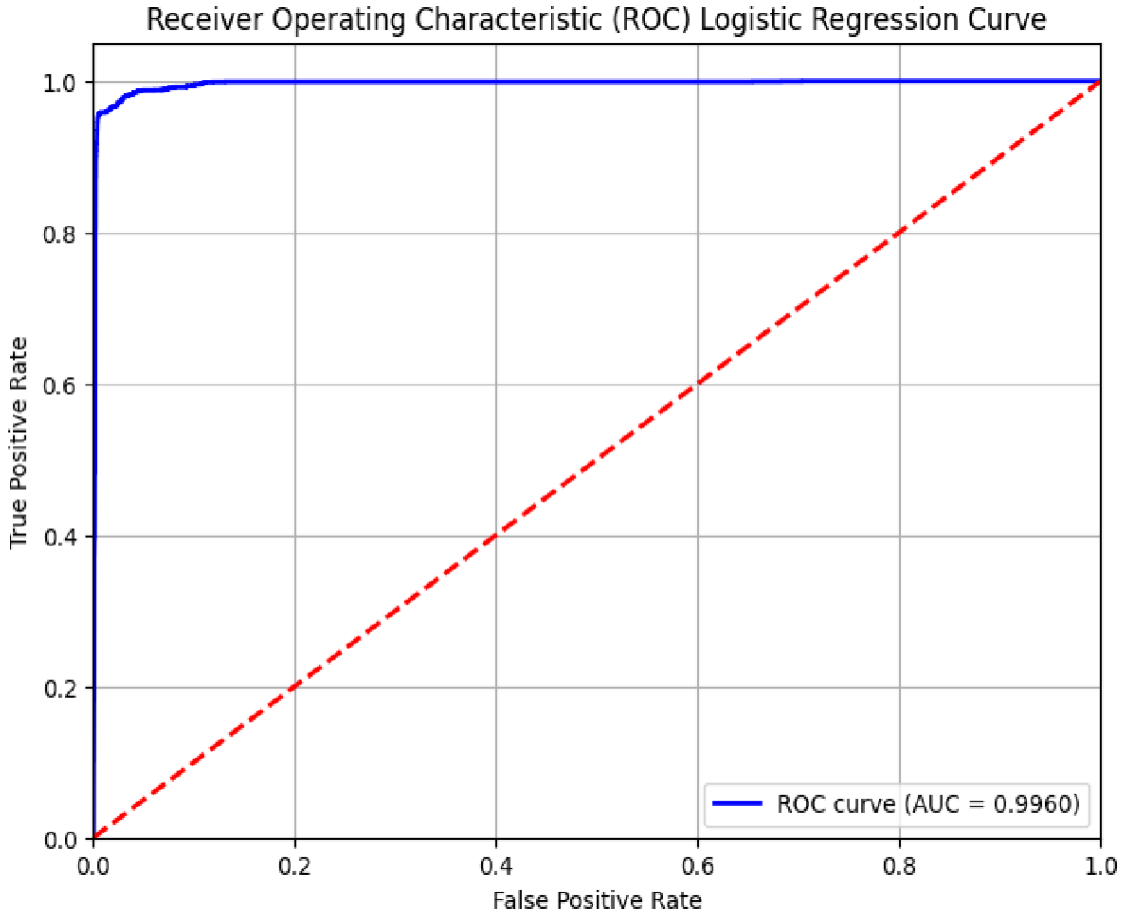
Model Performance

Logistic Regression:

Logistic Regression is a supervised machine learning algorithm that is used for solving classification tasks where the goal is to predict the probability that an instance belongs to class or not. It is a statistical algorithm which analyses the relationship between two data factors.

- Accuracy : 98.15%
- Precision : 99%
- Recall : 97%
- F1-Score : 98%

ROC curve for Logistic Regression:

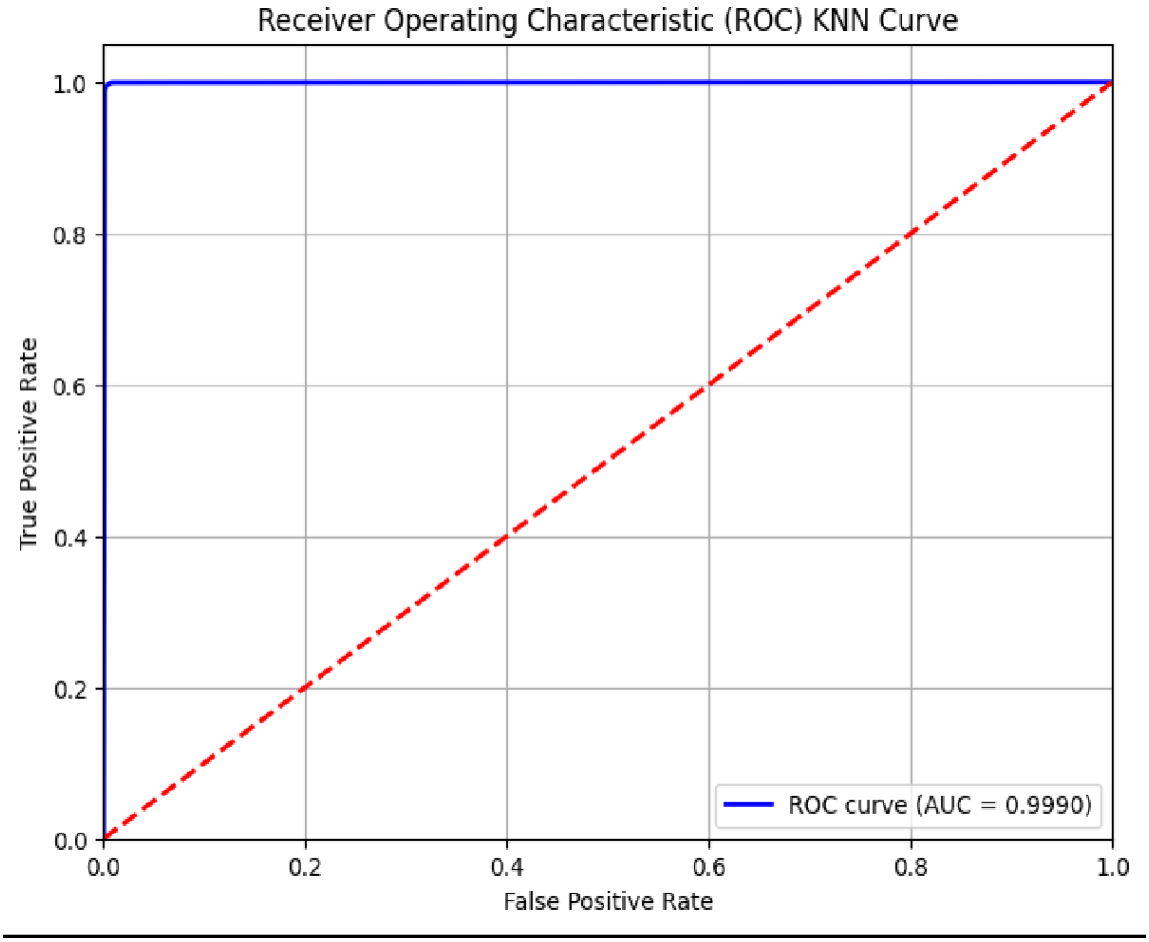


K-Nearest Neighbour(KNN):

K-Nearest Neighbour is a supervised machine learning algorithm technique. KNN algorithm assumes the similarity between the new case and available cases and put the new case into the category that is most similar to the available categories. KNN algorithm can be used for Regression as well as for Classification but mostly used for the Classification problems.

- Accuracy : 99.51%
- Precision : 99%
- Recall : 100%
- F1-Score :100%

ROC curve for KNN:

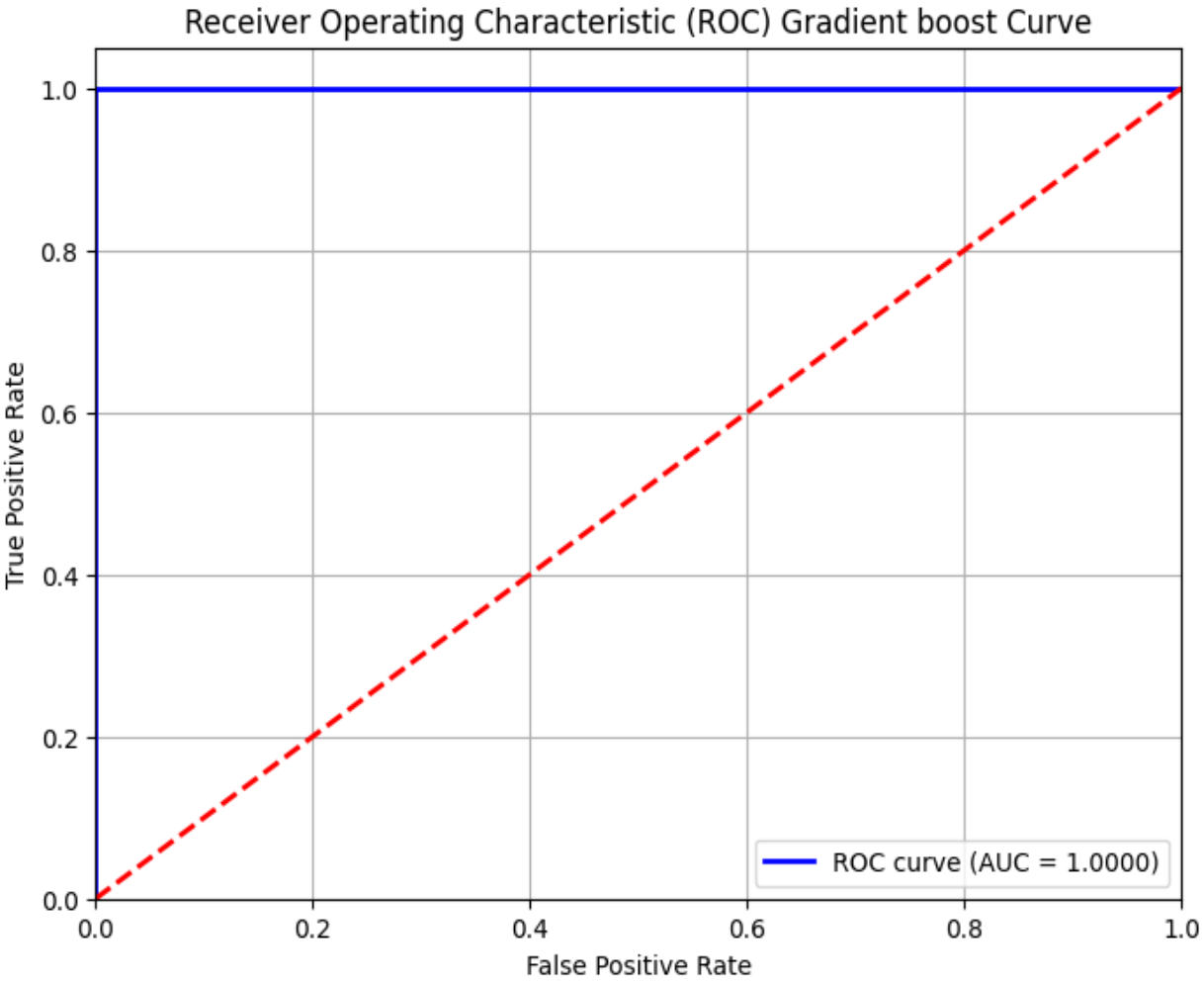


Gradient Boosting:

Gradient boosting is a popular boosting algorithm which is used for classification and regression tasks. Boosting is one kind of ensemble learning method which trains the model sequentially and each new model tries to correct the previous model.

- Accuracy : 99.99%
- Precision :100%
- Recall : 100%
- F1-Score :100%

ROC curve for the Gradient Boosting:



Artificial Neural Network (ANN):

Artificial Neural Network is a computational model based on the biological neural networks of animal brains. Artificial Neural Networks contain artificial neurons which are called units. These units are arranged in a series of layers that together constitute the whole Artificial Neural Network in a system.

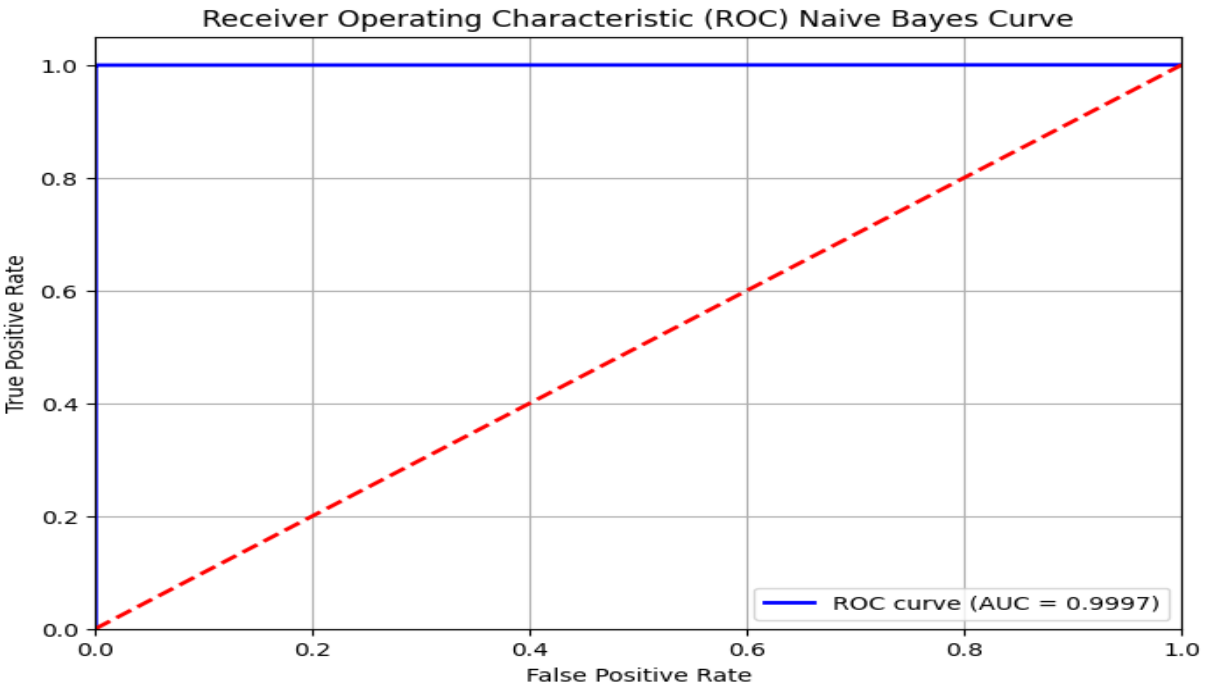
- Accuracy : 99.97%

Naïve bayes:

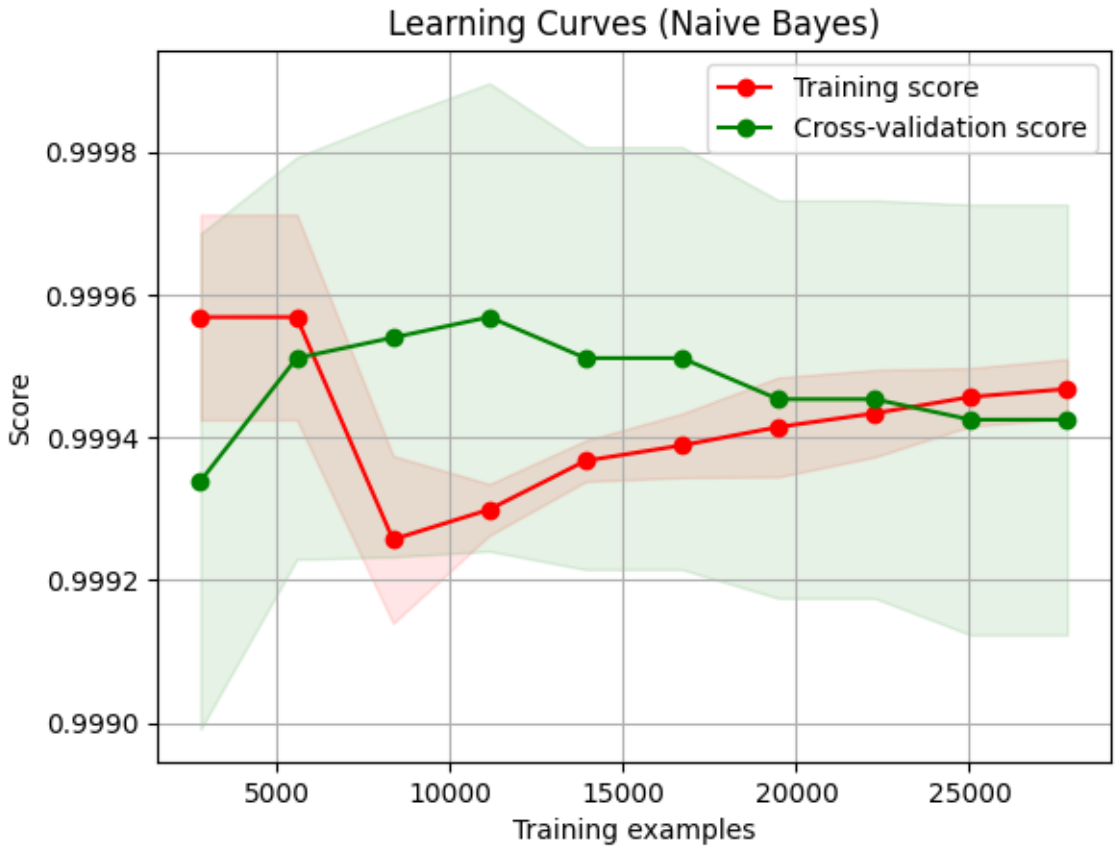
Naïve Bayes is an supervised machine learning algorithm. It is mainly used for the classification problems. It is highly used in text classification tasks. It is based on bayes theorem. It is a probabilistic classifier, which means that it predicts on the basis of the probability of an object.

- Accuracy : 99.99%
- Precision : 100%
- Recall : 100%
- F1-Score : 100%

ROC curve for Naïve Bayes:



Learning curve for naïve bayes:



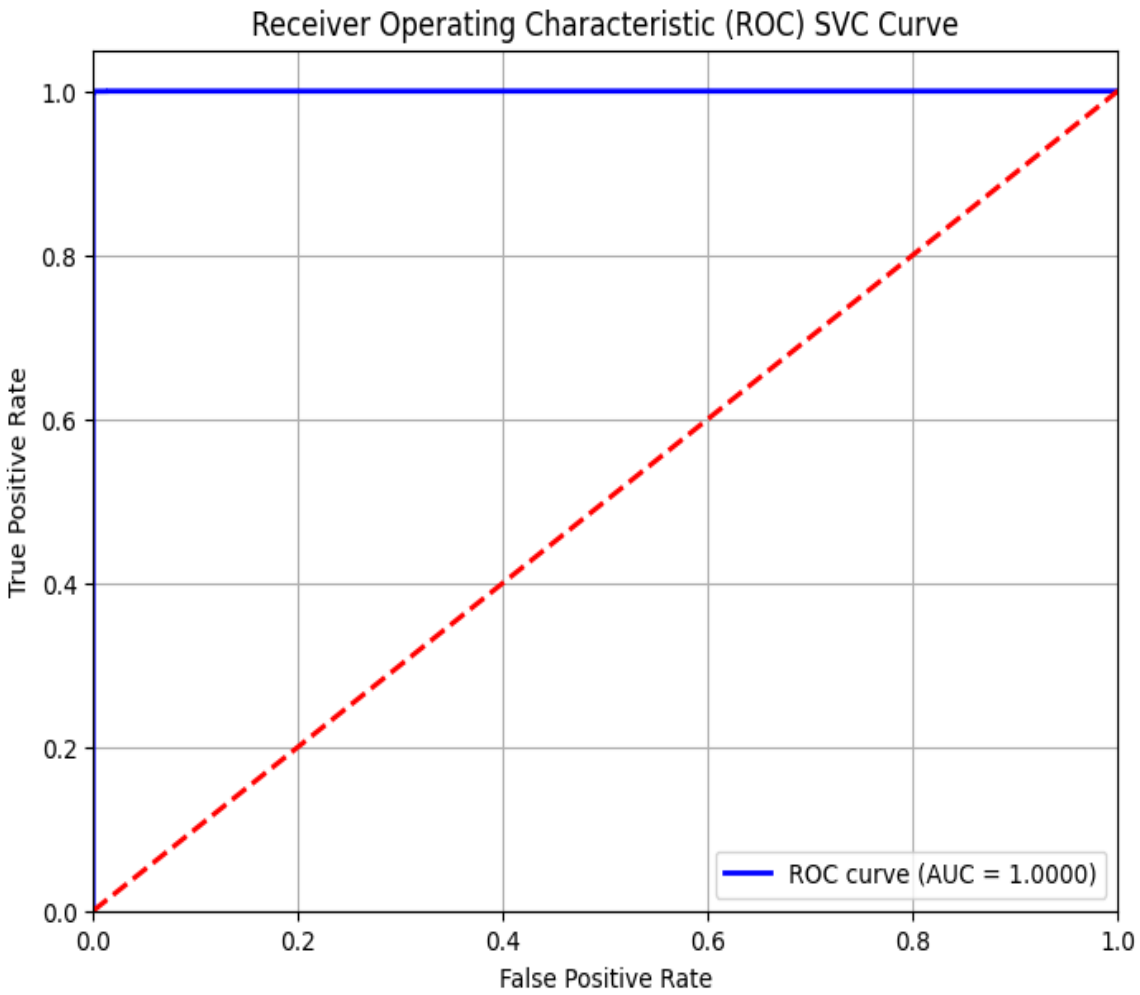
Support Vector Classifier (SVC):

Support Vector Classifier is a machine learning algorithm that's a specific implementation of the support vector machine algorithm.

SVMs are used for the classification, regression, and outlier detection tasks. SVC is designed for classification tasks, and it seeks to find the hyperplane that best separates data points into different classes.

- Accuracy : 99.96%
- Precision : 100%
- Recall : 100%
- F1-Score : 100%

ROC curve for SVC:

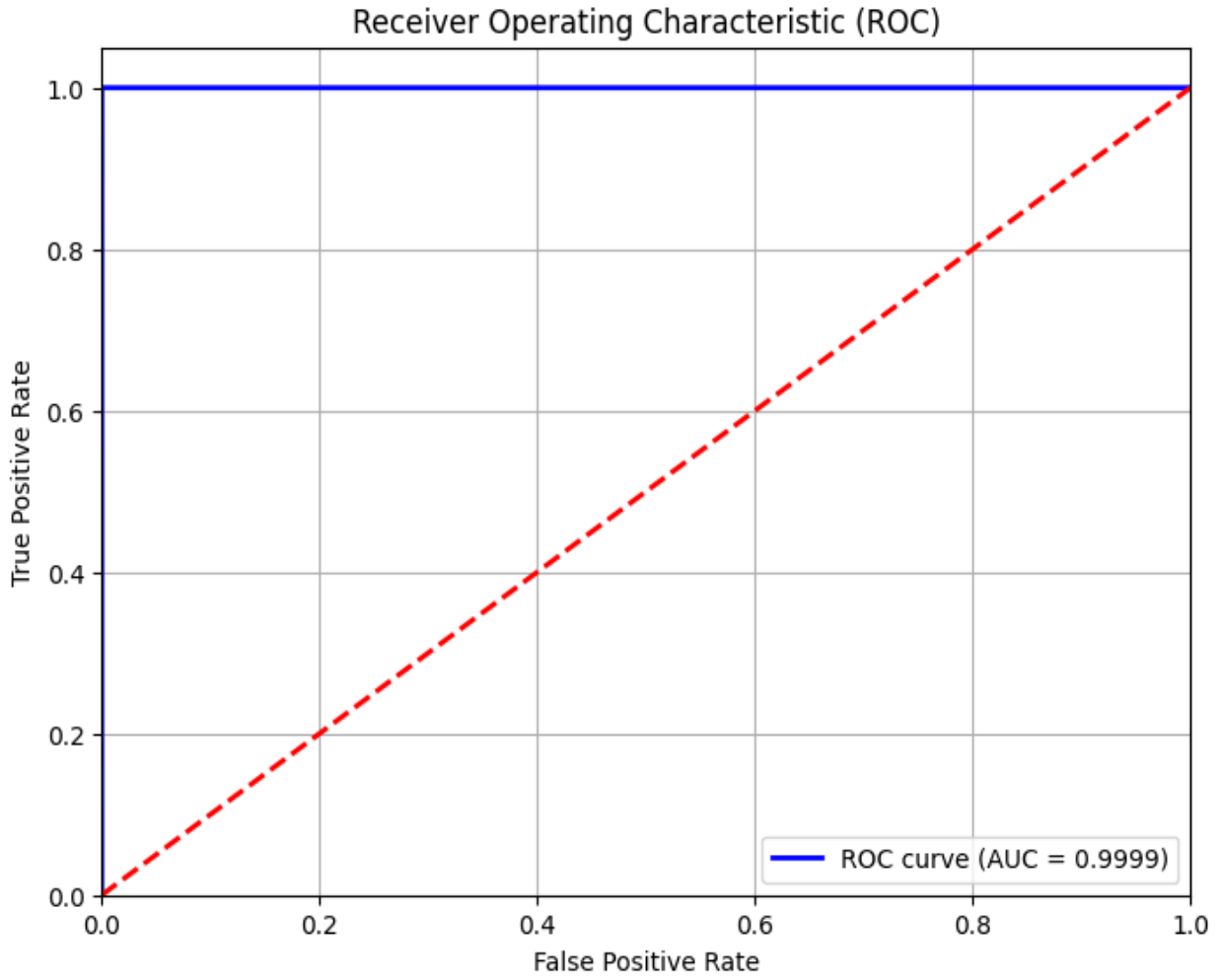


Decision Tree Classifier:

Decision Tree is a supervised machine learning algorithm. It is used both for classification and regression tasks. They provide a clear way to make decisions based on data by modeling the relationship between different variables. A decision tree is a flowchart like structure used to make decisions and predictions.

- Accuracy : 99.99%
- Precision : 100%
- Recall : 100%
- F1-Score : 100%

ROC curve for the Decision Tree:



ACKNOWLEDGMENT:

I would like to express my sincere gratitude to my advisor, Dr. Narendran Rajagopalan , Associate Professor , for their invaluable guidance and support throughout 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.