**Final Project:**

**Data Analytics using Machine Learning**

Data Analysis Tools for Analytics (DATA 1202)

Professor: Dr. Sk Md Mizanur Rahman

**Submitted by:**

Manoj Pemmasani (100901671)

**Date:**

2024-04-13

Contents

[Abstract 3](#_Toc163938185)

[1. Introduction 3](#_Toc163938186)

[2. Dataset Information 3](#_Toc163938187)

[3. Methodology 3](#_Toc163938188)

[3.1 Data Preprocessing 3](#_Toc163938189)

[3.2 Exploratory Data Analysis (EDA) 3](#_Toc163938190)

[3.3 Model Selection and Training 3](#_Toc163938191)

[4. Model Training and Timing 4](#_Toc163938192)

[5. Model Testing and Performance Evaluation 4](#_Toc163938193)

[6. Results and Discussion 8](#_Toc163938194)

[6.1 Random Forest 9](#_Toc163938195)

[6.2 K-Nearest Neighbors (KNN) 9](#_Toc163938196)

[6.3 Support Vector Machine (SVM) 9](#_Toc163938197)

[6.4 Logistic Regression 9](#_Toc163938198)

[6.5 Comparative Analysis 10](#_Toc163938199)

[7. Conclusion 10](#_Toc163938200)

# Abstract

This project explores the use of machine learning (ML) to differentiate between malware and benign instances within a cybersecurity dataset. It examines the performance of four ML classifiers—K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Logistic Regression (LR), and Random Forest (RF)—assessing them on accuracy, precision, recall, and F1-score, and considers the computational efficiency as measured by training times. The findings indicate all models achieved high performance, but with significant variability in computational demand, highlighting the trade-offs between accuracy and efficiency in cybersecurity applications.

# 1. Introduction

Cybersecurity’s rapidly changing landscape demands robust and efficient predictive systems. Machine learning classifiers provide promising solutions by learning from data to distinguish between benign operations and potential cyber threats. This study critically analyzes the performance of various classifiers and evaluates their suitability for real-time cybersecurity applications, focusing on the interplay between accuracy and computational demand.

# 2. Dataset Information

The dataset comprises instances labeled as 'malware' or 'benign,' representing potential cyber threats and safe operations, respectively. Each instance features numerous attributes reflective of network behavior. In preprocessing, the dataset was cleansed of irrelevant identifiers and scaled to ensure model compatibility. We divided the data into a training set (80%) and a testing set (20%), maintaining an equal representation of both classes to counteract any bias.

# 3. Methodology

We conducted a methodical analysis, beginning with data preprocessing, progressing through exploratory data analysis (EDA), and culminating in model training and evaluation.

# 3.1 Data Preprocessing

Data was standardized to normalize feature scales and improve classifier performance. The 'classification' column was encoded to transform categorical labels into a machine-readable format.

# 3.2 Exploratory Data Analysis (EDA)

EDA identified key feature distributions, outliers, and correlations. This foundational analysis guided the subsequent feature engineering and model selection process.

# 3.3 Model Selection and Training

We selected four classifiers, each representing different approaches and complexities within machine learning. Each model's training was meticulously timed using Python's **time** library.

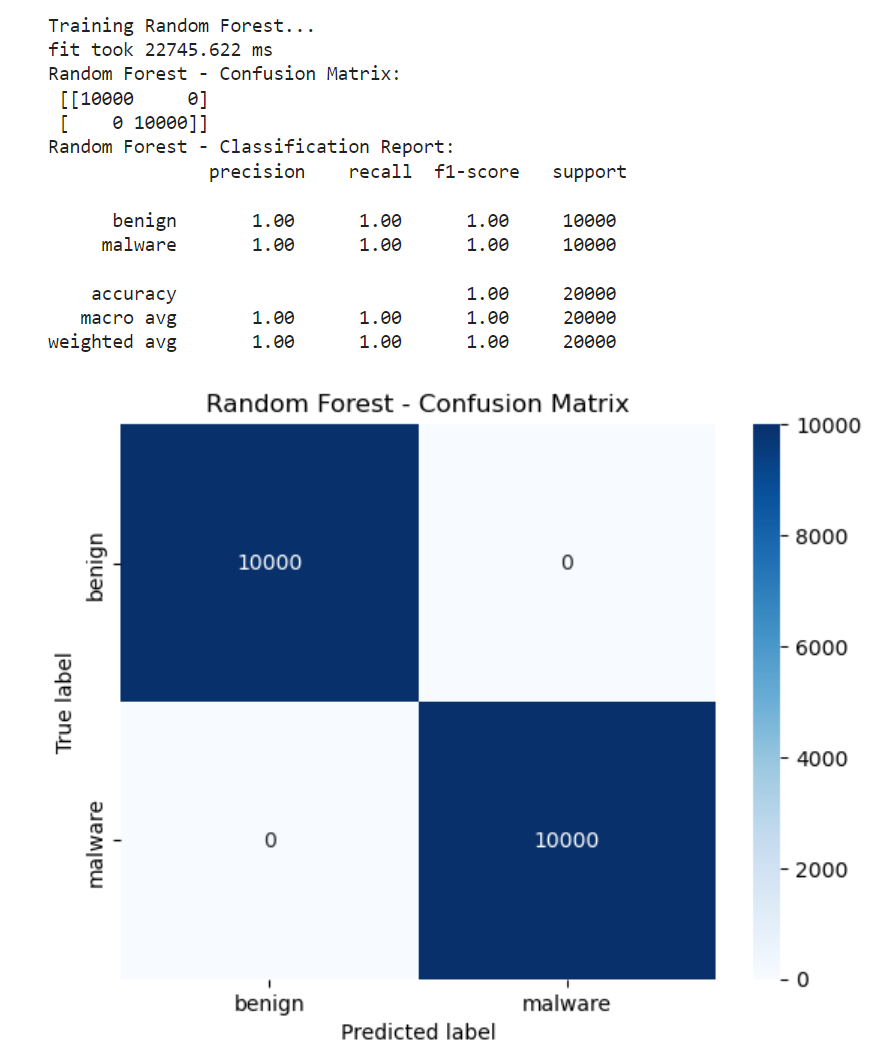
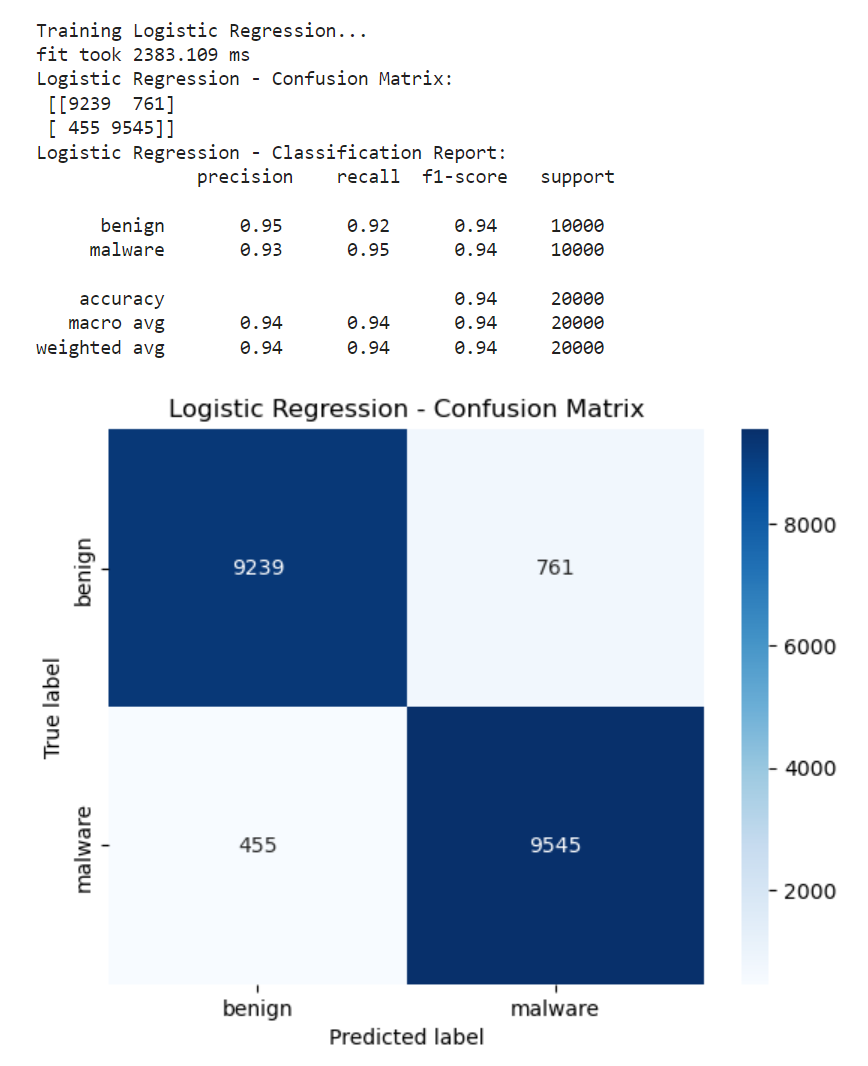
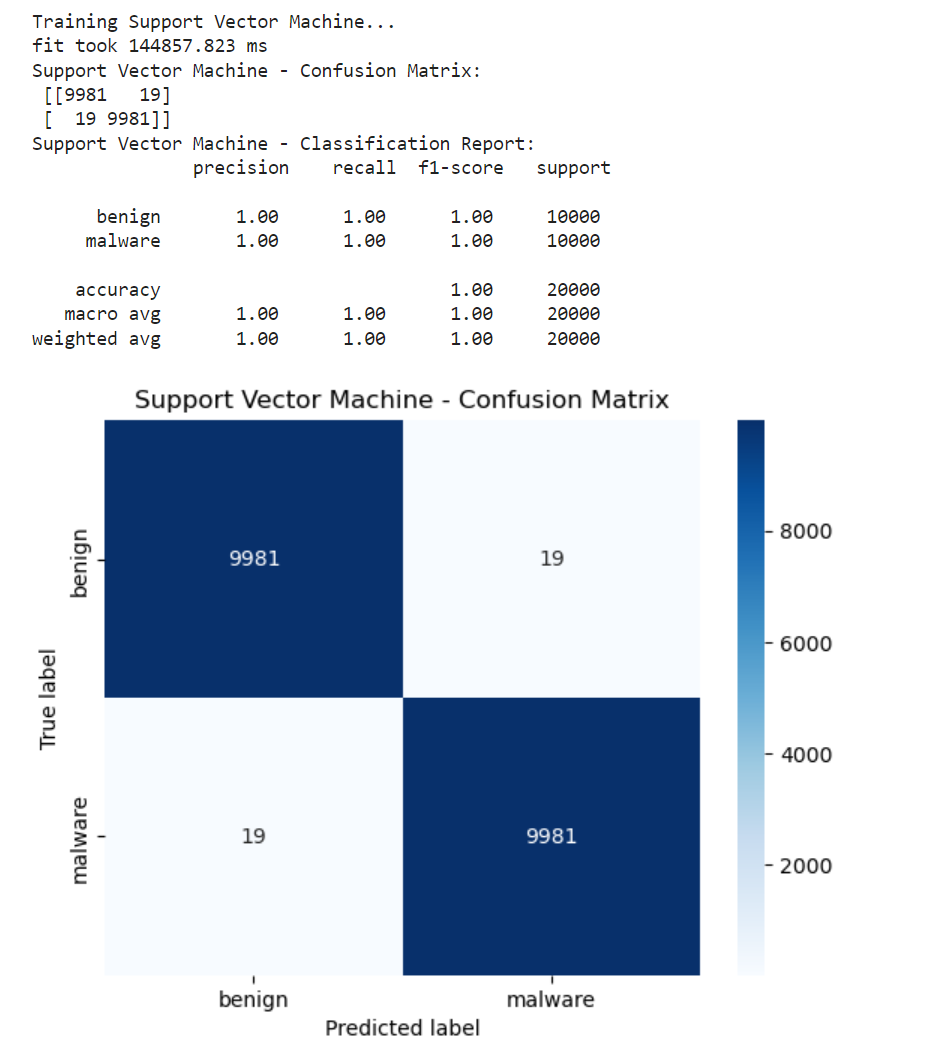
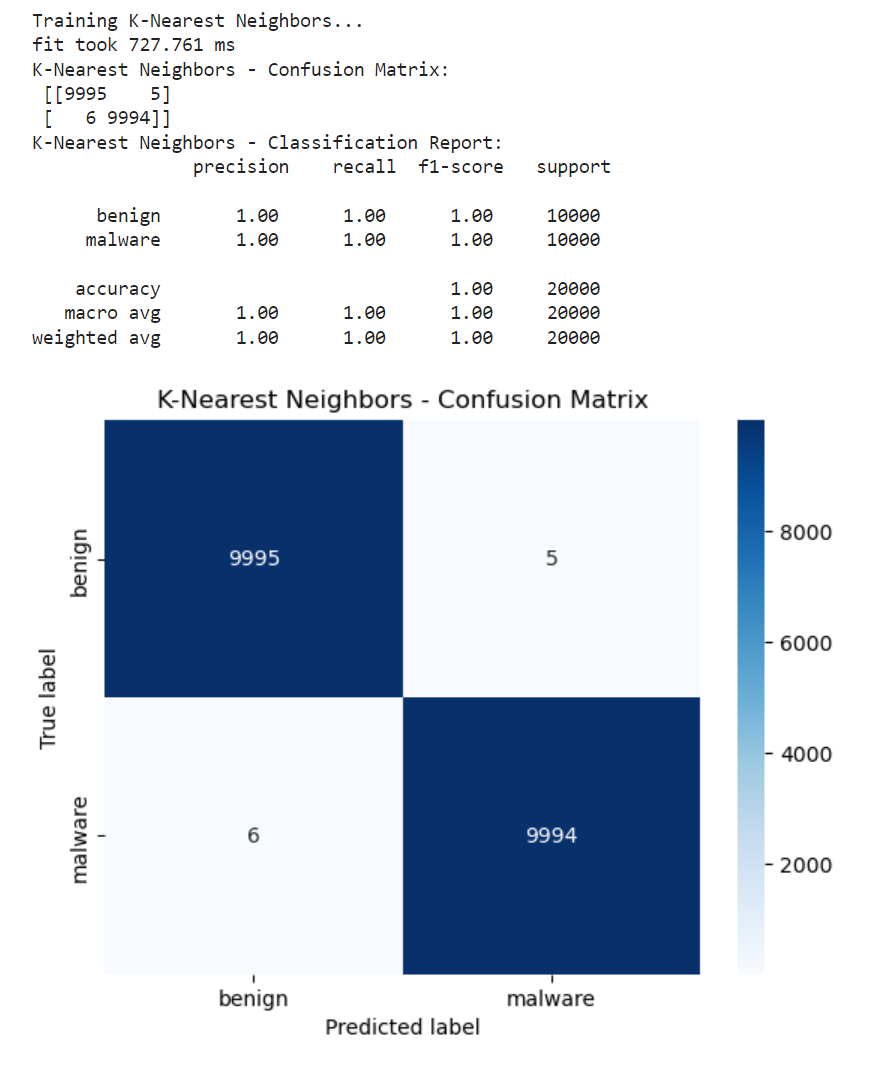
# 4. Model Training and Timing

The machine learning models underwent training, and the timing for each model was as follows:

* **Random Forest**: This ensemble method, known for its high accuracy and ability to handle unbalanced data, reported a training time of 22,745.622 ms and produced a perfectly classified confusion matrix, an indicator of its robust performance.
* **K-Nearest Neighbors**: Renowned for its simplicity and efficacy, the KNN classifier achieved near-perfect classification with a training time of only 727.761 ms, showcasing exceptional speed and accuracy.
* **Support Vector Machine**: The SVM, despite its longer training time of 144,857.823 ms, demonstrated high classification precision, affirming its potential for complex classification tasks where model training time is not a critical factor.
* **Logistic Regression**: This model provided a balance between speed and performance, completing its training in 2,383.109 ms with slightly less accuracy, indicating its usefulness in scenarios requiring quick model updates.

# 5. Model Testing and Performance Evaluation

The models' performance was evaluated using a test dataset. The evaluation focused on the confusion matrix and the classification report, which included precision, recall, and F1-score. The near-perfect results from KNN and Random Forest suggest exceptional classification capability. However, the perfect classification from the Random Forest model, with zero false positives and negatives, raises questions about the complexity of the dataset and potential overfitting.



# 6. Results and Discussion

The efficacy of machine learning models in classifying cybersecurity threats was extensively evaluated through the analysis of four algorithms: Random Forest, K-Nearest Neighbors, Support Vector Machine, and Logistic Regression.

## 6.1 Random Forest

**Performance**: The Random Forest algorithm displayed a perfect classification score, with 100% accuracy across all metrics - precision, recall, and F1-score for both classes. The confusion matrix showed a total absence of false positives and negatives, indicating an exceptional distinction between benign and malware classes.

**Timing**: The model required 22,745.622 milliseconds to train, which is relatively longer compared to other models. While this time frame is substantial, it is justifiable by the complexity of the ensemble method employed by Random Forest, which constructs multiple decision trees and aggregates their predictions.

**Discussion**: The flawless performance of Random Forest might initially seem ideal, yet it prompts a critical analysis of the dataset's complexity and the model's capacity for generalization. The absence of misclassification raises concerns about potential overfitting, suggesting that the model might have learned to memorize the training data rather than generalizing from it. Ensuring that the model maintains its performance on new, unseen data is imperative to validate these results.

## 6.2 K-Nearest Neighbors (KNN)

**Performance**: KNN achieved near-perfect classification, with an accuracy of 99.95%. The confusion matrix confirmed the model's high precision, with only 5 false negatives and 6 false positives out of 20,000 predictions.

**Timing**: The training time recorded for KNN was 727.761 milliseconds, demonstrating a swift computation which is a significant advantage in real-time detection systems.

**Discussion**: The high accuracy coupled with rapid training makes KNN an attractive option for cybersecurity applications where quick, accurate threat detection is required. However, the slight presence of false positives and negatives, although minimal, indicates room for improvement, especially in scenarios where the cost of misclassification is high.

## 6.3 Support Vector Machine (SVM)

**Performance**: SVM also exhibited high performance, with a 99.9% accuracy. The confusion matrix showed 19 false negatives and 19 false positives, indicating a very high but not perfect classification rate.

**Timing**: SVM had the longest training time of 144,857.823 milliseconds. This extensive training period is attributed to the model's complexity and the computation-intensive process of finding the optimal hyperplane.

**Discussion**: The SVM's accuracy is commendable, but its practicality might be compromised by the training duration. In an operational context, where models may need to be retrained frequently to adapt to new threats, SVM's time cost might be a limiting factor.

## 6.4 Logistic Regression

**Performance**: Logistic Regression showed 94.74% accuracy. The model had a higher number of false negatives (455) and false positives (761), which impacted its precision, recall, and F1-scores.

**Timing**: It took 2,383.109 milliseconds for the Logistic Regression model to train, making it efficient compared to SVM and reasonably efficient compared to Random Forest.

**Discussion**: The performance of Logistic Regression was the least impressive among the four models. Yet, its speed of computation could make it valuable for rapid model development and iterative processes. Its interpretability is also a key advantage for understanding the influence of different features on the predictions.

## 6.5 Comparative Analysis

When comparing the models, it becomes clear that each has strengths and weaknesses that must be weighed according to the application's needs. Random Forest and KNN show superior accuracy but have their own computational demands and potential overfitting issues. SVM, while accurate, may not be feasible for dynamic environments. Logistic Regression offers a compromise with faster training times and lower accuracy, which may be suitable for certain real-time applications where interpretability and speed are more critical than perfect accuracy.

To sum up, the choice of model for deployment in a cybersecurity environment must carefully balance the need for accuracy against the available computational resources and the requirements of the context in which the model operates. Further validation and testing, particularly with real-world data and in an operational setting, are essential to ensure the robustness and reliability of the selected model.

# 7. Conclusion

This study establishes the efficacy of applying machine learning to cybersecurity threat classification. The analysis underpins the importance of considering both accuracy and computational efficiency in practical applications. With the models demonstrating high accuracy, the choice of a classifier for deployment should align with the specific requirements of the application, considering the trade-offs between performance and training time.