Phishing Website Detection: An ML-Based Predictive Analysis.

-Ritik Mathpal

Table of Contents

1.Introduction

1.1. Background and Motivation

Phishing websites are a major cybersecurity threat, tricking users into revealing sensitive information like passwords and credit card details by imitating legitimate sites. These fraudulent sites contribute to identity theft, financial fraud, and data breaches. Traditional detection methods, such as blacklists and rule-based systems, are often ineffective against newly created phishing sites, which evolve rapidly to bypass security measures.

1.2. Project Objectives

In this project, **machine learning (ML) is used to classify websites as either phishing or legitimate** based on various features like URL structure, domain age, SSL certificate status, and webpage content. The goal is to build an accurate classification model that can automatically identify phishing websites by learning patterns from the data.

This project involves:

- Analyzing a phishing website dataset.
- Extracting and selecting relevant features.
- Training and evaluating different ML models.
- Comparing their performance using accuracy, precision, recall, and other metrics.

The results from this project will help in understanding which ML techniques are most effective for phishing detection and how website characteristics contribute to classification accuracy.

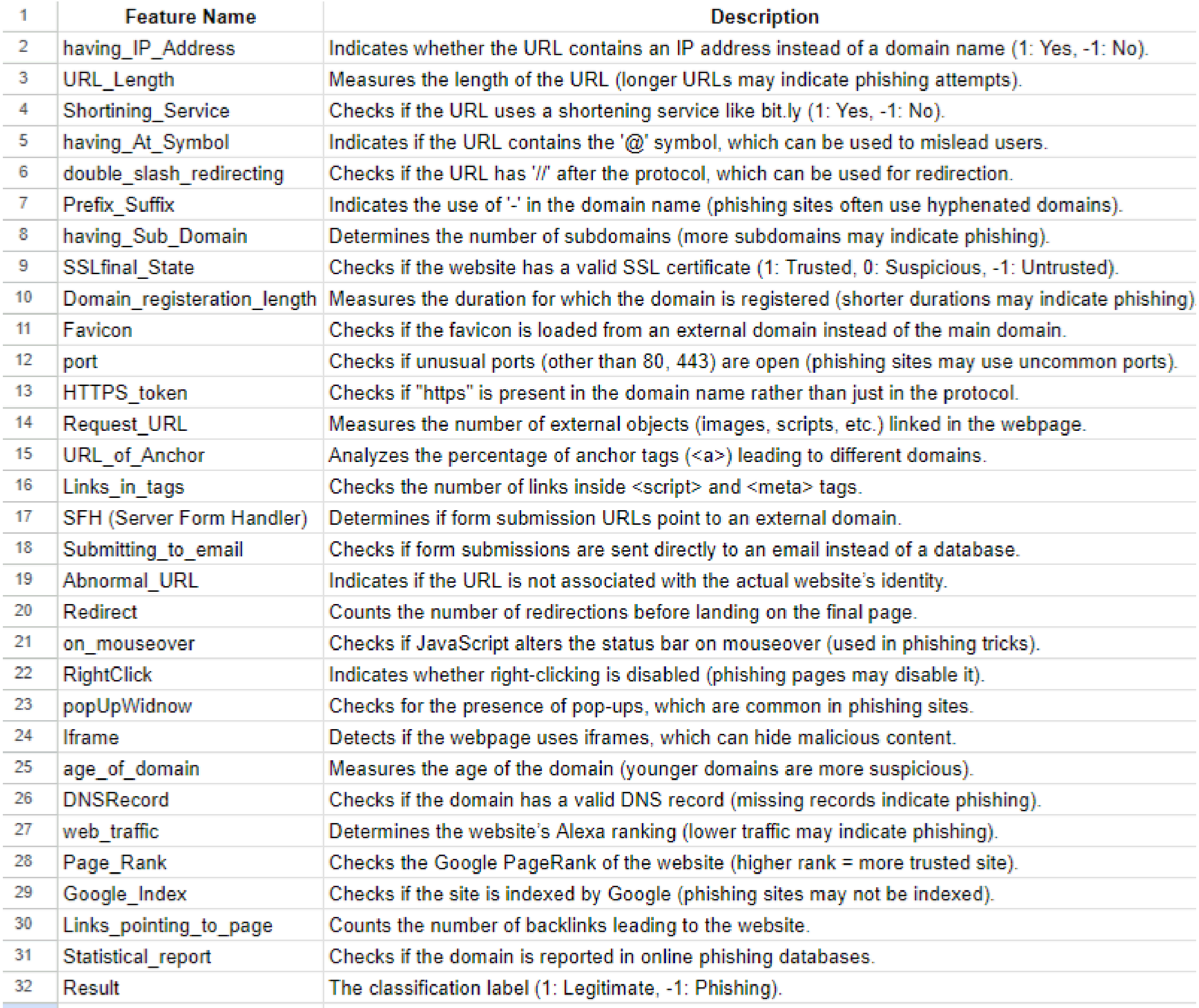
2.Dataset Description

2.1. Source of the Dataset

The dataset used in this project was obtained from UCI Machine Learning Repository. It consists of labeled website data, where each entry is classified as either **phishing** or **legitimate** based on various features.

2.2. Overview of Features

The dataset contains **11055** rows and **31** features, all being categorical variables. Some of the key features include:



1	Feature Name	Description
2	having_IP_Address	Indicates whether the URL contains an IP address instead of a domain name (1: Yes, -1: No).
3	URL_Length	Measures the length of the URL (longer URLs may indicate phishing attempts).
4	Shortining_Service	Checks if the URL uses a shortening service like bit.ly (1: Yes, -1: No).
5	having_At_Symbol	Indicates if the URL contains the '@' symbol, which can be used to mislead users.
6	double_slash_redirecting	Checks if the URL has '//' after the protocol, which can be used for redirection.
7	Prefix_Suffix	Indicates the use of '-' in the domain name (phishing sites often use hyphenated domains).
8	having_Sub_Domain	Determines the number of subdomains (more subdomains may indicate phishing).
9	SSLfinal_State	Checks if the website has a valid SSL certificate (1: Trusted, 0: Suspicious, -1: Untrusted).
10	Domain_registeration_length	Measures the duration for which the domain is registered (shorter durations may indicate phishing)
11	Favicon	Checks if the favicon is loaded from an external domain instead of the main domain.
12	port	Checks if unusual ports (other than 80, 443) are open (phishing sites may use uncommon ports).
13	HTTPS_token	Checks if "https" is present in the domain name rather than just in the protocol.
14	Request_URL	Measures the number of external objects (images, scripts, etc.) linked in the webpage.
15	URL_of_Anchor	Analyzes the percentage of anchor tags (<a>) leading to different domains.
16	Links_in_tags	Checks the number of links inside <script> and <meta> tags.
17	SFH (Server Form Handler)	Determines if form submission URLs point to an external domain.
18	Submitting_to_email	Checks if form submissions are sent directly to an email instead of a database.
19	Abnormal_URL	Indicates if the URL is not associated with the actual website's identity.
20	Redirect	Counts the number of redirections before landing on the final page.
21	on_mouseover	Checks if JavaScript alters the status bar on mouseover (used in phishing tricks).
22	RightClick	Indicates whether right-clicking is disabled (phishing pages may disable it).
23	popUpWidnow	Checks for the presence of pop-ups, which are common in phishing sites.
24	Iframe	Detects if the webpage uses iframes, which can hide malicious content.
25	age_of_domain	Measures the age of the domain (younger domains are more suspicious).
26	DNSRecord	Checks if the domain has a valid DNS record (missing records indicate phishing).
27	web_traffic	Determines the website's Alexa ranking (lower traffic may indicate phishing).
28	Page_Rank	Checks the Google PageRank of the website (higher rank = more trusted site).
29	Google_Index	Checks if the site is indexed by Google (phishing sites may not be indexed).
30	Links_pointing_to_page	Counts the number of backlinks leading to the website.
31	Statistical_report	Checks if the domain is reported in online phishing databases.
32	Result	The classification label (1: Legitimate, -1: Phishing).

2.3. Data Preprocessing

Before training machine learning models, the dataset underwent preprocessing steps.

- No columns with **missing** values found.

- Converting row entries from string data type to int.
- The dataset was divided into **training** and **testing** sets using an **80-20 split**.

3.Model Selection

3.1. Introduction

Model selection is a crucial step in the machine learning pipeline, ensuring that the chosen model generalizes well to unseen data. This section discusses the models considered, evaluation criteria, and the selection process.

3.2. Considered Models

The following models were evaluated based on their suitability for the given dataset and problem:

- Logistic Regression: A simple yet effective model for binary classification.
- Decision Tree: A non-parametric model that captures complex patterns.
- Random Forest: An ensemble learning method that improves robustness.
- K Nearest Neighbour: A distance-based model that classifies based on neighboring data points.
- Support Vector Machine (SVM): A powerful algorithm that finds the optimal decision boundary for classification.

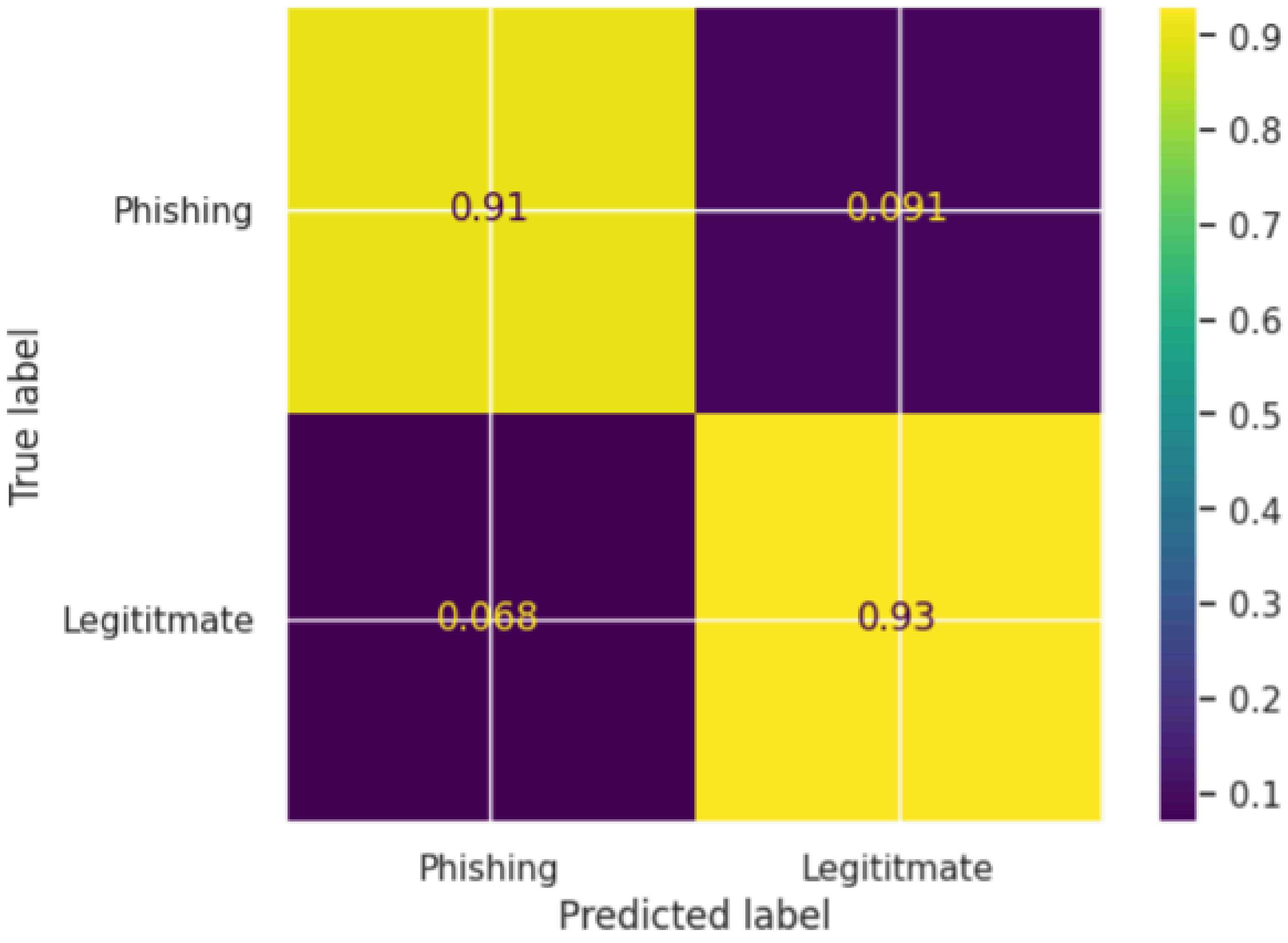
3.3. Model Evaluation Criteria

To compare model performance, the following metrics were used:

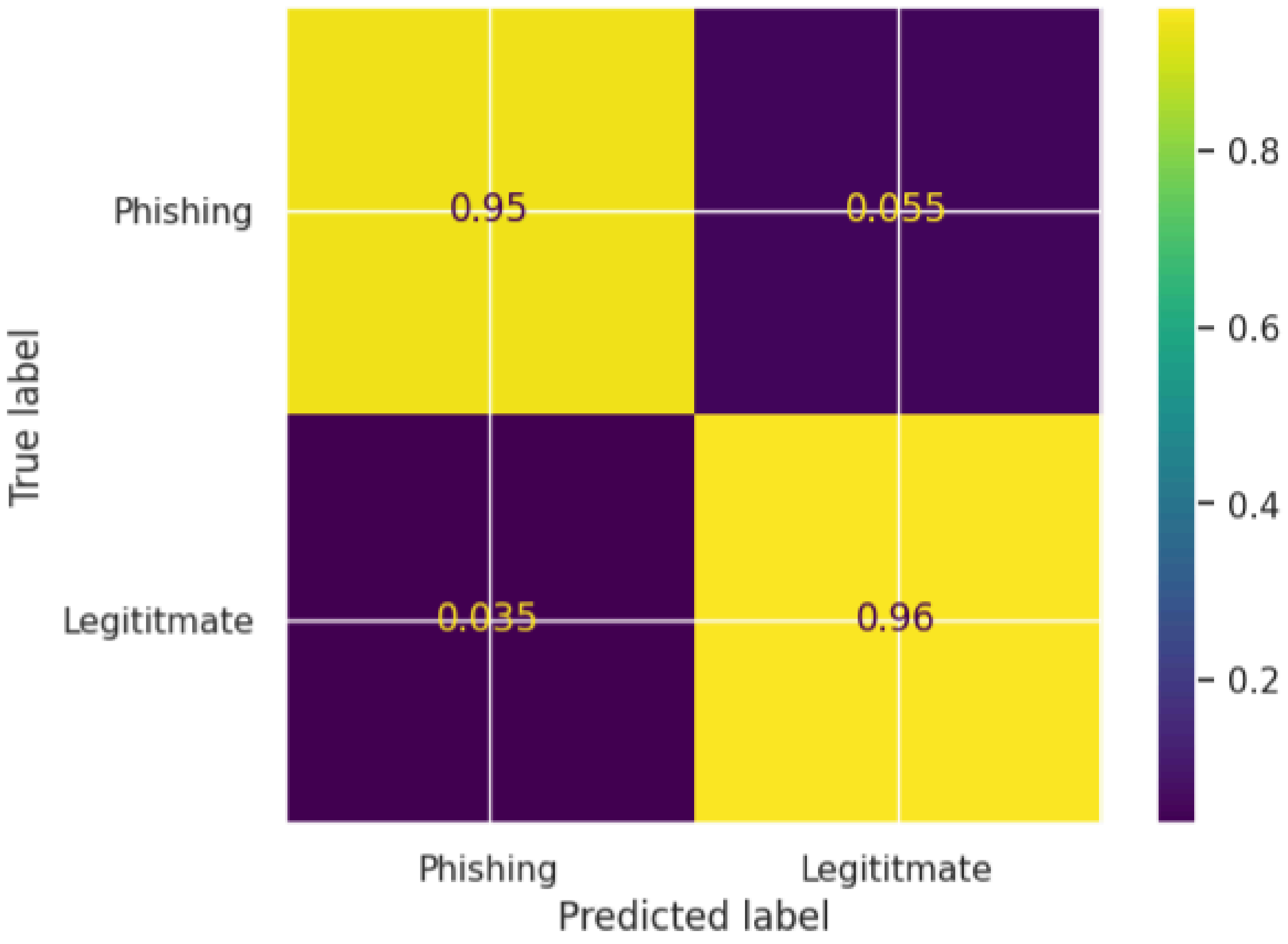
- Accuracy: Measures overall correctness.
- Precision: Important when minimizing false positives.
- Recall: Critical for minimizing false negatives.
- F1-score: Balances precision and recall.
- AUC-ROC: Evaluates classifier performance across thresholds.

3.4. Confusion Matrices

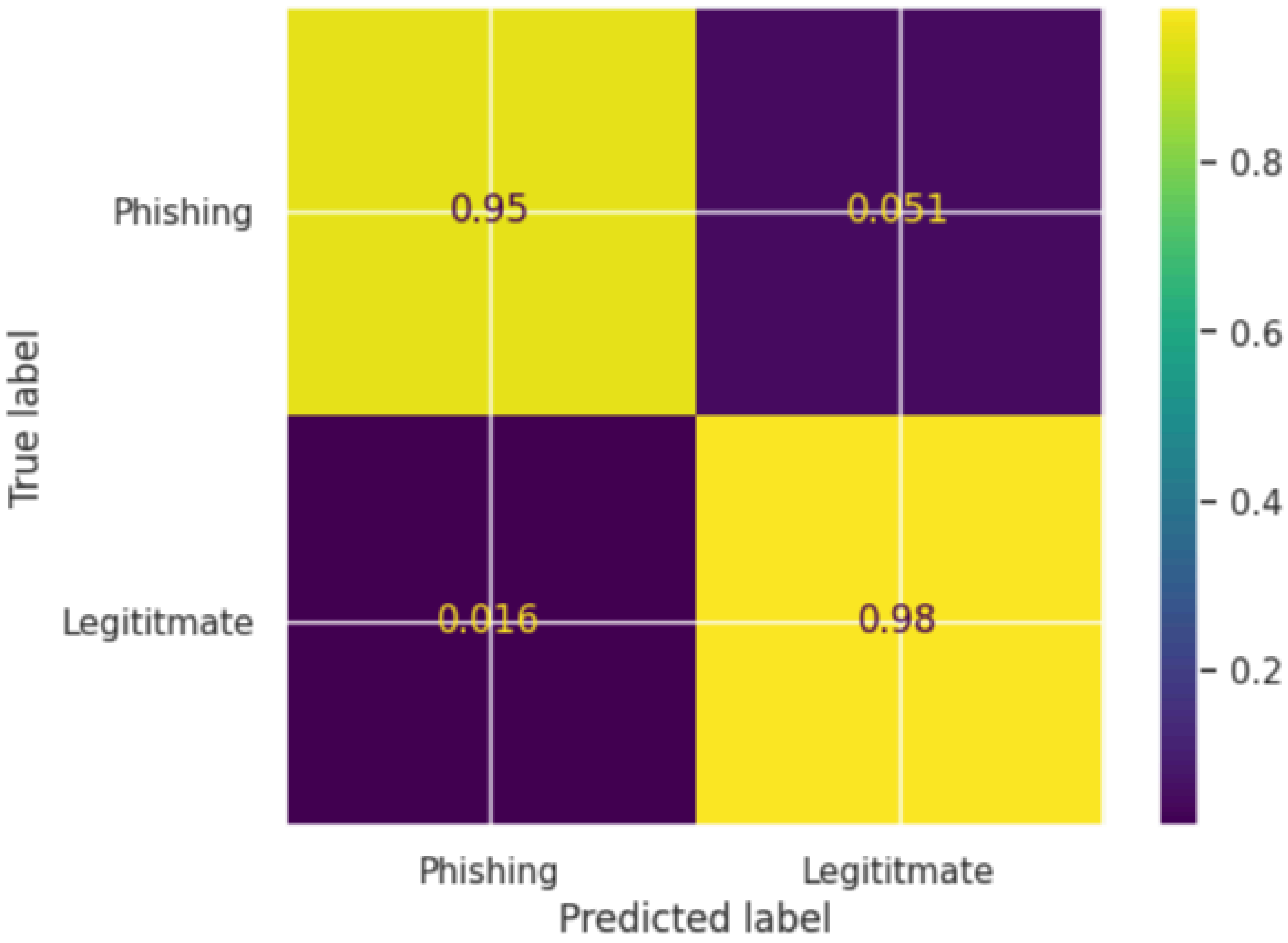
3.4.1. Logistic Regression



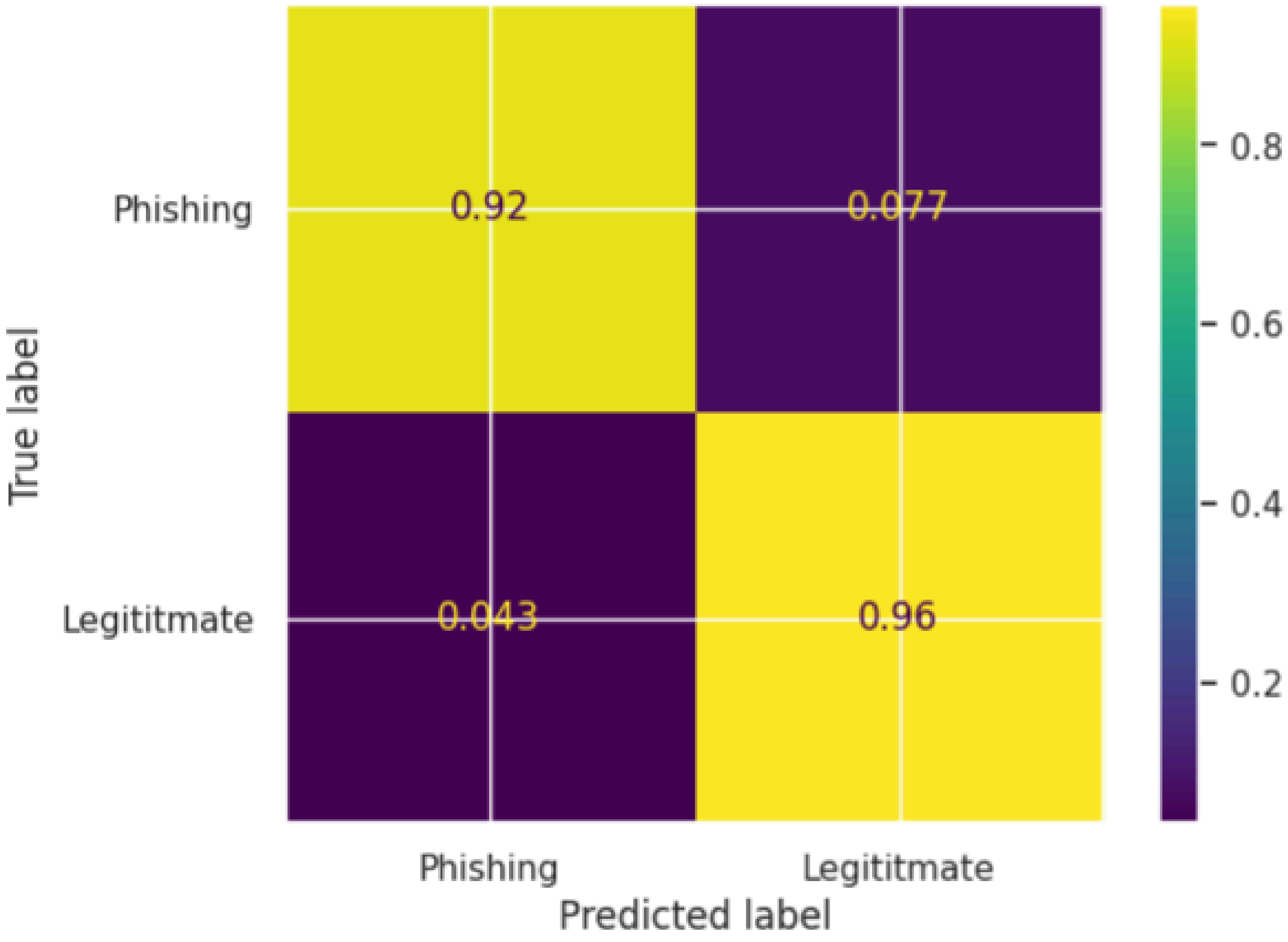
3.4.2. Decision Tree Classifier



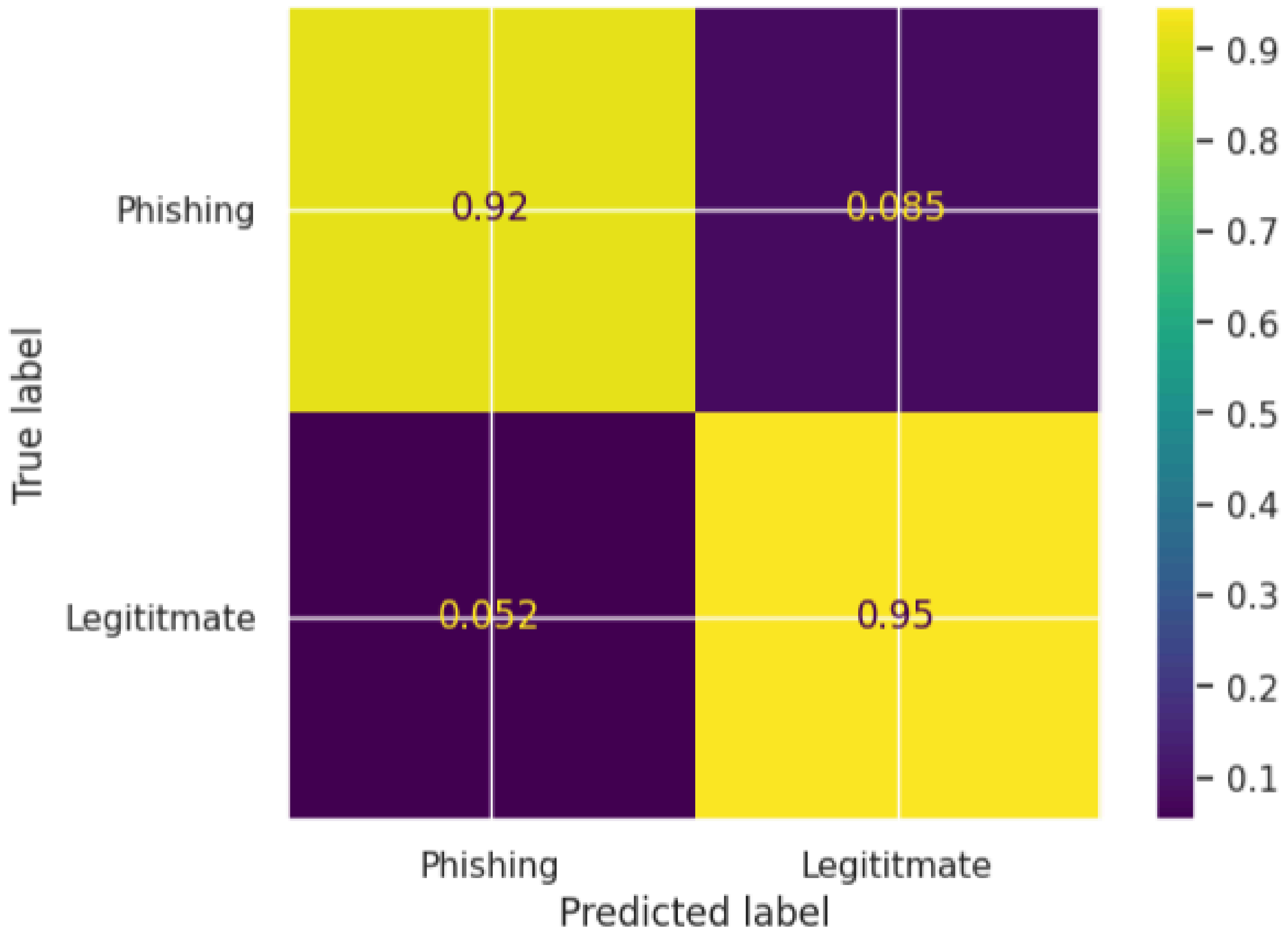
3.4.3 Random Forest Classifier



3.4.4. Support Vector Machines

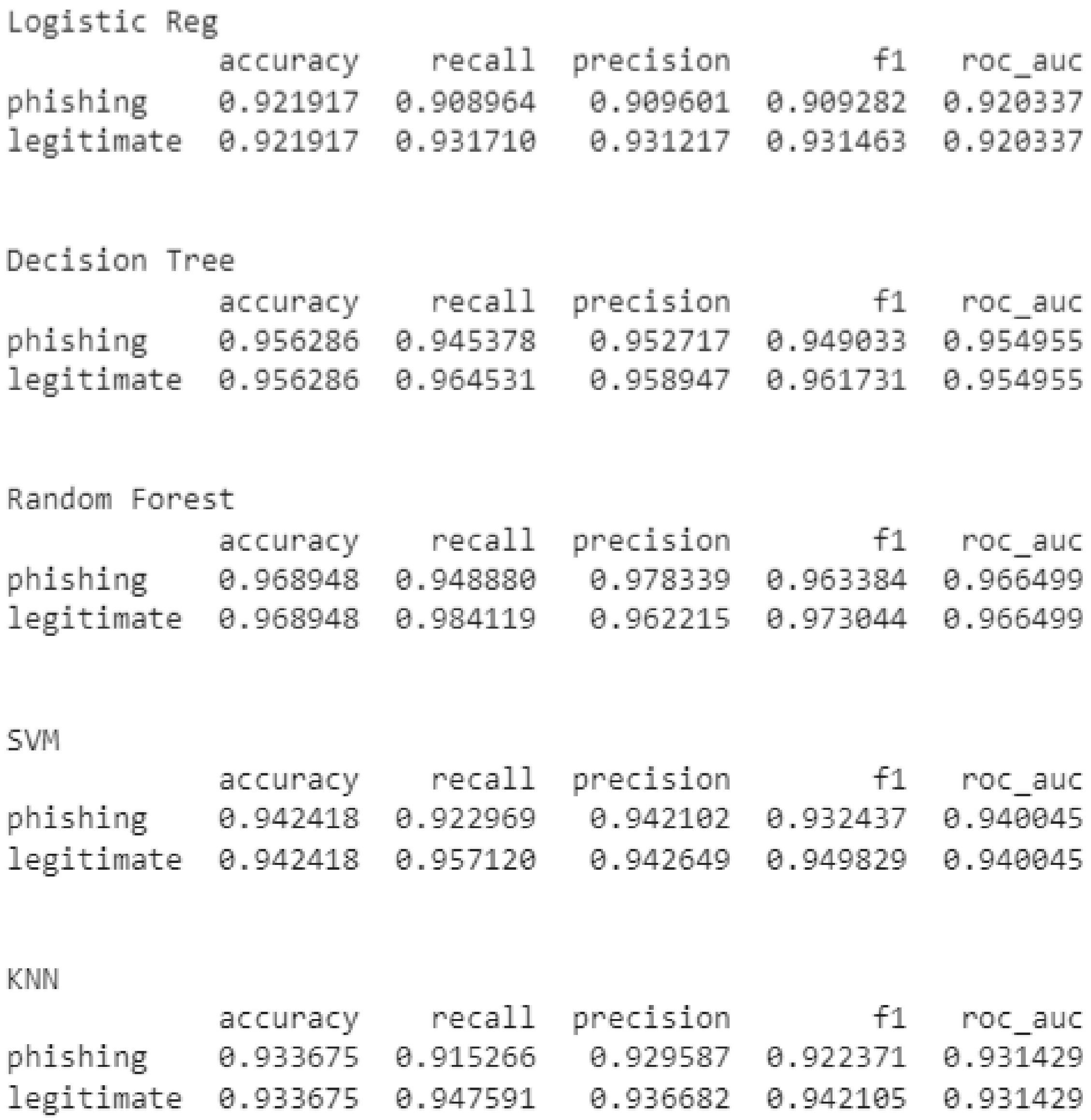


3.4.5 K Nearest Neighbours



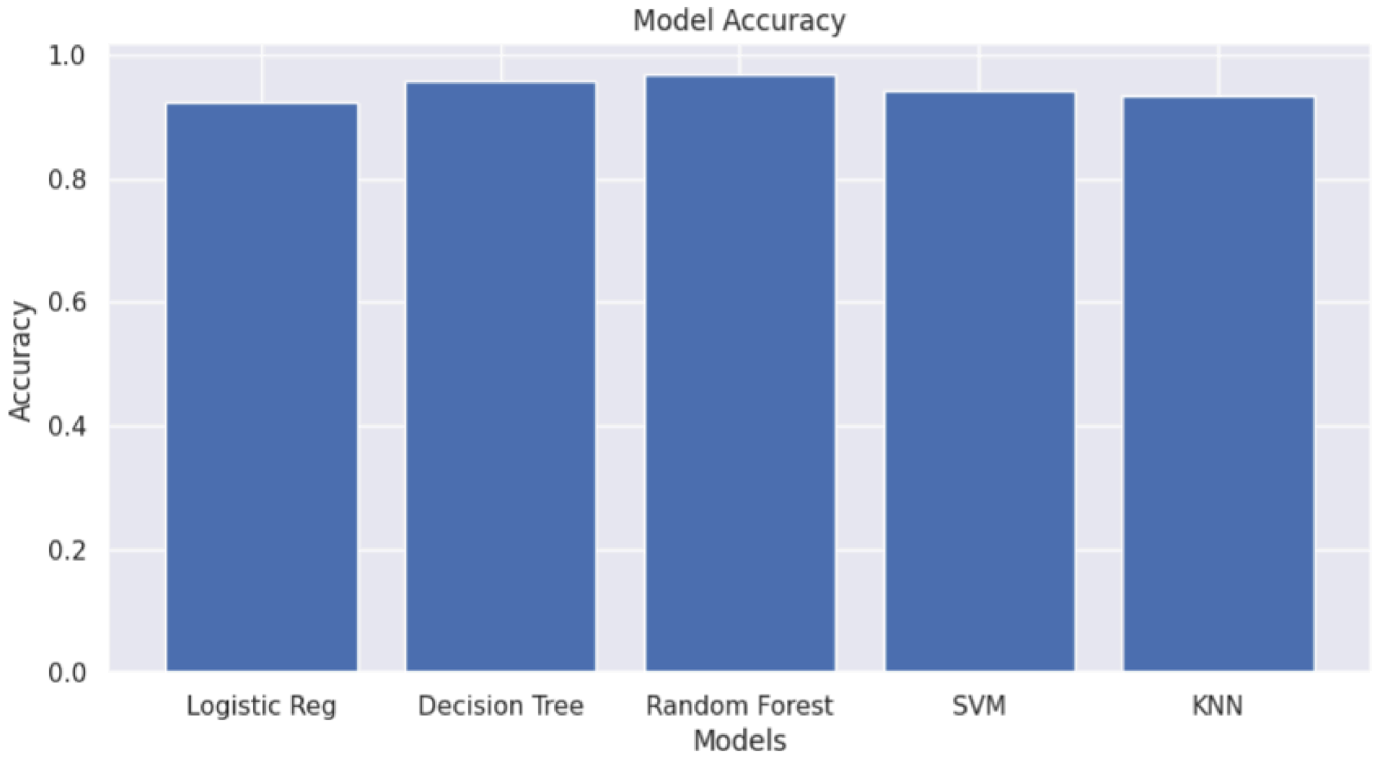
3.5. Metrics Comparison

3.5.1. Evaluation Table

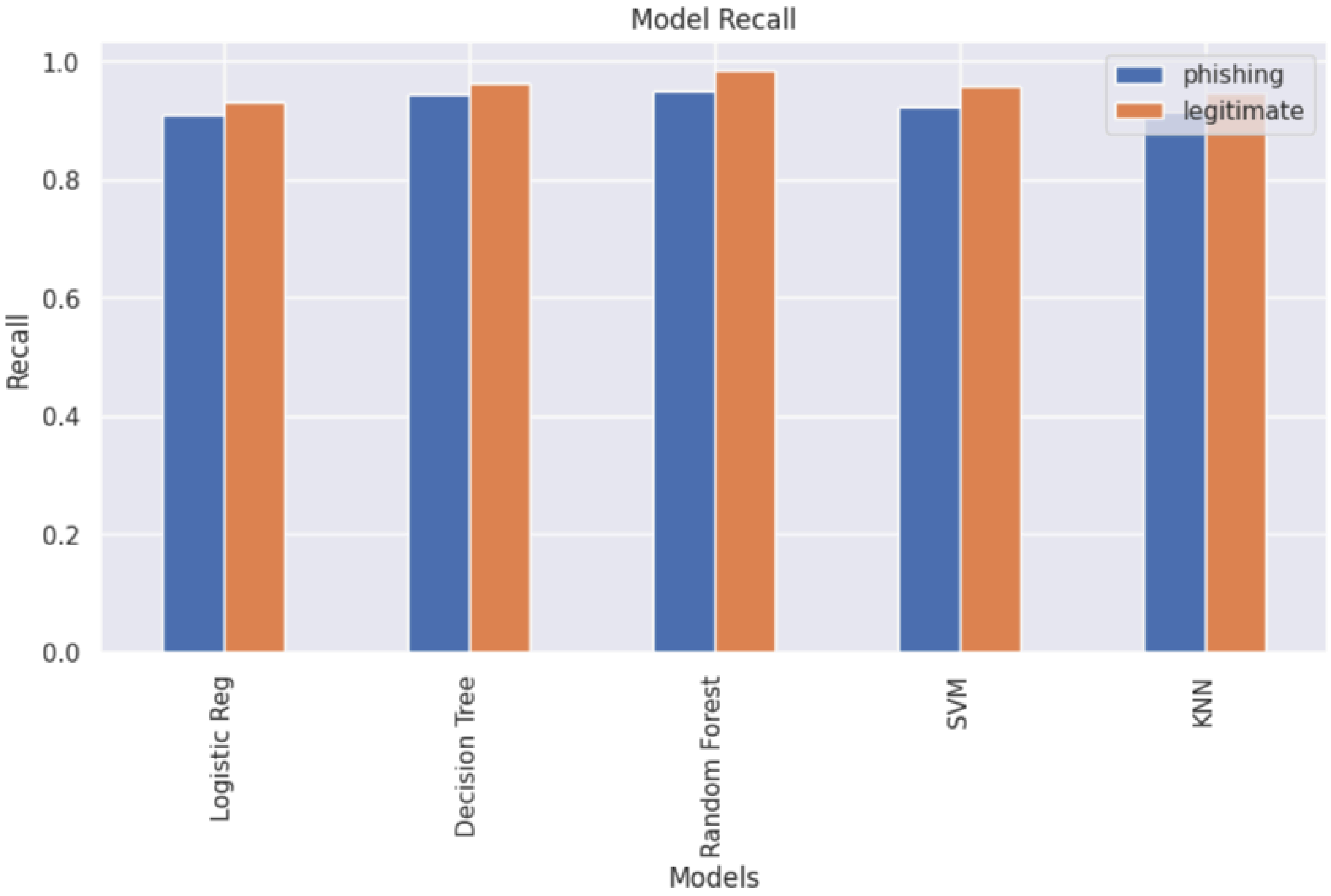


```
Logistic Reg
            accuracy    recall  precision        f1   roc_auc
phishing    0.921917  0.908964   0.909601  0.909282  0.920337
legitimate  0.921917  0.931710   0.931217  0.931463  0.920337


Decision Tree
            accuracy    recall  precision        f1   roc_auc
phishing    0.956286  0.945378   0.952717  0.949033  0.954955
legitimate  0.956286  0.964531   0.958947  0.961731  0.954955


Random Forest
            accuracy    recall  precision        f1   roc_auc
phishing    0.968948  0.948880   0.978339  0.963384  0.966499
legitimate  0.968948  0.984119   0.962215  0.973044  0.966499


SVM
            accuracy    recall  precision        f1   roc_auc
phishing    0.942418  0.922969   0.942102  0.932437  0.940045
legitimate  0.942418  0.957120   0.942649  0.949829  0.940045


KNN
            accuracy    recall  precision        f1   roc_auc
phishing    0.933675  0.915266   0.929587  0.922371  0.931429
legitimate  0.933675  0.947591   0.936682  0.942105  0.931429
```

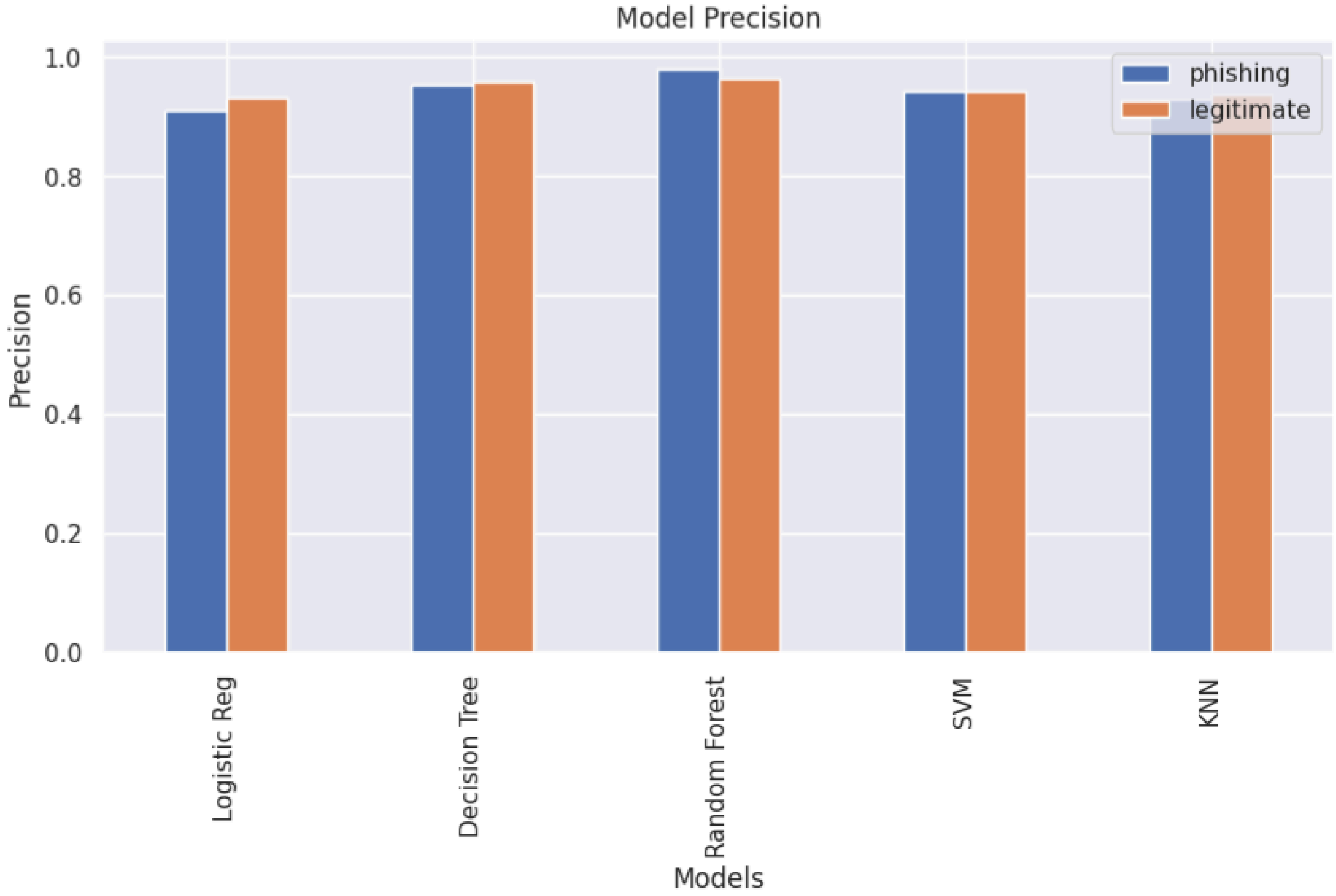
3.5.2 Accuracy



3.5.3 Recall



3.5.4. Precision


Model Precision
Precision
1.0
0.8
0.6
0.4
0.2
0.0
phishing
legitimate
Logistic Reg
Decision Tree
Random Forest
SVM
KNN
Models

4.Conclusion and Future Work

4.1. Project Overview

- The project aimed to build and evaluate multiple machine learning models for classification.
- Several models, including Logistic Regression, Decision Tree, Random Forest, KNN, and SVM, were compared based on key performance metrics.

4.2. Model Performance

- **Random Forest emerged as the best-performing model**, demonstrating a strong balance of accuracy, precision, and recall.
- **Decision Tree followed closely**, offering strong interpretability and decent generalization.
- **SVM ranked third**, performing well with optimized decision boundaries.
- **Logistic Regression and KNN had nearly equal performance**, serving as strong baselines but with slightly lower generalization.

4.3. Future Improvements

- Experimenting with **more advanced models** like Gradient Boosting or XGBoost.
- Implementing **feature selection techniques** to further optimize input variables.
- Exploring **deep learning approaches** for potential improvements in classification.
- Enhancing data preprocessing methods to handle missing values and outliers more effectively.

4.4. Conclusion

- The project successfully demonstrated a structured approach to model selection and evaluation.
- The best-performing model (Random Forest) will be further tested for deployment on unseen data.
- Continuous refinement through hyperparameter tuning and feature engineering can lead to even better results.