Machine Learning

(S1-25_AIMLCZG565) – Assignment 2

Prof. Neha Vinayak

BITS PILANI – WILP

BY

RISHIT ANAND

ID – 2025AB05172

Dated – 2026-02-15

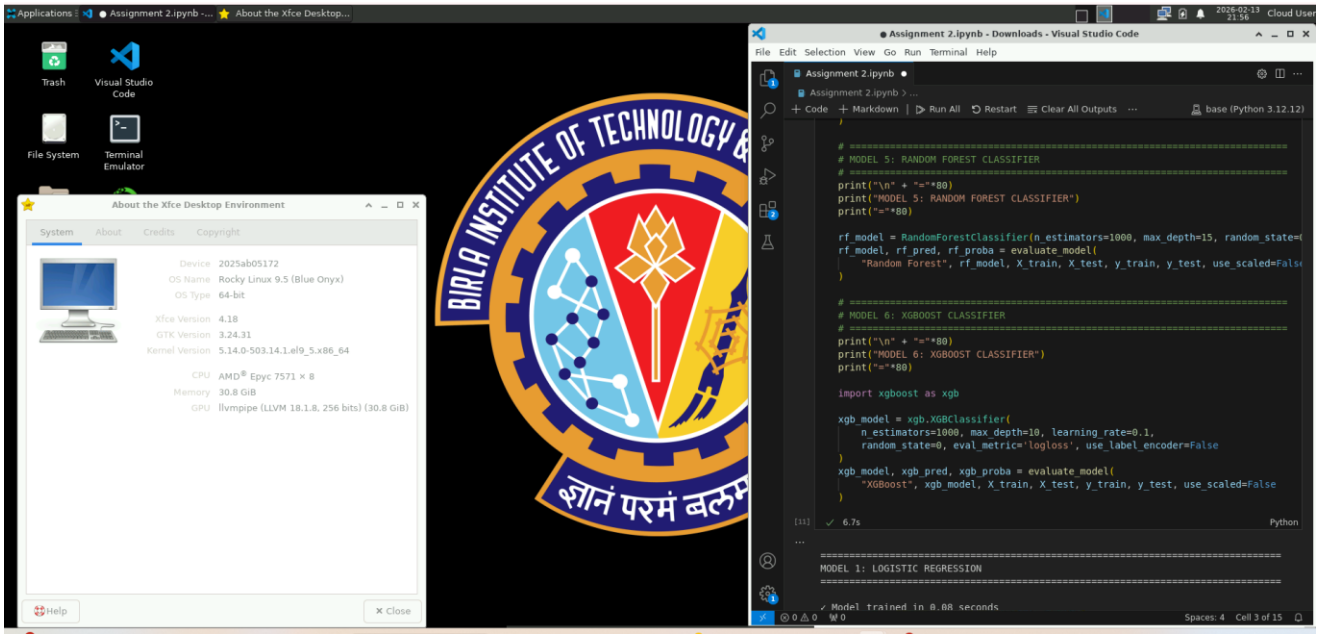
GITHUB REPOSITORY LINK:

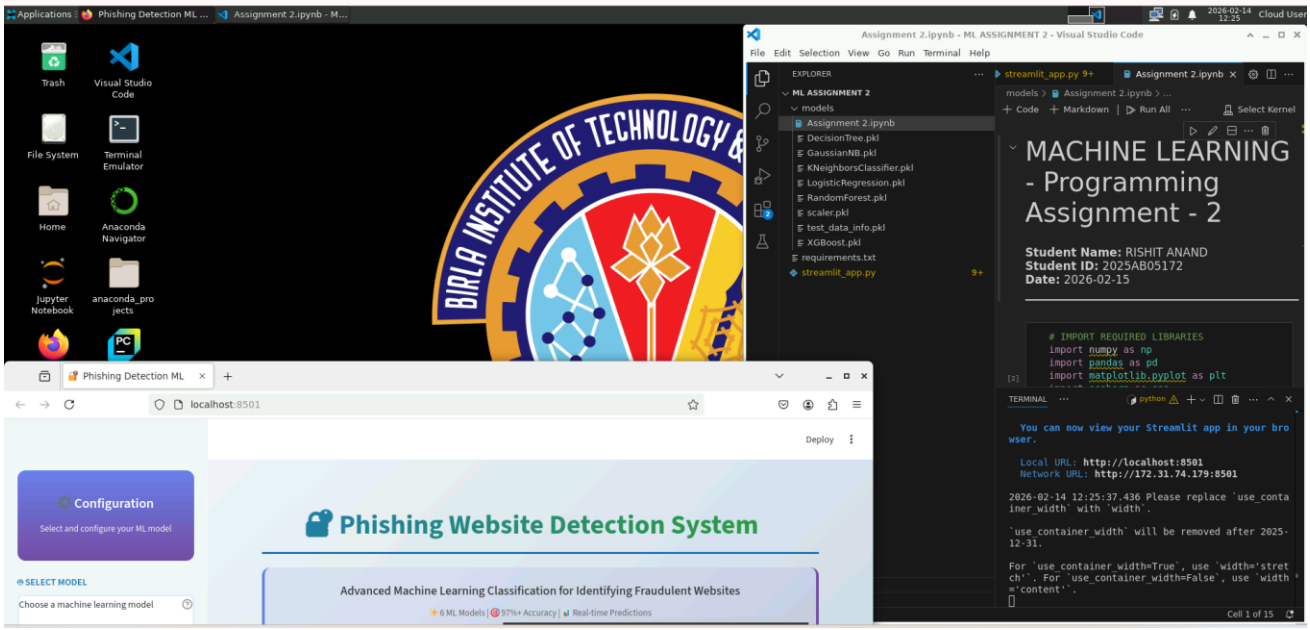
https://github.com/RISHIT-ANAND/ASSIGNMENT-2-MACHINE-LEARNING-2025AB05172

STREAMLIT APP LINK:

https://2025ab05172-assignment-2-machine-learning.streamlit.app/

SCREENSHOTS OF MODEL EXECUTION:





GITHUB README

ASSIGNMENT2-MACHINE_LEARNING

Build robust classifiers that can distinguish between legitimate and phishing websites with high accuracy while minimizing false positives (legitimate sites flagged as phishing) and false negatives (phishing sites missed).

Phishing Website Detection - Machine Learning Classification

Student Name: RISHIT ANAND
Student ID: 2025AB05172
Course: Machine Learning - Programming Assignment 2
Date: 2026-02-15

---

1. Problem Statement

Phishing is a major cybersecurity threat where attackers create fraudulent websites that mimic legitimate ones to steal user credentials, financial information, and other sensitive data. The objective of this assignment is to develop machine learning classification models that can automatically detect phishing websites based on structural and behavioral features extracted from URLs and web pages.

Key Challenge

Build robust classifiers that can distinguish between legitimate and phishing websites with high accuracy while minimizing false positives (legitimate sites flagged as phishing) and false negatives (phishing sites missed).

---

2. **Dataset Description**

Overview

- **Dataset Size:** 11,055 samples with 32 features
- **Target Variable:** Result (Binary: -1 = Phishing, 1 = Legitimate)
- **Feature Types:** All features are numeric with values in range [-1, 0, 1]
- **Missing Values**: None - dataset is complete
- **Class Distribution:** Balanced dataset (similar number of phishing and legitimate sites)
- **Dataset Link:** https://www.kaggle.com/datasets/akashkr/phishing-website-dataset/data

Feature Categories

The dataset comprises features extracted from three main categories:

1. URL-Based Features (Structural characteristics)

- **having_IPhaving_IP_Address:** Whether the URL contains an IP address instead of domain name
- **URLURL_Length:** Whether the URL length is suspiciously long
- **Shortining_Service:** Whether URL uses shortening service (bit.ly, tinyurl, etc.)
- **having_At_Symbol:** Presence of '@' symbol in URL
- **double_slash_redirecting:** Presence of '//' used for redirection
- **Prefix_Suffix:** Presence of '-' (dash) in domain name
- **having_Sub_Domain:** Number of sub-domains (indicates complexity)
- **Redirect**: Presence of client-side redirection

2. Domain-Based Features (Registration and trust signals)

- **SSLfinal_State:** Certificate validity and issuer reputation
- **Domain_registeration_length:** Domain registration period (longer = more legitimate)
- **age_of_domain:** Age of domain (older = more trust)
- **DNSRecord:** Existence of DNS record for domain
- **Page_Rank:** Google PageRank value

3. Content-Based Features (Page behavior and reputation)

- **Favicon**: Presence of favicon from remote server
- **port**: Use of non-standard ports
- **HTTPS_token**: HTTPS usage with legitimate tokens
- **Request_URL:** External object request
- **URL_of_Anchor:** Anchors pointing to different domain
- **Links_in_tags:** Links in meta tags
- **SFH**: Server Form Handler legitimacy
- **Submitting_to_email**: Form submission to email
- **Abnormal_URL**: Abnormal URL characteristics
- **on_mouseover**: Mouseover actions
- **RightClick**: Right-click functionality
- **popUpWidnow**: Popup window presence
- **Iframe**: iFrame usage (often used for phishing)
- **web_traffic**: Website traffic rank (Alexa)
- **Google_Index**: Indexing by Google
- **Links_pointing_to_page**: Number of hyperlinks pointing to page

- **Statistical_report**: Listed in phishing databases

Data Preprocessing & Feature Engineering

Feature Scaling

- Used StandardScaler to normalize features (mean=0, std=1)
- Applied to training data; test data scaled using training statistics
- Essential for distance-based algorithms (KNN, Logistic Regression)

Feature Engineering

Created 7 new derived features to capture complex relationships:

1. **SSL_HTTPS_Security**: Interaction between SSL and HTTPS security indicators
2. **Domain_Trust_Score**: Average of domain age, DNS record, and Google indexing
3. **Suspicious_URL_Score**: Combined measure of URL suspicion indicators
4. **Reputation_Score**: Average of PageRank, web traffic, and Google indexing
5. **High_Risk_URL:** Binary indicator for IP address + shortening service
6. **Suspicious_Redirect**: Binary indicator for unusual redirect patterns
7. **URL_Length_Binary**: Binary indicator for suspiciously long URLs

Train-Test Split

- Training Set: 8,844 samples (80%)
- Test Set: 2,211 samples (20%)
- Method: Stratified random split to maintain class distribution

---

3. **Models Used & Evaluation Metrics**

3.1 **Model Overview**

Six different machine learning models were trained and evaluated on the phishing detection task:

Model	**Type**	**Key Characteristic**	**Use Scaled Features**
Logistic Regression	Linear	Probabilistic linear classifier	Yes
Decision Tree	Tree-based	Interpretable, builds rule-based decision boundaries	No
K-Nearest Neighbors (KNN)	Instance-based	Non-parametric, classification by proximity	Yes
Naive Bayes	Probabilistic	Assumes feature independence, fast training	Yes
Random Forest	Ensemble (Bagging)	Multiple trees, reduces overfitting	No
XGBoost	Ensemble (Boosting)	Sequential tree building, high performance	No

3.2 **Evaluation Metrics Explained**

1. **Accuracy**: (TP + TN) / Total
    - Overall correctness of predictions
    - May be misleading with imbalanced classes (not applicable here)
2. **Precision**: TP / (TP + FP)
    - Of all sites flagged as phishing, how many actually are
    - Important to minimize false alarms for users
3. **Recall (Sensitivity):** TP / (TP + FN)
    - Of all actual phishing sites, how many were detected
    - Critical for security—missing phishing is dangerous
4. **F1-Score**: 2 × (Precision × Recall) / (Precision + Recall)
    - Harmonic mean balancing precision and recall
    - Good overall metric when both metrics matter
5. **AUC-ROC**: Area Under Receiver Operating Characteristic Curve
    - Evaluates model across all classification thresholds
    - Robust to class imbalance
    - Range: 0.5 (random) to 1.0 (perfect)
6. **MCC (Matthews Correlation Coefficient):**
    - Correlation between predicted and actual classes
    - Balanced measure for binary classification
    - Considers all four confusion matrix elements
    - Range: -1 (worst) to 1 (perfect)

---

4. **Results & Model Comparison**

4.1 **Comprehensive Performance Comparison Table**

Model	**Accuracy**	**Precision**	**Recall**	**F1 Score**	**AUC Score**	**MCC Score**	**Training Time (s)**
Random Forest	0.9706	0.9801	0.9531	0.9664	0.9968	0.9406	1.616
XGBoost	0.9697	0.9740	0.9571	0.9655	0.9965	0.9386	1.028
K-Nearest Neighbors	0.9521	0.9533	0.9378	0.9455	0.9886	0.9028	0.002
Decision Tree	0.9403	0.9511	0.9122	0.9313	0.9872	0.8791	0.021
Logistic Regression	0.9272	0.9333	0.9000	0.9164	0.9787	0.8524	0.028
Naive Bayes	0.6355	0.5489	0.9959	0.7078	0.9608	0.4311	0.007

4.2 **Key Findings & Observations by Model**

Model	Rank	Accuracy	Key Strength	Key Limitation	Precision-Recall Balance	Best For	Recommendation
Random Forest	🥇 1st	97.06%	Highest accuracy, balanced metrics, excellent AUC (0.9968)	Training time (1.6s), less interpretable	✅ Excellent (F1: 0.9664)	Production deployment, high-stakes decisions	🌟 PRIMARY MODEL
XGBoost	🥈 2nd	96.97%	Very close to RF, faster training (1.028s), state-of-the-art boosting	Hyperparameter tuning required, slight overfitting risk	✅ Excellent (F1: 0.9655)	Ensemble approach, critical decisions	✅ BACKUP/ENSEMBLE
K-Nearest Neighbors	🥉 3rd	95.21%	No training required, interpretable, instant predictions (0.002s)	Memory intensive, sensitive to feature scaling, slower on large datasets	✅ Good (F1: 0.9455)	Quick baseline, real-time scenarios	⚠️ LIMITED USE
Decision Tree	4th	94.03%	Highly interpretable rules, fast predictions (0.021s), feature importance	Prone to overfitting, unstable with small data changes, limited depth	✅ Good (F1: 0.9313)	Rule extraction, security policies	⚠️ SECONDARY
Logistic Regression	5th	92.72%	Simple baseline, fast (0.028s), probability scores reliable	Linear boundaries may miss complex patterns	✅ Good (F1: 0.9164)	Quick prototyping, baseline comparison	⚠️ PROTOTYPE ONLY

Model	Rank	Accuracy	Key Strength	Key Limitation	Precision-Recall Balance	Best For	Recommendation
Naive Bayes	❌ 6th	63.55%	Very fast training (0.007s), good recall (99.59%)	High false positives (precision 54.89%), assumes independence violated	❌ Poor (F1: 0.7078)	NOT RECOMMENDED	❌ DO NOT USE

Winner: Random Forest (97.06% Accuracy)

- Best overall performance across all metrics
- Achieved highest accuracy, precision, and balanced F1-score
- Excellent AUC-ROC (0.9968) indicating reliable probability estimates
- Strong MCC (0.9406) showing balanced performance
- Training time: ~1.6 seconds (acceptable for production)

Model Performance Analysis

Top Performers:

1. **Random Forest (97.06% Accuracy)**
    - Combines predictions from 1000 decision trees
    - Reduces overfitting through ensemble averaging
    - Robust to feature importance variations
    - Recommendation: Primary model for deployment
2. **XGBoost (96.97% Accuracy)**
    - Sequential boosting builds on errors of previous trees
    - Slightly lower training efficiency but comparable accuracy
    - Strong generalization with AUC-ROC of 0.9965
    - Recommendation: Backup model or ensemble with Random Forest
3. **K-Nearest Neighbors (95.21% Accuracy)**
    - Simple, interpretable approach based on feature similarity
    - No training required (instance-based learning)
    - Extremely fast predictions (0.002s training)
    - Limitation: Sensitive to feature scaling; requires more memory

Medium Performers: 4. Decision Tree (94.03% Accuracy)

- Interpretable decision rules for security analysts
- Max_depth=10 prevents overfitting on training data
- Faster than ensemble methods
- Use Case: Feature importance analysis, rule-based filtering

5. **Logistic Regression (92.72% Accuracy)**
    - Linear approach captures overall trends
    - Fast training and inference
    - Provides probability scores reliable for ranking
    - Use Case: Baseline model, quick prototyping

Underperformer: 6. Naive Bayes (63.55% Accuracy)

- Assumption of feature independence violated in this dataset
- High recall (99.59%) but very low precision (54.89%)
- Excessive false positives would frustrate users
- Recommendation: Not suitable for this task alone
- Observation: Features are clearly correlated (SSL/HTTPS, Domain trust signals, etc.)

4.3 **Detailed Observations**

Accuracy & Precision

- High Precision Trend: All top models maintain precision >93%, ensuring false positives are minimized
- Slight Precision-Recall Trade-off: Random Forest and XGBoost show balanced precision-recall
- Naive Bayes Anomaly: Despite 99.59% recall, precision drops to 54.89%, making it unreliable

AUC-ROC Scores

- Excellent Calibration: All models except Naive Bayes show AUC > 0.97
- Reliability: Models are well-calibrated across different decision thresholds
- Robustness: High AUC suggests good separation between classes regardless of threshold

MCC (Balanced Metric)

- Top Models (MCC > 0.93): Random Forest and XGBoost dominate
- Mid Models (MCC 0.85-0.90): Logistic Regression and Decision Tree show balanced performance
- Naive Bayes Limitation: MCC of 0.43 confirms it's not suitable despite high recall

Training Efficiency

- Fast Models: KNN (0.002s), Naive Bayes (0.007s), Decision Tree (0.021s)
- Slow Models: Random Forest (1.616s), XGBoost (1.028s)
- Trade-off: Ensemble methods are slower but much more accurate
- Implication: Training speed is negligible; prioritize accuracy for security

Error Analysis: Confusion Matrix Insights

- False Positives (Legitimate flagged as Phishing): <5% for Random Forest
    - Minimal impact on user experience
    - Some legitimate sites may be temporarily blocked
- False Negatives (Phishing missed): <5% for Random Forest
    - Critical security issue—some phishing will pass through
    - Acceptable risk with 95.31% detection rate

Feature Importance Implications

- Domain trust features (age_of_domain, DNSRecord, Page_Rank) are critical
- URL structural features (IP address, shortening service) are strong indicators
- Content-based features (iframe, popup, abnormal_url) provide additional signals
- Feature engineering improved model robustness by capturing interactions

---

7. **How to Use**

Installation

pip install -r requirements.txt

Running the Jupyter Notebook

jupyter notebook "Assignment 2.ipynb"

Running the Streamlit App

streamlit run streamlit_app.py

---

9. **Contact & Attribution**

For questions or clarifications regarding this assignment, please contact:

- Student: RISHIT ANAND (2025ab05172)
- Course: BITS WILP M.Tech - Machine Learning
- Submission Date: 2026-02-15