# Team Analytix Report for Binary Classification

**Problem Statement**

The rise and fall of ShopSecure serves as a cautionary tale in the world of e-commerce and cybersecurity. Once a trusted platform boasting cutting-edge encryption technologies and AI-powered fraud detection, ShopSecure faced a devastating downfall due to security lapses introduced by unmonitored third-party advertisements. These vulnerabilities exposed users to phishing scams, malware, and data breaches, eroding customer trust and tarnishing the company’s reputation.

To rebuild credibility and secure their platform, ShopSecure now seeks to develop a **robust phishing detection system**. This system will use machine learning to classify URLs as either **legitimate** or **malicious**, thereby safeguarding users from cyber threats. The task requires a systematic approach to analyze the provided dataset, understand the patterns in URL characteristics, and build an efficient model to predict phishing attempts.

**Dataset Overview**

The dataset contains information about **11,000 URLs**, each described by **32 features**. These features are encoded as:

* -1: **Suspicious**
* 0: **Phishing**
* 1: **Legitimate**

The dataset presents a unique opportunity to leverage these characteristics to distinguish between legitimate and malicious URLs. The features are specifically tailored to highlight potential markers of phishing, enabling a data-driven approach to cybersecurity.

**Tasks at Hand**

**Task 1: Exploratory Data Analysis (EDA)**

The first step involves understanding the structure and quality of the data to lay a strong foundation for modeling:

1. **Explore Data Distributions**:
   * Analyze the distribution of feature values (-1, 0, 1) using histograms.
   * Utilize heatmaps to visualize correlations between features and detect redundancy or multicollinearity.
2. **Dataset Insights**:
   * Determine the total number of samples and calculate the count of unique values for each feature.
   * Check for missing or null values, as these could impact model training and evaluation.
3. **Feature Selection**:
   * Identify highly correlated features using a correlation matrix with a predefined threshold.
   * Randomly remove one feature from each highly correlated pair to reduce redundancy while preserving information.

**Task 2: Building the Classification Model**

The ultimate objective is to create a robust machine learning model to classify URLs into **legitimate** or **malicious** categories. The subtasks include:

1. **Model Building**:
   * Train and evaluate multiple binary classification models (e.g., Logistic Regression, Random Forest, and Decision Tree) to detect phishing URLs.
2. **Model Evaluation**:
   * Plot ROC curves to illustrate the diagnostic performance of each model.
   * Compare metrics such as **Accuracy** and **ROC-AUC** to identify the best-performing model.
3. **Validation**:
   * Implement the **K-Fold Cross-Validation** technique to validate the accuracy and robustness of each model.
4. **Final Output**:
   * Present a classification system with optimized features and parameters that achieves the highest accuracy on the validation dataset.

**Key Challenges**

1. **Feature Redundancy**:
   * Identifying and removing highly correlated features to avoid overfitting and improve computational efficiency.
2. **Class Distribution**:
   * Understanding the balance of legitimate, suspicious, and phishing samples in the dataset to ensure fair model performance.
3. **Model Generalization**:
   * Ensuring the classification system performs well not just on training data but also on unseen URLs, particularly through cross-validation.

**Objective**

The overarching goal is to deliver a **reliable phishing detection system** that protects users from cyber threats while maintaining the scalability and robustness required for real-world applications. Through rigorous exploratory analysis, model building, and validation, the solution aims to restore trust and security to ShopSecure's platform.

**Task 1: Exploratory Data Analysis (EDA)**

Plotted histograms ([Figures Section](#_FIGURES_:_Correlation)) for the feature values in sets of 4, ensuring clarity and readability. Visualized how features are distributed across the three target classes (-1: Suspicious, 0: Phishing, 1: Legitimate). This helped identify skewness, outliers, or unusual patterns in the data.

Checked the total number of samples and features to get an overview of the dataset's size. Total samples size = 11055 over 32 features. Used data.info() to examine feature data types and detect inconsistencies. Also checked for unique elements in all features which came to be -1,0,1 . Verified if any features contained null values using data.isnull().sum() to ensure data completeness, there are 0 null values in the whole dataset.

Column Result was reclassified into not having any suspicious entry, and rather tag all suspicious entries as malicious to comply with binary classification. We are labelling any suspicious link as malicious to make a more strict and more secure classification system with as high of accuracy rate as possible for safe websites.

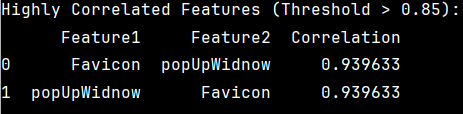
Generated a correlation matrix ([Figures Section](#_FIGURES_:_Correlation)) to analyze relationships between features used in both task 1 and 2. For Task 2 preprocessing we mainly focused on detecting highly correlated features that might introduce multicollinearity or redundancy. Used a heatmap to visualize the correlation matrix for easier interpretation. Identified feature pairs with correlation coefficients exceeding a threshold (e.g., > 0.85), indicating strong linear relationships. Favicon and popUpWindow showed high linear relationships.

Figure 1. High Correlation Features

One of them was removed from the dataset to avoid overfitting of the model due to redundancy. Anyone could be removed from the dataset, doesnot affect the model’s capability in the bigger scale.

**TASK 2: Binary Classification Models:** Cleaned Data (removed index) was divided into Train and Test. 20 percent of the data has been kept as test data and rest 80 percent is training data.

Two Machine Learning Models have been employed here to test this Binary Classification. Decision Tree and Random Forest.

A Decision Tree splits the dataset into smaller subsets based on feature thresholds, forming a tree structure. Each node in the tree represents a decision based on a feature, and leaf nodes represent the output class. Splitting is based on reducing impurity (e.g., Gini Index or Entropy).

Random Forest is an ensemble learning method that combines multiple decision trees to improve accuracy and reduce overfitting. Each tree is trained on a random subset of the data (using bootstrap sampling) and selects a random subset of features for splitting nodes. Predictions are made by aggregating, Classification: Majority vote of the trees or Regression: Average prediction of the trees. Math Behind Decision Trees (Core Component): At each node, the algorithm selects the feature and split point that maximizes information gain or minimizes impurity. Impurity is measured by 2 mthods or measures, Gini Index and Entropy.

A black and white math equation

Description automatically generatedA black and white math equation

Description automatically generatedRandom Forest is advantageous because of its ability to handle non linear data effectively. It is highly resistant to overfitting due to the estimators argument which can be set and fine tuned to give near perfect results but this comes with very high computation cost.

Figure 2. Entropy

Figure 3. Gini Index

Table 1. When to Choose the Models

|  |  |  |
| --- | --- | --- |
| **Criterion** | **Random Forest** | **Decision Tree** |
| **Data Size** | Medium to Large | Small to Medium |
| **Relationship** | Non-Linear | Non-Linear |
| **Accuracy** | High | Moderate |
| **Interpretability** | Low | High |
| **Handling of Non Linear Features** | Excellent | Excellent |
| **Overfitting** | Low (If sufficient trees made) | Requires Regularization of max depth |
| **Computation Cost** | High | Lesser |

A graph with a line

Description automatically generatedROC curve to be plotted. A ROC Curve is a graphical representation used to evaluate the performance of a binary classification model. It plots the True Positive Rate (TPR) against the False Positive Rate (FPR) at various thresholds. It plots the True Positive Rate (TPR) against the False Positive Rate (FPR) at various thresholds showing its ability to differentiate between the positive (malicious) and negative (legitimate) classes. A threshold determines the point at which a probability prediction is classified as positive or negative. Adjusting thresholds affects the trade-off between TPR and FPR. Each point on the ROC curve corresponds to a specific threshold for classifying positive and negative cases. We plotted the ROC Curves of our Random Forest and Decision Tree Models.

Figure 4. ROC Curves

**Random Forest** achieved the highest AUC (~0.99), showcasing its exceptional ability to distinguish between legitimate and malicious URLs. **Decision Tree** had lower AUC (~0.98), reflecting good performance but lower reliability compared to the random forest model. Both of them performed **more or less extremely similarly**, though this can **vary time to time and attempt after attempt, depending on the training set that is decided**. Even **accuracy values can change depending on the training and testing sets**. Hence an average of the value which is in the range of 0.96 to 0.99 is safe to assume in our case.

A black background with white text

Description automatically generatedA black background with white text

Description automatically generatedAccuracy of models are to be:

Figure 5. Random Forest Results

Figure 6. Decision Tree Results

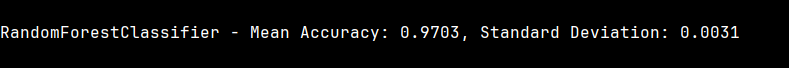
****We thereafter perform the K-Fold cross-validation to cross validate our results.

Figure 7. K-Fold Validation for Random Forest Classification

Figure 8. K-Fold Validation for Decision Tree Classification

K-Fold ensures that the reported performance is a realistic estimate of the model’s ability to generalize. By applying the same K-Fold process to all models, one can compare their robustness and accuracy fairly. Low variability in performance (low standard deviation) indicates the model is reliable and suitable for real-world deployment.

In our case we can confirm that our accuracies of the models are very well complemented by K-Fold Validation Technique and it can be inferred from this that the models are statistically accurate and trustworthy.

From above accuracies we can state that Random Forest Model is the most accurate model for our binary classification case scenario with an accuracy of 98.73 percent, which is impressive in itself.

# A screenshot of a graph Description automatically generatedFIGURES : Correlation and Histograms

Figure 9. CORRELATION MATRIX

A graph of a bar chart

Description automatically generated with medium confidenceA graph of a bar chart

Description automatically generated with medium confidence  
  
  
A graph of a bar graph

Description automatically generated with medium confidence  
A graph of a bar chart

Description automatically generated with medium confidence  
A graph of a bar graph

Description automatically generated with medium confidence  
  
  
  
  
  
  
  
  
  
  
  
  
A graph of a bar graph

Description automatically generated with medium confidence  
A graph of a bar graph

Description automatically generated with medium confidenceA graph of a person and person

Description automatically generated with medium confidence