Safeguarding What You Click

Somanshu Mahajan
2023800053 A4

Phishing and legitimate URL Detection

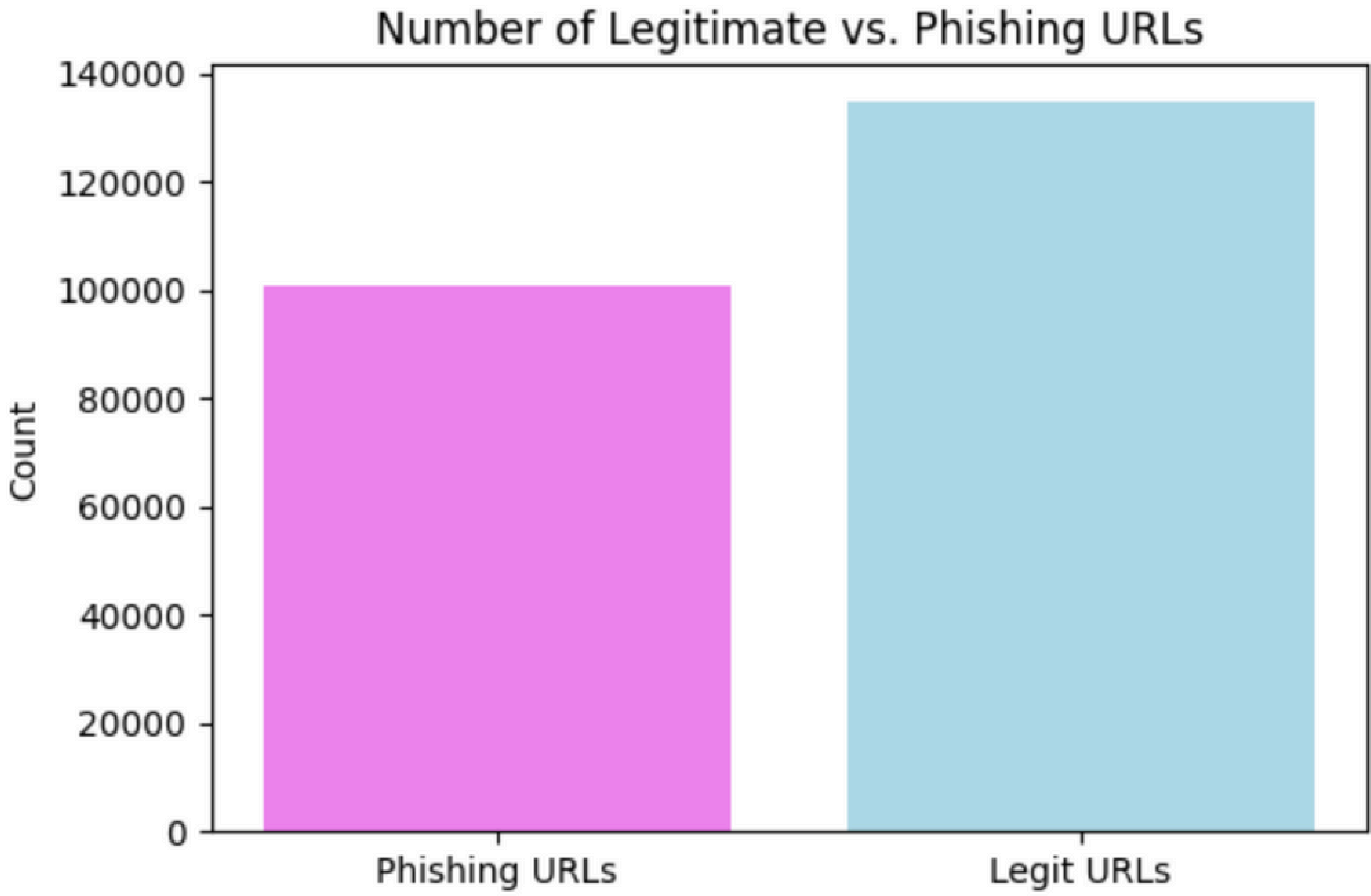
Introduction

The dataset contains set of URLs classified as legitimate and phishing.

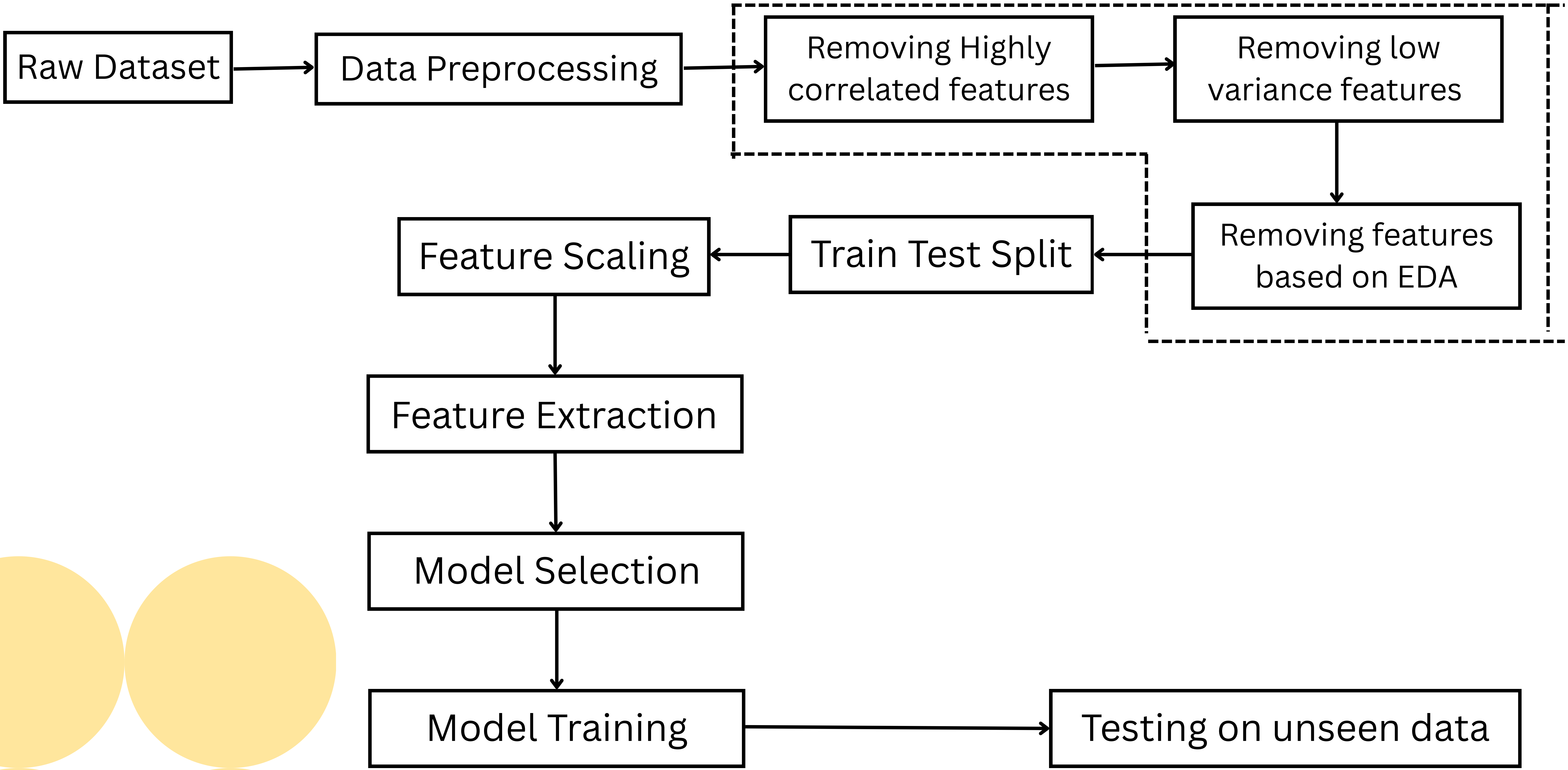
The dataset is outsourced from Kaggle. This dataset was contributed by **Arvind Prasad** and **Shalini Chandra** as their part of work on a technique of phishing URL detection.

This dataset has many URL features extracted based on:

- Metadata Features (Responsiveness, Favicon, etc.)
- Script Based Features (Lines of Code, etc)
- Length Based Features (URL length, TLD length, etc)
- Domain Features (TLD Legit Probability, etc)
- Title Features (Title Match Scores, etc)
- Security Features (Robots, Crypto, HTTPS, etc)
- DOM-Based Features (Redirection, Count of JS Files)



Workflow


Raw Dataset
Data Preprocessing
Removing Highly correlated features
Removing low variance features
Removing features based on EDA
Train Test Split
Feature Scaling
Feature Extraction
Model Selection
Model Training
Testing on unseen data

Data Preprocessing

NULL VALUES

No null values present.

DUPLICATES

No Duplicates Present.

DROPPING COLUMNS

Categorical columns dropped except TLD column.

ENCODING

Frequency encoding done for columns TLD containing the Top-Level Domain of all URLs.

OUTLIERS

Almost quarter of outlier present in 3-4 columns. No outlier handling performed due to impact on correlation.

DATA TYPE CASTING

Reduced the datatype of numeric columns from int64 to int32 and int8.

Feature Extraction

Random Forest Classifier

Feature importance by measuring performance drop after feature shuffling.

Permutation Importance

It works by measuring how much the model's performance decreases when the values of a particular feature are randomly shuffled.

Extra Trees Classifier

Random forest variant; more randomized tree building for decorrelation.

How features are ranked?

The features selected by each method where arranged in the manner of their rank in their respective scale and the average of rank of feature was taken and arranged in ascending specifying least rank, more importance.

Averaging importances of three methods was not used because each method use different importance scale.

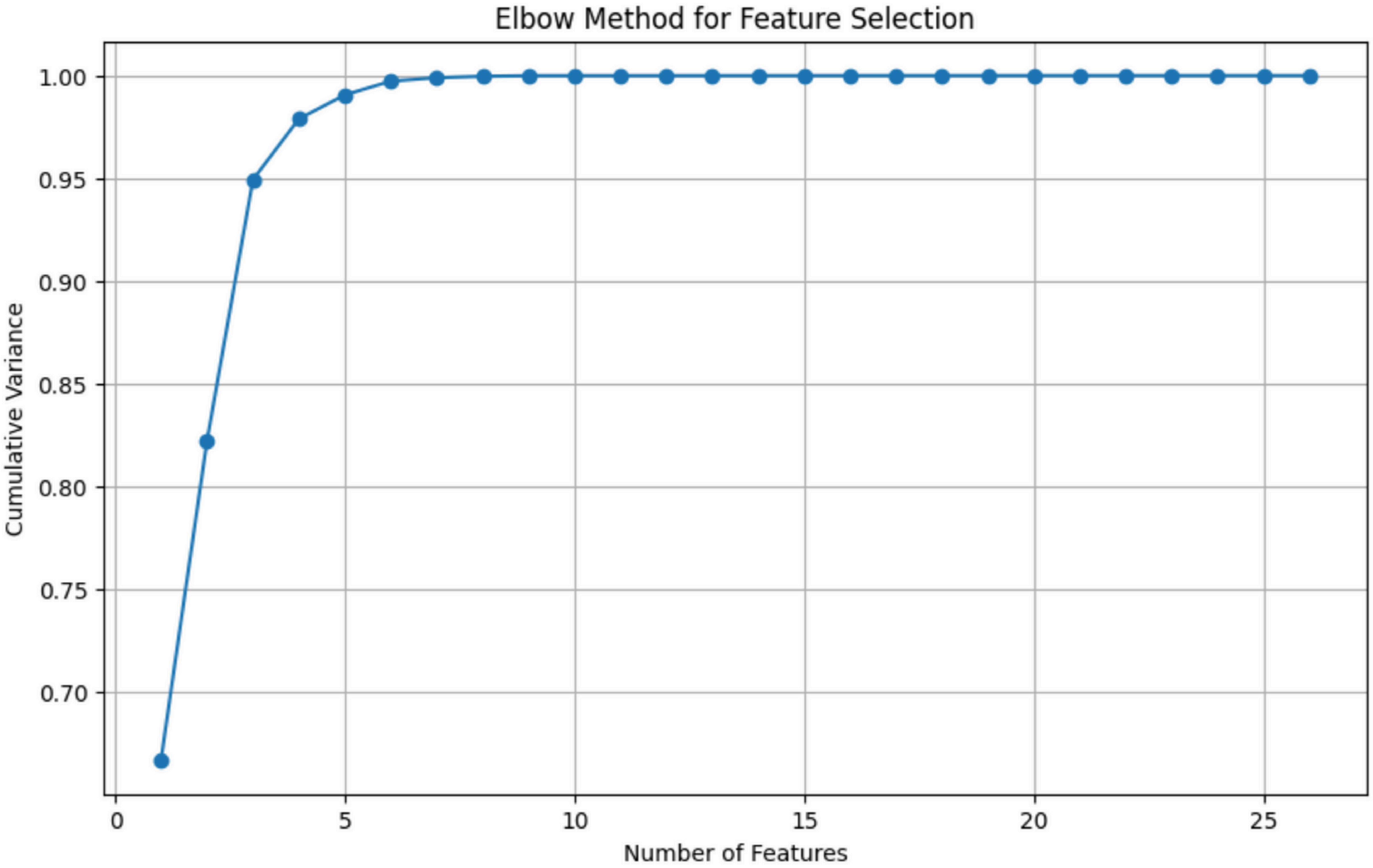
Top K features

To determine how much top features we take so that it does not cause overfitting was done using elbow method by plotting number of features against cumulative variance.

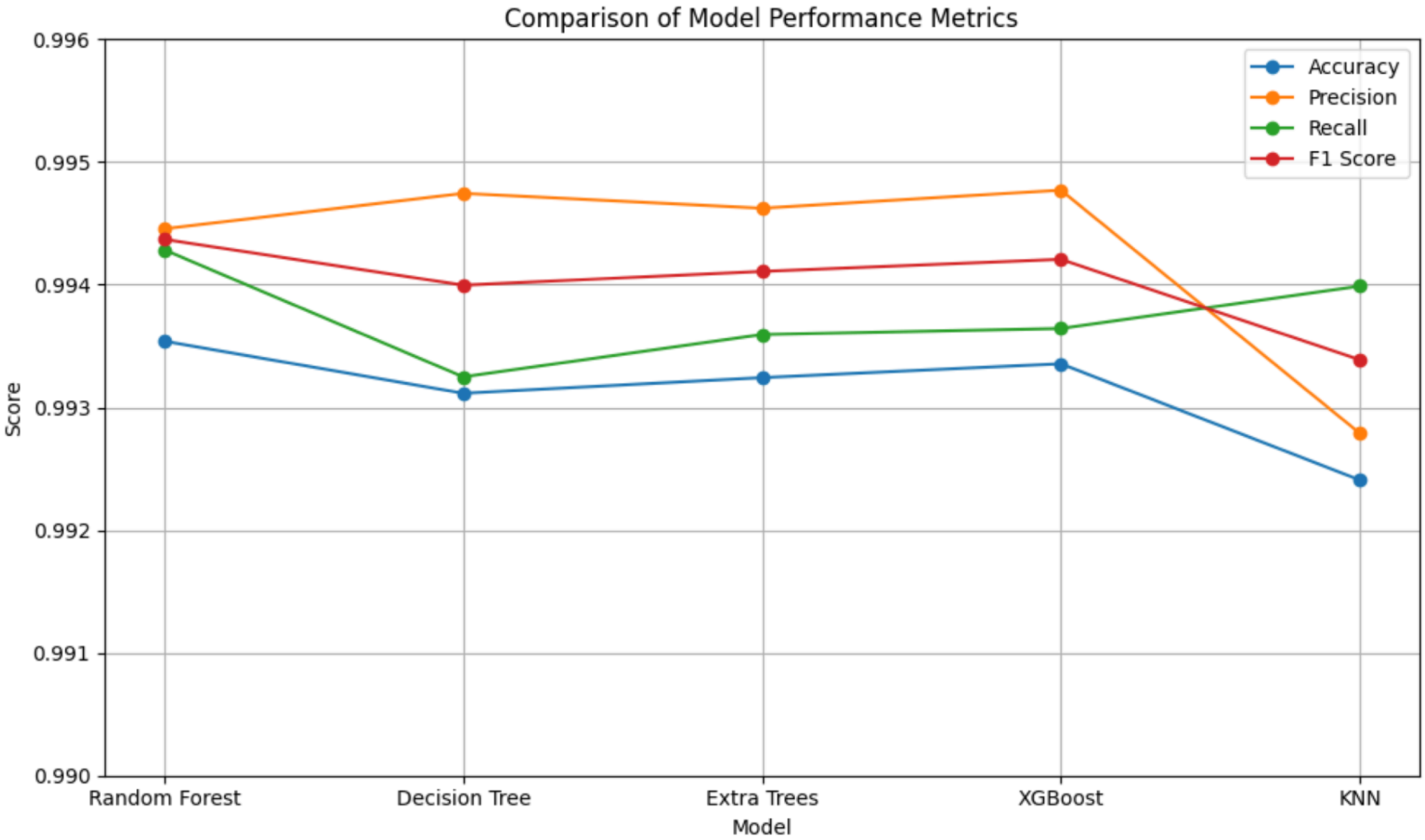
By the graph it was decided to take 6 features because beyond that it does not improve the model performance.

Top 6 features :

- HasSocialNet
- IsHTTPS
- HasCopyrightInfo
- NoOfSelfRef
- HasDescription
- URLLength



Evaluation Metrics of various Models



Hyperparameter Tuning

Hyperparameter tuning is needed to optimize model performance, balance bias and variance, speed up training, and ensure the model generalizes well to new, unseen data.

n_estimators :	200
max_depth :	20
min_samples_split :	2
min_samples_leaf :	1
max_features :	log2
bootstrap :	False

Best Combination of Hyperparameters

Reducing False Positive Rate

There are more false positives than the false negatives. One will prioritize false negatives over false positives. This is because it will be fine if a legitimate URL is marked phishing but not vice versa.

Threshold	False Positives	False Negatives
0.6	184	253

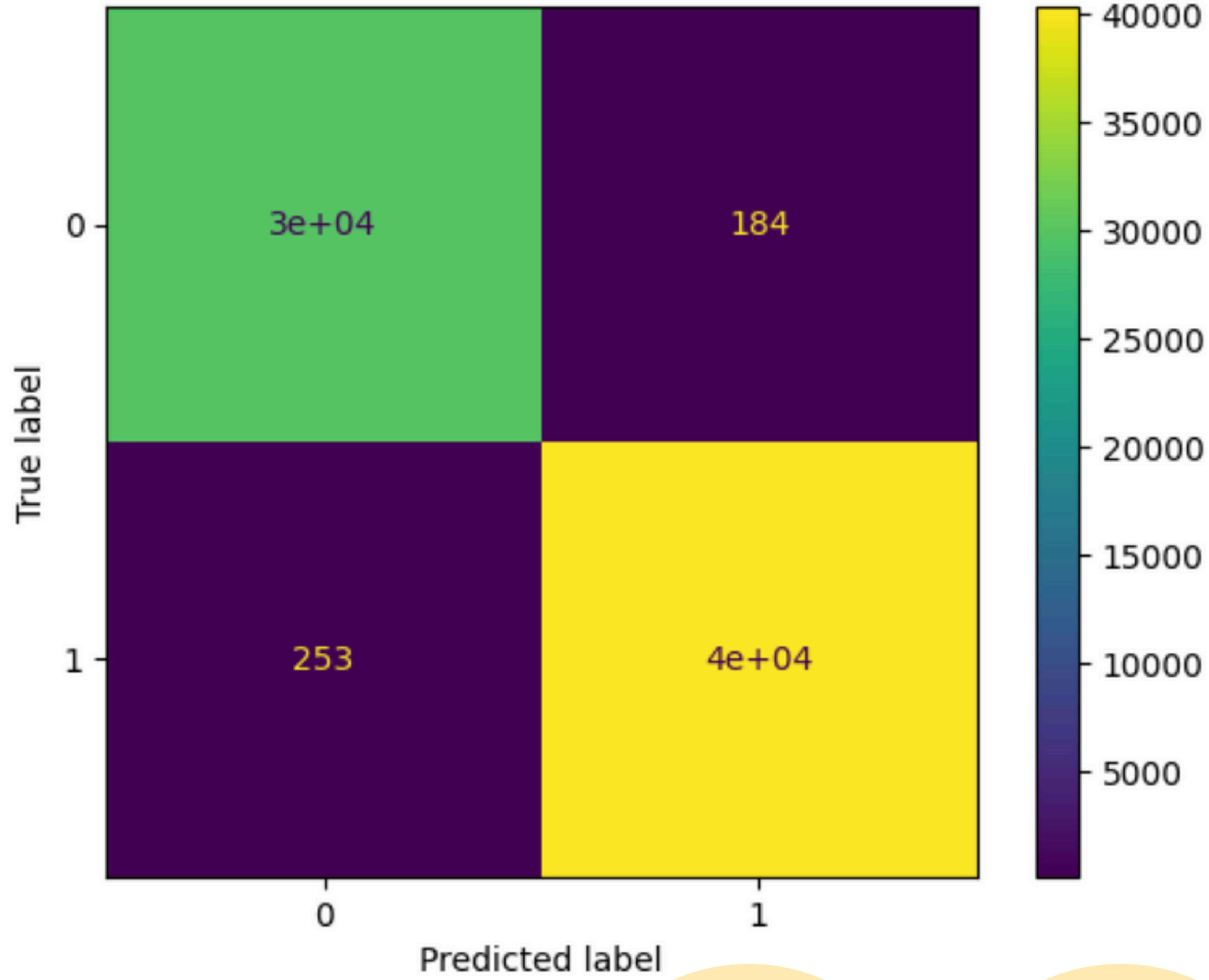
I have used threshold tuning to achieve this and got best results at 0.6.

Final Analysis

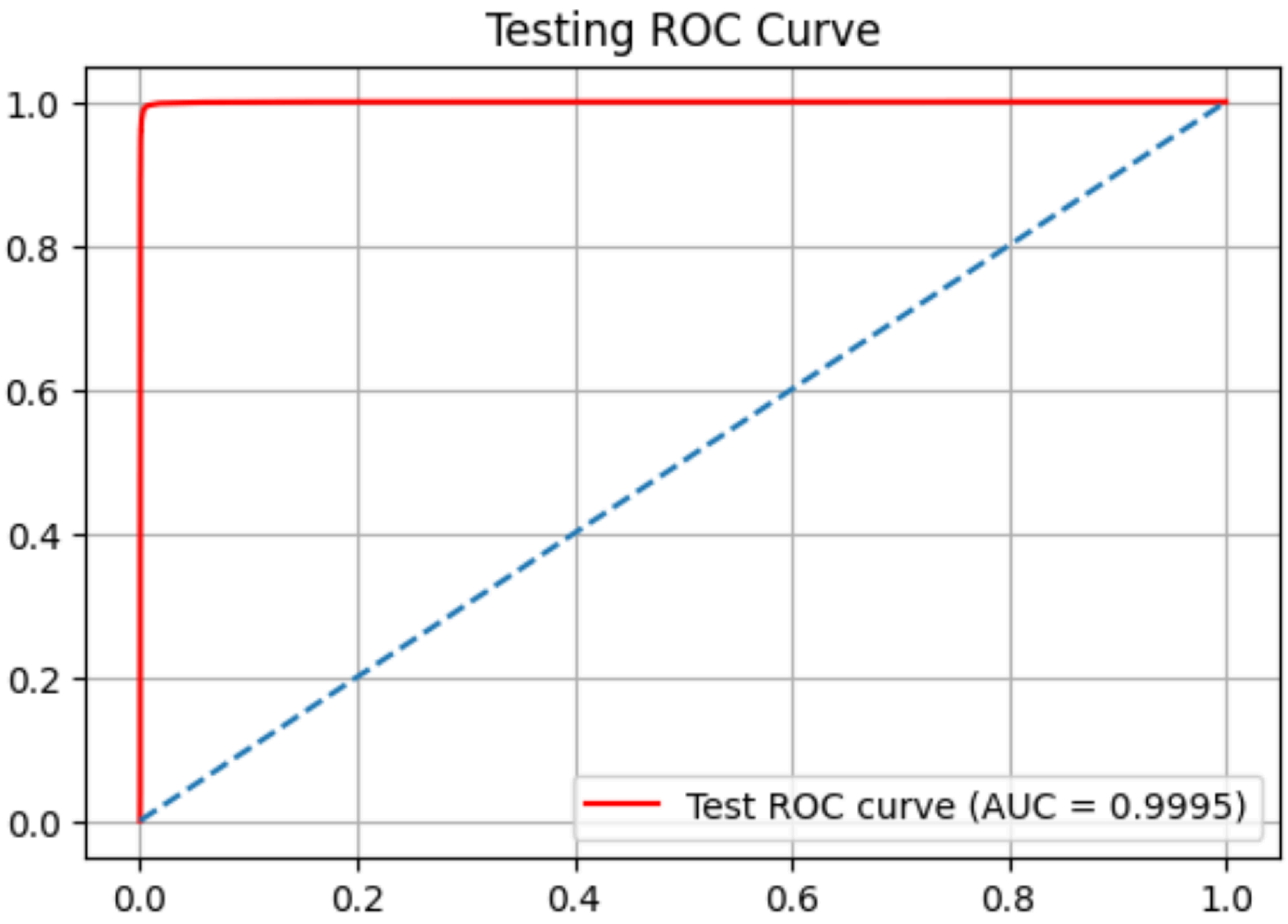
Training Accuracy : 0.9949

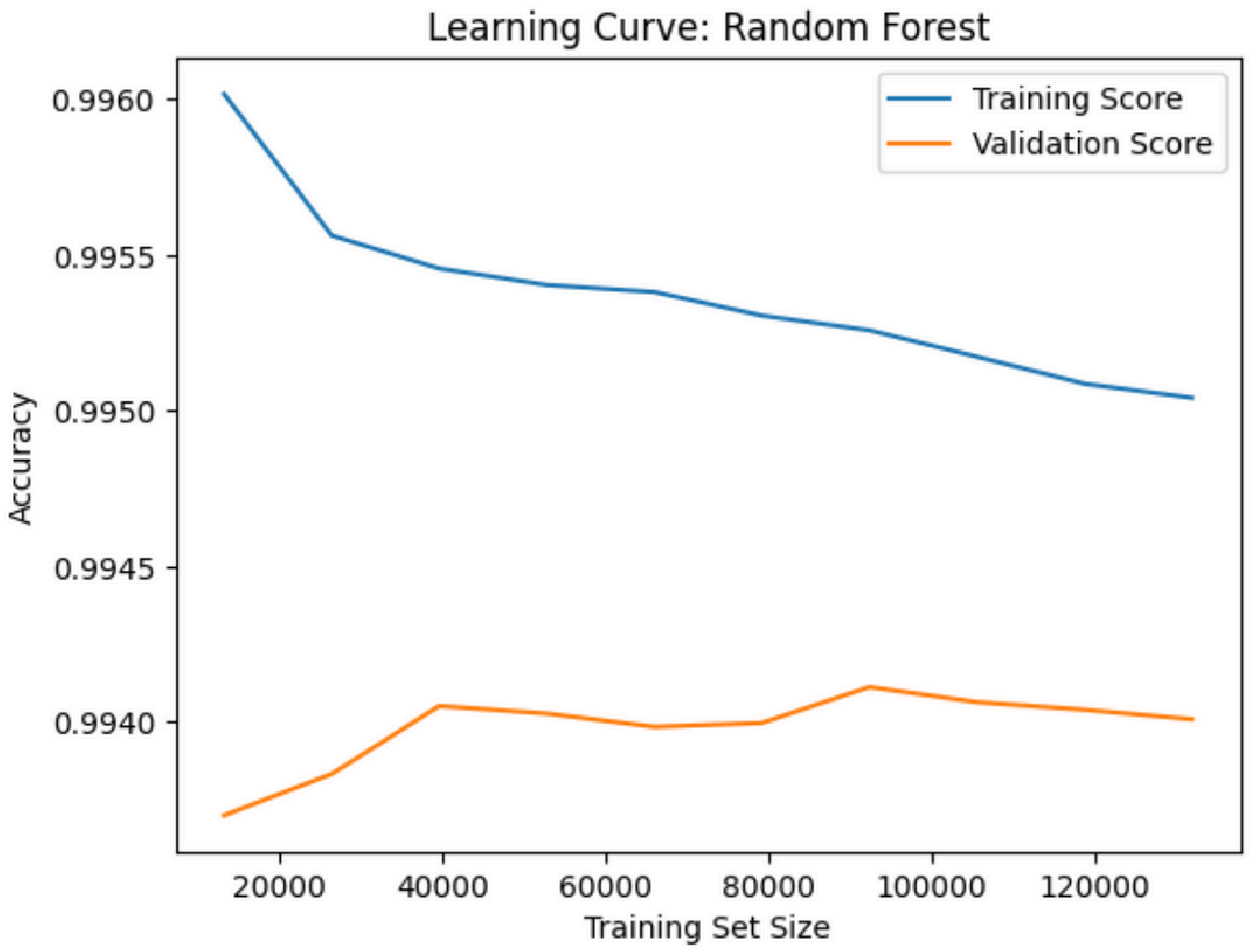
Test Accuracy : 0.9938

	0	1
Precision	0.9916	0.9955
Recall	0.9939	0.9938
F1-Score	0.9928	0.9946



The test accuracy has remained same though training accuracy has decreased by 0.01. This is a plus that the gap between training and test score is decreased again meaning the model is generalizing well.





The red line represents the ROC curve. The ROC curve is very close to the ideal curve, hugging the top-left corner. This shows that the model has a very high ability to distinguish between positive and negative classes.

The decrease in accuracy of training score as the set size increase is showing that model is generalizing well. Validation score is also increasing with set size which indicates it is generalizing good on unseen data. Also gap between both scores is less meaning model is not suffering overfitting. Both curves seem to plateau as the training set size increases.

Safeguarding What You Click

Somanshu Mahajan
2023800053 A4

Thank You

References:
Training Dataset : https://www.kaggle.com/datasets/joebeachcapital/phiusiil-phishing-url
Code File: https://github.com/Somanshu-Mahajan/Phishing-URL-Detection.git