PHISHING WEBSITE IDENTIFICATION

USING MACHINE LEARNING

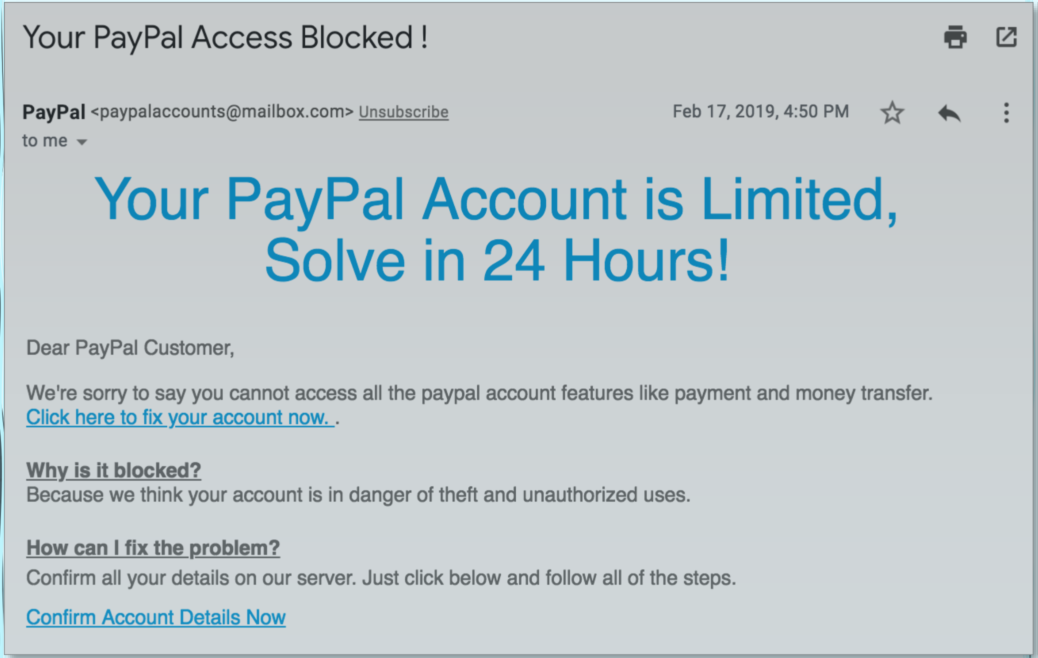
BEAMING TEAM - STOR 565 SPRING 2024
Ishmael Benjamin Torres Aguilar, Izzet Egemen Elver,
Dilay Ozkan, Malavika Mampally, Trung Nghia Nguyen

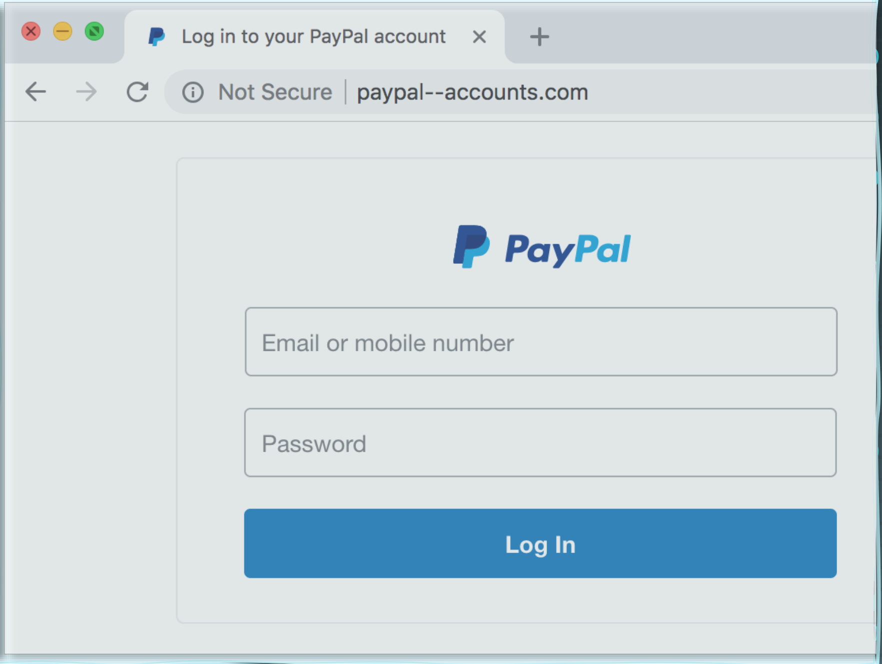
TABLE OF CONTENTS

01

INTRODUCTION & DATA DESCRIPTION

MOTIVATION





Wait... This looks like a phishing website!!!

MOTIVATION

- Phishing: *"persuade potential victims into divulging sensitive information such as credentials, or bank and credit card details"* (European Union Agency for Cybersecurity, 2024).

- Often occur over
  - malicious webpages
  - e-mails
  - instant messages that appear to be originating from a legitimate source.

MOTIVATION

- Even cyber experts are sometimes not able to identify a website with malicious intent.
  - Study shows that 97% of security experts fail to recognize phishing emails from genuine emails (Business Wire, 2015).

- Data from the FBI Internet Crimes Report illustrated that in 2022:
  - 300,497 phishing victims
  - Lost $52,089,159 just in the U.S. (FBI, 2022).

GOAL & BENEFITS

- Develop binary classification models to categorize if a web page has malicious objectives, i.e. phishing, or if it is, in fact, legitimate.
  - Traditional supervised methods: K-NN, SVM, logistic regression, tree-based methods, etc.
  - Neural-network based methods: MLP, TABNET.

- Benefits of this study:
  - Service providers/web browser developers
  - Authorities

DATA DESCRIPTION

11,430 URLs

legitimate

phishing

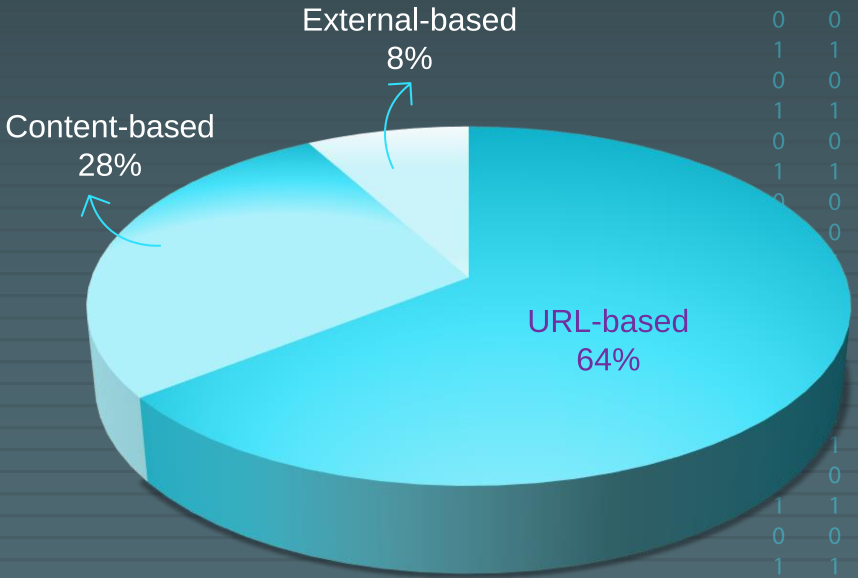
Balanced


Mendeley Data

Web page phishing detection

Published: 25 June 2021 | Version 3 | DOI: 10.17632/c2gw7fy2j4.3
Contributors: Abdelhakim Hannousse, Salima Yahiouche

- 87 extracted features based on three categories:
    - URL-based: contain structural and statistical features.
    - Content-based: split into hyperlinks and abnormal content-based features.
    - External-based: page rank, domain age, etc.

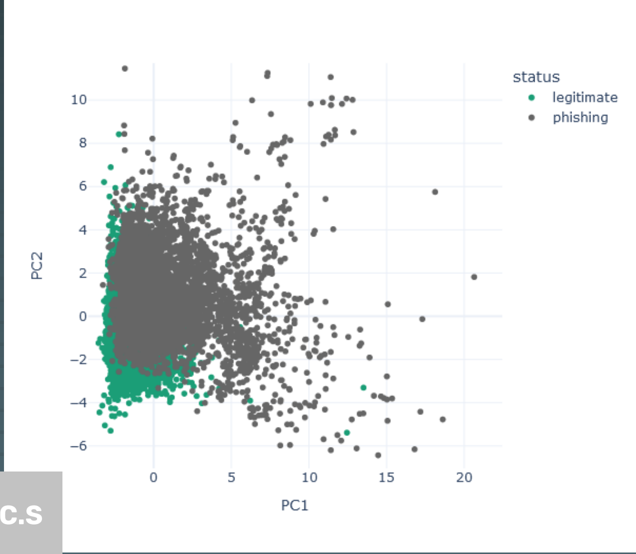


02

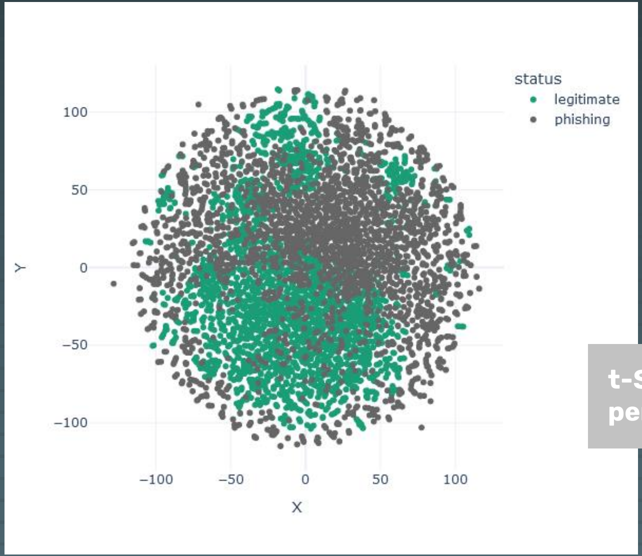
EXPLORATORY DATA ANALYSIS

Dimensionality Reduction

PCA and t-SNE were carried out to inspect how distinct the classes appear to be.



First 2 P.C.S



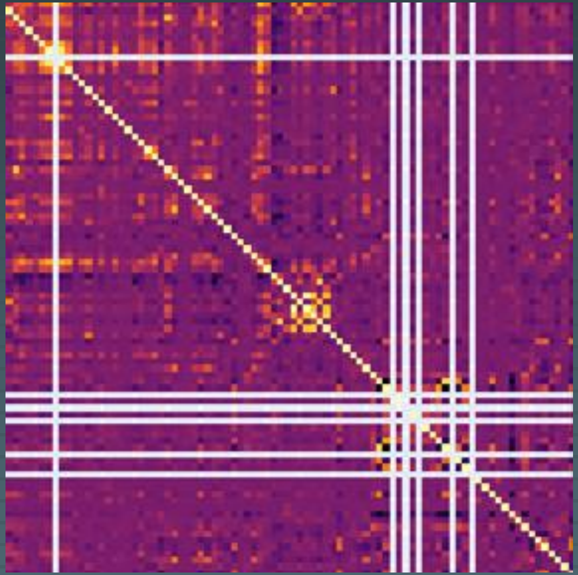
t-SNE with perplexity = 3

- Slight separation of classes observed with no clear distinction
- 52 components were required to explain 90% variance

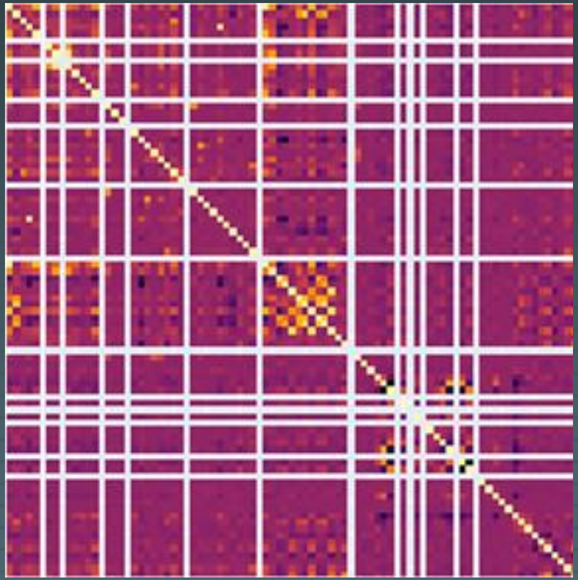
Correlation among variables

Presence of 80+ variables made it difficult to visualise all of the correlations and make a judgement.

PHISHING

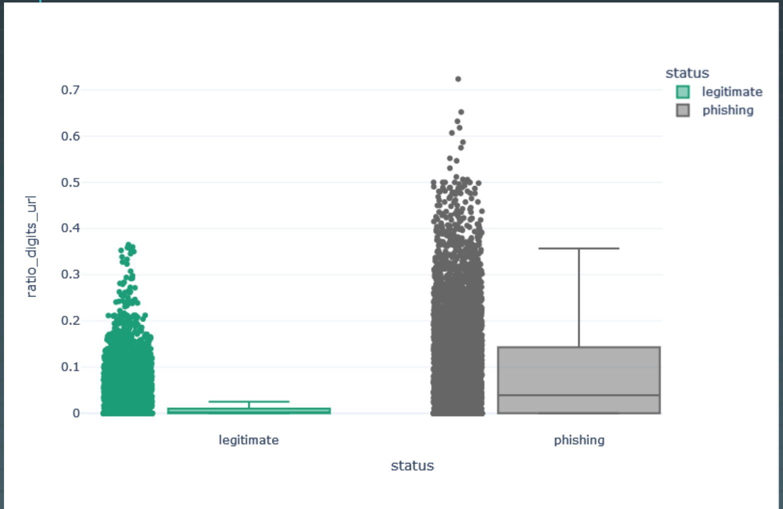


LEGITIMATE



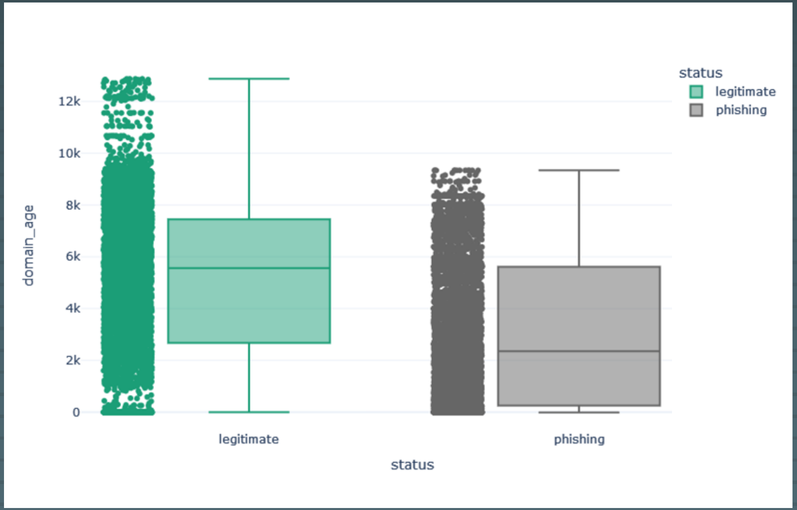
But the **white grid lines** indicate the presence of constant columns or variables with 0 variance. **THESE VARIABLES ARE QUICKEST DETECTORS OF LEGITIMATE WEBSITES!** For eg. nb_stars and nb_dollars

EDA Insights..



Number of digits were drastically more in phishing websites.

Age of the domain was greater for legitimate websites which suggests that phishing websites are more recently created.

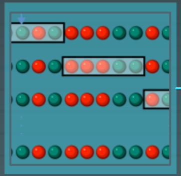


03

MODELS & RESULTS

K-Nearest Neighbors

LOOCV



Cross validation over a range of values of k
k= 1 to 40

Optimum Value

k

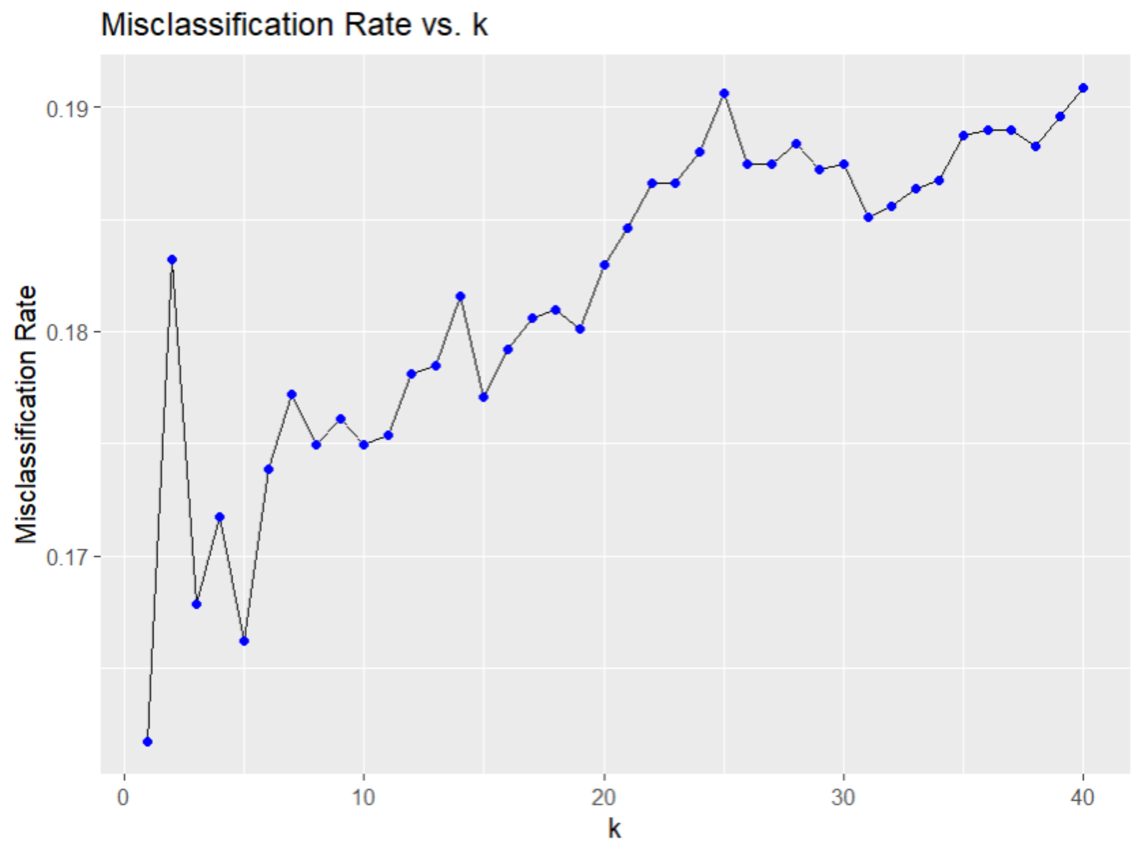
Choose k based on mis - classification rate

Model fitting

Use the chosen k value and interpret accuracy

Test

Finally check how it works on the test data



Acceptable values of k are 1, 3 and 5.

For k=1

Accuracy	84.37%
Precision	83.65%
Recall	85.51%
AUC	84.39%

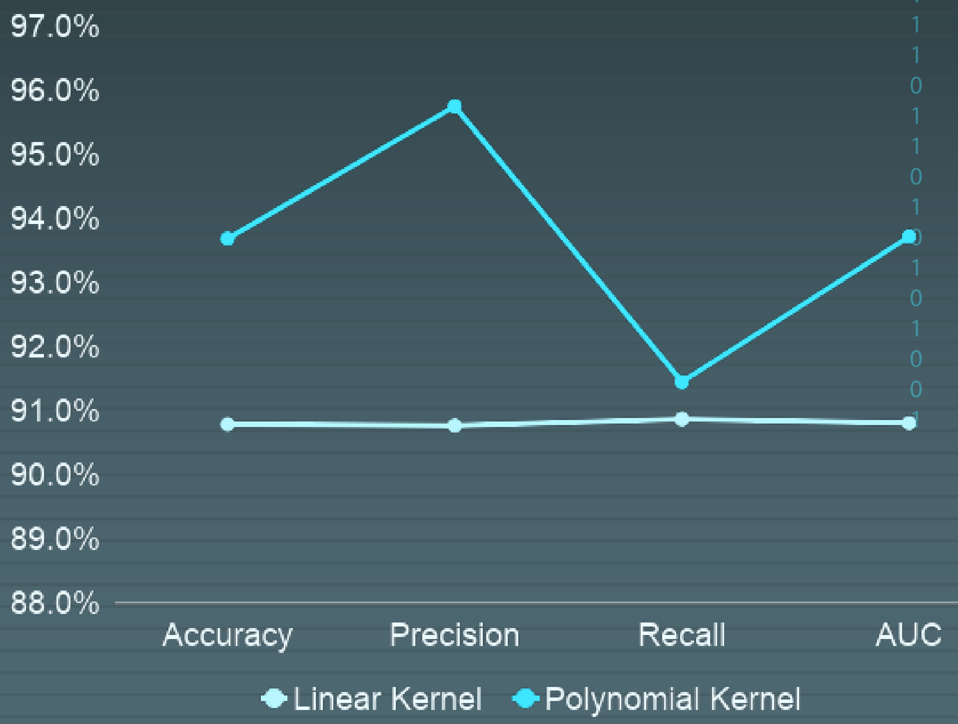
Support Vector Machines

Linear Kernel: cost = 0.5 and 1552 support vectors

cost	error	dispersion
0.1	0.07461642	0.006856030
0.5	0.07311626	0.006412525
1	0.07361626	0.006552537

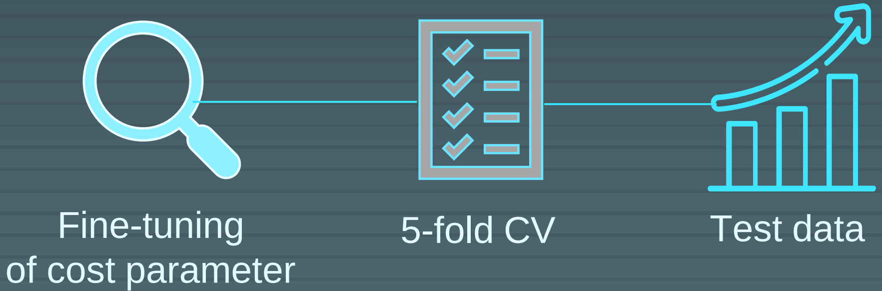
Polynomial Kernel: cost = 1, degree = 3 and 2803 support vectors

cost	degree	error	dispersion
0.5	2	0.09373611	0.008662970
1	2	0.08448814	0.007509416
0.5	3	0.07298861	0.007990297
1	3	0.06299017	0.008019329
0.5	4	0.18410190	0.014801183
1	4	0.14048049	0.012321569



Logistic Regression

- Logistic regression with no regularization
- LASSO logistic regression
- Ridge logistic regression



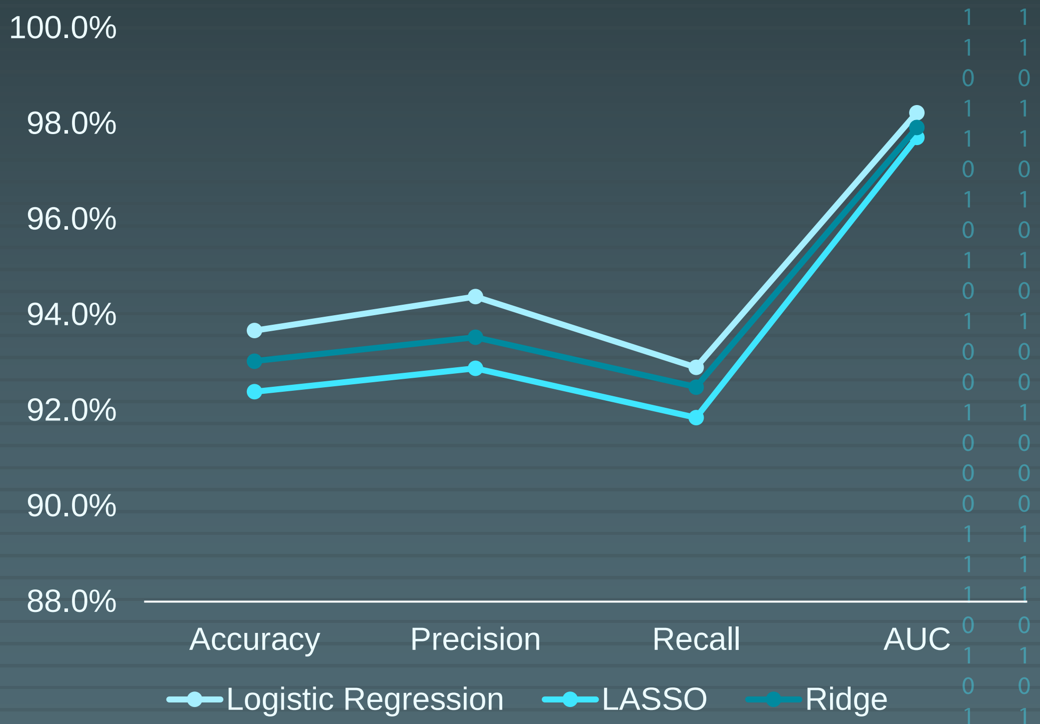
Logistic Regression

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

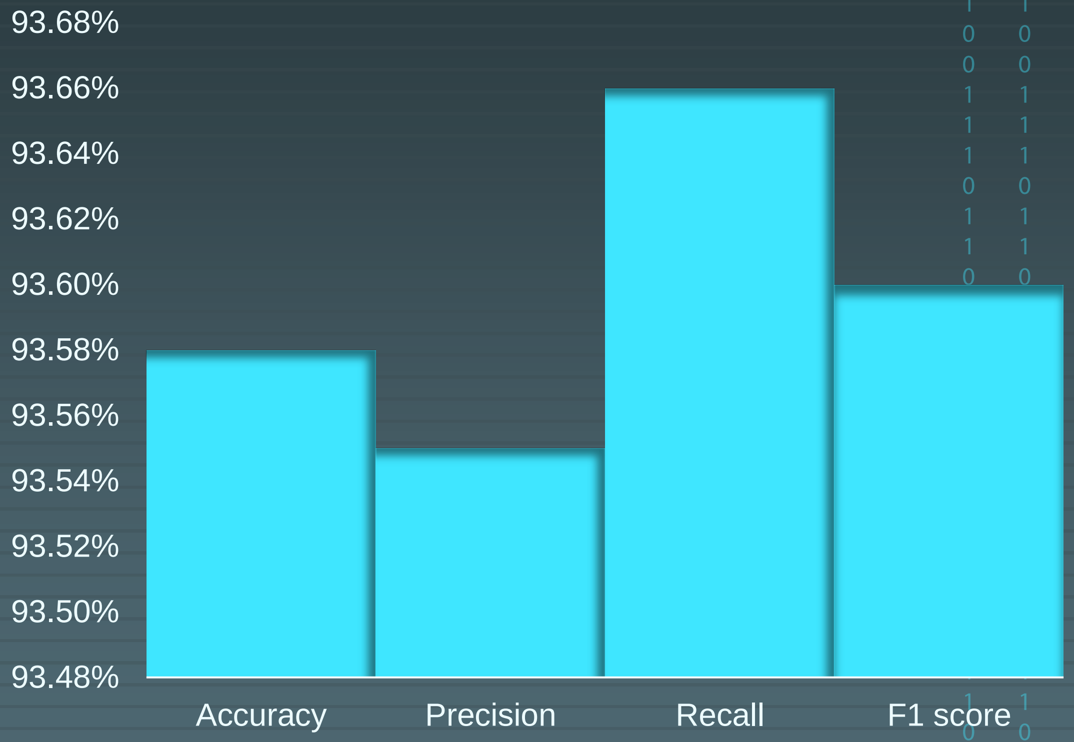
- Legitimate URLs

+ Phishing URLs



Decision Trees

- **Classification Trees**
  - cost complexity pruning



XGBOOST

Random Search

Generated random set of hyperparameter s

Optimum Hyperparamat ers

Choose the best model on validation data

Model fitting

Use best model on validation

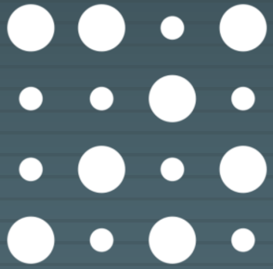
Test

Finally check how it works on the test data

XGBOOST

Tuning on model description

We looked with different models with d**ifferent complexities and flexibility** with parameters such as maximum depth of weak learners, shrinkage size etc.



We found **over 1,000 different models** using random search to obtain the final model evaluating on validation set with a split 60 train, 20 val, 30 test.

XGBOOST

Performance on test data

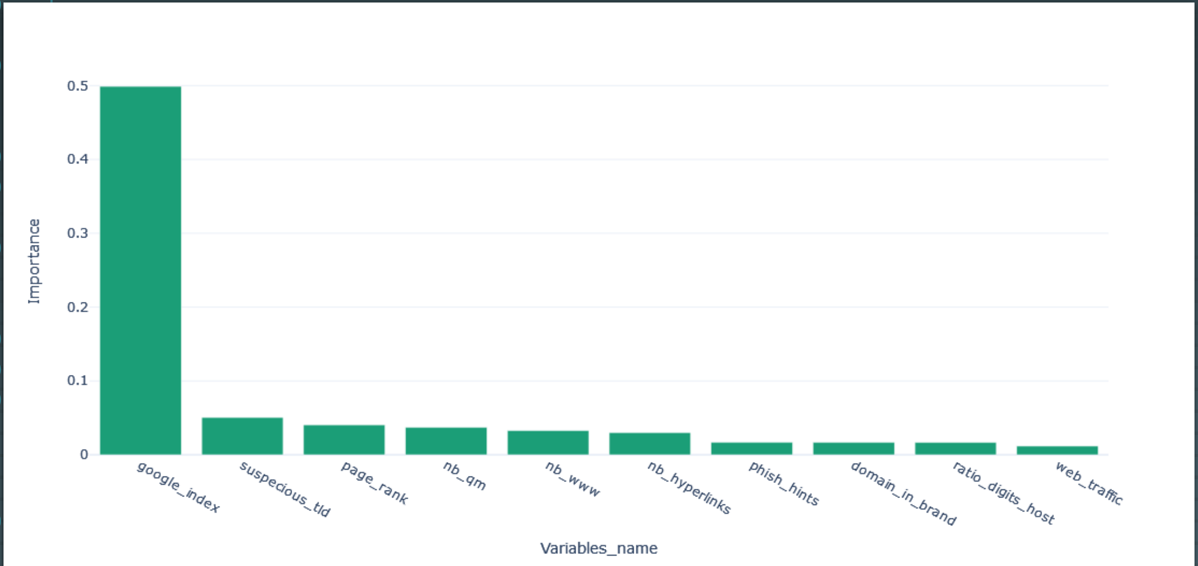
A preliminary performance on validation data was around 98%.

The final performance on test data was **96.2%**

Accuracy	Precision	Recall
96%	97%	96%

XGBOOST

Most importance features



Google Index represent around 50% of the feature importance between 87 features.

This could represent a **robustness issue** as it depends mostly on Google's algorithm.

Random Forest

4 Fold Cross Validation

Less parameter space did not require random search

Optimum Hyperparamaters

Choose the best parmeters

Model fitting

Use best model found by CV

Test

Finally check how it works on the test data

Random Forest

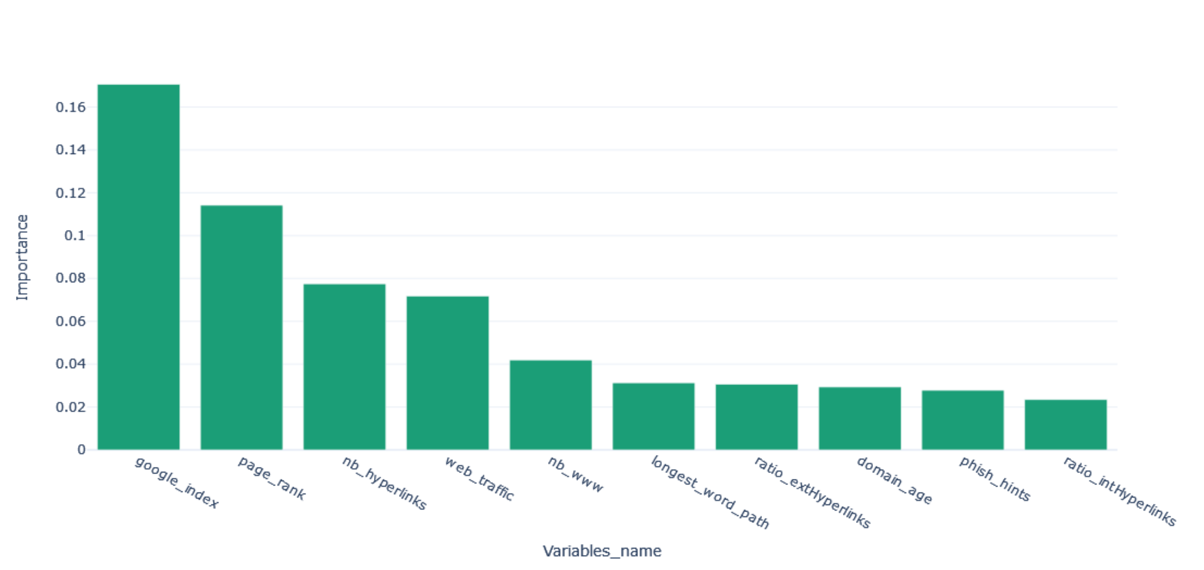
Performance on test data

The final performance for this model on test data was **96.29% the higher recorded across methods.**

Accuracy	Precision	Recall
96.29%	97%	96%

Random Forest

Most importance features

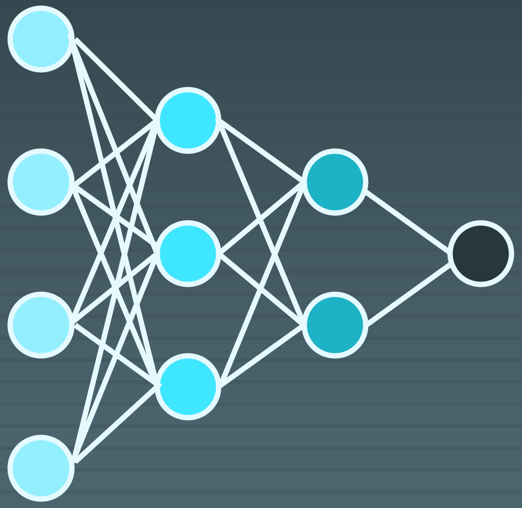


This model is a **more robust predictor** as the importance is more distributed among variables.

Nevertheless, **Google variables still represents 26% of the importance**.

Multilayer Perceptron

Model Description



- Input layer: 87 nodes, output layer: 1 node
- ReLU activation between layers
- `Batchnorm` for data normalization
- Sigmoid activation at output layer
- Dropout regularization (p = 0.2)

Model	Number of hidden layers	Number of nodes per each hidden layer
Model 1	1	(300)
Model 2	2	(300, 100)
Model 3	3	(500, 300, 100)
Model 4	4	(500, 300, 100, 50)

Multilayer Perceptron

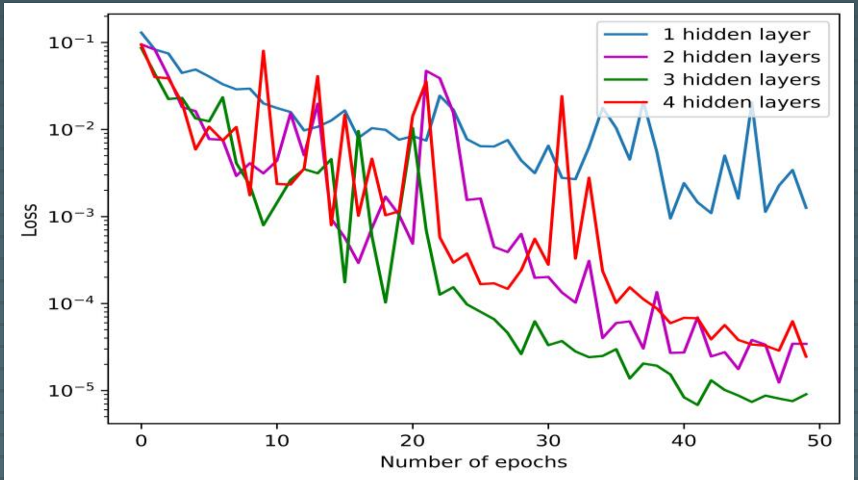
Training Parameters and Results

- Binary Cross-Entropy (BCE) loss:

$$L_{BCE} = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log\left(p(y_i)\right) + (1-y_i)\log(1-p(y_i))\right].$$

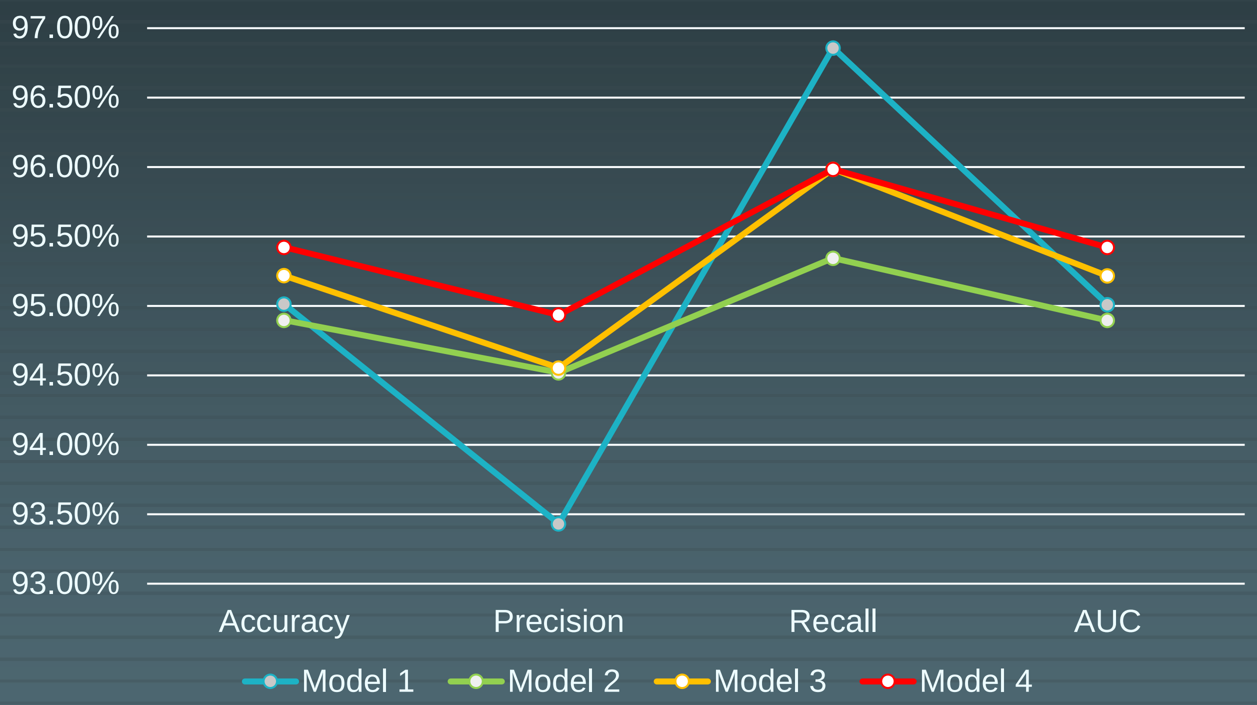
- Adam optimizer, learning rate = 0.001.
- 50 epochs.

Model	Last-iteration loss value
Model 1	0.0012583367060869932
Model 2	3.4345404856139794e-05
Model 3	9.056506314664148e-06
Model 4	2.4474804376950487e-05



Multilayer Perceptron

Performance on Test Dataset



Best MLP accuracy on test dataset: **95.42%** (4-hidden-layer model).

04

CONCLUSIONS & LIMITATIONS

Conclusions and Limitations

- Random forest outperforms other methods for its flexibility and high power for tabular data.
- Most of the algorithms heavily depends on the variables from external sources (Google index and Page rank).
- This may be an issue as the predictors depend on a third source algorithm.
- The dataset was balanced, but in real life legitimate websites are much more abundant than phishing websites.

THANKS!

Do you have any questions?