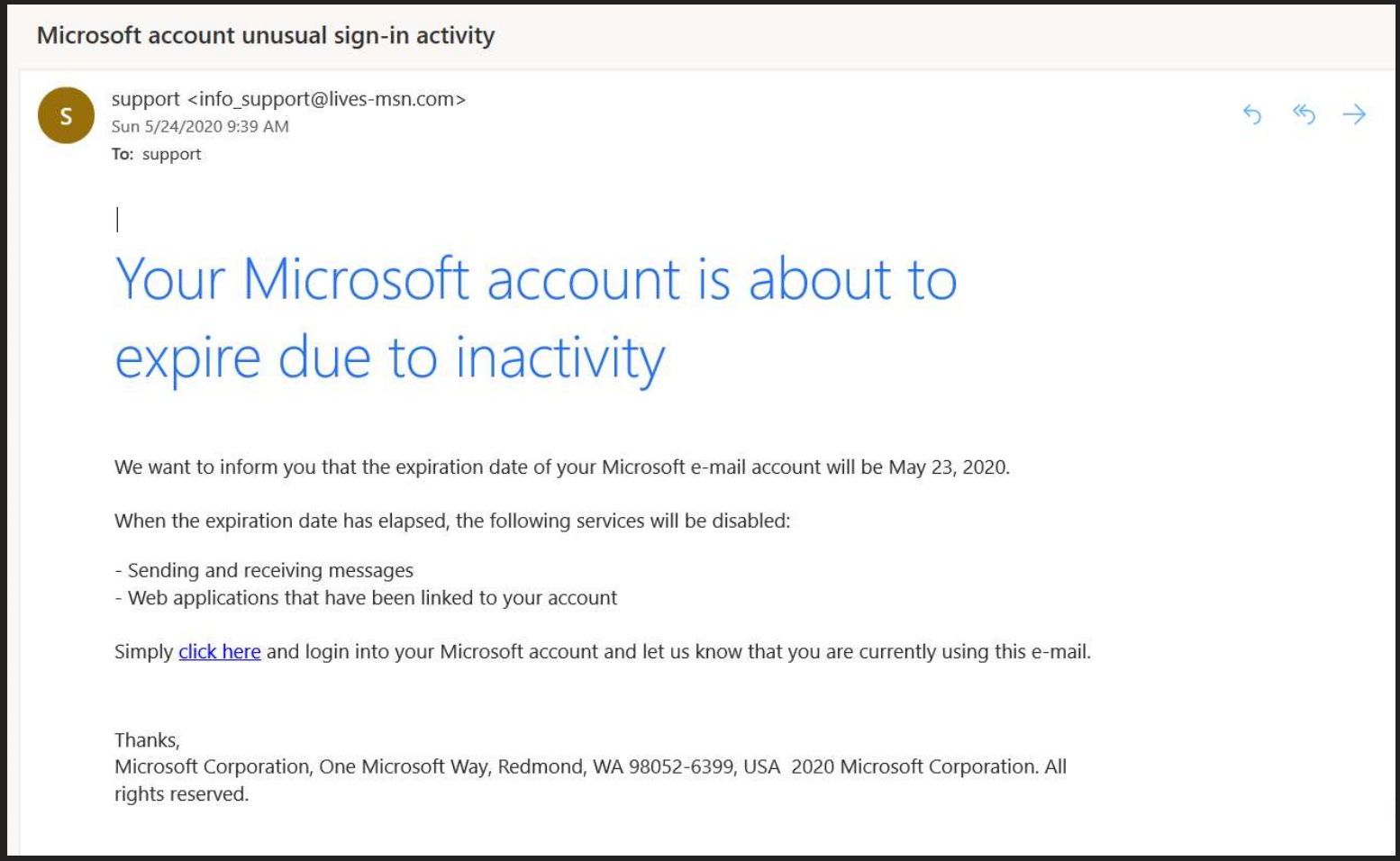
DETECING PHISHI
WEBSITES USIN
MACHINE LEARNI

Zach Fechko

MOTIVATION



Microsoft account unusual sign-in activity

S

support <info_support@lives-msn.com>
Sun 5/24/2020 9:39 AM
To: support

Your Microsoft account is about to expire due to inactivity

We want to inform you that the expiration date of your Microsoft e-mail account will be May 23, 2020.

When the expiration date has elapsed, the following services will be disabled:

- Sending and receiving messages
- Web applications that have been linked to your account

Simply click here and login into your Microsoft account and let us know that you are currently using this e-mail.

Thanks,
Microsoft Corporation, One Microsoft Way, Redmond, WA 98052-6399, USA 2020 Microsoft Corporation. All rights reserved.

- As someone who uses the internet on a daily basis, I've gotten my fair share of phi
- I wanted to see if there was a way to detect "phishy" websites using machine learn
- Having a tool that tells you if a website is a phishing website would be a huge bene and organizations.

ML PROBLEM AND TECHNICAL APPROACH

The goal of this project is to find the optimal machine learning model that can detec
the highest accuracy possible based on the features of the website and its URL.

Technical approach

1. Gather and preprocess dataset
2. Train each model on training set and test them on the testing set
3. Gather accuracies and store for later comparison

WEBSITE DATA

- The dataset comes from Kaggle
- It contains the domain of, and information about the features of 10000 websites and classifies them as either phishy (

Domain	Have_IP	Have_At	URL_Length	URL_Depth	Redirection	https_Domain	TinyURL	
tobogo.net	0	0	1	2	0	0	0	0
teat09.com	0	0	0	3	0	0	0	0
depositphotos.com	0	0	1	1	0	0	0	0
superuser.com	0	0	1	3	0	0	0	0
web.de	0	0	1	6	0	0	0	0

- The dataset is broken up into 3 main chunks
  - Address bar based features
  - Domain based features
  - HTML/JavaScript based features
- I randomly sampled the dataset into a training and testing set along an 80/20 split
  - Training set made up of 8000 entries
  - Test set made up of 2000 entries

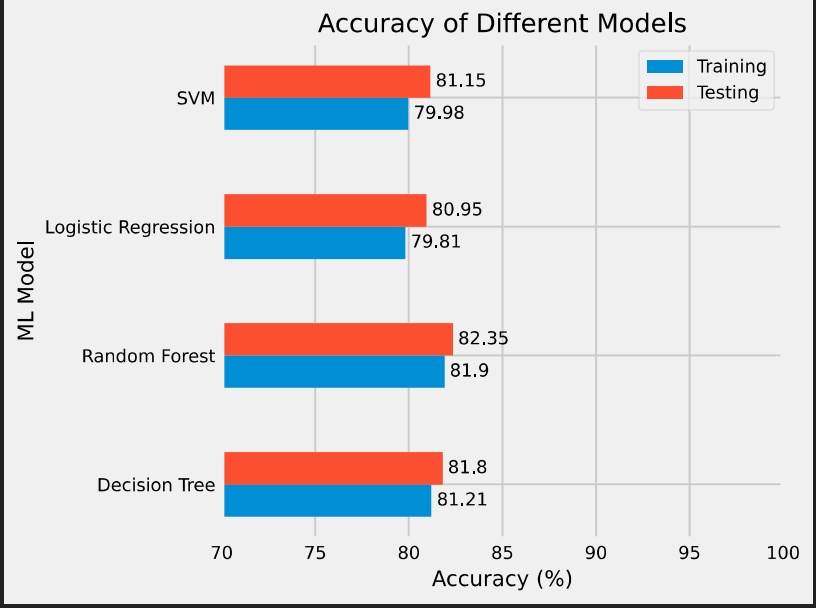
MODELS USED

- Decision Tree
  - Used multiple depths and found that `maxdepth = 5` was the most optimal
- Random Forest
  - Used multiple depths and found that `maxdepth = 5` gave the best results
- Binary Logistic Regression
  - Used 1000 iterations
- SVM
  - Used Linear kernel with regularization parameter `C = 1`

MODEL EVALUATION & RESULTS

- Each model is fit on the training set and then evaluated on the testing set, where its accuracy score on both the trainir compared after all evaluation is complete

Graph | Table



Out of all the models the Random Forest had the highest training and testing accuracy

- Training Accuracy of 81.9%
- Testing Accuracy of 82.35%

Some models like the SVM and Logistic Regression show some slight underfitting with the testing accuracy being aroun accuracy

WHAT'S NEXT?

- Build a new dataset using entries from open-source datacenters like PhishTank
- Try deep learning models
  - Neural Networks
  - Multilayer perceptrons