VALIMAIL

TRUST YOUR EMAIL

Data Science Practicum Final Presentation

Jialiang Shi, Joy Qi

22 May 2019

Master of Data Science, University of San Francisco

Agenda

- Summary
- Roadmap & Our Storyline
- Data Collected & Web-Scraping
- Model approaches & Comparisons
- Report on Distributions
- Future Work

Summary

TRUST YOUR EMAIL

Accomplished 6 Data Science Sprints

- October 2018 to May 2019

Developed Web-scraping scripts on:

- Social Media links
- Wappalyzer tool (Web technologies)
- Security Trails (Historical WHOIS records)

Built classification models:

- On Random Forest and Logistic Regression
- With 95% accuracy on the Unknowns
- Automating the rule-based manual process

Initial Roadmap

Orientation & Project Context	DNS data familiarity	Research and recommend additional data	Augmenting dataset, start automation	Machine learning Model POC
Understand the Context of the Project What does Valimail do as a business? What is the scope and objective of Defend? How does domain classification fit into Defend?	Get familiar with the DNS data from scanning domains Understand correlations between the good and bad domains.	Research additional data points and tools that can be useful in predicting good or bad domains Recommendation for these additional data sets as to usefulness in resolution	Augment DNS data based on recommendation from phase 3 with scripting tools Identify and automate signals that can be automated, and what cannot be	Train machine learning algorithms on the curated dataset you have built Test it on unknown domains Recommend a model for production

Our Storyline

TRUST YOUR EMAIL

Sprint 1 | Sprint 2 | Sprint 3 | Sprint 4 | Sprint 5 | Sprint 6 | Future Sprint: 7

Research & Learn

25/Nov/18

1
Understand monitored domains
Research on features
Identify correlations

7/Jan/19

2
Additional data signals
Integrate datasets

Explore & Finetune

31/Jan/19

3
Integrate scraped data
Researching on clustering

27/Feb/19

4
Integrate scraped data
Look at model performance

27/Mar/19

6
Explore Domain IQs
Augmenting dataset
Re-train the model
Test model on Unknowns

24/Apr/19

5
Report on distributions
Prediction & probabilities
Building Data Pipeline

Validate & Report

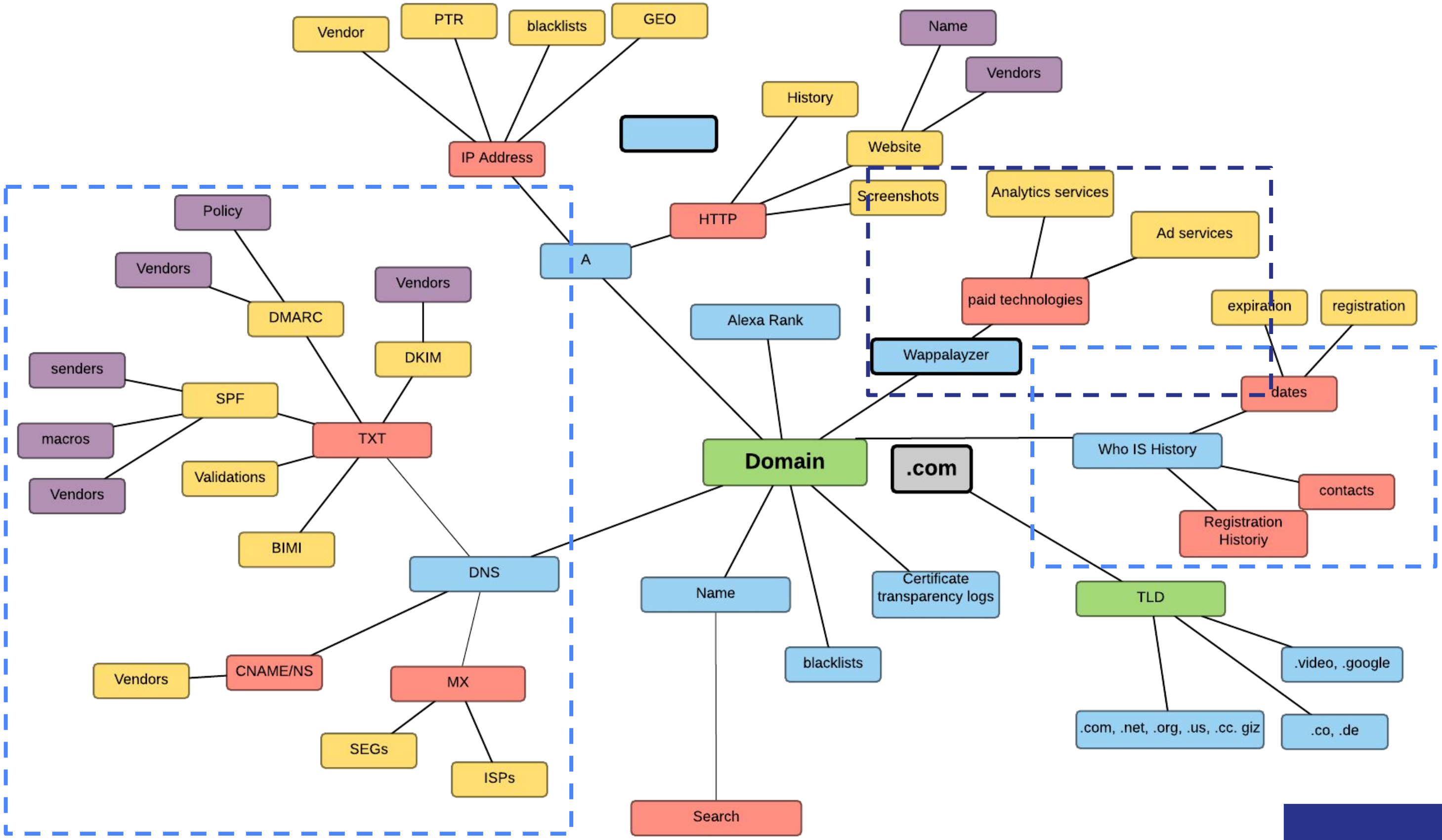
Present

7
Build data pipeline
Google Knowledge Graph
More features on scraped data

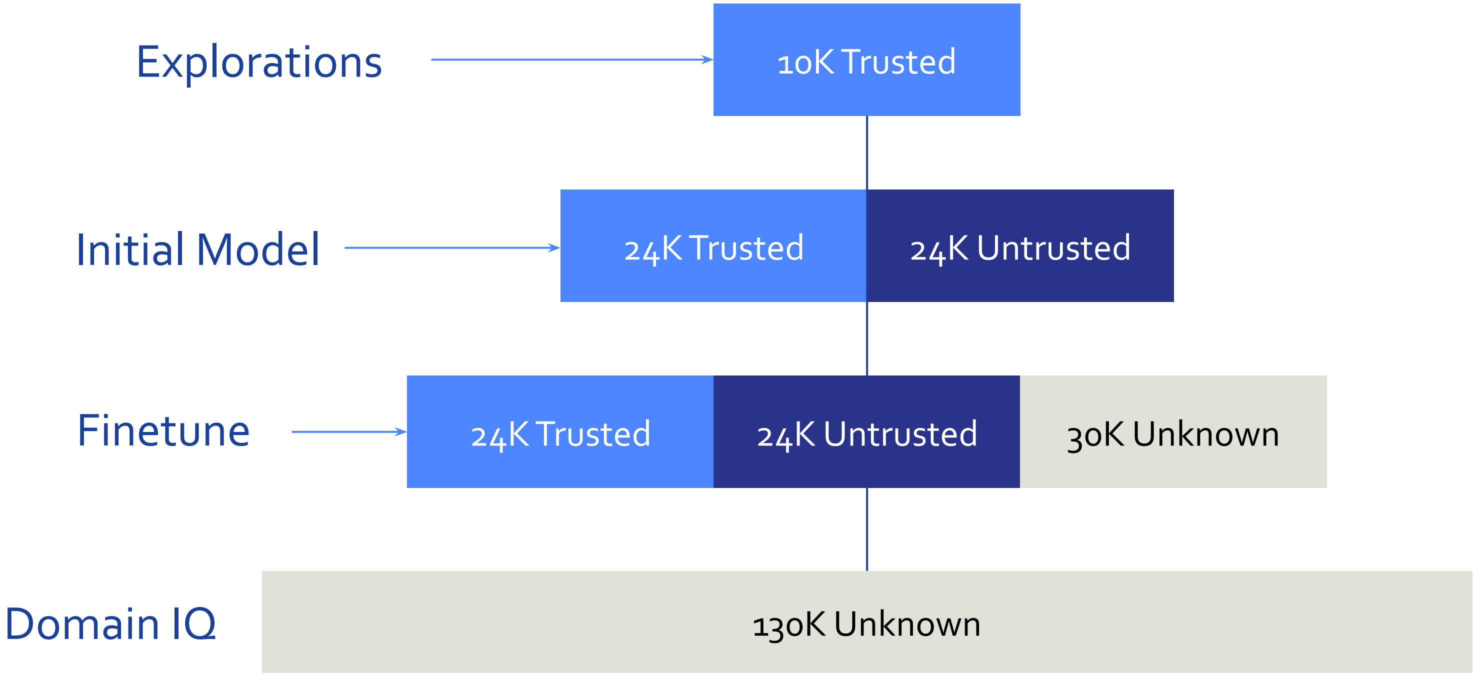
Practicum Ends
29/June/19

Data Collected & Web-Scraping

Domain Signals



Domain data used



Data Collected

130K Unknowns: DNS from Domain IQ + Web Scraped data

DNS

- DMARC
- SPF
- DKIM
- TLD
- ...

134 Boolean Columns

+

Social Media Links

- Linkedin
- Facebook
- Twitter
- Youtube
- Instagram

+

Wappalyzer, Security Trails

Wappalyzer

- List of web app
- web APPs exist
- Web APP counts

Security Trails

- List of security trails
- Security Trails exist
- WHOIS counts

Domain IQ

Web scraped Data

Web Scraping

Social Media Links

Approach	BeautifulSoup
Multiprocessing	24 pools
Script Speed	5 hours / 130k data

```python
from bs4 import BeautifulSoup
import pandas as pd
import requests
import re
from multiprocessing import Pool
import time
import random

def read_excel(file_path):
    domains = pd.read_csv(file_path)
    urls = ['https://www.' + u for u in domains['Domain']]
    return urls

def scrape(url):
    l, f, t, y, i = "", "", "", "", ""
    try:
        page = requests.get(url, timeout=10)
        soup = BeautifulSoup(page.text, features="lxml")
    except:
        return url, l, f, t, y, i

    try:
        l = soup.find('a', attrs={'href': re.compile("^https://www.linkedin.com")})['href']
    except:
        pass
    try:
        f = soup.find('a', attrs={'href': re.compile("^https://www.facebook.com")})['href']
    except:
        pass
    try:
        t = soup.find('a', attrs={'href': re.compile("^https://twitter.com")})['href']
    except:
        pass
    try:
        y = soup.find('a', attrs={'href': re.compile("^https://www.youtube.com")})['href']
    except:
        pass
    try:
        i = soup.find('a', attrs={'href': re.compile("^https://www.instagram.com")})['href']
    except:
        pass

    return url, l, f, t, y, i
```

Web Scraping

Wappalyzer, Security Trails

	Wappalyzer	Security Trails
API Quota	100k queries / month	200k queries / month
Multiprocessing	20 pools	20 pools
API Speed	6 hours / 130k data	5 hours / 130k data

```python
def get_page_tech(domain):
    """
    Given a single domain and using the api_key from wappalyzer,
    scrape the page technology application information
    return the result in a list of list
    containing [[[category0], app0, app1...], [[category1], app0, app1...]]
    """
    api_key = "GP93gT9IfV8KKx4fe0DIw46j4cbI1kq47ZFGfd1y"
    headers = {"X-Api-Key": api_key}
    print("begin", domain)
    r = requests.get(url="https://api.wappalyzer.com/lookup/v1/?url=https://"
                     + domain, headers=headers)
    results = r.json()
    if isinstance(results, list):
        latest_month = results[0]
        app_list = latest_month['applications']
        if len(app_list) > 0:
            df_app = pd.DataFrame.from_dict(app_list)
            dict_app = df_app[['categories', 'name']].to_dict('split')
            return dict_app['data']
```

```python
def get_security_trails(domain):
    """
    Given a single domain and using the api_key from security trails,
    scrape the page technology application information
    return the result in a list of list
    containing [[[category0], app0, app1...], [[category1], app0, app1...]]
    """
    api_key = 'JJUgjk3i9Q0vADQQsYWx8ekGbWc8u0Wb'
    headers = {'Content-type': 'application/json'}
    url = 'https://api.securitytrails.com/v1/domain/' \
          + domain + '/associated?apikey=' + api_key
    print("begin", domain)
    r = requests.get(url, headers=headers)
    trails_dict = r.json()
    key = 'records'
    if key in trails_dict:
        records = trails_dict[key]
        return records
```

Model Approach & Comparisons

TRUST YOUR EMAIL

Previous Model

Model based on whitelist and blacklist

```
[29]: rf3 = RandomForestClassifier(n_estimators=100)
      rf3.fit(X_train, y_train)

      print(classification_report(rf3.predict(X_test), y_test))
```

```
              precision    recall  f1-score   support

         bad       0.95      0.95      0.95      5903
        good       0.95      0.95      0.95      5744

   micro avg       0.95      0.95      0.95     11647
   macro avg       0.95      0.95      0.95     11647
weighted avg       0.95      0.95      0.95     11647
```

Model Comparison Based on Unknown Dataset

Logistic Regression

```
lr = LogisticRegression(solver='liblinear')
lr.fit(X_train, y_train)
print(classification_report(lr.predict(X_test), y_test))
```

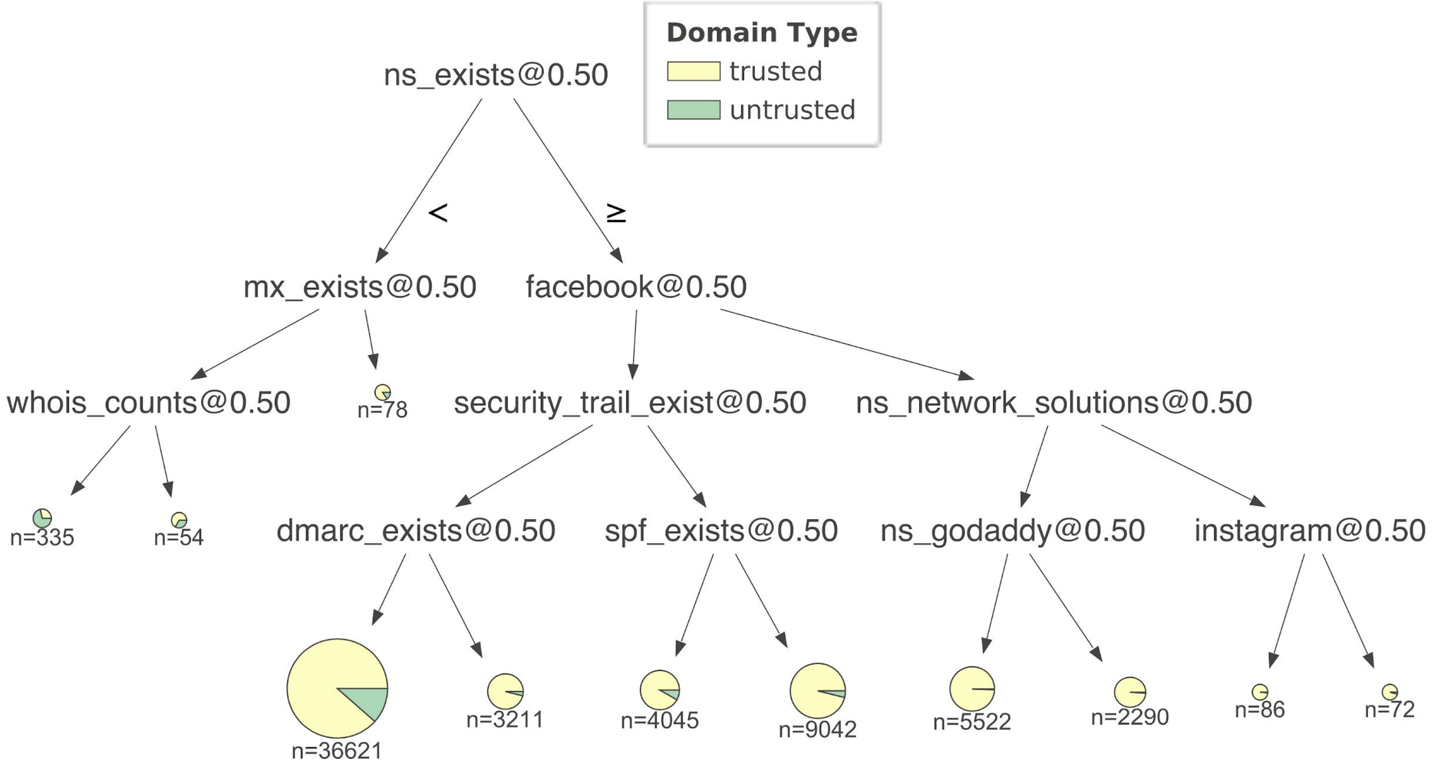
	precision	recall	f1-score	support
trusted	1.00	0.91	0.95	20144
untrusted	0.04	0.73	0.08	104
micro avg	0.91	0.91	0.91	20248
macro avg	0.52	0.82	0.52	20248
weighted avg	0.99	0.91	0.95	20248

Random Forest

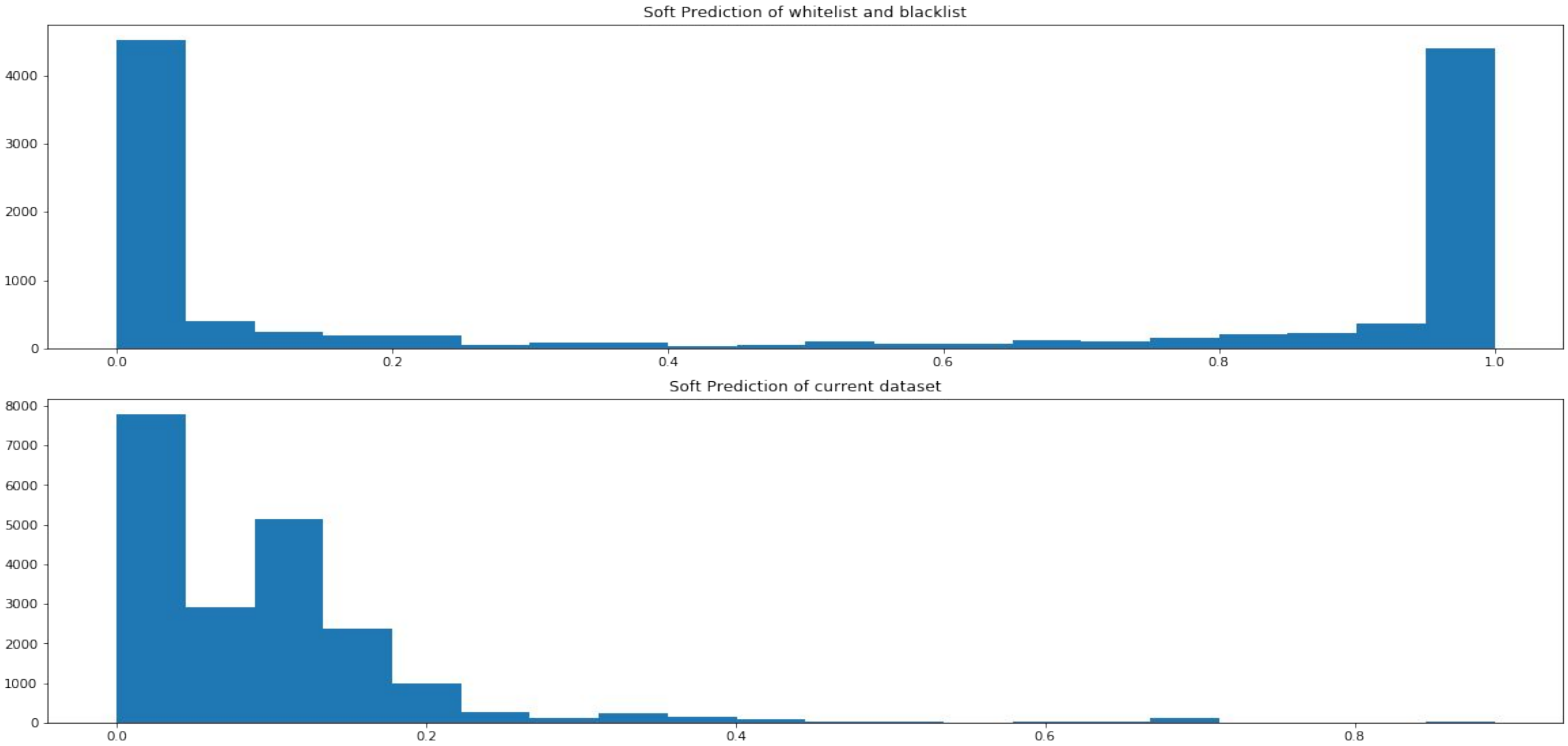
```
rf = RandomForestClassifier(n_estimators=500)
rf.fit(X_train, y_train)
print(classification_report(rf.predict(X_test), y_test))
```

	precision	recall	f1-score	support
trusted	1.00	0.91	0.95	20064
untrusted	0.06	0.56	0.10	184
micro avg	0.91	0.91	0.91	20248
macro avg	0.53	0.74	0.53	20248
weighted avg	0.99	0.91	0.95	20248

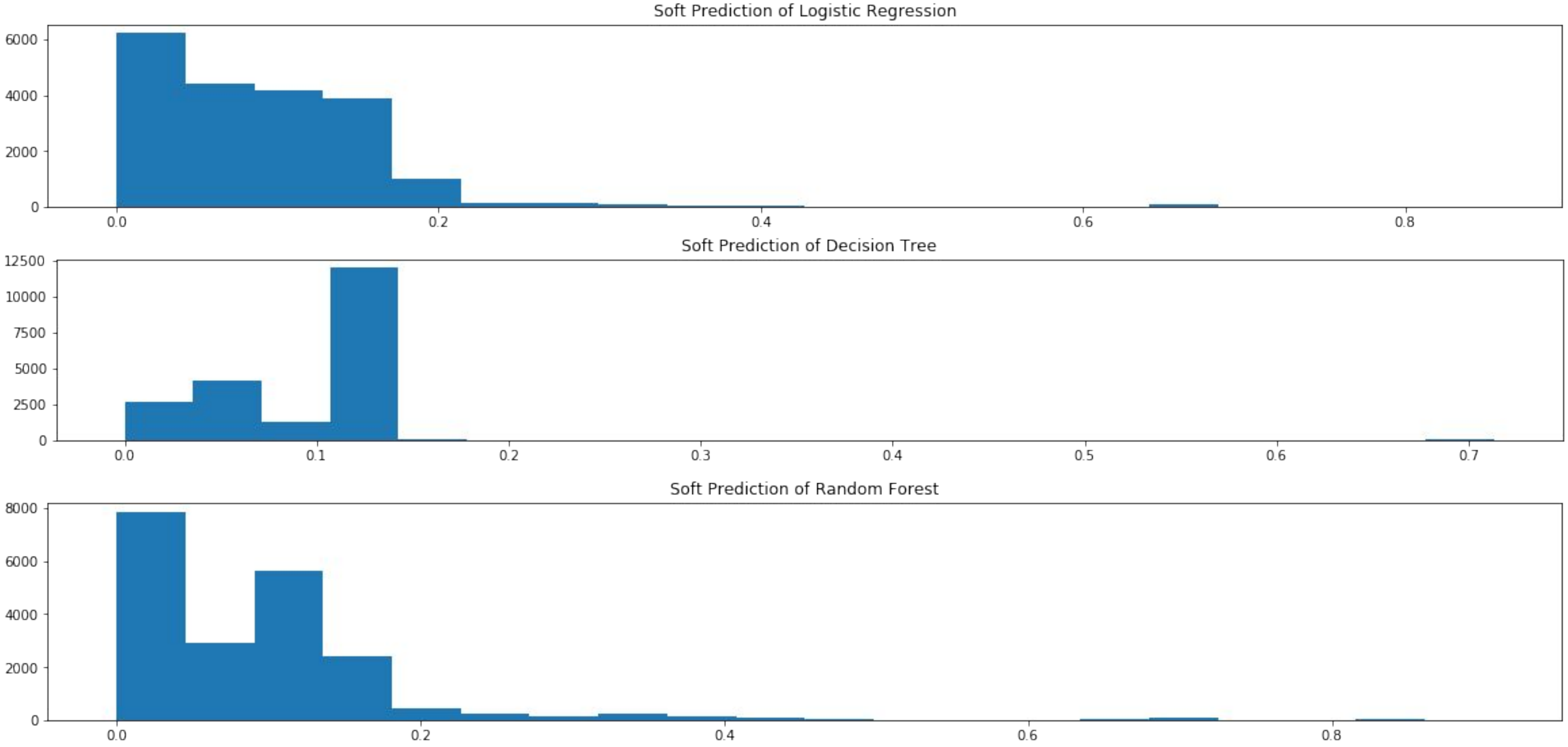
Decision Tree Visualization



Soft Prediction of Different Dataset



Soft Prediction of Different Models



Report on Distributions

Feature Importance Ranking

No.1 feature: app_list_exist (0.196)

No.2 feature: security_trail_exist (0.139)

No.3 feature: whois_counts (0.123)

No.4 feature: ns_exists (0.0855)

No.5 feature: mx_exists (0.0702)

No.6 feature: ns_amazon_route53 (0.0473)

No.7 feature: facebook (0.0413)

No.8 feature: ns_godaddy (0.0381)

No.9 feature: twitter (0.0242)

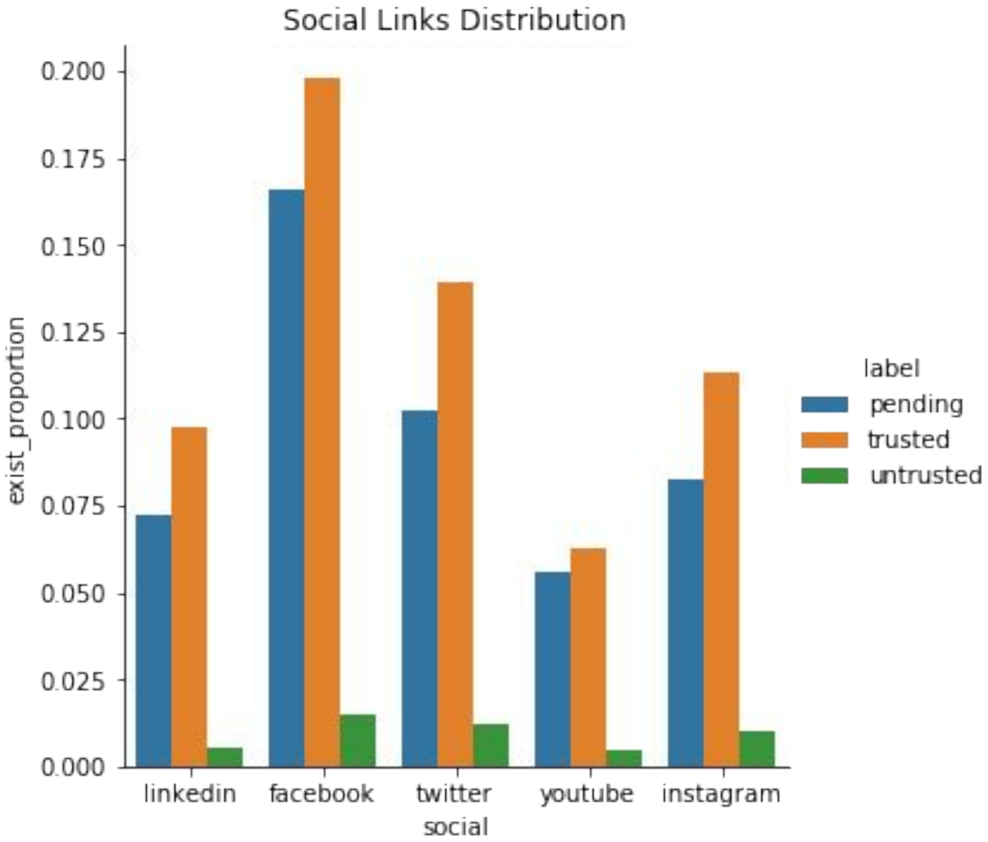
No.10 feature: mx_internal (0.0212)

Wappalyzer + Security Trails

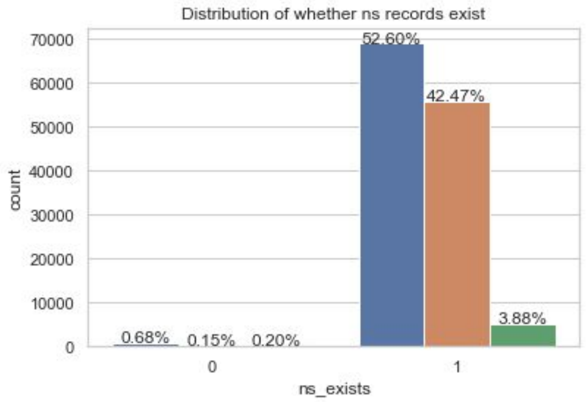
DNS Records

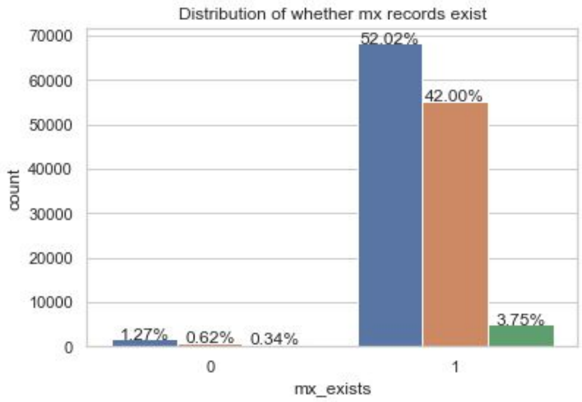
Social Media Links

Social Media Links Distribution

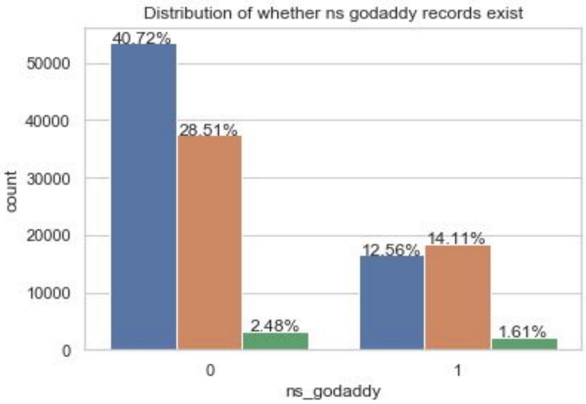


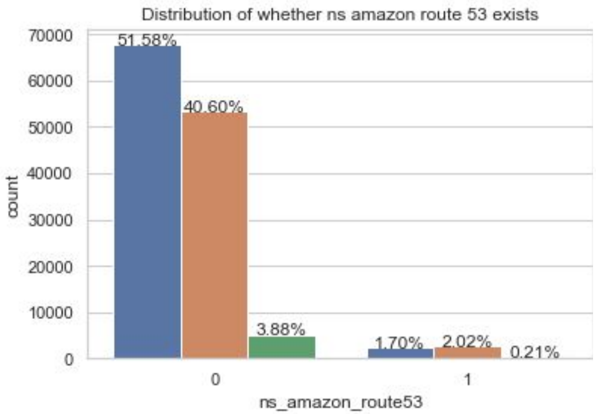
TRUST YOUR EMAIL

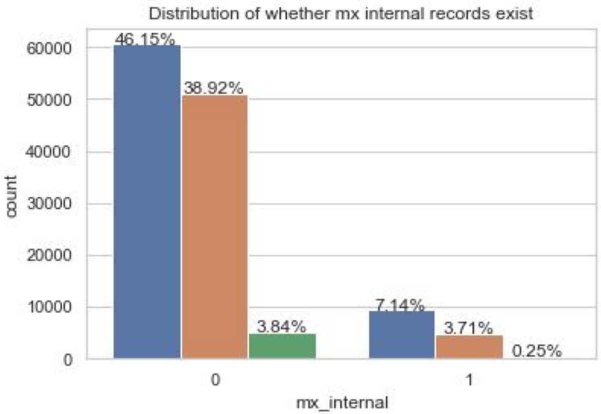




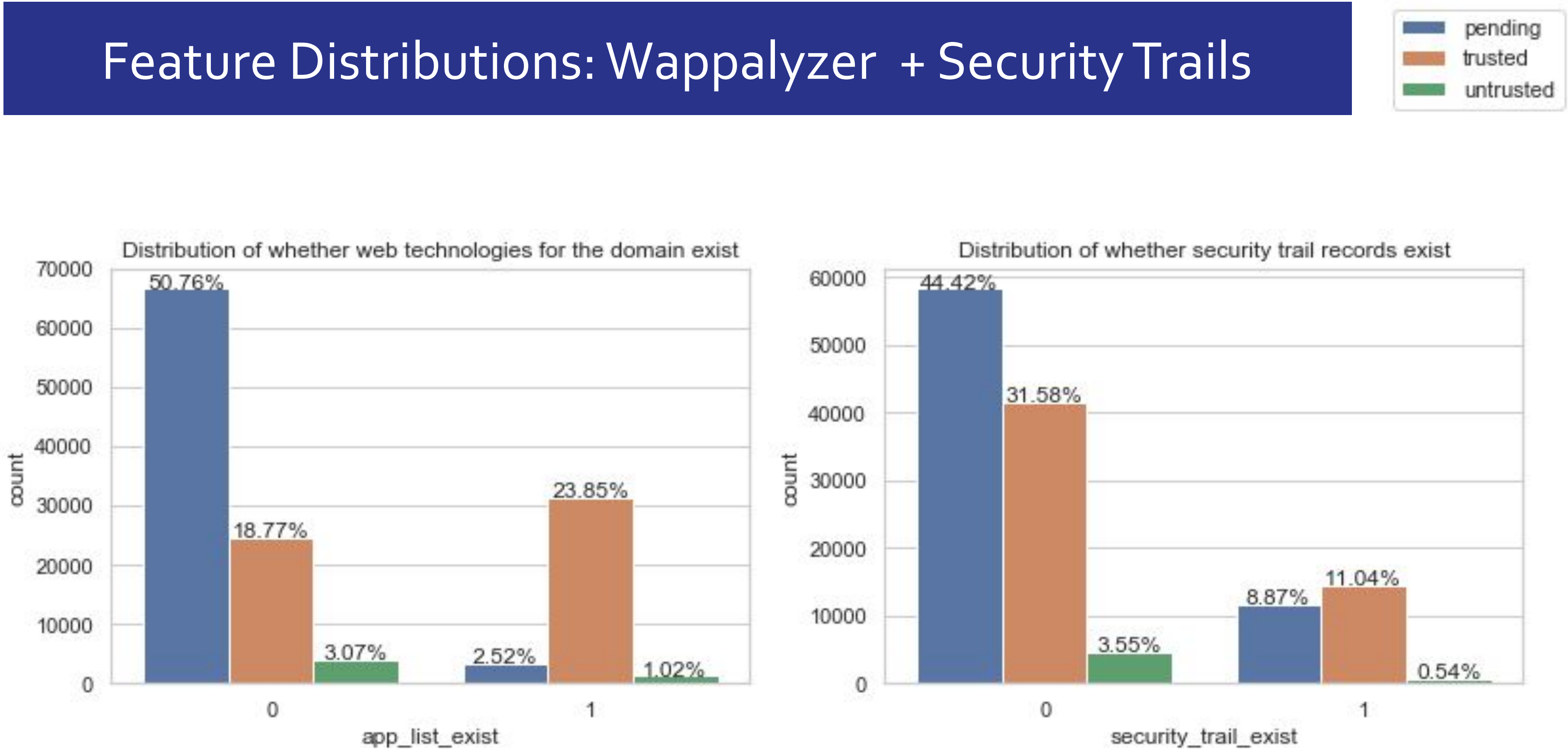
Feature Distributions: DNS Data







Feature Distributions: Wappalyzer + Security Trails



Future Work

- Explore Security Trails Data
    - First Seen Record
    - Valid or not (did it expire?)
- Explore Data from Google Knowledge Graph
- Build Data pipeline

VALIMAIL
TRUST YOUR EMAIL

SPECIAL THANK YOU

To

Claire, Gio, Carlos, Brian
and the Engineering team