TITLE

DETECTION OF MALICIOUS URLS

Submitted to: Mr. Sagar Pandey

Name: Khwairakpam Bidyananda Singh

Student ID: 11806898

Roll no.: 64

Section : K18KK

Email Address : boinaokh3@gmail.com

GitHub Link: https://github.com/Bidyananda/AI_CA

Table of contents

Abstract

The Detection of Malicious URLs (DMU) is a system which could point out malicious and fishy looking URLs..

The project I submitted consist of the following:

1. The code: A python file
2. The data set : urldata.csv

❖ Introduction:

WHAT DOES MY PROJECT CONSISTS OF?

The DMU combines the following into a single unit.

Contains a URLs detecting program which further combine with the python code helps us differentiate between malicious URLs.

❖ Objective:

The DMU will help us differentiate between malicious URLs.

It will help us stay away from virus from those malicious URLs which as a result will make our system vulnerable. This can result in our system being hacked and privacy compromised.

DMU will help prevent phishing attacks by hackers to void our privacy.

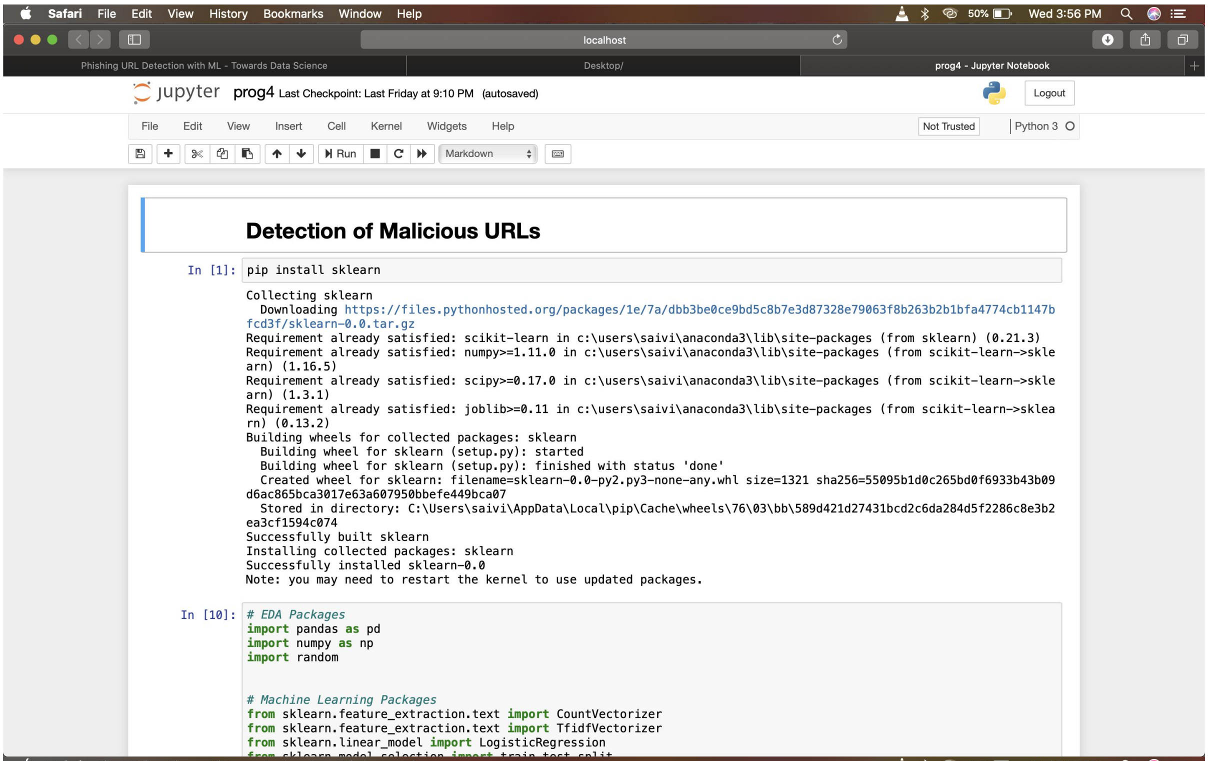
❖ Motivation:

This project will help us prevent phishing attack and being vulnerable to hackers which void our privacy.

Lost in capital will also be reduced by using this project as DMU will detect the malicious URLs and warn user of the danger/risk.

❖ Implementation of the project

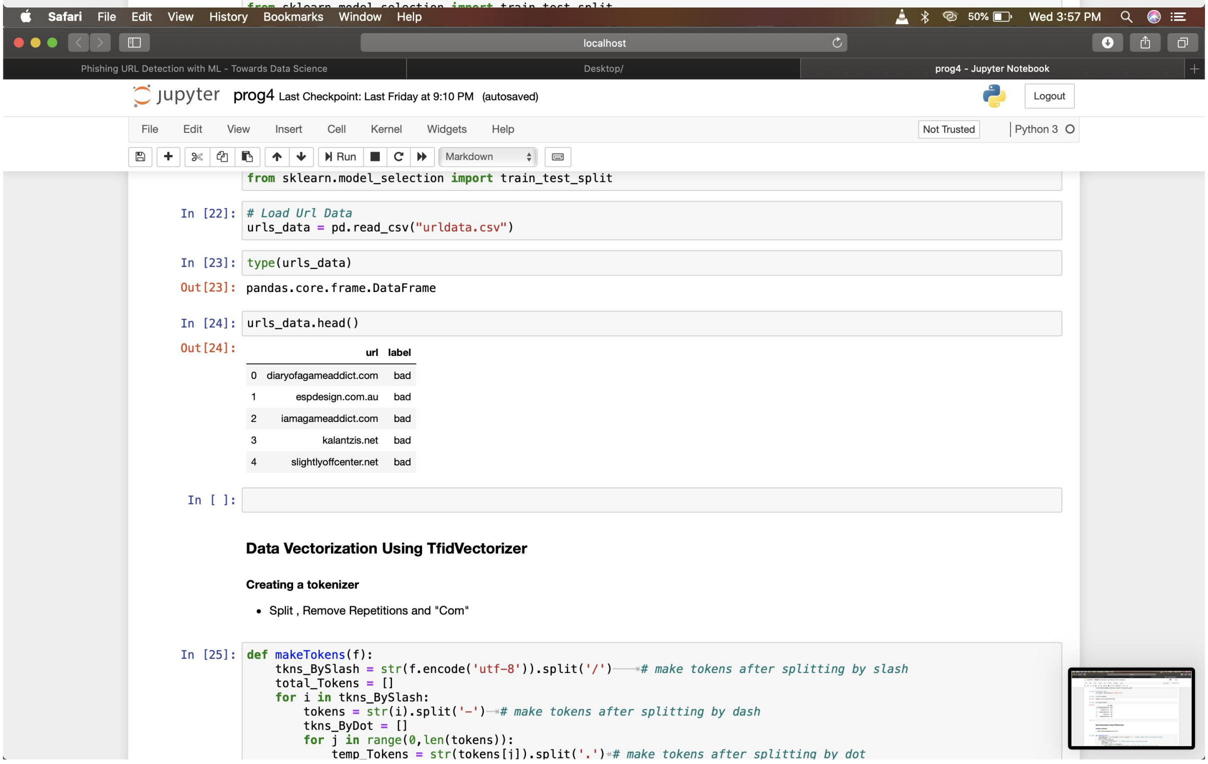
PYTHON:


Safari File Edit View History Bookmarks Window Help
50% Wed 3:56 PM
localhost
Phishing URL Detection with ML - Towards Data Science
Desktop/
prog4 - Jupyter Notebook
jupyter prog4 Last Checkpoint: Last Friday at 9:10 PM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3
Run
Markdown

Detection of Malicious URLs

```
In [1]: pip install sklearn

Collecting sklearn
  Downloading https://files.pythonhosted.org/packages/1e/7a/dbb3be0ce9bd5c8b7e3d87328e79063f8b263b2b1bfa4774cb1147b
fcd3f/sklearn-0.0.tar.gz
Requirement already satisfied: scikit-learn in c:\users\saivi\anaconda3\lib\site-packages (from sklearn) (0.21.3)
Requirement already satisfied: numpy>=1.11.0 in c:\users\saivi\anaconda3\lib\site-packages (from scikit-learn->skle
arn) (1.16.5)
Requirement already satisfied: scipy>=0.17.0 in c:\users\saivi\anaconda3\lib\site-packages (from scikit-learn->skle
arn) (1.3.1)
Requirement already satisfied: joblib>=0.11 in c:\users\saivi\anaconda3\lib\site-packages (from scikit-learn->sklea
rn) (0.13.2)
Building wheels for collected packages: sklearn
  Building wheel for sklearn (setup.py): started
  Building wheel for sklearn (setup.py): finished with status 'done'
  Created wheel for sklearn: filename=sklearn-0.0-py2.py3-none-any.whl size=1321 sha256=55095b1d0c265bd0f6933b43b09
d6ac865bca3017e63a607950bbefe449bca07
  Stored in directory: C:\Users\saivi\AppData\Local\pip\Cache\wheels\76\03\bb\589d421d27431bcd2c6da284d5f2286c8e3b2
ea3cf1594c074
Successfully built sklearn
Installing collected packages: sklearn
Successfully installed sklearn-0.0
Note: you may need to restart the kernel to use updated packages.
```

```
In [10]: # EDA Packages
         import pandas as pd
         import numpy as np
         import random


         # Machine Learning Packages
         from sklearn.feature_extraction.text import CountVectorizer
         from sklearn.feature_extraction.text import TfidfVectorizer
         from sklearn.linear_model import LogisticRegression
```


Safari File Edit View History Bookmarks Window Help
50% Wed 3:57 PM
localhost
Phishing URL Detection with ML - Towards Data Science
Desktop/
prog4 - Jupyter Notebook
jupyter prog4 Last Checkpoint: Last Friday at 9:10 PM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3
Run
Markdown

```
         from sklearn.model_selection import train_test_split

In [22]: # Load Url Data
         urls_data = pd.read_csv("urldata.csv")

In [23]: type(urls_data)

Out[23]: pandas.core.frame.DataFrame

In [24]: urls_data.head()

Out[24]:
```

	url	label
0	diaryofagameaddict.com	bad
1	espdesign.com.au	bad
2	iamagameaddict.com	bad
3	kalantzis.net	bad
4	slightlyoffcenter.net	bad

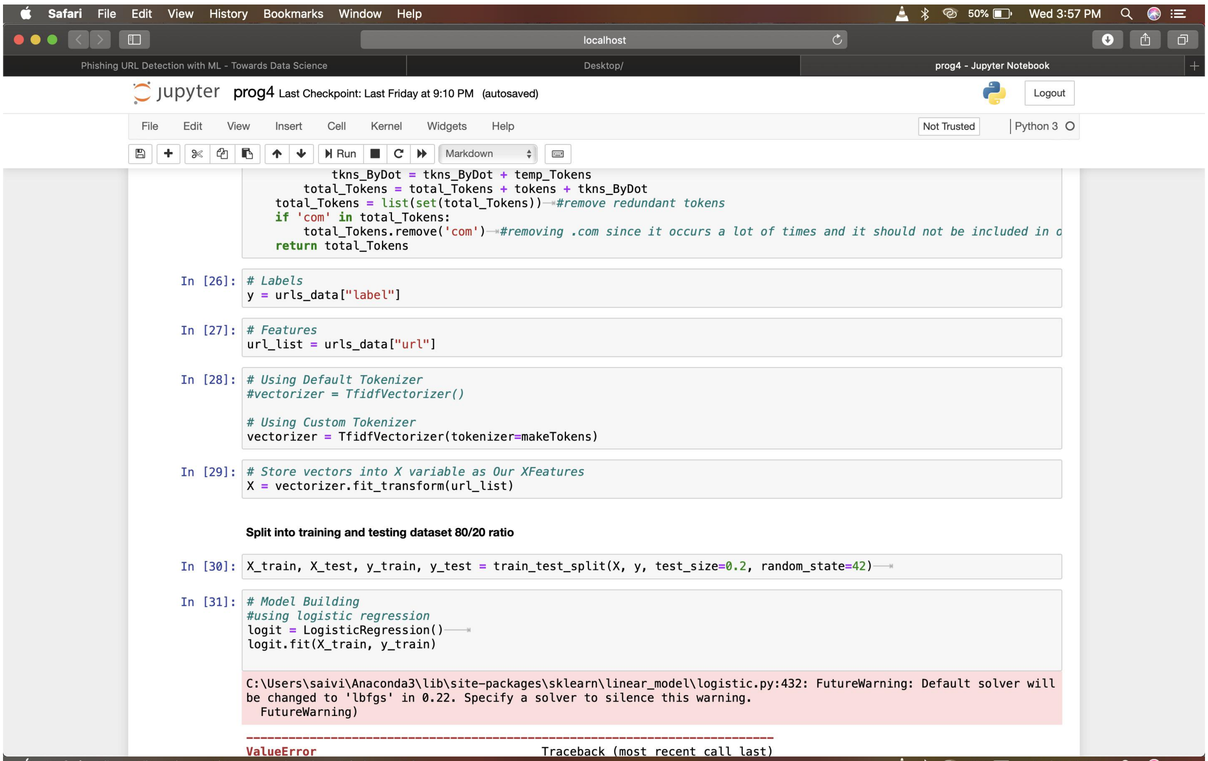
```
In [ ]:
```

Data Vectorization Using TfidVectorizer

Creating a tokenizer

- Split , Remove Repetitions and "Com"

```
In [25]: def makeTokens(f):
             tkns_BySlash = str(f.encode('utf-8')).split('/')	# make tokens after splitting by slash
             total_Tokens = []
             for i in tkns_BySlash:
                 tokens = str(i).split('-')	# make tokens after splitting by dash
                 tkns_ByDot = []
                 for j in range(0,len(tokens)):
                     temp_Tokens = str(tokens[j]).split('.')	# make tokens after splitting by dot
```


Safari File Edit View History Bookmarks Window Help
50% Wed 3:57 PM
localhost
Phishing URL Detection with ML - Towards Data Science
Desktop/
prog4 - Jupyter Notebook
jupyter prog4 Last Checkpoint: Last Friday at 9:10 PM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3
Run
Markdown

```
                tkns_ByDot = tkns_ByDot + temp_Tokens
            total_Tokens = total_Tokens + tokens + tkns_ByDot
        total_Tokens = list(set(total_Tokens))    #remove redundant tokens
        if 'com' in total_Tokens:
            total_Tokens.remove('com')    #removing .com since it occurs a lot of times and it should not be included in o
        return total_Tokens
```

```
In [26]: # Labels
         y = urls_data["label"]
```

```
In [27]: # Features
         url_list = urls_data["url"]
```

```
In [28]: # Using Default Tokenizer
         #vectorizer = TfidfVectorizer()

         # Using Custom Tokenizer
         vectorizer = TfidfVectorizer(tokenizer=makeTokens)
```

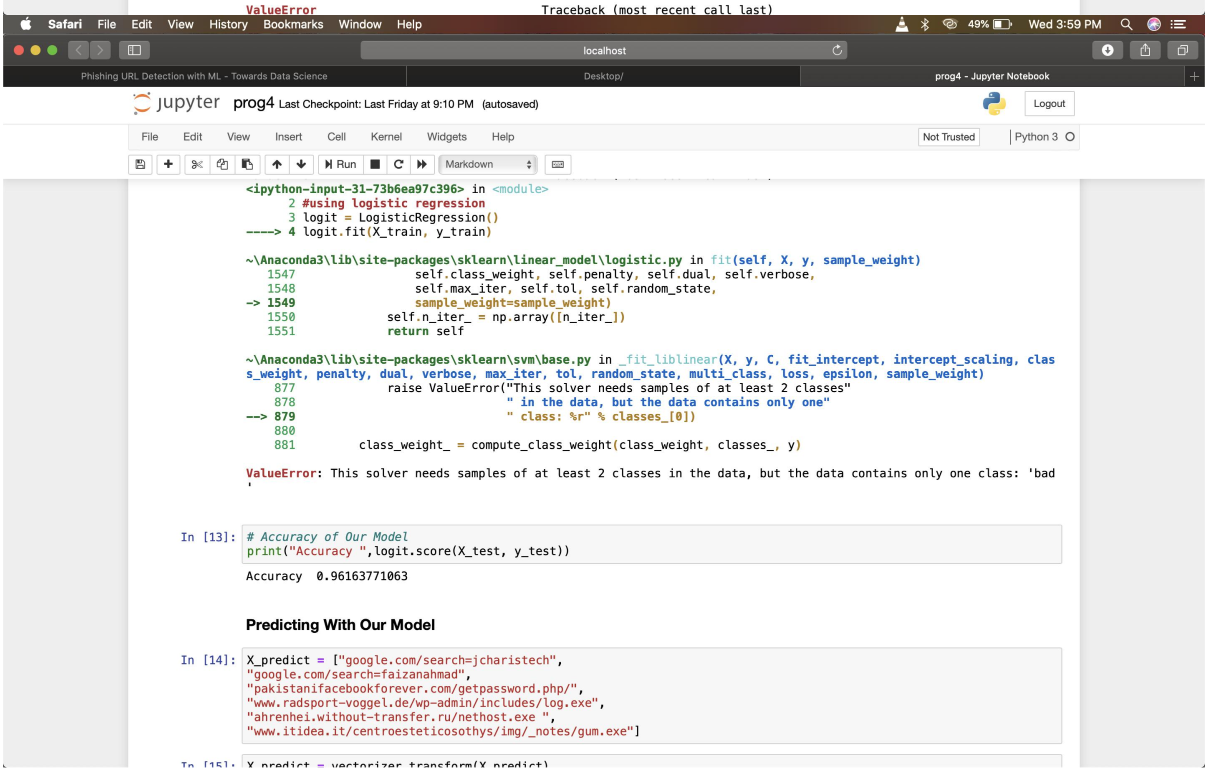
```
In [29]: # Store vectors into X variable as Our XFeatures
         X = vectorizer.fit_transform(url_list)
```

Split into training and testing dataset 80/20 ratio

```
In [30]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
In [31]: # Model Building
         #using logistic regression
         logit = LogisticRegression()
         logit.fit(X_train, y_train)
```

```
C:\Users\saivi\Anaconda3\lib\site-packages\sklearn\linear_model\logistic.py:432: FutureWarning: Default solver will
be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)

---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
```


Safari File Edit View History Bookmarks Window Help
49% Wed 3:59 PM
localhost
Phishing URL Detection with ML - Towards Data Science
Desktop/
prog4 - Jupyter Notebook
jupyter prog4 Last Checkpoint: Last Friday at 9:10 PM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3
Run
Markdown

```
<ipython-input-31-73b6ea97c396> in <module>
      2 #using logistic regression
      3 logit = LogisticRegression()
----> 4 logit.fit(X_train, y_train)

~\Anaconda3\lib\site-packages\sklearn\linear_model\logistic.py in fit(self, X, y, sample_weight)
   1547                 self.class_weight, self.penalty, self.dual, self.verbose,
   1548                 self.max_iter, self.tol, self.random_state,
-> 1549                 sample_weight=sample_weight)
   1550             self.n_iter_ = np.array([n_iter_])
   1551             return self

~\Anaconda3\lib\site-packages\sklearn\svm\base.py in _fit_liblinear(X, y, C, fit_intercept, intercept_scaling, clas
s_weight, penalty, dual, verbose, max_iter, tol, random_state, multi_class, loss, epsilon, sample_weight)
    877             raise ValueError("This solver needs samples of at least 2 classes"
    878                              " in the data, but the data contains only one"
--> 879                              " class: %r" % classes_[0])
    880 
    881         class_weight_ = compute_class_weight(class_weight, classes_, y)

ValueError: This solver needs samples of at least 2 classes in the data, but the data contains only one class: 'bad
'
```

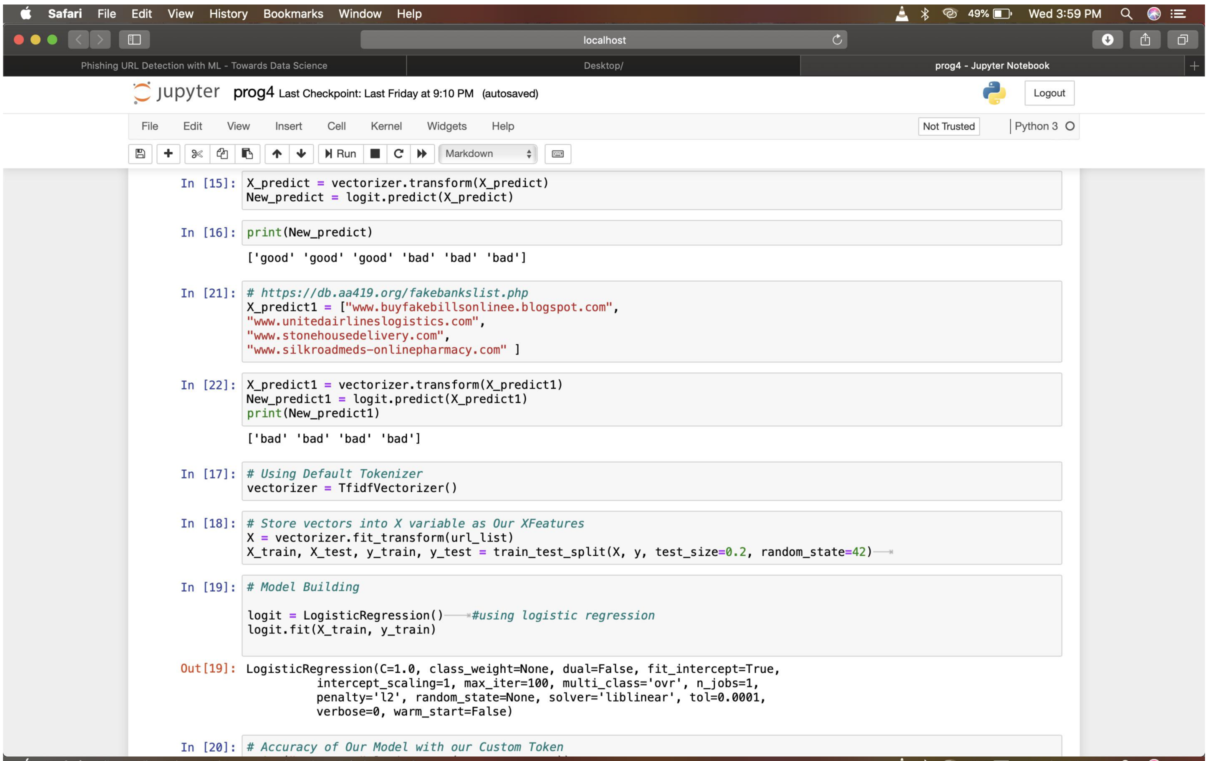
```
In [13]: # Accuracy of Our Model
         print("Accuracy ",logit.score(X_test, y_test))
```

```
Accuracy  0.96163771063
```

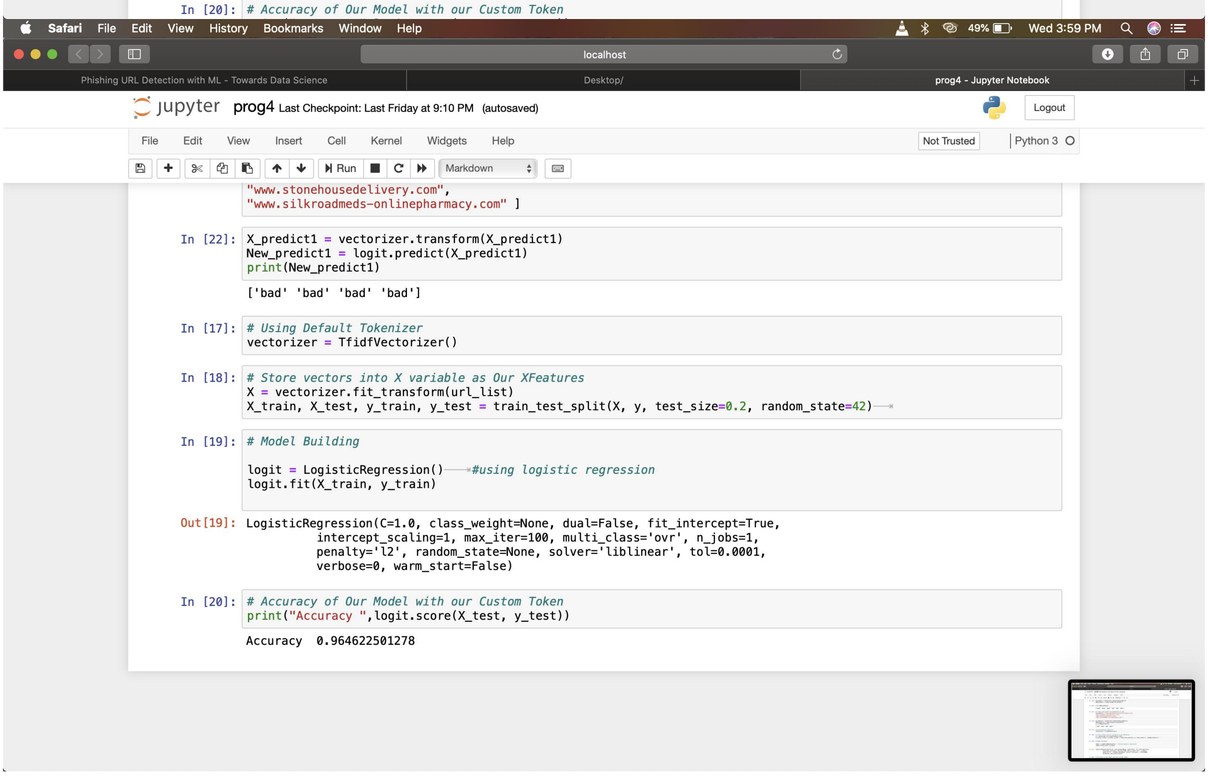
Predicting With Our Model

```
In [14]: X_predict = ["google.com/search=jcharistech",
         "google.com/search=faizanahmad",
         "pakistanifacebookforever.com/getpassword.php/",
         "www.radsport-voggel.de/wp-admin/includes/log.exe",
         "ahrenhei.without-transfer.ru/nethost.exe ",
         "www.itidea.it/centroesteticosothys/img/_notes/gum.exe"]
```

```
In [15]: X_predict = vectorizer.transform(X_predict)
```


Safari File Edit View History Bookmarks Window Help
49% Wed 3:59 PM
localhost
Phishing URL Detection with ML - Towards Data Science
Desktop/
prog4 - Jupyter Notebook
jupyter prog4 Last Checkpoint: Last Friday at 9:10 PM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3
Run
Markdown

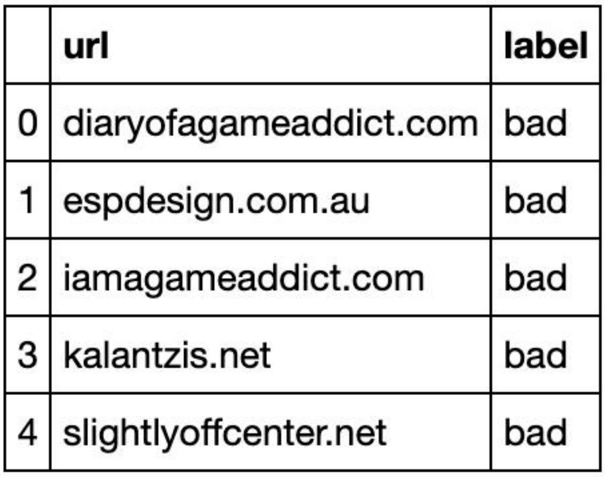
```
In [15]: X_predict = vectorizer.transform(X_predict)
         New_predict = logit.predict(X_predict)

In [16]: print(New_predict)

         ['good' 'good' 'good' 'bad' 'bad' 'bad']

In [21]: # https://db.aa419.org/fakebankslist.php
         X_predict1 = ["www.buyfakebillsonlinee.blogspot.com",
         "www.unitedairlineslogistics.com",
         "www.stonehousedelivery.com",
         "www.silkroadmeds-onlinepharmacy.com" ]

In [22]: X_predict1 = vectorizer.transform(X_predict1)
         New_predict1 = logit.predict(X_predict1)
         print(New_predict1)

         ['bad' 'bad' 'bad' 'bad']

In [17]: # Using Default Tokenizer
         vectorizer = TfidfVectorizer()

In [18]: # Store vectors into X variable as Our XFeatures
         X = vectorizer.fit_transform(url_list)
         X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

In [19]: # Model Building

         logit = LogisticRegression()    #using logistic regression
         logit.fit(X_train, y_train)

Out[19]: LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                   intercept_scaling=1, max_iter=100, multi_class='ovr', n_jobs=1,
                   penalty='l2', random_state=None, solver='liblinear', tol=0.0001,
                   verbose=0, warm_start=False)

In [20]: # Accuracy of Our Model with our Custom Token
```


Safari File Edit View History Bookmarks Window Help
49% Wed 3:59 PM
localhost
Phishing URL Detection with ML - Towards Data Science
Desktop/
prog4 - Jupyter Notebook
jupyter prog4 Last Checkpoint: Last Friday at 9:10 PM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3
Run
Markdown

```
         "www.stonehousedelivery.com",
         "www.silkroadmeds-onlinepharmacy.com" ]

In [22]: X_predict1 = vectorizer.transform(X_predict1)
         New_predict1 = logit.predict(X_predict1)
         print(New_predict1)

         ['bad' 'bad' 'bad' 'bad']

In [17]: # Using Default Tokenizer
         vectorizer = TfidfVectorizer()

In [18]: # Store vectors into X variable as Our XFeatures
         X = vectorizer.fit_transform(url_list)
         X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

In [19]: # Model Building

         logit = LogisticRegression()    #using logistic regression
         logit.fit(X_train, y_train)

Out[19]: LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                   intercept_scaling=1, max_iter=100, multi_class='ovr', n_jobs=1,
                   penalty='l2', random_state=None, solver='liblinear', tol=0.0001,
                   verbose=0, warm_start=False)
```

```
In [20]: # Accuracy of Our Model with our Custom Token
         print("Accuracy ",logit.score(X_test, y_test))

         Accuracy  0.964622501278
```

❖ Output:



	url	label
0	diaryofagameaddict.com	bad
1	espdesign.com.au	bad
2	iamagameaddict.com	bad
3	kalantzis.net	bad
4	slightlyoffcenter.net	bad

❖ Scope:

The DMU project can be used in the following ways:

- Detect phishing attacks
- Virus detection
- Detect Spam

❖ Work Distribution:

As this was a one-person project, all of the work was undertaken by me.

❖ Libraries Used:

1. Sklearn
2. Numpy
3. Pandas
4. Random

❖ GitHub link:

https://github.com/Bidyananda/AI_CA

❖ References:

1. https://towardsdatascience.com/phishing-domain-detection-with-ml-5be9c99293e5