# “DETECTING PHISHING WEBSITE USING MACHINE LEARNING”

**Project submitted in partial fulfillment of the requirements for the**

**award of the Degree of**

# Bachelor of Computer Applications of

**SASTRA DEEMED UNIVERSITY**

# Submitted by

# “S. SIDDHARTHAN”

**Register No: 21113070562**

# Under the Supervision and Guidance of

**Prof. L. GOWRI**



# School of Computing SASTRA Deemed University

**(Under section 3 of the UGC Act, 1956)**

# Thanjavur – 613 401

**January – 2024**

# School of Computing SASTRA University

**(Under section 3 of the UGC Act, 1956)**

# Thanjavur – 613 401



**Bonafide Certificate** Certified that this project report entitled

**“DETECTING PHISHING WEBSTE USING MACHINE LEARNING”**

Is a bonafide record of work done by

# “S.SIDDHARTHAN”

**Register No. 21113070562**

In partial fulfillment of the requirements for award of the Degree of **Bachelor of Computer Applications**

During the year 2021 -2024

# Project Guide: Prof. L. Gowri

**Submitted for Project viva-voce examination held on --------**

# Examiner -I Examiner-II

Acknowledgements

I would like to thank our Honorable Chancellor Prof.R.Sethuraman for providing me with an opportunity and the necessary infrastructure for carrying out this project as a part of our curriculum.

I would like to thank our Honorable Vice-Chancellor Dr.S.Vaidhyasubramaniam and Dr.S.Swaminathan, Dean, Planning & Development, for the encouragement and strategic support at every step of our college life.

I extend my sincere thanks to Dr.R.Chandramouli, Registrar, SASTRA Deemed to be University for providing the opportunity to pursue this project.

I extend my heartfelt thanks to Dr.V.S.Shankar Sriram, Dean, School of Computing, Dr.R.Muthaiah, Associate Dean-Research, School of Computing Dr.K.Ramkumar, Associate Dean-Academics, School of Computing Dr.D.Manivannan, Associate Dean-Infrastructure School of Computing and Dr.R.Alageswaran, Associate Dean-Student Welfare, School of Computing for their motivation and support offered in materializing this project.

My guide Prof. L. Gowri, School of Computing was the driving force behind this whole idea from the start. Her deep insight in the field and invaluable suggestions helped me in making progress throughout our project work. I also thank the project review panel members for their valuable comments and insights which made this project better.

I would like to extend my gratitude to all the teaching and non-teaching faculties of the School of Computing who have either directly or indirectly helped me in the completion of the project.

I gratefully acknowledge all the contributions and encouragement from my family and friends resulting in the successful completion of this project. I thank you all for providing me an opportunity to showcase my skills through project.

**ABSTRACT**

This research delves into the realm of machine learning methodologies applied to the detection of phishing websites through an in-depth analysis of URL features. The primary focus lies in the meticulous examination and discrimination between legitimate and phishing sites, utilizing advanced techniques to bolster the accuracy of detection mechanisms. The evaluation encompasses the efficacy of decision trees, random forests, and deep learning algorithms, aiming to identify the most suitable approach for precise and reliable detection of phishing activities. By undertaking this investigation, the study significantly contributes to cybersecurity measures by enhancing the capabilities of phishing detection systems, thereby fortifying online security protocols and providing a robust defense against evolving cyber threats.

# Table of Contents

|  |  |  |
| --- | --- | --- |
| S.NO | CONTENT | PG NO |
| 1 | Introduction  Problem Statement  Objective | 1  2  3 |
| 2 | **System Design**  Architecture Diagram | 4 |
| 3 | **System Requirements**  3.1 Hardware requirements  3.2 Software requirements | 5 |
| 4 | **System Analysis**  Data Flow Diagram | 6 |
| 5 | **Approach** | 9 |
| 6 | **Coding**  6.1 Feature Extraction  6.2 Detecting Phishing Website | 10  23 |
| 7 | **Front End**  7.1 Home Page  **7.2** Legitimate URL  7.3 Phishing URL | 40  41  42 |
| 8 | **Future plans** | 43 |
| 9 | **Bibliography** | 44 |

# Introduction

The escalating threat of phishing websites poses an ever-growing risk to the landscape of online security, necessitating robust and proactive measures to counteract the evolving tactics employed by malicious entities. Recognizing the pivotal role of machine learning algorithms in fortifying cybersecurity, this study is dedicated to exploring their indispensable application for the proactive identification and mitigation of phishing sites. The overarching objective is to implement advanced measures that leverage the power of machine learning for real-time detection, thereby significantly reducing the risks posed by phishing websites. Through a focused examination of cutting-edge technologies, the research aims to contribute to the development of a resilient defense mechanism that protects users from the constantly changing and increasingly sophisticated landscape of cyber threats.

**Problem Statement**

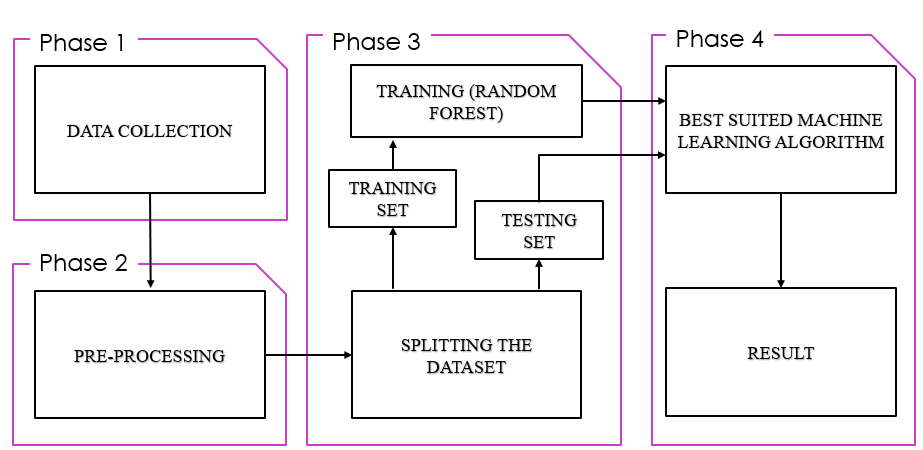
The relentless growth of phishing threats has surpassed the efficacy of traditional identification methods, demanding an urgent and proactive response to safeguard online security. In this context, the research takes a pioneering approach by focusing on the strategic utilization of machine learning algorithms for a meticulous and robust analysis of URL features. The primary goal is to develop an advanced model that goes beyond merely identifying phishing threats but excels in effectively distinguishing between legitimate and malicious websites. By integrating state-of-the-art methodologies, the study aims to address the dynamic nature of phishing tactics, providing a resilient and adaptive solution that significantly elevates cybersecurity standards. Furthermore, the research emphasizes the importance of real-time detection capabilities, aiming to create a system that not only reacts to known threats but also proactively identifies emerging phishing techniques. This multifaceted approach seeks to contribute to the ongoing evolution of cybersecurity practices, offering a comprehensive defense against the multifarious and sophisticated challenges posed by phishing threats in the contemporary digital landscape.

**Objective**

This research is centered around the comprehensive development and assessment of machine learning algorithms, specifically decision trees, random forests, and deep learning, with a targeted application in proactive phishing detection. The primary objective is to thoroughly evaluate the efficacy of these models in distinguishing between legitimate and malicious websites. The overarching goal is to significantly enhance online security, going beyond immediate applications to contribute valuable insights to the continually evolving field of cybersecurity. By delving into the nuances of machine learning algorithms, this study aims to advance the current state-of-the-art in proactive phishing detection, fostering a deeper understanding of the complex dynamics involved in discerning authentic from fraudulent online entities. The research aspires to fortify online security practices, ensuring a resilient defense against the ever-expanding landscape of phishing threats, and to contribute knowledge that informs effective cybersecurity strategies in the face of dynamic tactics employed by cyber adversaries.

**2. SYSTEM DESIGN**

**Architecture Diagram**



**3. SYSTEM REQUIREMENT**

**3.1 HARDWARE REQUIREMENTS:**

* System: Intel Pentium IV 2.80 GHz.
* Monitor: LED.
* Mouse: Logitech.
* Ram: 4.00 GB or above 4.00 GB
* Hard Disk: 250 GB

**3.2 SOFTWARE REQUIREMENTS:**

* Operating system: Windows 7, Ubuntu
* Language: Python 3

**4. SYSTEM ANALYSIS**

**DATA FLOW DIAGRAM**

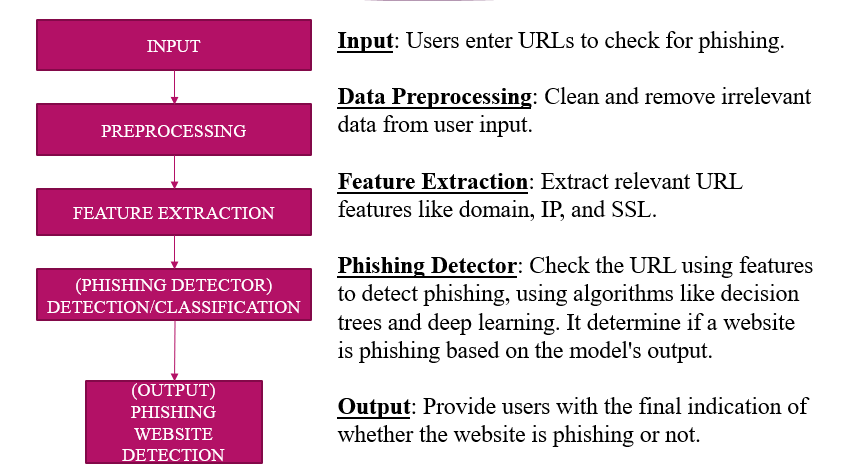
**Level 0:**

Input URL

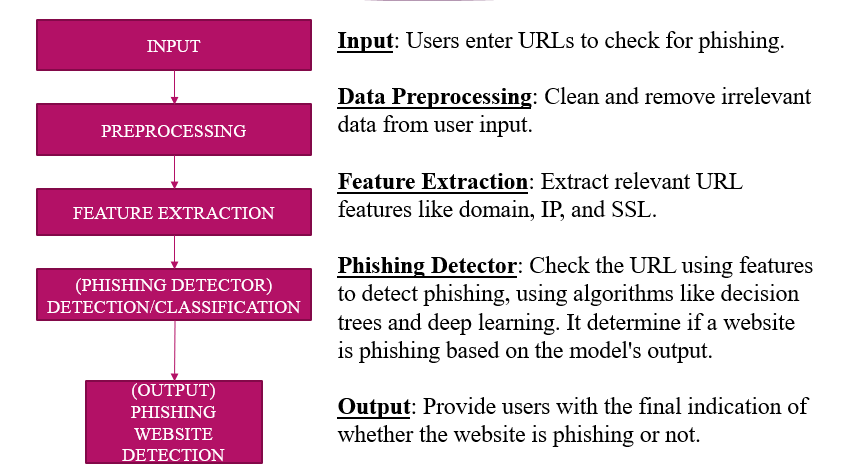
Process the URL

Output (Phishing Website Detection)

**Level 1:**

****

**LEVEL 2:**

****

**5. APPROACH**

Below mentioned are the steps involved in the completion of this project:

• Collect dataset containing phishing and legitimate websites from the open-source platforms.

• Write a code to extract the required features from the URL database.

• Analyze and preprocess the dataset by using EDA techniques.

• Divide the dataset into training and testing sets.

• Run selected machine learning and deep neural network algorithms like SVM, Random Forest, Autoencoder on the dataset. • Write a code for displaying the evaluation result considering accuracy metrics.

• Compare the obtained results for trained models and specify which is better.

**6. CODING**

**6.1. Feature Extraction**

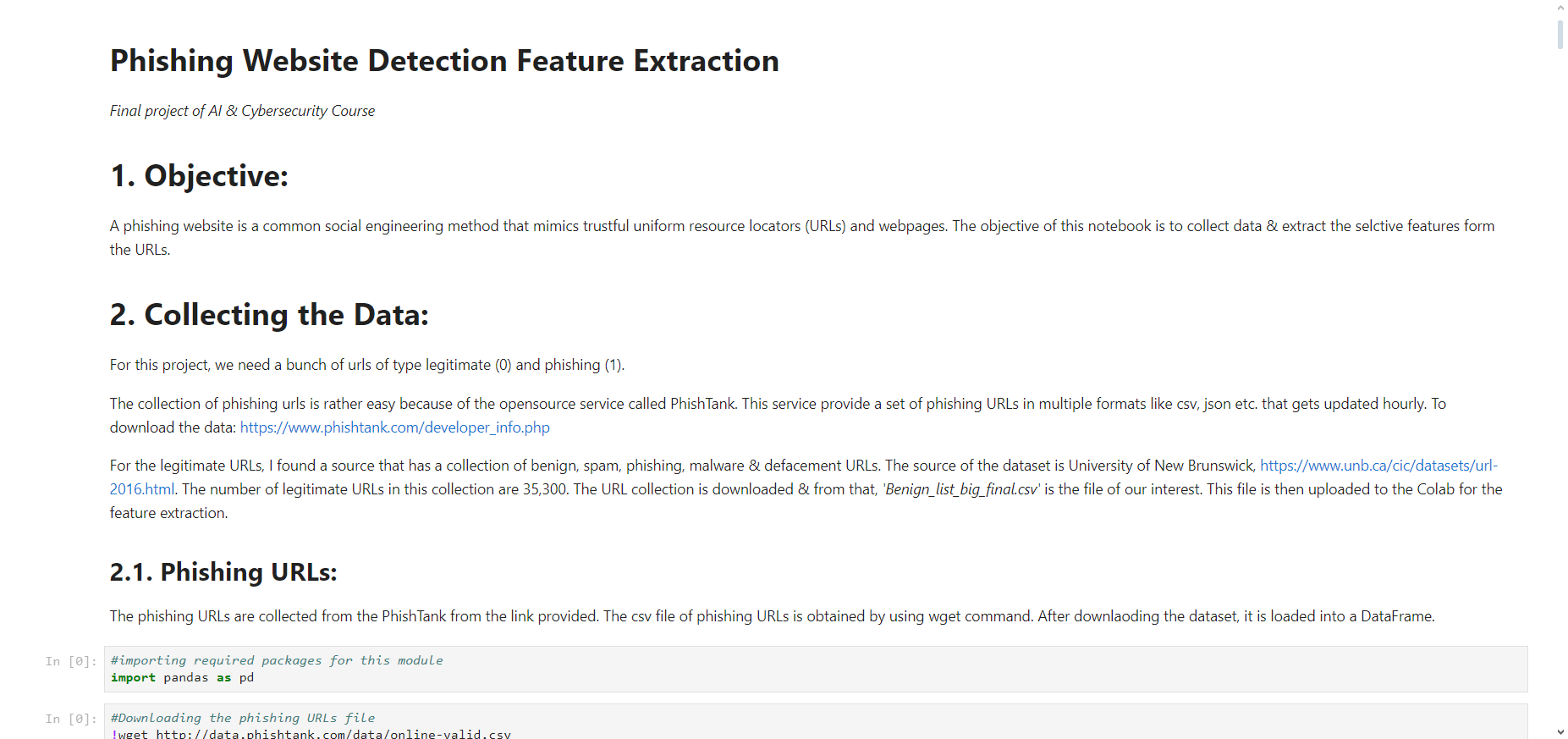
**1. Data Collection:**

For this project, we need a bunch of URLs of type legitimate (0) and phishing (1).

The collection of phishing URLs is rather easy because of the opensource service called Phish Tank. This service provides a set of phishing URLs in multiple formats like csv, json etc. that gets updated hourly. To download the data: https://www.phishtank.com/developer\_info.php

For the legitimate URLs, I found a source that has a collection of benign, spam, phishing, malware & defacement URLs. The source of the dataset is University of New Brunswick, https://www.unb.ca/cic/datasets/url-2016.html. The number of legitimate URLs in this collection are 35,300. The URL collection is downloaded & from that, 'Benign\_list\_big\_final.csv' is the file of our interest. This file is then uploaded to the Colab for the feature extraction.

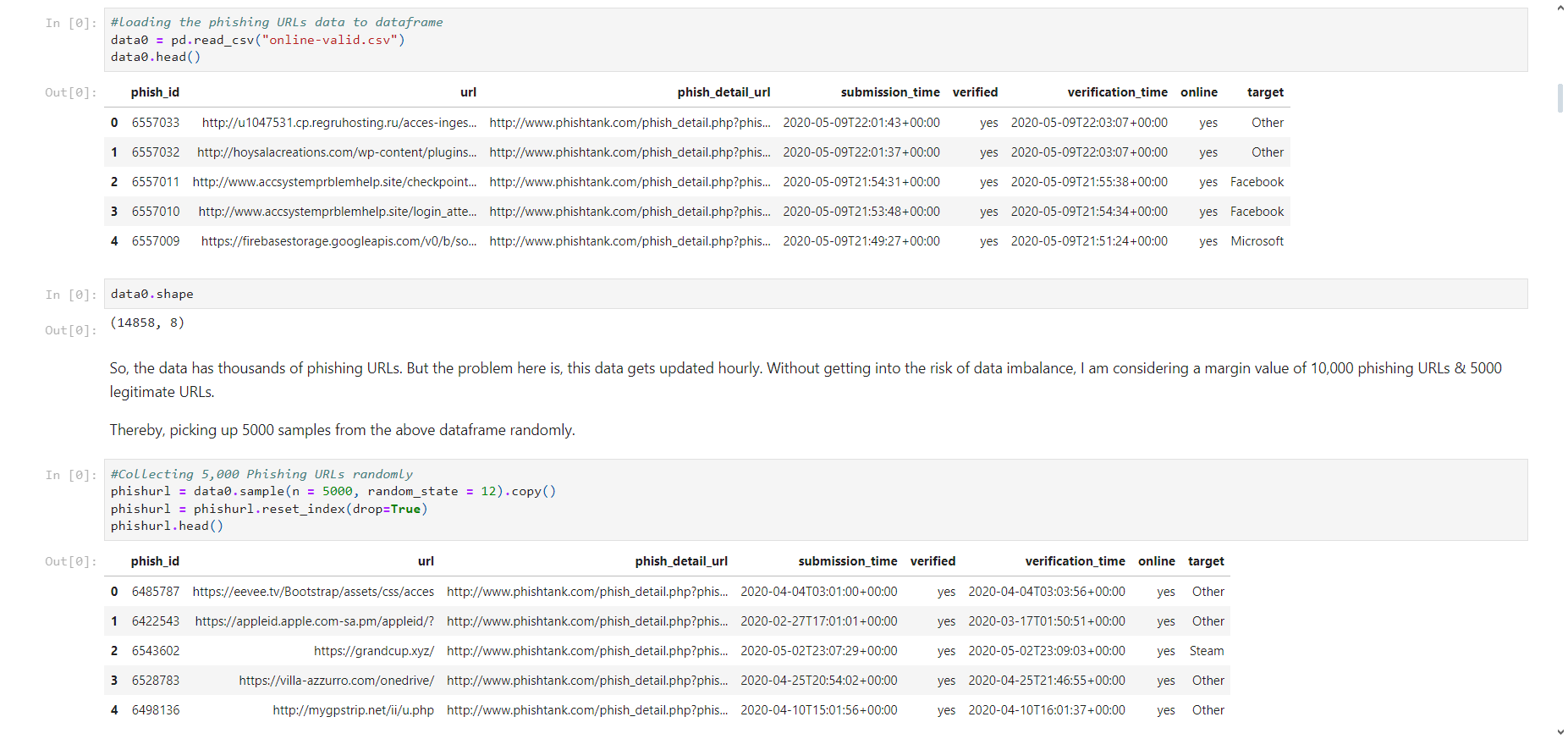
First Step is importing pandas.

****

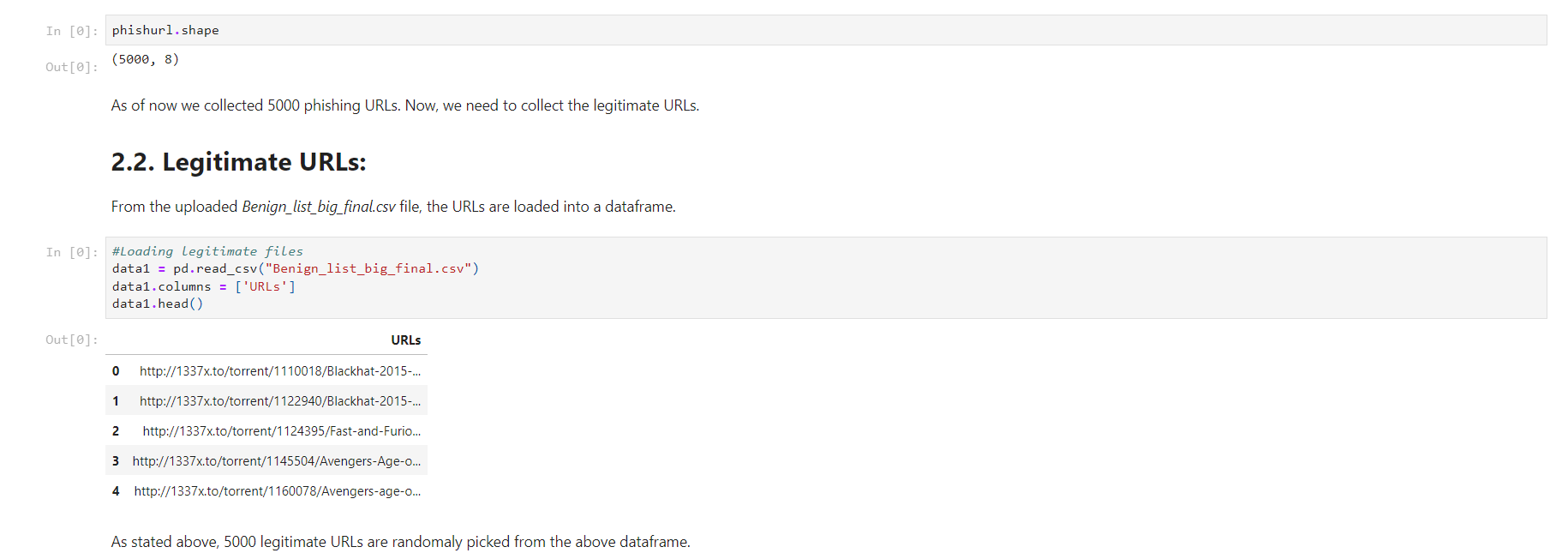
Now downloading the phishing websites.

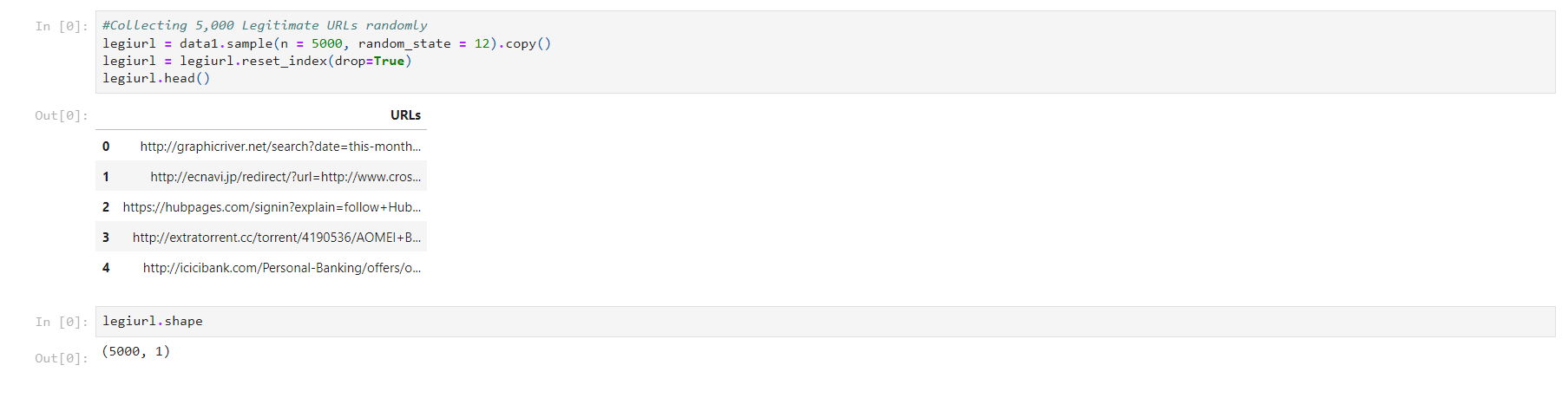
****

**2. Samples**

Taking 5000 sample phishing website

Uploading the Legitimate URL excel file which is got from https://www.unb.ca/cic/datasets/url-2016.html



Taking 5000 Samples of Legitimate website

**3. Feature Extraction:**

In this step, features are extracted from the URLs dataset.

The extracted features are categorized into

1. Address Bar based Features

2. Domain based Features

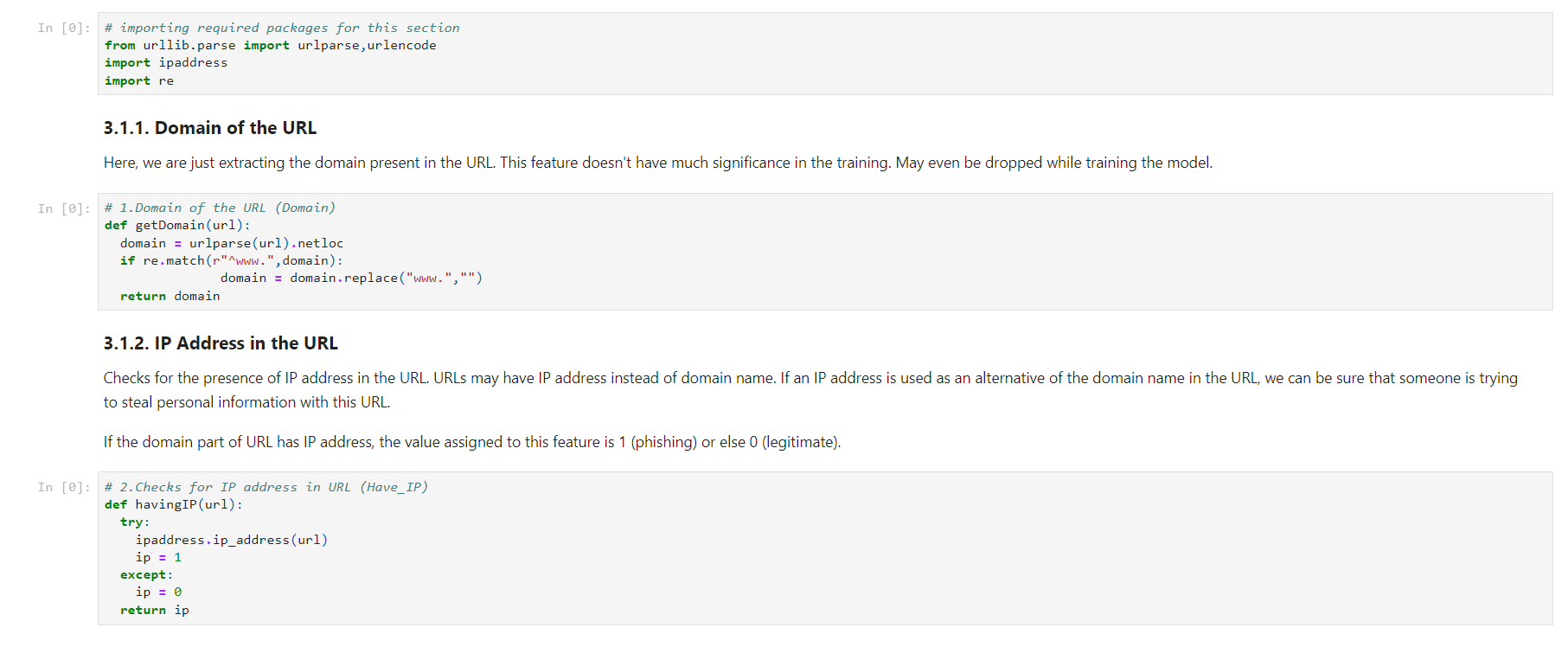
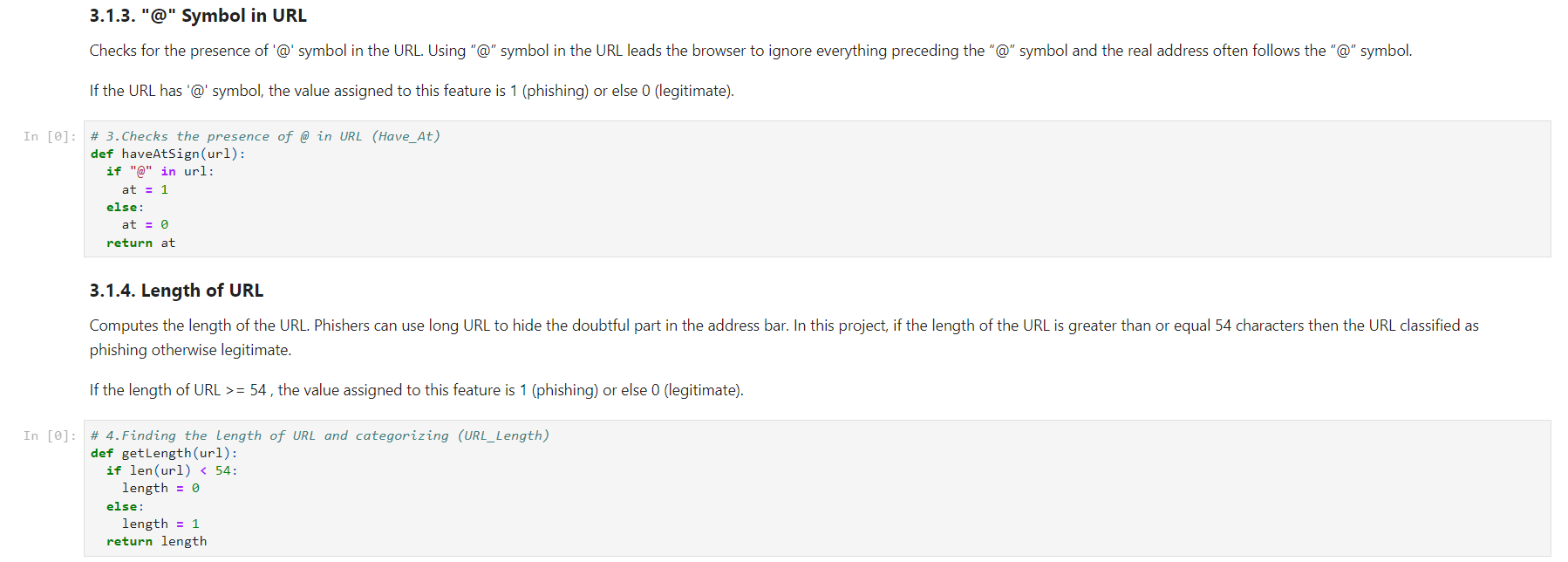
3. HTML & Javascript based Features

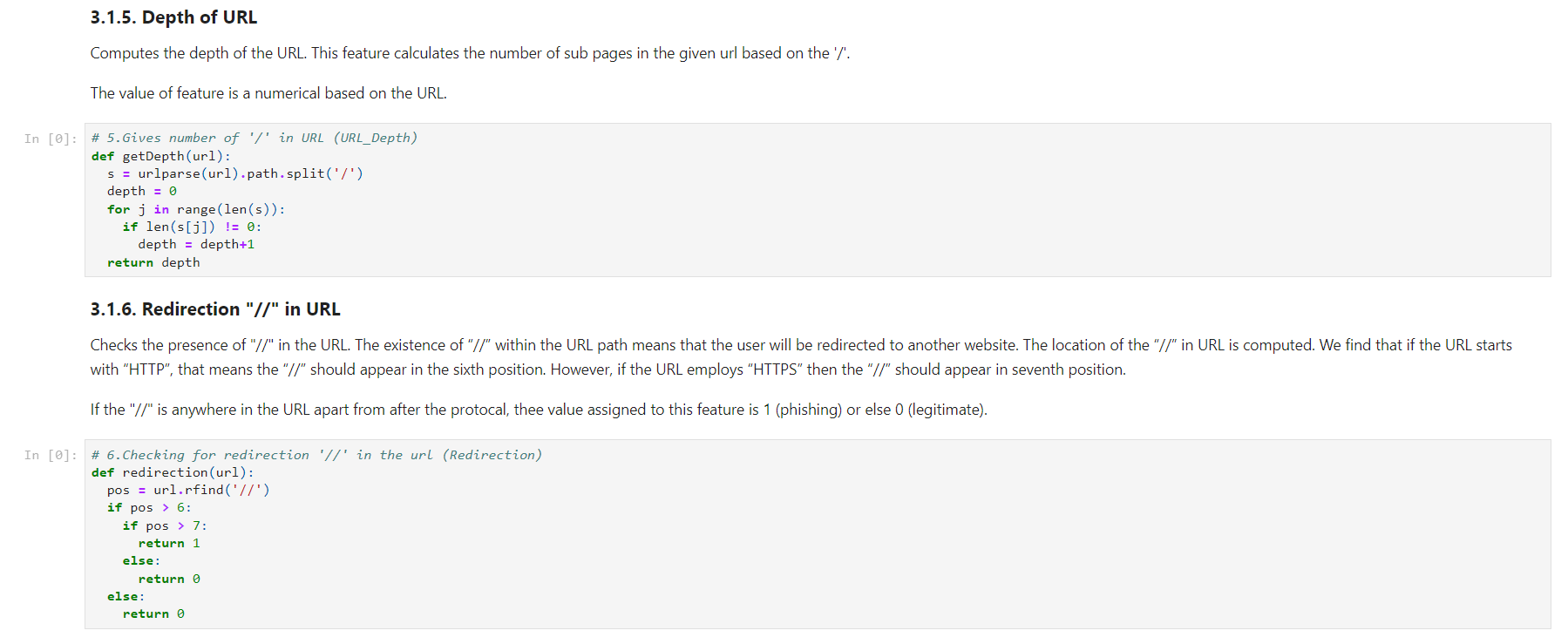
**3.1. Address Bar Based Features:**

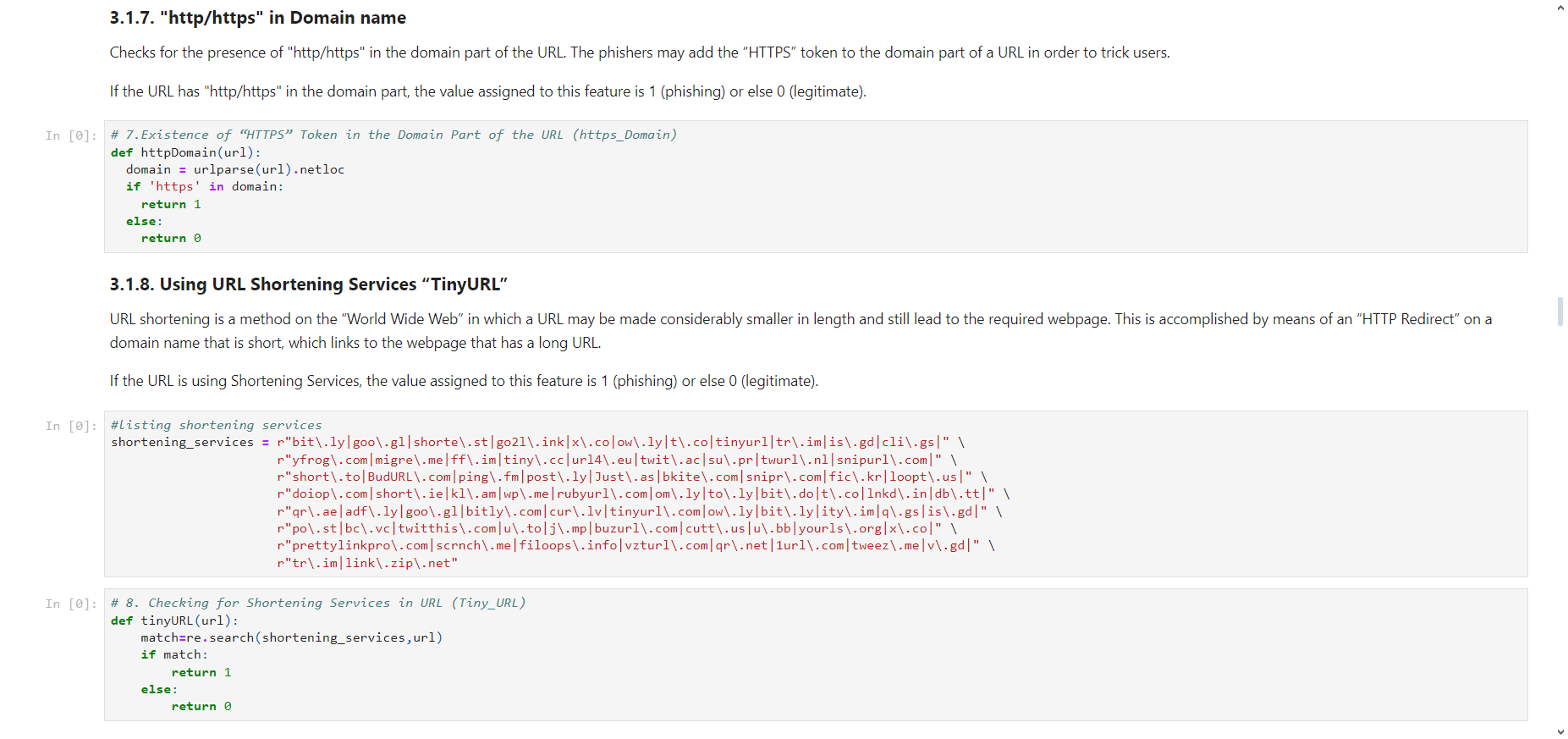
Many features can be extracted that can be considered as address bar base features. Out of them, below mentioned were considered for this project.

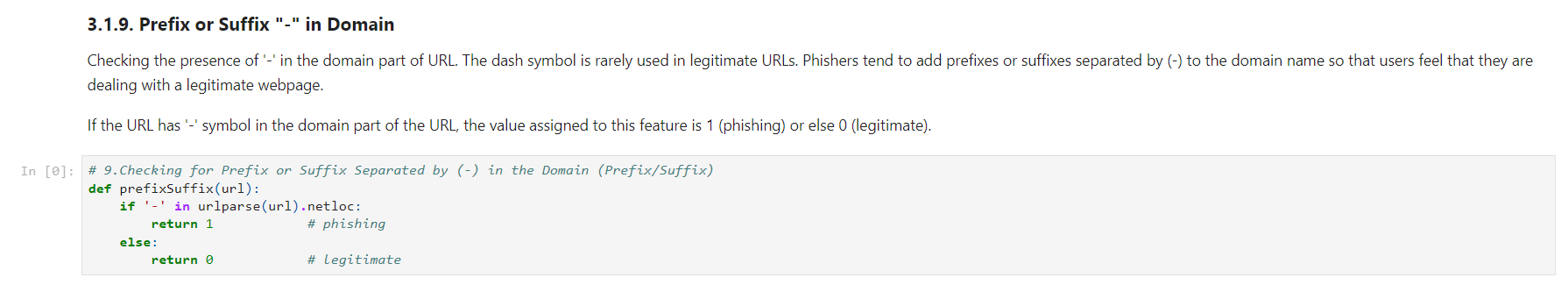
* Domain of URL
* IP Address in URL
* "@" Symbol in URL
* Length of URL
* Depth of URL
* Redirection "//" in URL
* "http/https" in Domain name
* Using URL Shortening Services “TinyURL”
* Prefix or Suffix "-" in Domain

Each of these features are explained and the coded below:







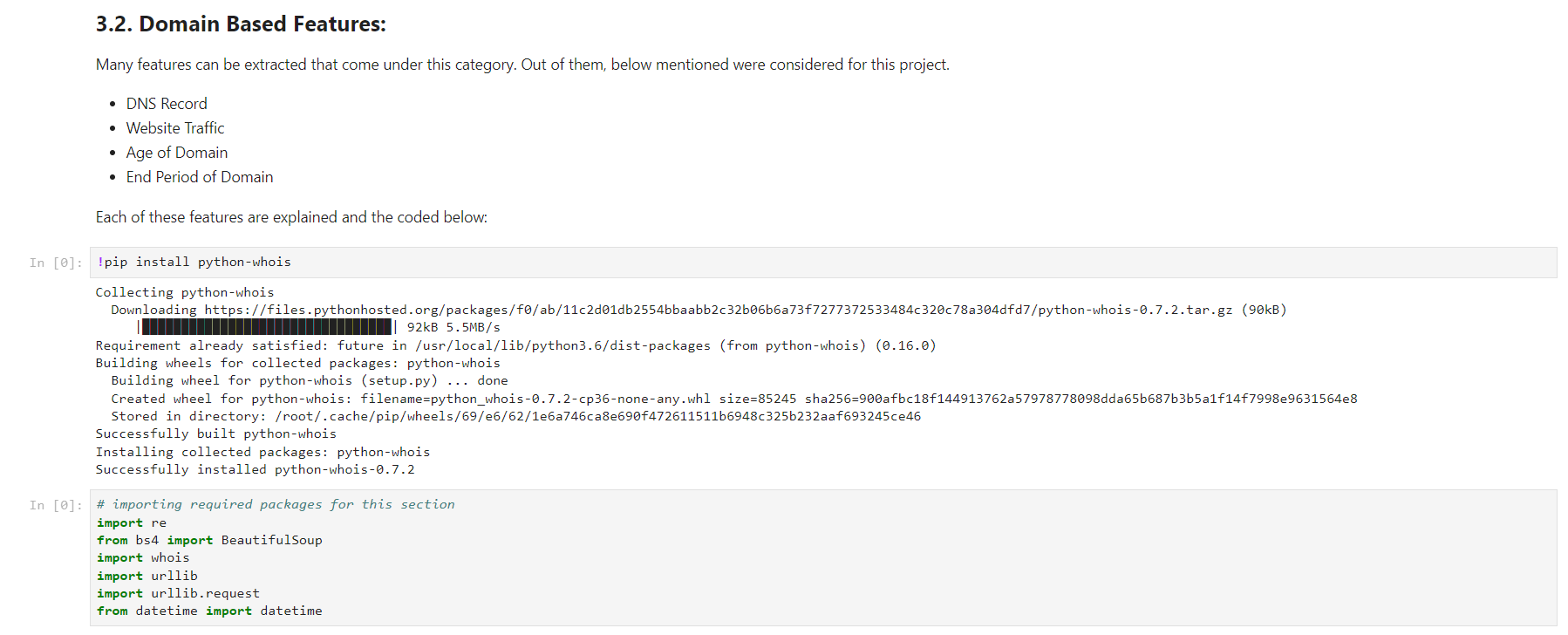


**3.2. Domain Based Features:**

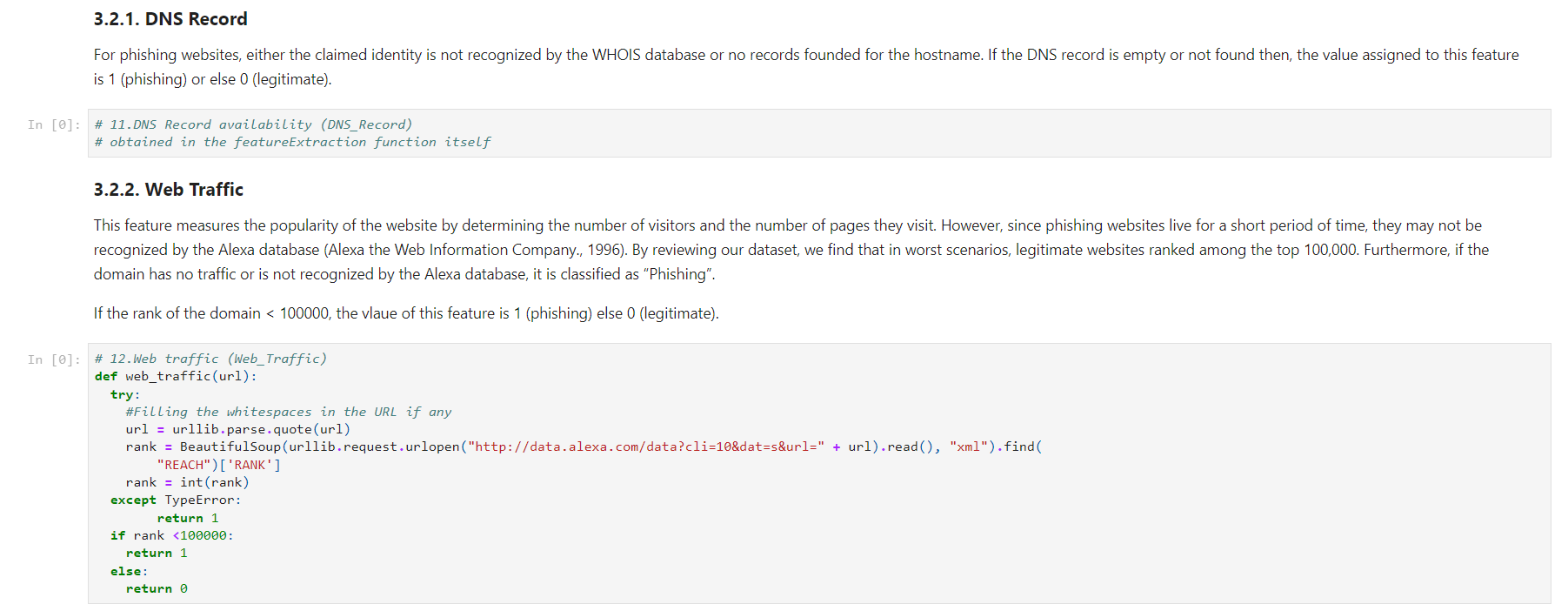
Many features can be extracted that come under this category. Out of them, below mentioned were considered for this project.

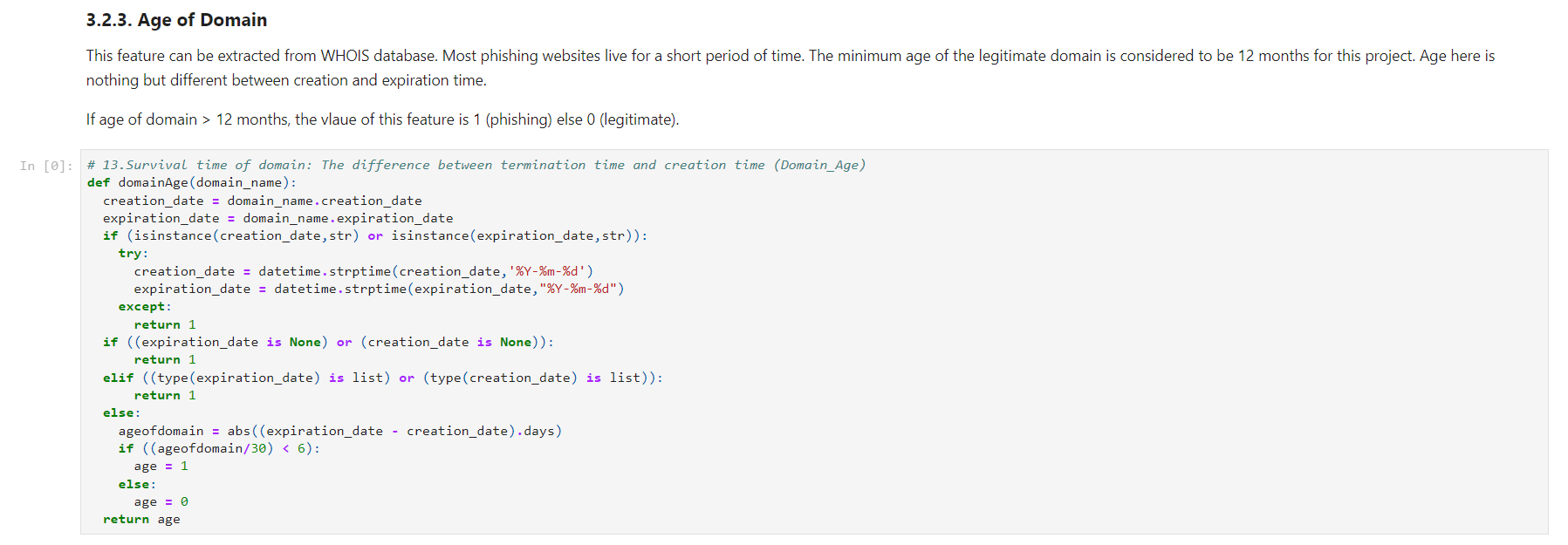
* DNS Record
* Website Traffic
* Age of Domain
* End Period of Domain

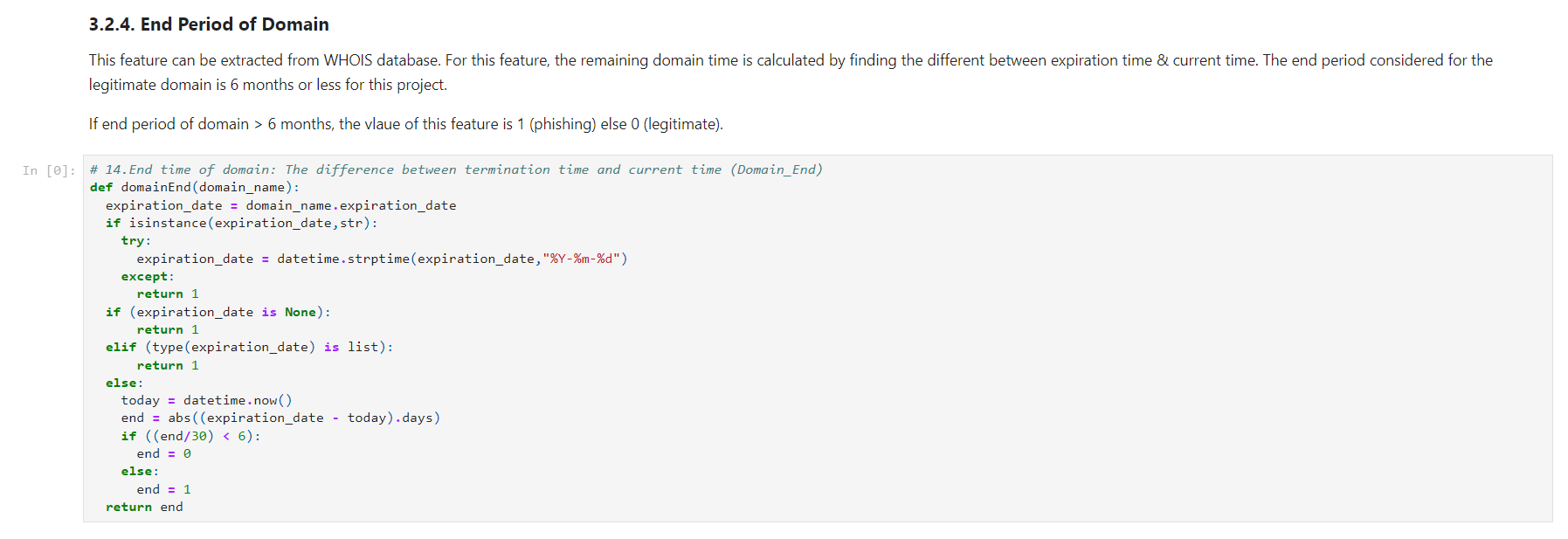
Installing & importing required data



Each of these features are explained and the coded below:



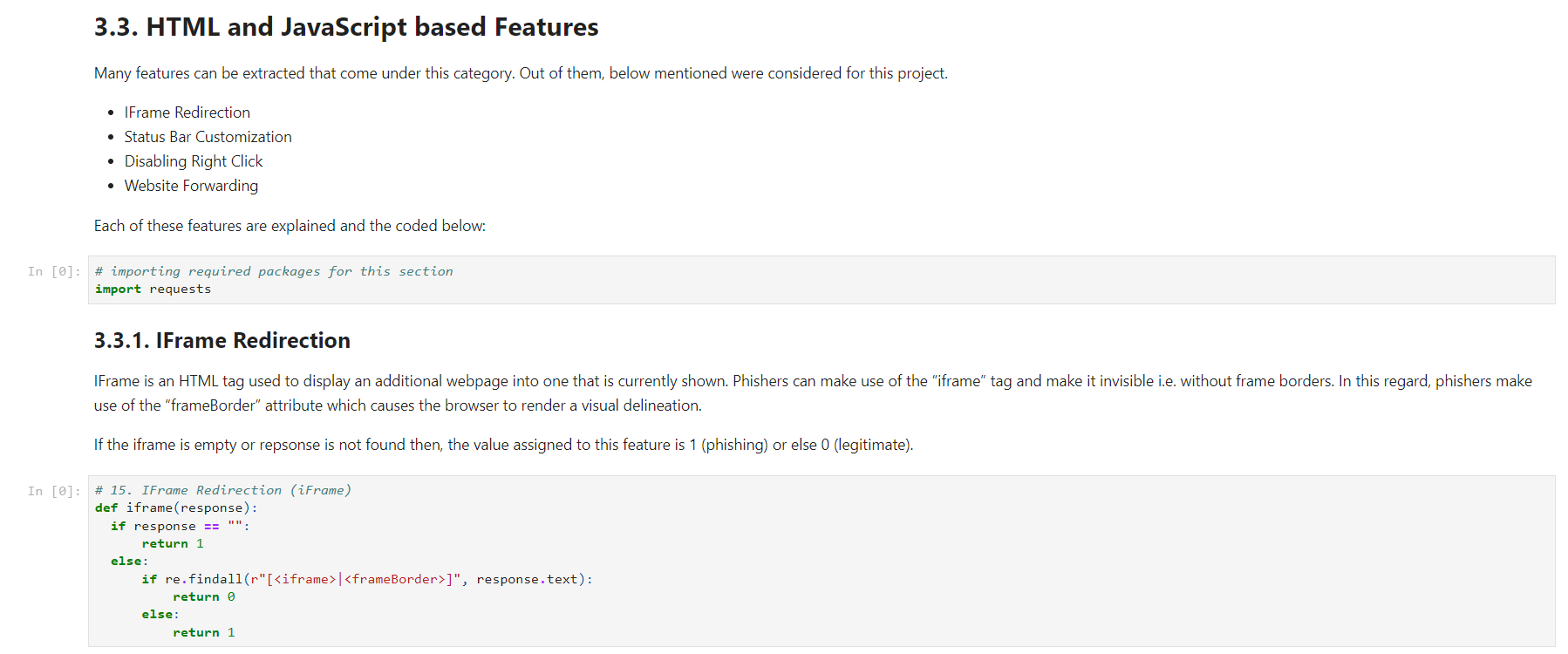


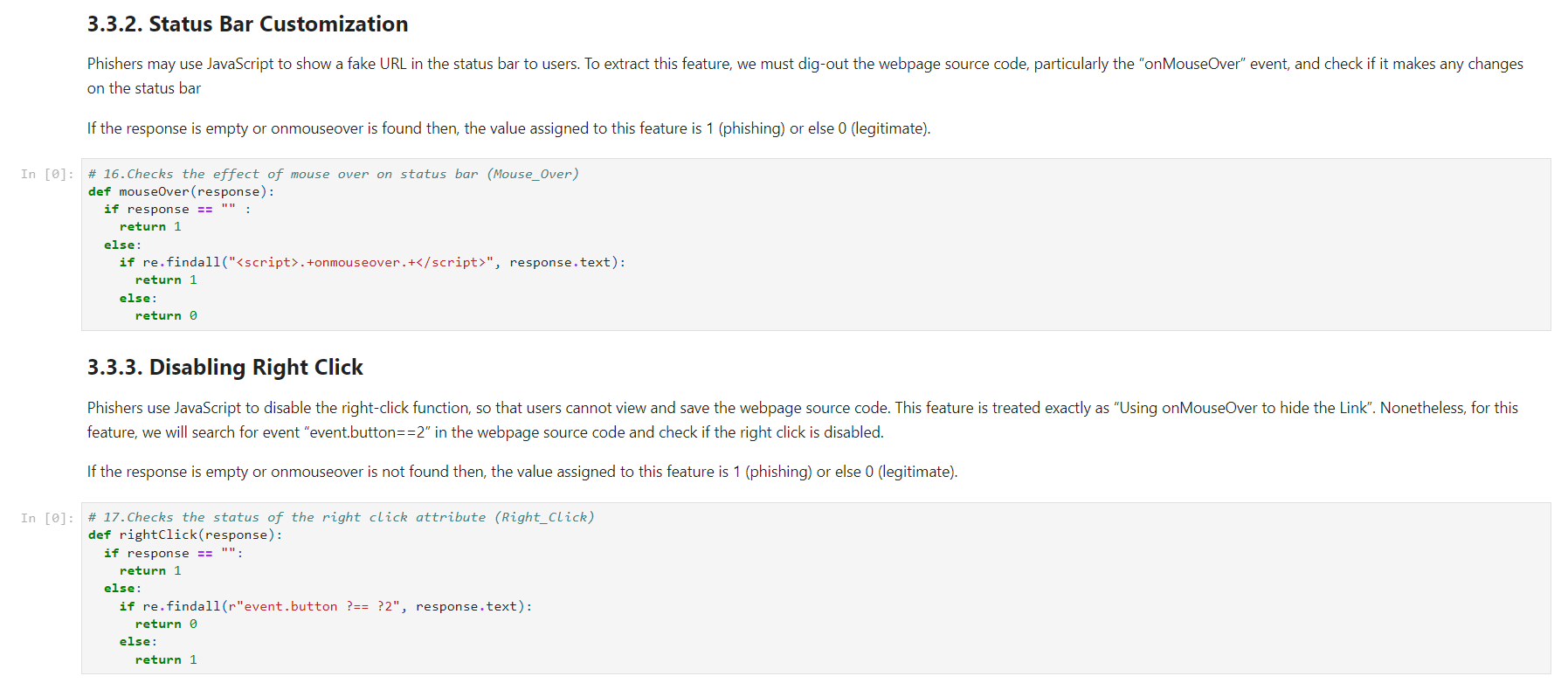


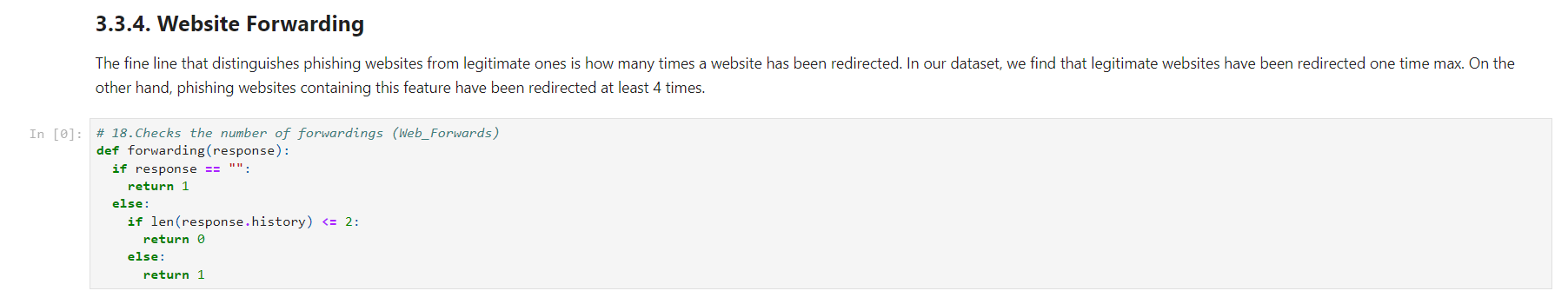
**3.3. HTML and JavaScript based Features**

Many features can be extracted that come under this category. Out of them, below mentioned were considered for this project.

* IFrame Redirection
* Status Bar Customization
* Disabling Right Click
* Website Forwarding

Each of these features are explained and the coded below:





**Computing URL Features**

Create a list and a function that calls the other functions and stores all the features of the URL in the list. We will extract the features of each URL and append to this list.

**Legitimate URLs:**

Now, feature extraction is done on legitimate URLs.



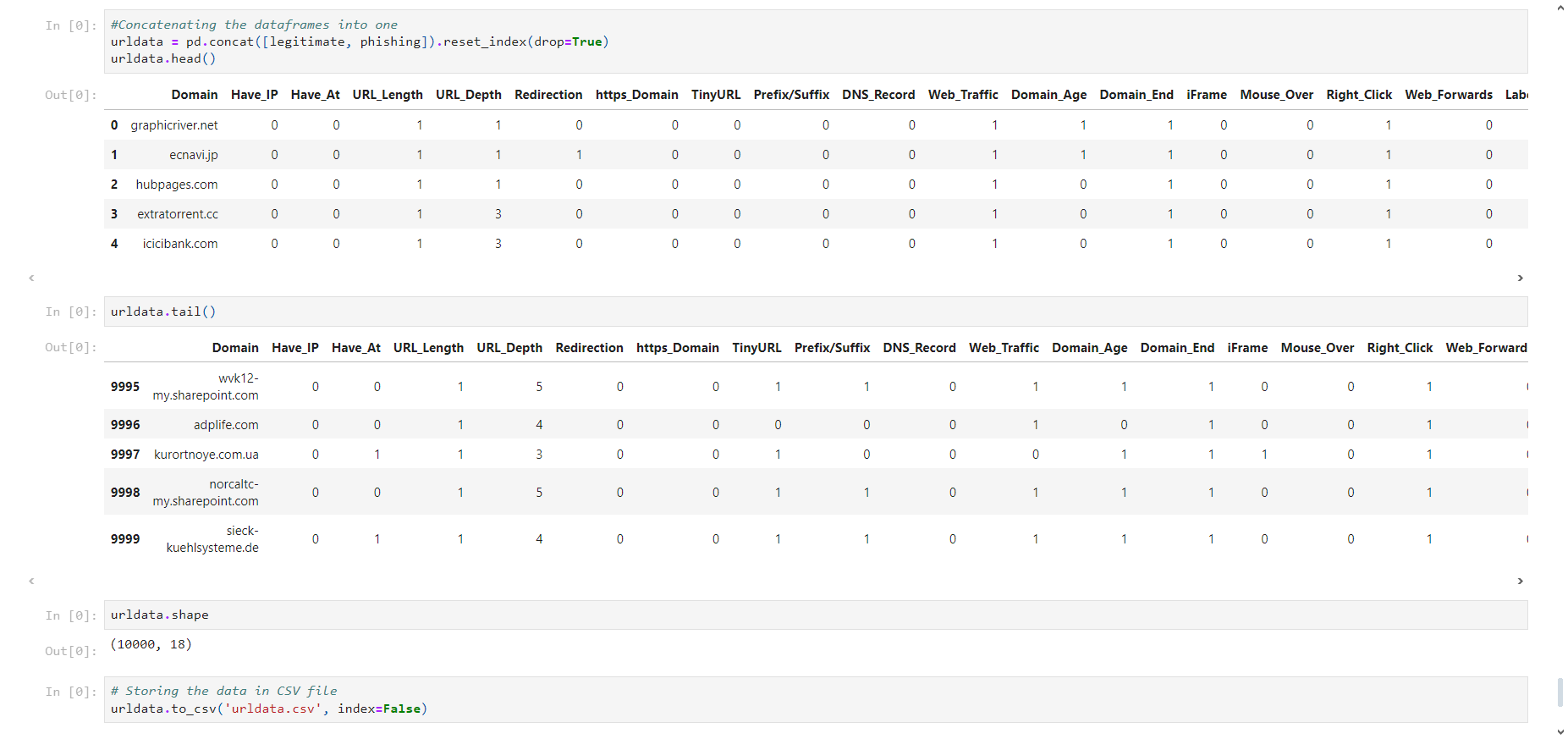
**Phishing URLs:**

Now, feature extraction is performed on phishing URLs.



**Final Dataset**

In the above section we formed two data frames of legitimate & phishing URL features. Now, we will combine them to a single data frame and export the data to csv file for the Machine Learning training done in other notebook.



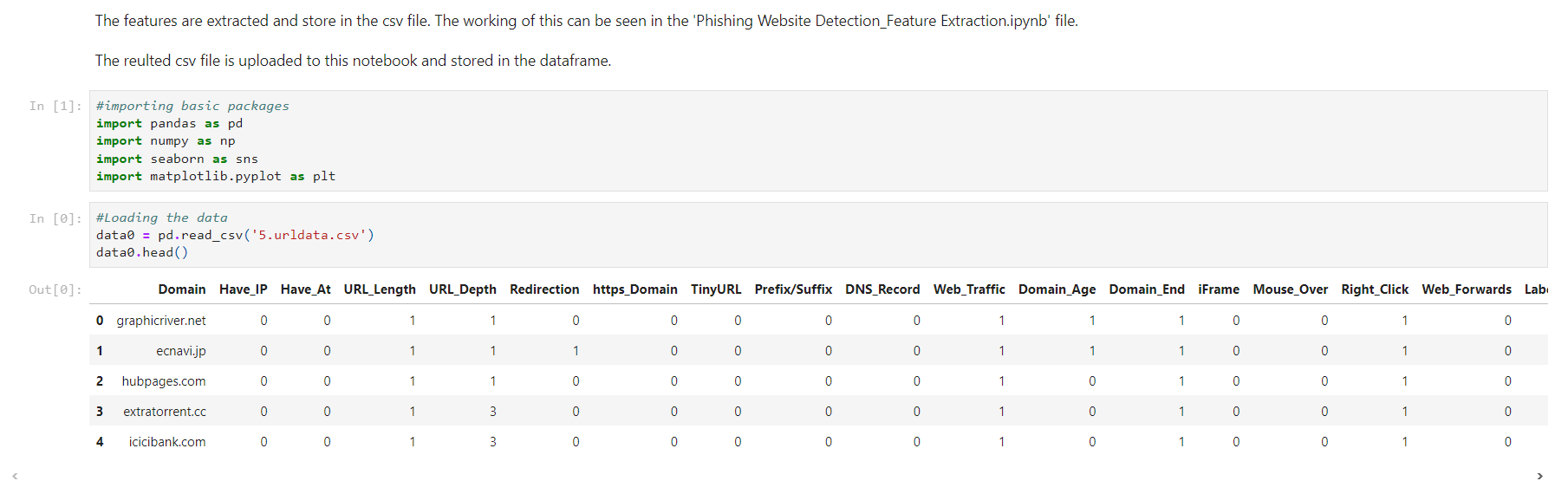
With this the objective of this notebook is achieved. We finally extracted 18 features for 10,000 URL which has 5000 phishing & 5000 legitimate URLs.

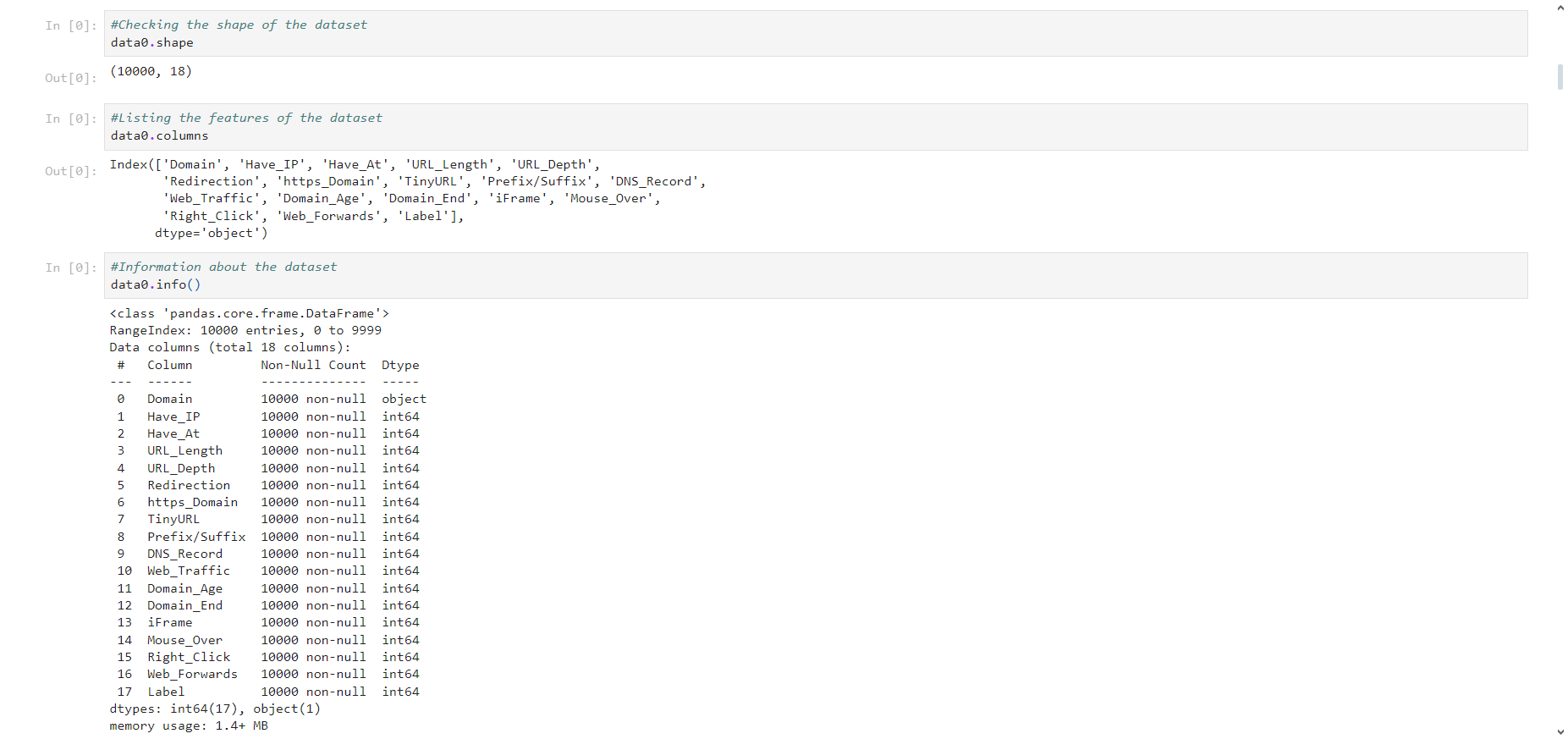
**6.2 Detecting Phishing Website**

**Loading Data:**

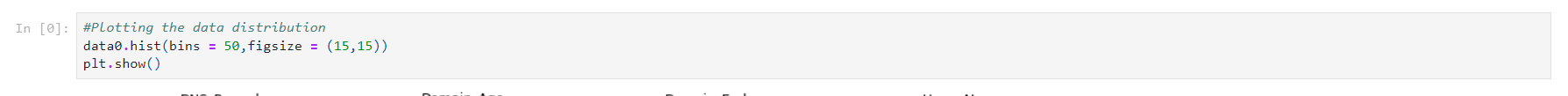
The features are extracted and store in the csv file. The working of this can be seen in the 'Phishing Website Detection\_Feature Extraction.ipynb' file.

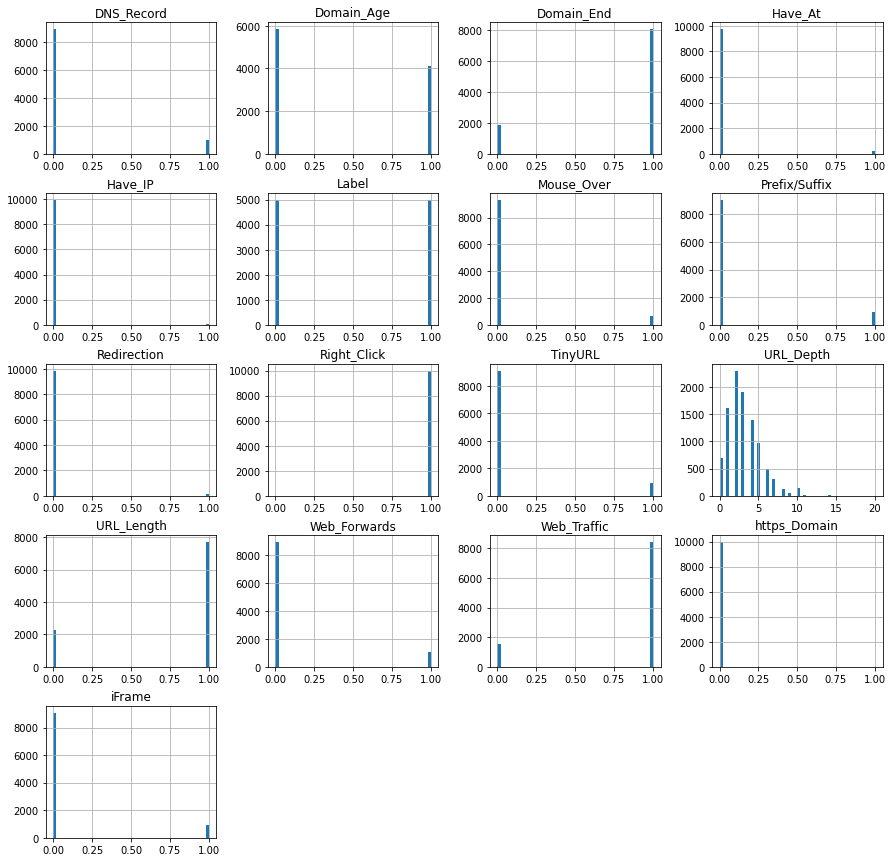
The resulted csv file is uploaded to this notebook and stored in the data frame.

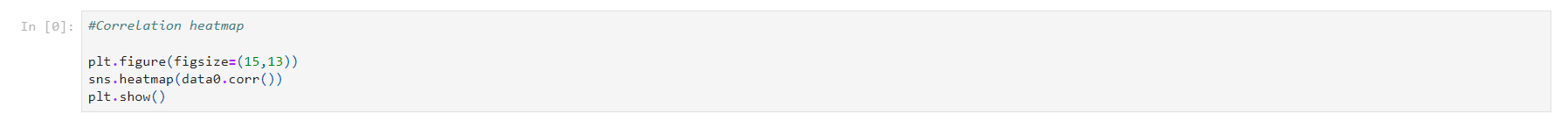


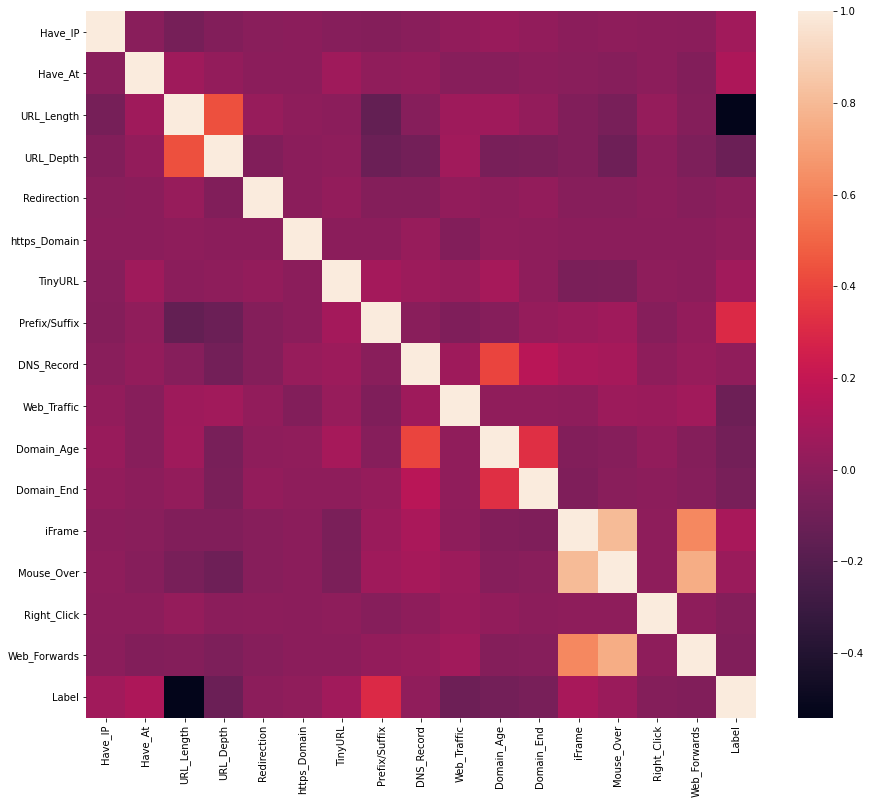


**Visualizing the data**

Few plots and graphs are displayed to find how the data is distributed and the how features are related to each other.

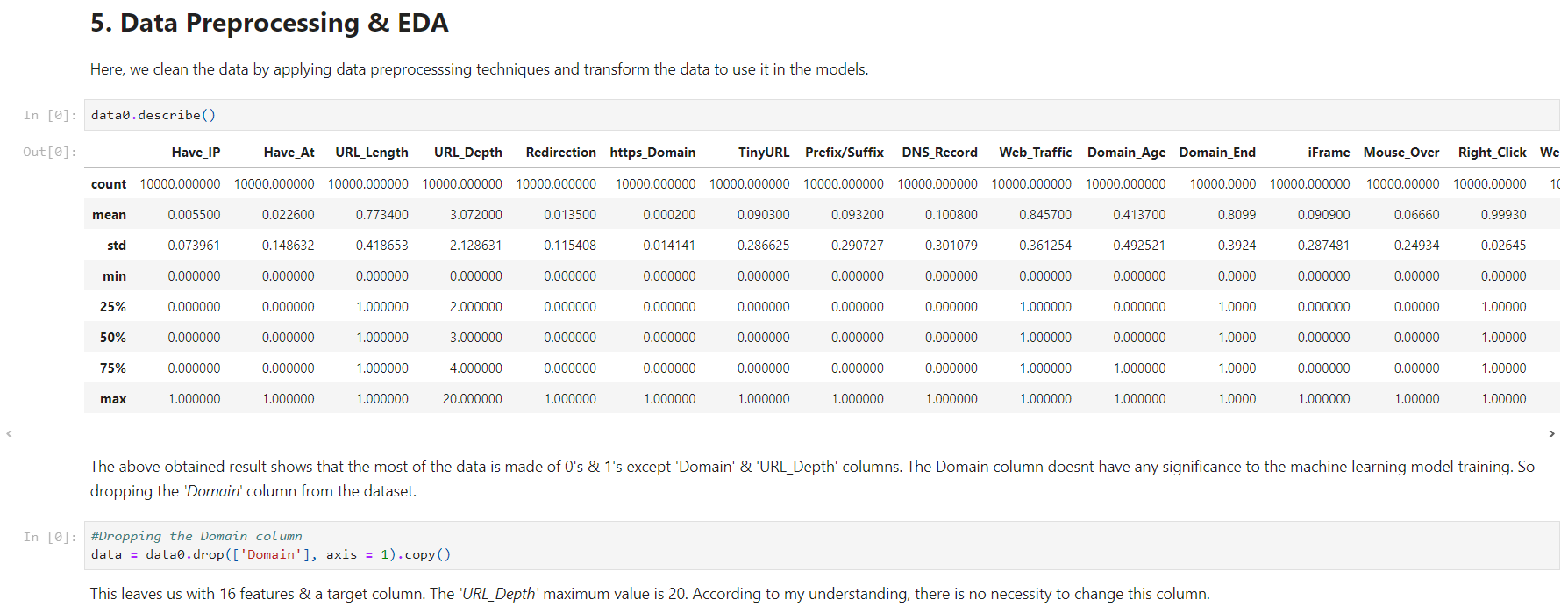


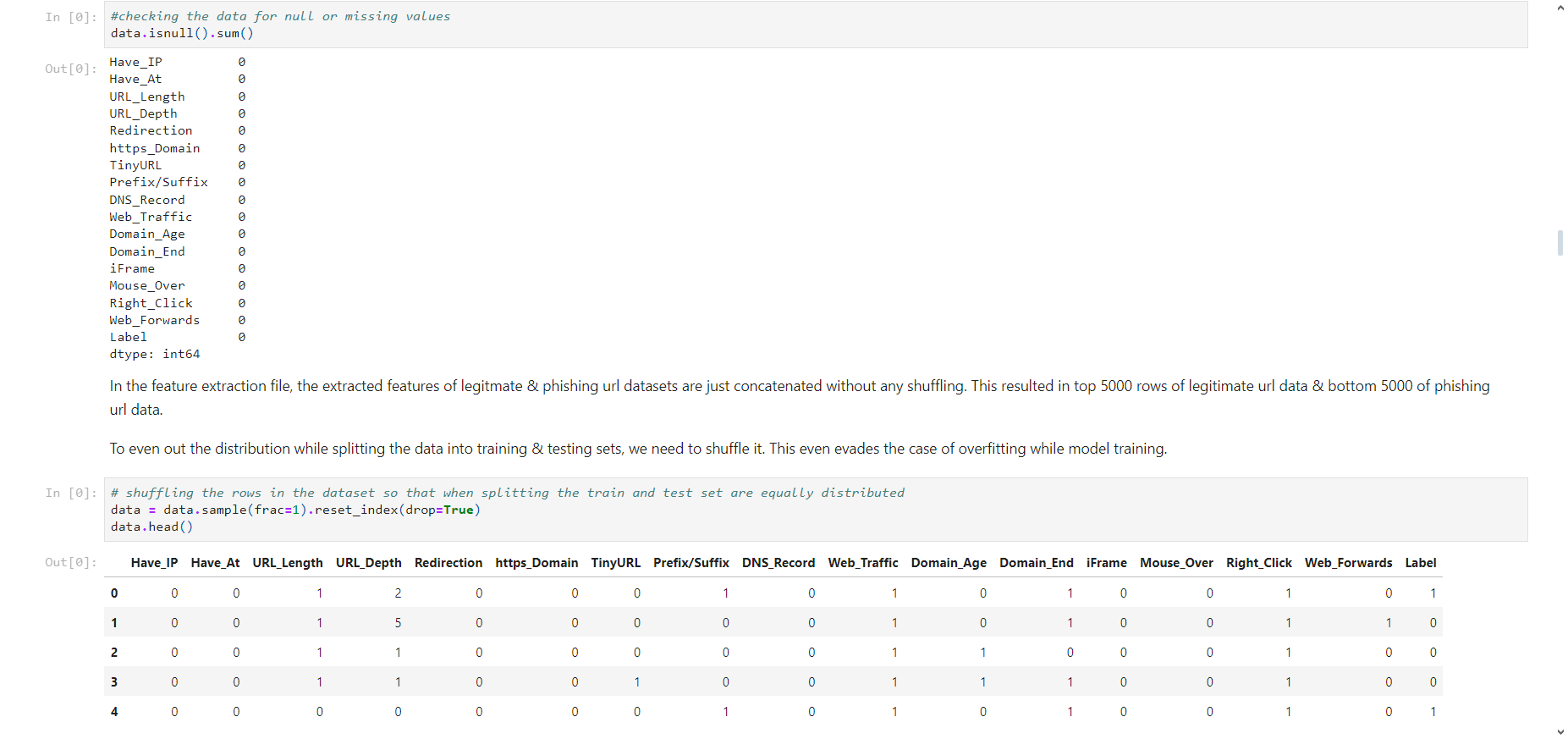




**Data Preprocessing & EDA**

Here, we clean the data by applying data preprocessing techniques and transform the data to use it in the models.





From the above execution, it is clear that the data doesn’t have any missing values.

By this, the data is thoroughly preprocessed & is ready for training.

**Splitting the Data**

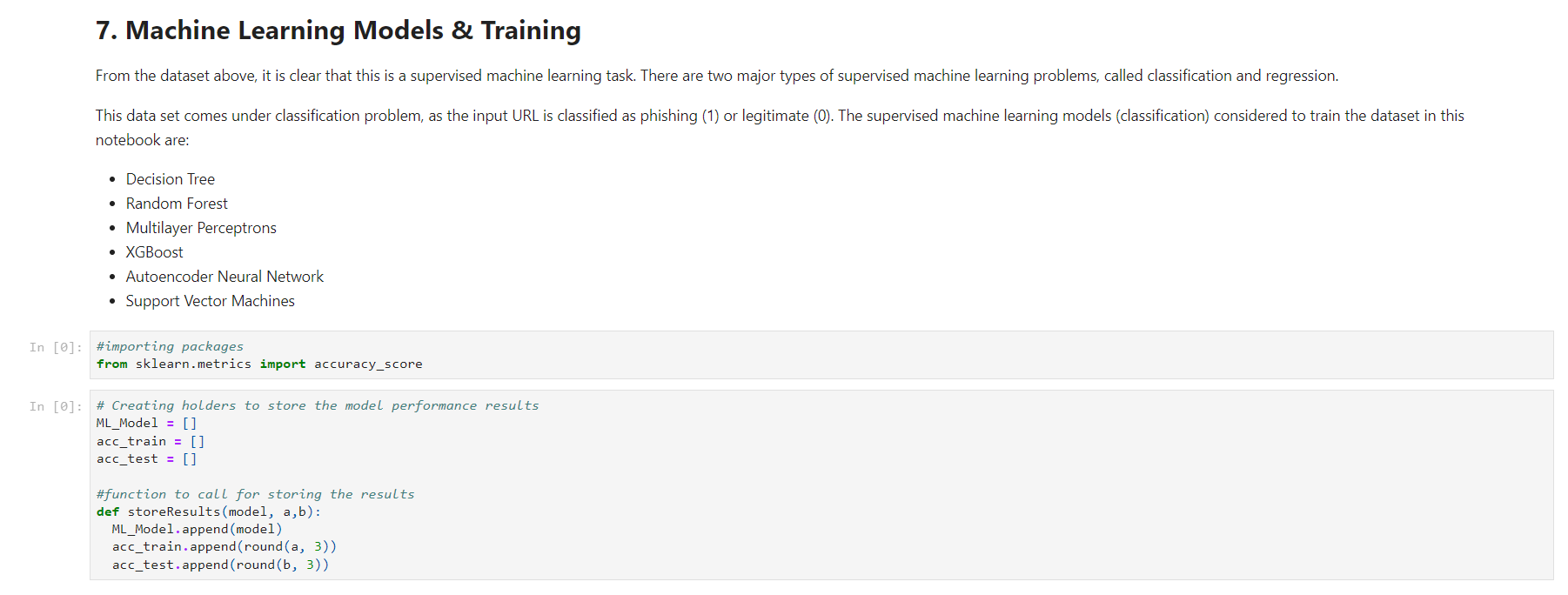


**Machine Learning Models & Training**

From the dataset above, it is clear that this is a supervised machine learning task. There are two major types of supervised machine learning problems, called classification and regression.

This data set comes under classification problem, as the input URL is classified as phishing (1) or legitimate (0). The supervised machine learning models (classification) considered to train the dataset in this notebook are:

* Decision Tree
* Random Forest
* Multilayer Perceptrons
* XGBoost
* Autoencoder Neural Network
* Support Vector Machines

****

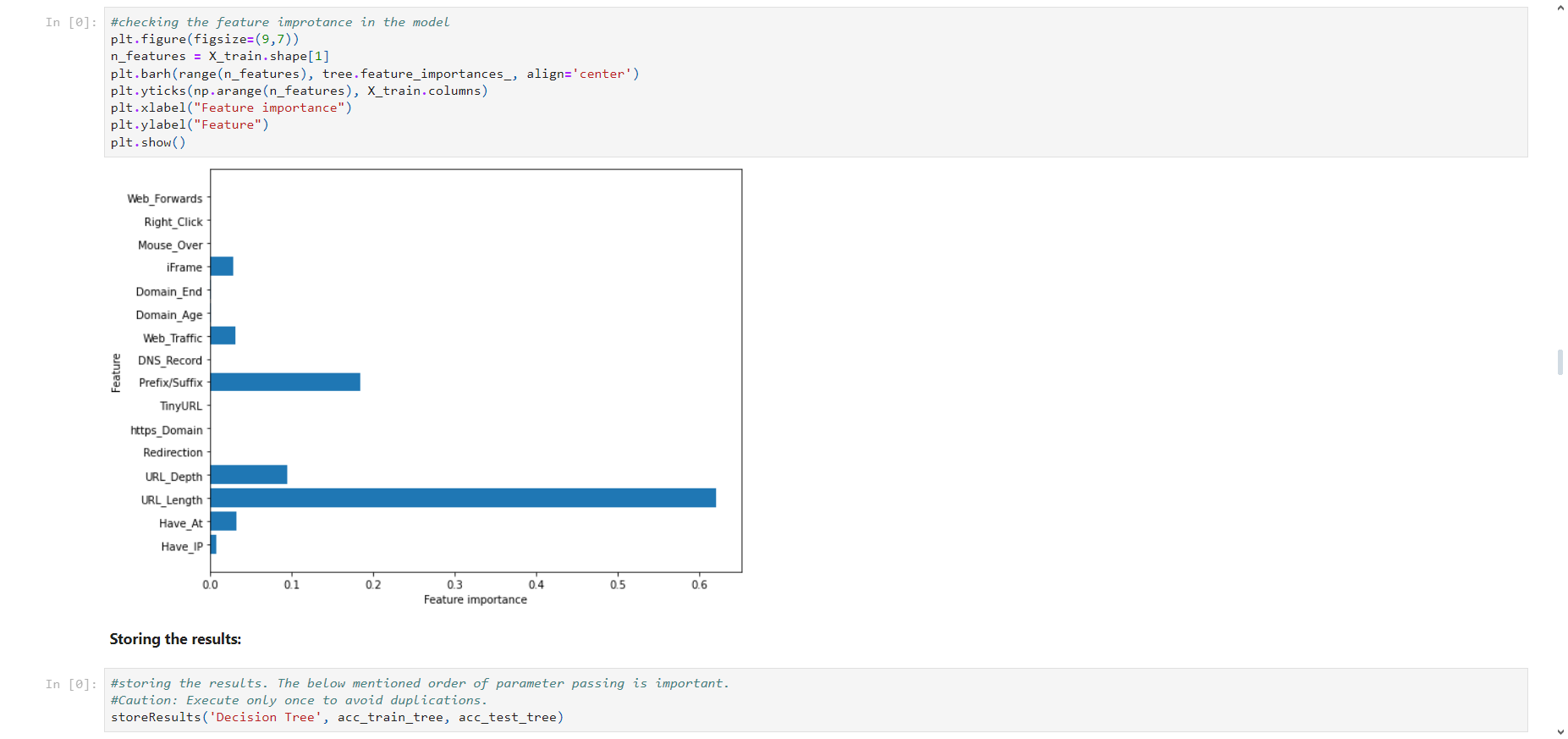
1. **Decision Tree Classifier**

Decision trees are widely used models for classification and regression tasks. Essentially, they learn a hierarchy of if/else questions, leading to a decision. Learning a decision tree means learning the sequence of if/else questions that gets us to the true answer most quickly.

In the machine learning setting, these questions are called tests (not to be confused with the test set, which is the data we use to test to see how generalizable our model is). To build a tree, the algorithm searches over all possible tests and finds the one that is most informative about the target variable.

****

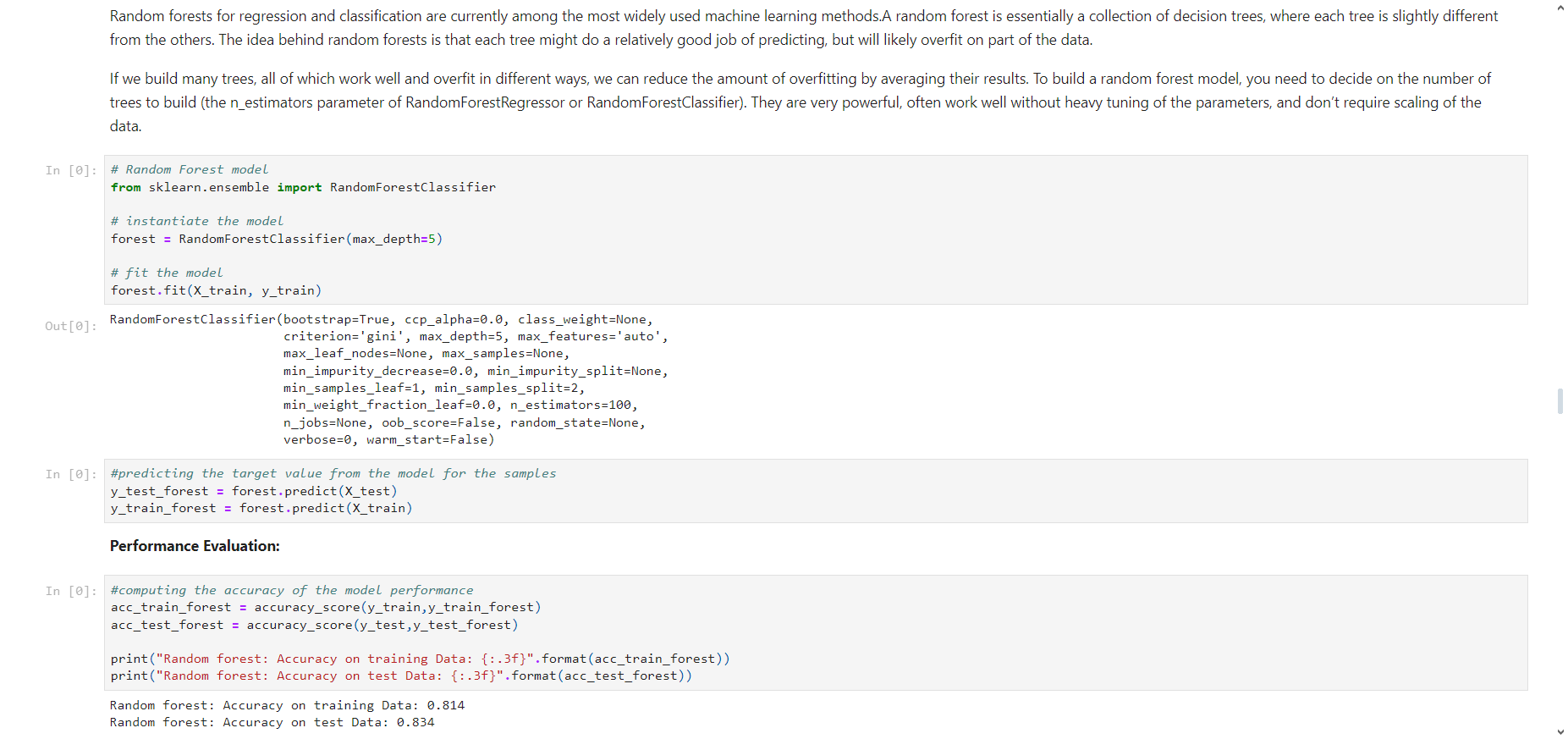
Checking the feature importance in the model & storing the results.



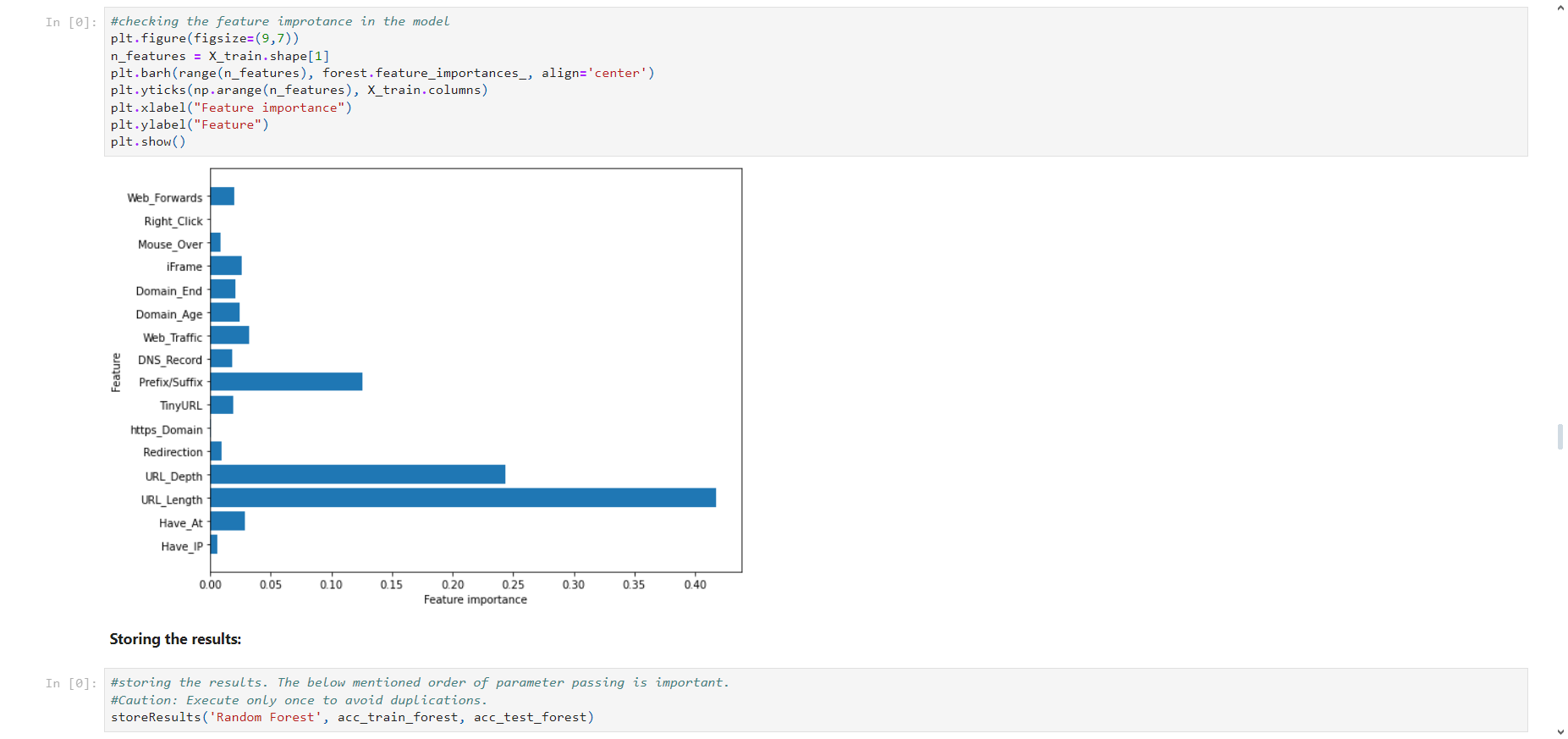
**2. Random Forest Classifier**

Random forests for regression and classification are currently among the most widely used machine learning methods. A random forest is essentially a collection of decision trees, where each tree is slightly different from the others. The idea behind random forests is that each tree might do a relatively good job of predicting, but will likely overfit on part of the data.

If we build many trees, all of which work well and overfit in different ways, we can reduce the amount of overfitting by averaging their results. To build a random forest model, you need to decide on the number of trees to build (the n\_estimators parameter of RandomForestRegressor or RandomForestClassifier). They are very powerful, often work well without heavy tuning of the parameters, and don’t require scaling of the data.



Checking the feature importance in the model & storing the results.



**3. Multilayer Perceptrons (MLPs): Deep Learning**

Multilayer perceptrons (MLPs) are also known as (vanilla) feed-forward neural networks, or sometimes just neural networks. Multilayer perceptrons can be applied for both classification and regression problems.

MLPs can be viewed as generalizations of linear models that perform multiple stages of processing to come to a decision.



**4. XGBoost Classifier**

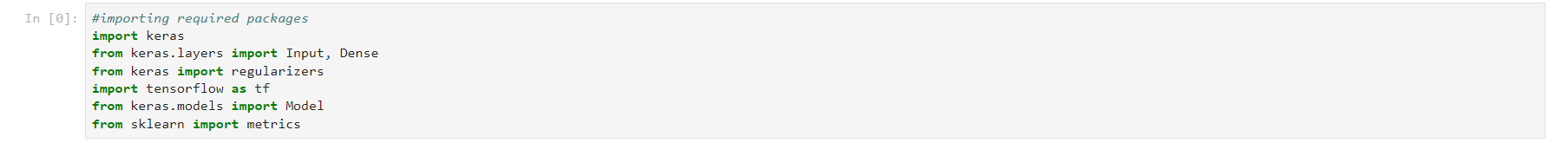
XGBoost is one of the most popular machine learning algorithms these days. XGBoost stands for eXtreme Gradient Boosting. Regardless of the type of prediction task at hand; regression or classification. XGBoost is an implementation of gradient boosted decision trees designed for speed and performance.

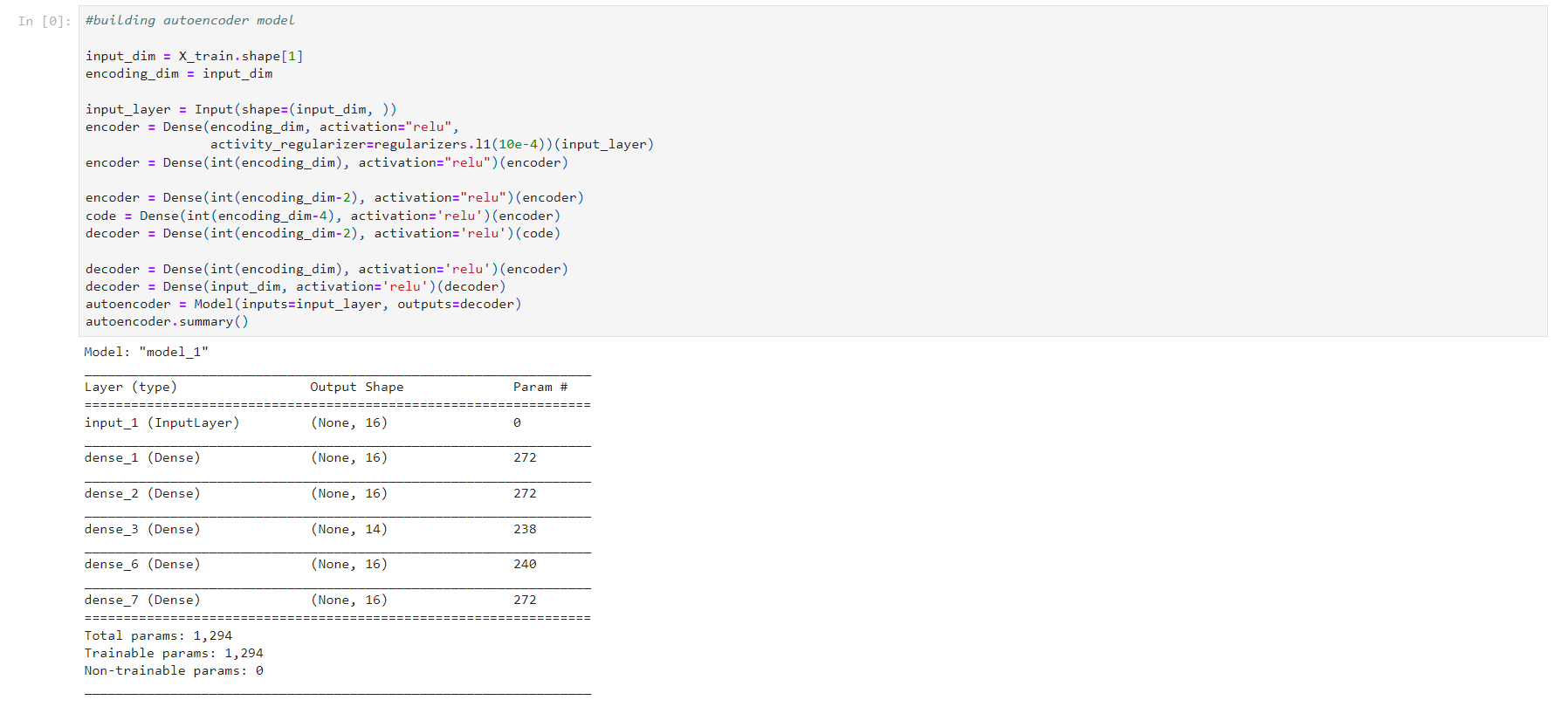


**5. Autoencoder Neural Network**

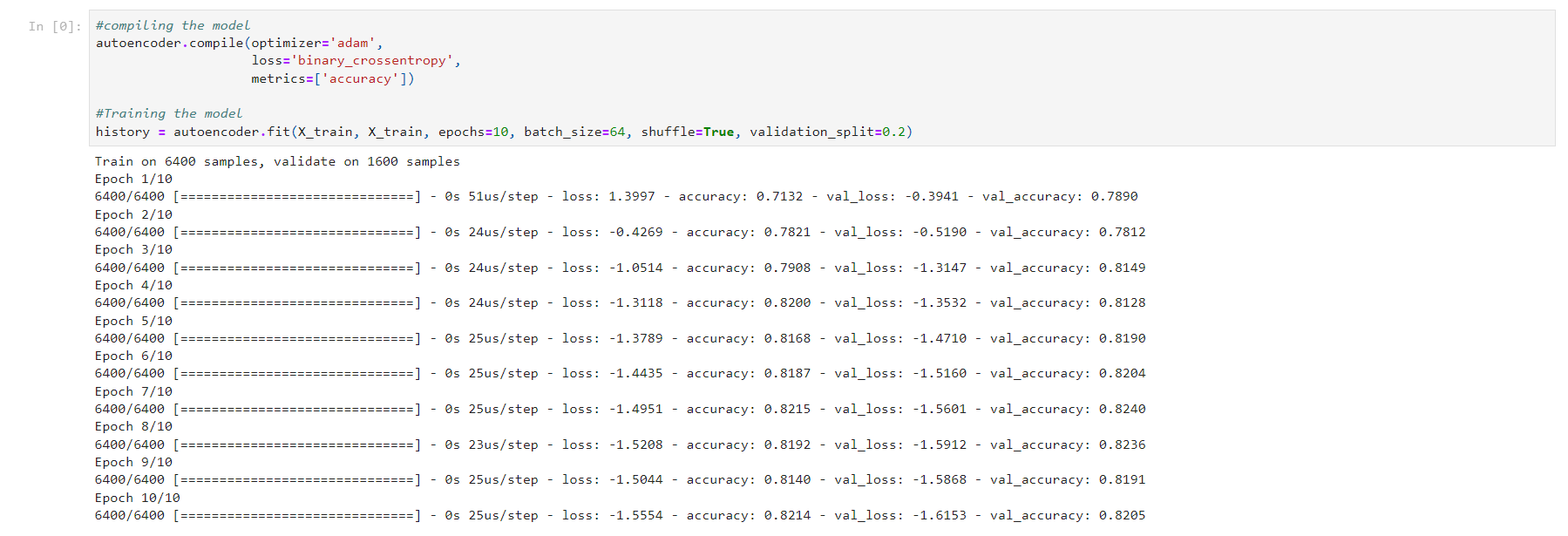
An auto encoder is a neural network that has the same number of input neurons as it does outputs. The hidden layers of the neural network will have fewer neurons than the input/output neurons. Because there are fewer neurons, the auto-encoder must learn to encode the input to the fewer hidden neurons. The predictors (x) and output (y) are exactly the same in an auto encoder.

Importing required packages,





Compiling the model





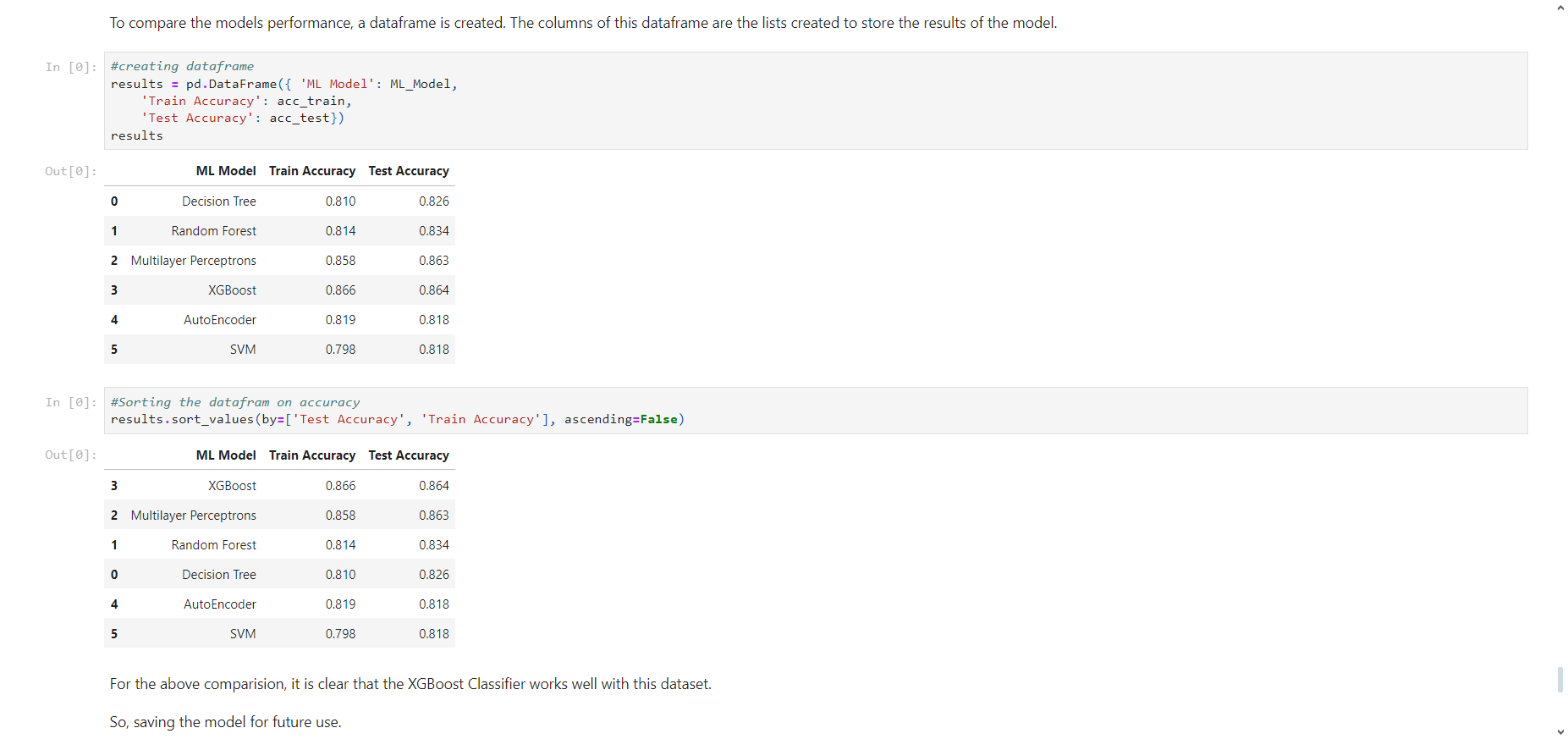
**6. Support Vector Machines**

In machine learning, support-vector machines (SVMs, also support-vector networks) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier.



**Comparison of Models**

To compare the models performance, a dataframe is created. The columns of this dataframe are the lists created to store the results of the model.

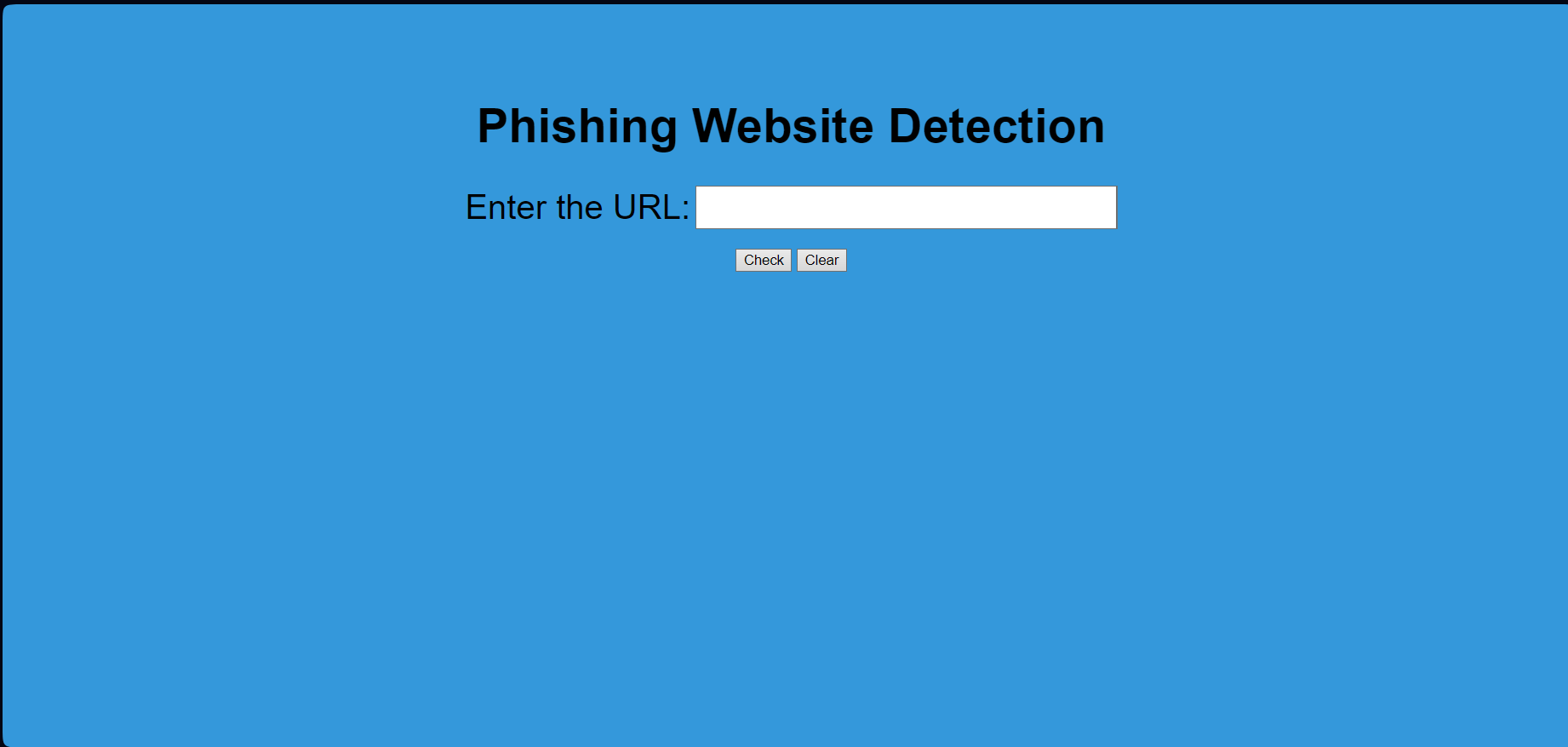


Saving the model file & testing the same.

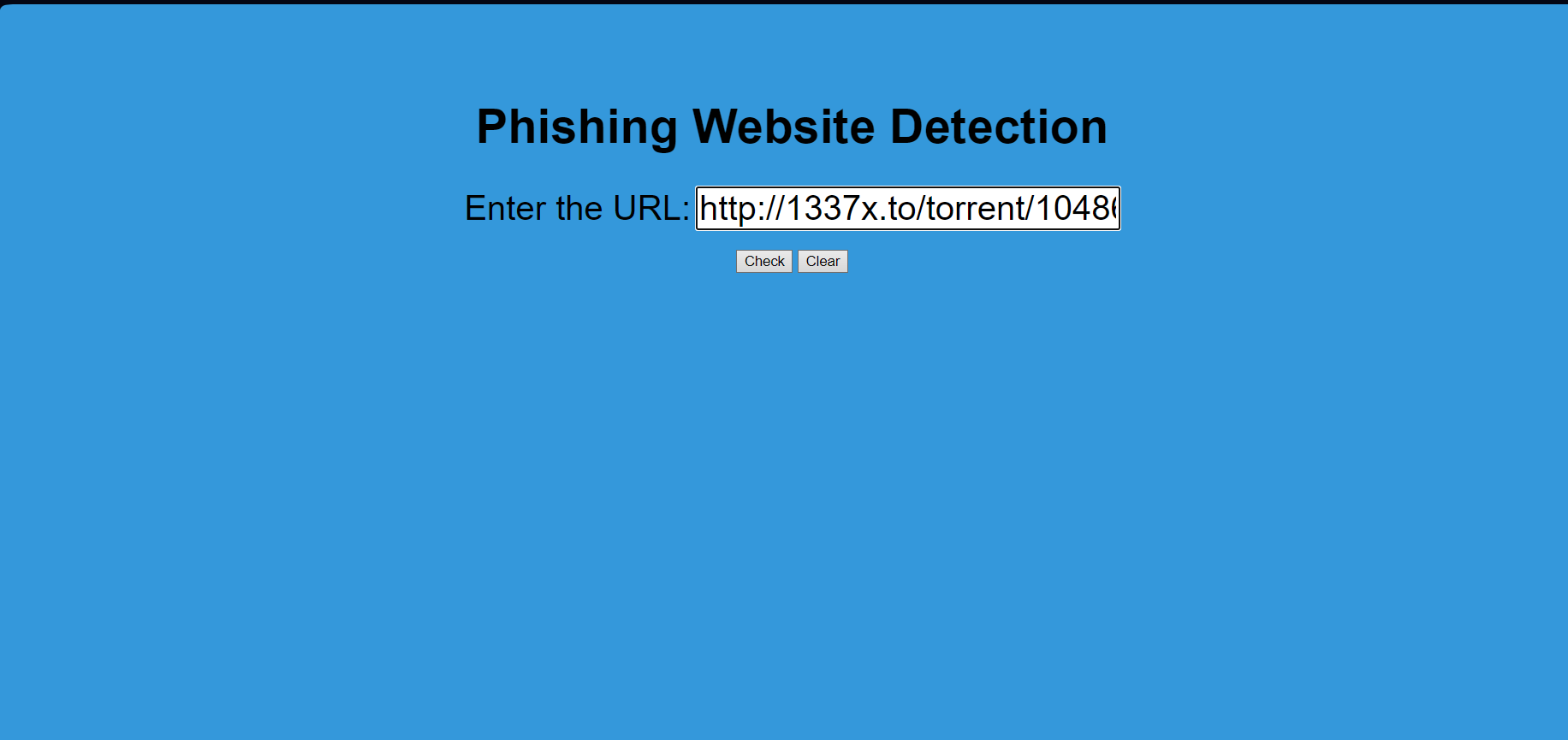


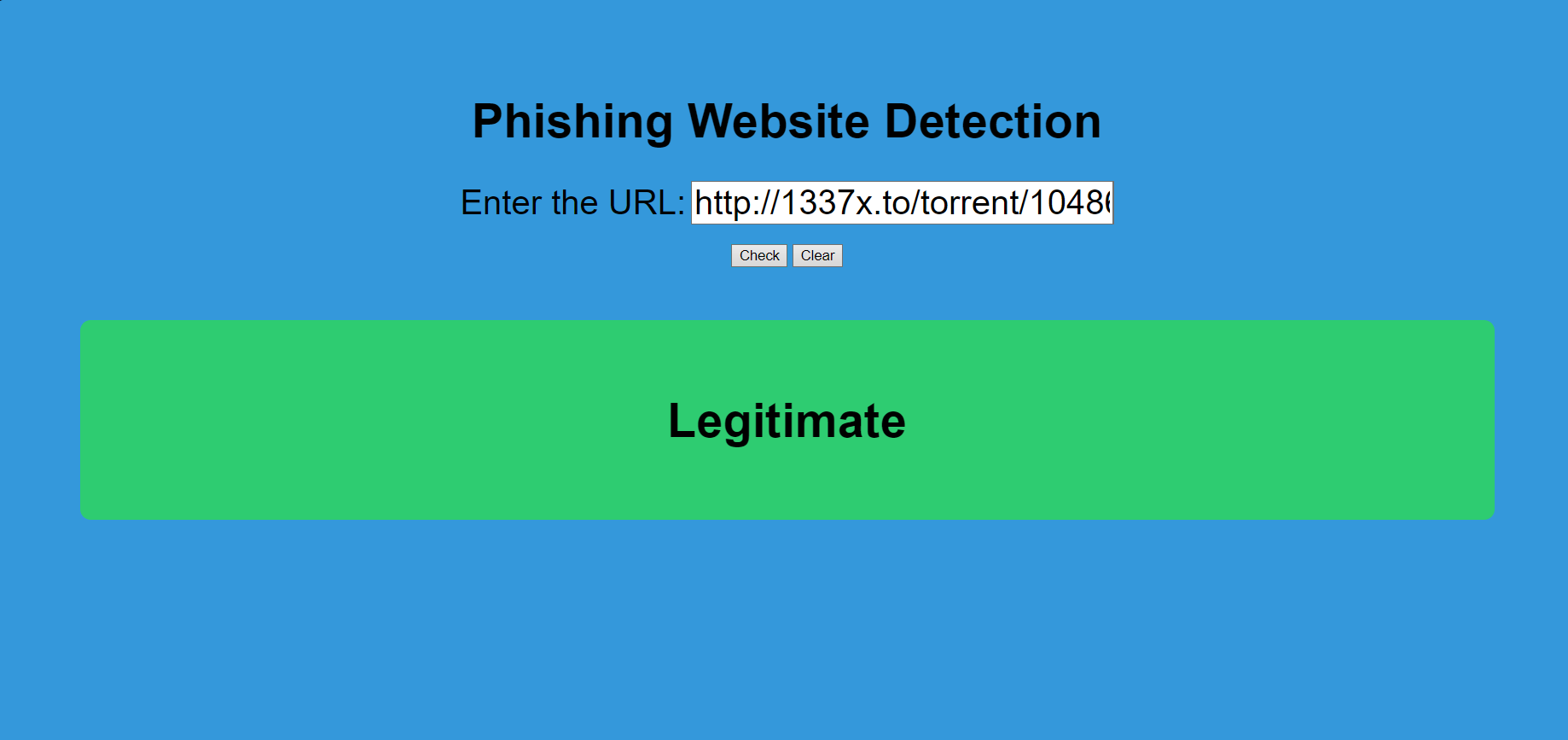
**7. Front Page**

**7.1 Home Page**

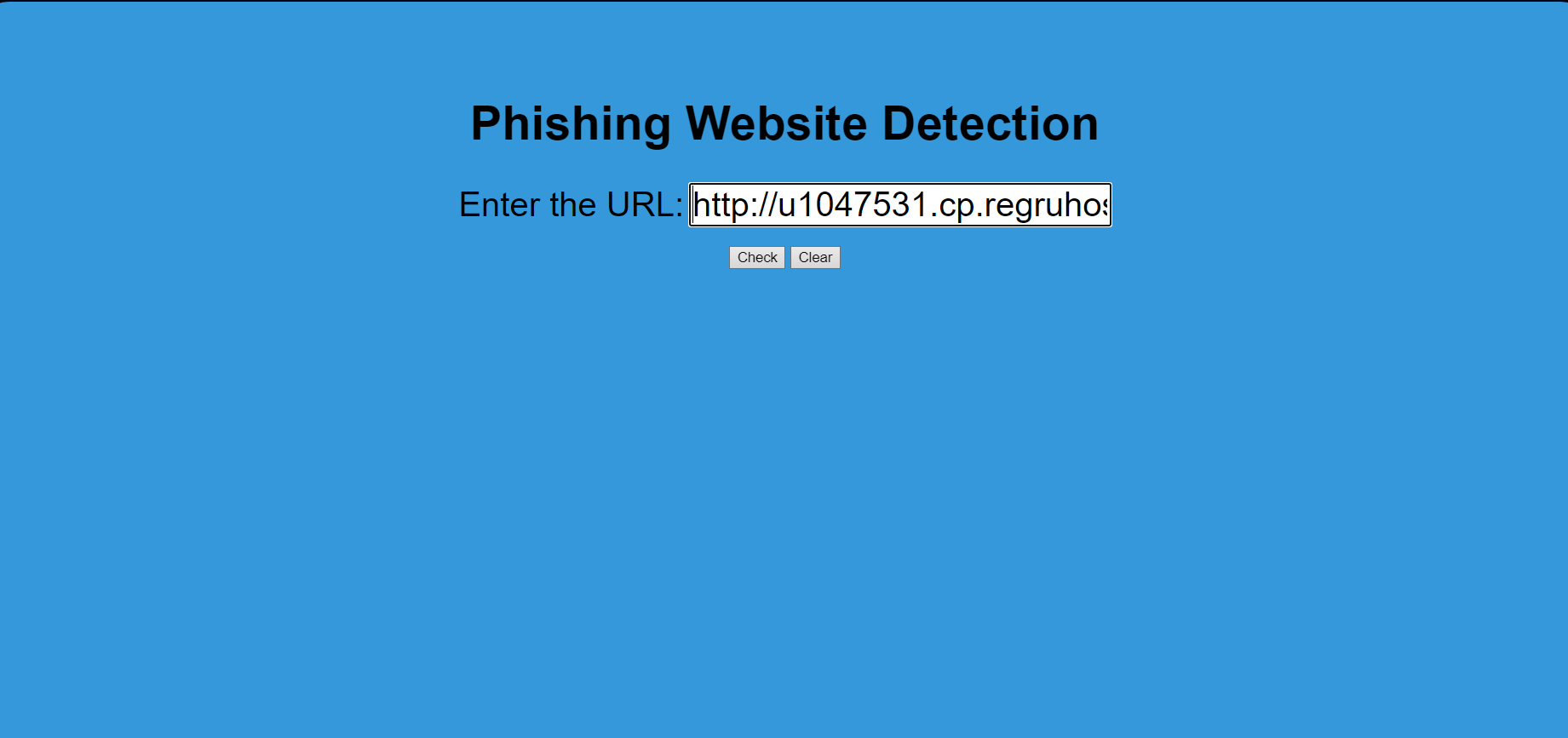
****

**7.2 Legitimate URL**

****

****

**7.3 Phishing URL**

****

****

**8. Future Plans**

Engaging in this project is not only intellectually stimulating but also highly rewarding, offering a wealth of knowledge and insights into the intricate realm of phishing websites and their differentiation from legitimate ones. As the project progresses, a deeper understanding of the nuanced features distinguishing these two categories unfolds, contributing to a broader comprehension of cybersecurity challenges. To extend the impact of this research, future steps may involve taking the project a step further by considering the development of browser extensions or creating a Graphical User Interface (GUI). Such extensions or interfaces could leverage the insights gained to classify inputted URLs, determining their legitimacy or phishing status using the trained and saved machine learning model. This extension or GUI would serve as a practical tool, providing users with real-time information and contributing to the broader initiative of fortifying online security measures. These next steps not only enhance the practical utility of the project but also open avenues for broader applications in the ongoing fight against evolving cyber threats.

**9. BIBLIOGRAPHY**

[1] Samuel Marchal, Jérôme François, Radu State, and Thomas Engel,

“PhishStorm: Detecting Phishing with Streaming Analytics,” IEEE Transactions on

Network and Service Management, vol. 11, issue: 4, pp. 458-471, December 2014

[2] Mohammed Nazim Feroz, Susan Mengel, “Phishing URL Detection Using URL

Ranking,” IEEE International Congress on Big Data, July 2015

[3] Mahdieh Zabihimayvan, Derek Doran, “Fuzzy Rough Set Feature Selection to

Enhance Phishing Attack Detection,” International Conference on Fuzzy Systems

(FUZZ-IEEE), New Orleans, LA,

USA, June 2019

[4] Moitrayee Chatterjee, Akbar-Siami Namin, “Detecting Phishing Websites

through Deep Reinforcement Learning,” IEEE 43rd Annual Computer Software

and Applications Conference (COMPSAC), July 2019

[5] Chun-Ying Huang, Shang-Pin Ma, Wei-Lin Yeh, Chia-Yi Lin, ChienTsung Liu,

“Mitigate web phishing using site signatures,” TENCON 2010-2010 IEEE Region

10 Conference, January 2011

**Website Reference:**

* https://blog.keras.io/building-autoencoders-in-keras.html
* https://en.wikipedia.org/wiki/Autoencoder
* https://mc.ai/a-beginners-guide-to-build-stacked-autoencoder-and-tying-weights-with-it/
* https://github.com/siddharthanshree/Sid
* https://machinelearningmastery.com/save-gradient-boosting-models-xgboost-python/
* https://archive.ics.uci.edu/ml/datasets/Phishing+Websites