**Phishing URL Detector**

The web is a vast and dynamic place, where billions of people access countless websites every day. Unfortunately, there are numerous frauds and phishing efforts that can result in money losses and other harm, and not all websites are reliable and safe. We created a web plugin with a 2 - level architecture to identify and stop phishing attempts in order to solve this issue.

The web URLs are immediately compared with a collection of whitelisted and blacklisted URLs at the initial level of the architecture. The blacklisted URLs are marked as potentially phishing assaults, whereas the whitelisted URLs are regarded as reliable. This strategy's foundation is a rapid and efficient method for locating and blocking well-known phishing websites.

However, relying just on a static list of well-known phishing URLs is insufficient to offer effective defence against the constantly changing strategies employed by cybercriminals. We have created a second level of the architecture to handle this problem, and this level uses a machine learning (ML) model to examine URLs and identify probable phishing attempts.

The logistic regression architecture, a well-liked and proven technique for binary classification issues, is used in the ML model. A huge dataset of 550,000 URLs was used to train the model, and tokenization and stemming techniques were used to extract useful attributes from the URLs. Tokenization is the process of dissecting a text string into discrete tokens, which are subsequently employed as features in the machine learning model. Stemming, which involves reducing words to their most basic form, can be used to identify differences in URLs that could otherwise go unnoticed.

The ML model can recognise patterns and elements in URLs that are suggestive of phishing attempts, such as the inclusion of suspect keywords or the use of obfuscation methods, by combining various techniques. The programme can then determine the possibility of each URL being a phishing attempt by assigning a probability score to it.

The plugin provides a thorough and effective phishing detection system by combining the outcomes of the two layers of the architecture. Users are given an additional degree of security against cybercrime by being prevented from accessing URLs that have been reported by either level as potential phishing assaults.

**TECHNOLOGIES USED:**

Python: With its simplicity and high-level programming capabilities, we utilise Python to create systems for web crawls and vulnerability detection. We use numerous libraries and modules that it provides, such as Scrapy and BeautifulSoup, which may be used for site crawling and data extraction. Furthermore, Python's regular expression processing capabilities can be used to search and extract critical data from websites, such as URLs, input fields, and other information that can be used to identify security issues.

Requests: The requests package is in Python for making HTTP requests. It hides the difficulties of making requests behind a beautiful, simple API, allowing you to concentrate on interacting with services and consuming data in your application. Flask: Flask is a Python-based lightweight web application framework. It is very popular for constructing small to medium-sized web apps since it is easy and adaptable.

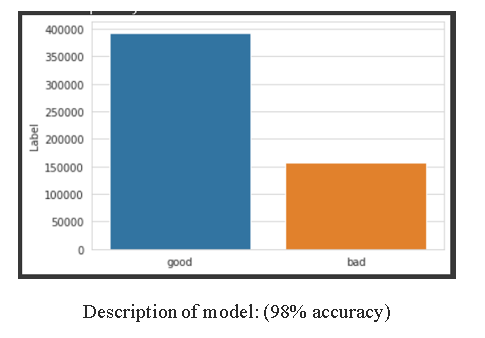
Manifest: A manifest file is a JSON file that contains data about a Chrome extension such as its name, version, icons, permissions, and other metadata. The Chrome browser uses the manifest file to determine how to install and operate the extension. JavaScript: JavaScript is a popular programming language for web development. It is used to provide interactivity and dynamic behaviour to web pages, as well as to construct a variety of web apps and utilities.

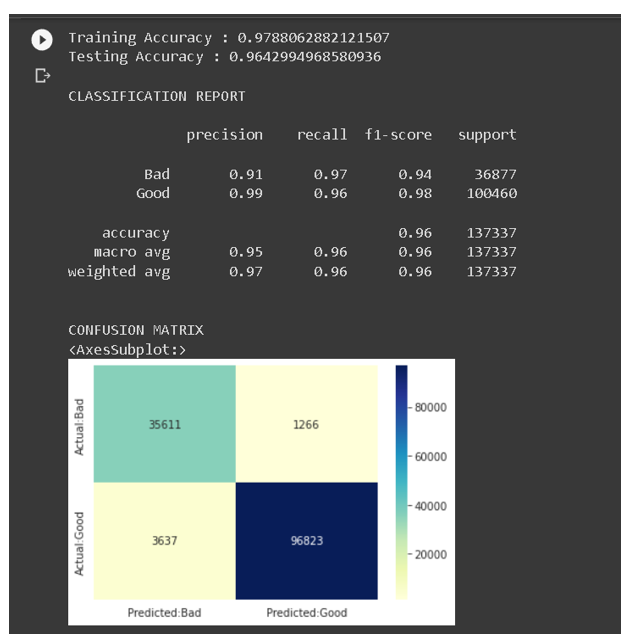
Logistic Regression: Logistic regression is a statistical method for analysing data with binary or categorical dependent variables. It predicts the likelihood of an event occurring based on one or more predictor variables.

Pickle: Pickle is a Python package for serialising and deserializing objects. It converts Python objects into a stream of bytes that can be saved in a file or transferred over a network. In Python, the pickle module can be used to save and load machine learning models and other complex data structures.

Tokenization: Tokenization is the process of converting a written document into smaller components known as tokens. Tokens are often words, but they can also be sentences, numerals, or other textual entities. Tokenization is an important step in natural language processing and machine learning because it allows algorithms to operate with individual words or phrases instead of the complete text content.

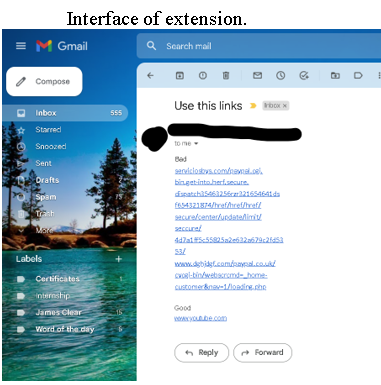
Stemming: The practice of reducing a word to its base or root form is known as stemming. For example, the words "running," "runs," and "ran" would all be reduced to the basic form "run" by stemming. Stemming is a technique used to increase the accuracy of text analysis and search results in natural language processing and information retrieval.

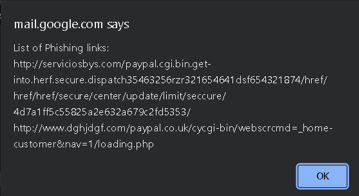
**Visualization**[](https://private-user-images.githubusercontent.com/81025229/252740263-7c028c05-bb03-4ddd-8307-736381c41164.png?jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..ZauXeAmLcnvDp0xqycgDz9qwU-2TlHM-Td4-mM_Gsxs)

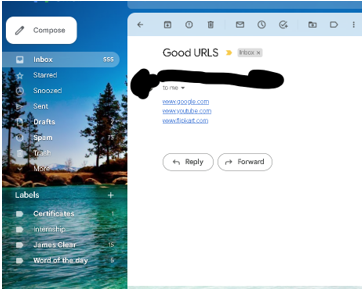
[](https://private-user-images.githubusercontent.com/81025229/252740291-8faf6596-1f44-4a7c-a31c-19ae35f7805e.png?jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..6Hfi1lwEFma5HDJCND6OqNueof-atkqdbQi4-gB0P5k)

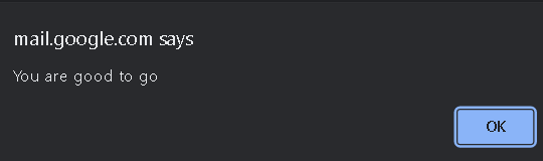
**Preview**

[](https://private-user-images.githubusercontent.com/81025229/252740480-08762e90-ce4d-48e7-8900-08594aae3930.png?jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..lIAG8Sp6eoJ2j0PvsBrXk4VtWkTcobPmoq2sr4-TpL0)

[](https://private-user-images.githubusercontent.com/81025229/252740522-eb28a5a2-2e6c-4171-a1fc-0baebf5861f1.png?jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..pOj9umvI0CiKP-hW2mWgblw4u7l2PUOMlAzPPAwSIkY)

[](https://private-user-images.githubusercontent.com/81025229/252740572-7084278a-25ca-4252-a5cb-5bd8f6a84e2f.png?jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..ZKa4Dnl2GhUxm6el_LbuCezp5jLL_iRplCaipBRktB4)

[](https://private-user-images.githubusercontent.com/81025229/252740596-96040b92-e95f-413a-8e01-819bd99b2318.png?jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..RN_4lYzQ9kCrem5tG7XBOm2e_SKiO5-c7B8n8cKdtYA)

[](https://private-user-images.githubusercontent.com/81025229/252740624-cf7dbeba-4fd6-45cf-ba08-ceb4d950af46.png?jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..r-FZgWDSEXoARtdn31Jb3JW9QpWR9Z65SDNLd31e2K8)

**Conclusion**

In conclusion, our web plugin employs a two-level design to recognise and stop phishing attempts. While the second level employs a machine learning model built on a huge dataset of URLs to analyse and detect possible phishing attempts, the first level depends on a static list of whitelisted and blacklisted URLs. The plugin gives users a strong and useful tool for keeping secure online by combining these two levels.