Project Report

Malicious and Benign Website Classification

Group-1

Abijith Prasanthan-AM.EN.U4CSE19102
Anand KS-AM.EN.U4CSE19106
Bharath Prathap Nair-AM.EN.U4CSE19113
Gautham Santhosh K-AM.EN.U4CSE19121
Rahan Manoj-AM.EN.U4CSE19144

Table of Contents

Malicious and Benign Website Classification

ABSTRACT

Today, we have millions of websites on the internet and it is important for users to be able to distinguish between potentially hazardous websites and genuine websites in order to safeguard one's personal information. This project aims to classify and detect malicious websites by analyzing various factors such as the URL of the website, IP address, the geographic location of where the website is hosted and other factors.

INTRODUCTION

The advancement in technology has opened up a new world, where all of our day-to-day activities such as banking, shopping, socialising etc happen in a virtual cyberspace. This also opens up an opportunity for more cybercrimes and hence it is imperative to have methods to protect an internet user from various threats like phishing, link spamming, redirection spamming and DNS spoofing.

Hence our project focuses on increasing internet users' security online by implementing a system that classifies websites into malicious and benign . A website is classified as malicious based on application layer and network layer characteristics .

BROAD CONTEXT - [Motivation]

Even though there are security tools used today to detect malicious websites, attackers use different methods to avoid detection by these methods. The most popular method to detect malicious websites is to keep a record of blacklisted URLs. But this method is useless when it comes to new websites being created because the list cannot keep up with the numerous number of websites being created every day.

Another method that is used for identifying malicious websites is Page Content Analysis. This is a more detailed analysis approach compared to the blacklist method. A downside to this method is the considerable amount of data that is to be collected about a particular website.

Our project aims to create a tool based on Machine learning and Data Analysis to classify a website as Malignant or Benign based on the data that is provided with lesser data and less time consumption.

DATASET AND ANALYSIS

The dataset for this project is taken from Kaggle. It contains 11 columns and over 3.5 Lakh rows. The data has been collected by crawling the Internet using MalCrawler and has been verified using google safe browsing API.

Columns:

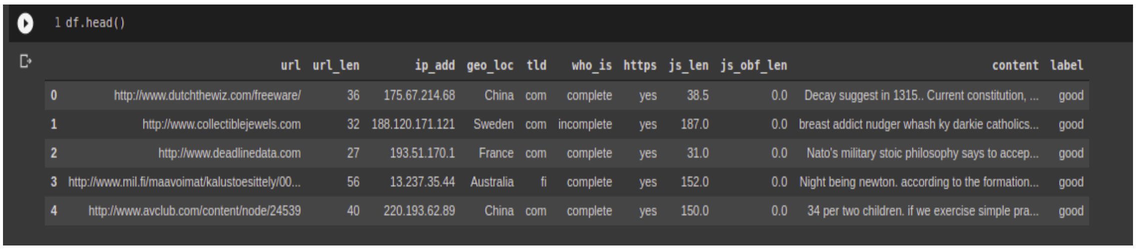
1. Index - (**Integer)**
2. url - URL of the webpage **(String)**
3. url_len - length of the URL **(Integer)**
4. ip_add - IP address of the URL **(String)**
5. geo_loc - the geographic location where the webpage is hosted **(String)**
6. tld - Top level domain of the webpage **(String)**
7. who_is - Whether the WHO IS domain information is complete or not. **(String)**

8. https - Whether the site uses https or http. **(String)**
9. js_len - Length of JavaScript code on the webpage. **(Float)**
10. js_obf_len - Length of obfuscated JavaScript code **(Float)**
11. label - class label for benign or malignant website **(String)**

METHODS

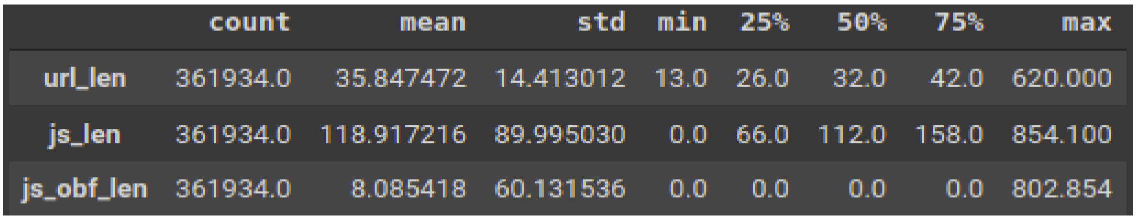
- **Dataset Summarisation**

A small subsection of the dataset containing 5 records is shown as well as a summary of data in each of the columns is shown below.

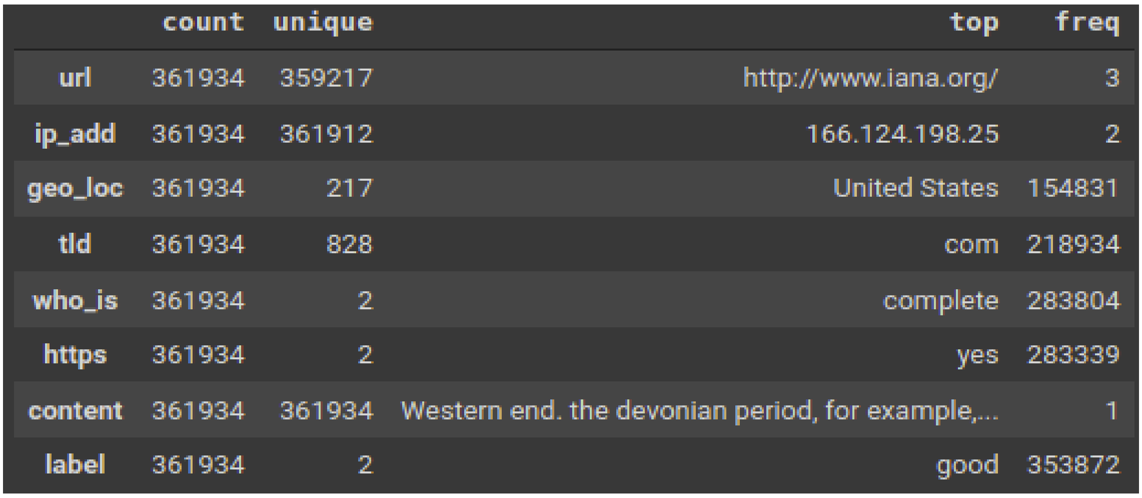


```
1 df.head()
```

	url	url_len	ip_add	geo_loc	tld	who_is	https	js_len	js_obf_len	content	label
0	http://www.dutchthewiz.com/freeware/	36	175.67.214.68	China	com	complete	yes	38.5	0.0	Decay suggest in 1315.. Current constitution, ...	good
1	http://www.collectiblejewels.com	32	188.120.171.121	Sweden	com	incomplete	yes	187.0	0.0	breast addict nudger whash ky darkie catholics...	good
2	http://www.deadlinedata.com	27	193.51.170.1	France	com	complete	yes	31.0	0.0	Nato's military stoic philosophy says to accep...	good
3	http://www.mil.fi/maavoimat/kalustoesittely/00...	56	13.237.35.44	Australia	fi	complete	yes	152.0	0.0	Night being newton. according to the formation...	good
4	http://www.avclub.com/content/node/24539	40	220.193.62.89	China	com	complete	yes	150.0	0.0	34 per two children. if we exercise simple pra...	good

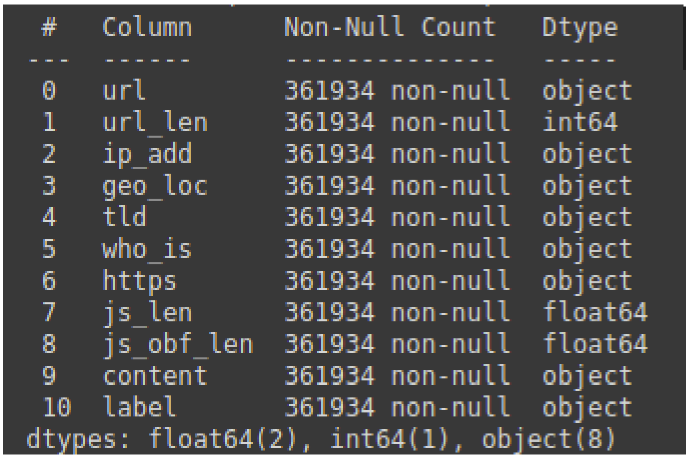


	count	mean	std	min	25%	50%	75%	max
url_len	361934.0	35.847472	14.413012	13.0	26.0	32.0	42.0	620.000
js_len	361934.0	118.917216	89.995030	0.0	66.0	112.0	158.0	854.100
js_obf_len	361934.0	8.085418	60.131536	0.0	0.0	0.0	0.0	802.854

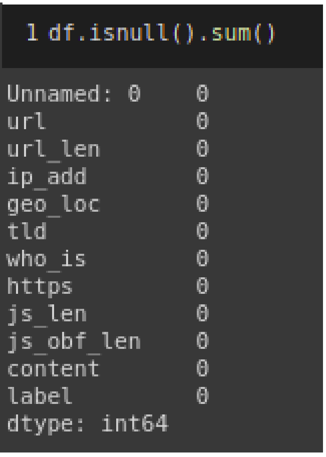


	count	unique	top	freq
url	361934	359217	http://www.iana.org/	3
ip_add	361934	361912	166.124.198.25	2
geo_loc	361934	217	United States	154831
tld	361934	828	com	218934
who_is	361934	2	complete	283804
https	361934	2	yes	283339
content	361934	361934	Western end. the devonian period, for example,...	1
label	361934	2	good	353872

We also checked whether there are any columns in the dataset with null values and it was observed that there are no null values within the dataset. This can be observed from the tables given below.



```
 #   Column      Non-Null Count   Dtype
---  ------      --------------   -----
 0   url         361934 non-null  object
 1   url_len     361934 non-null  int64
 2   ip_add      361934 non-null  object
 3   geo_loc     361934 non-null  object
 4   tld         361934 non-null  object
 5   who_is      361934 non-null  object
 6   https       361934 non-null  object
 7   js_len      361934 non-null  float64
 8   js_obf_len  361934 non-null  float64
 9   content     361934 non-null  object
 10  label       361934 non-null  object
dtypes: float64(2), int64(1), object(8)
```

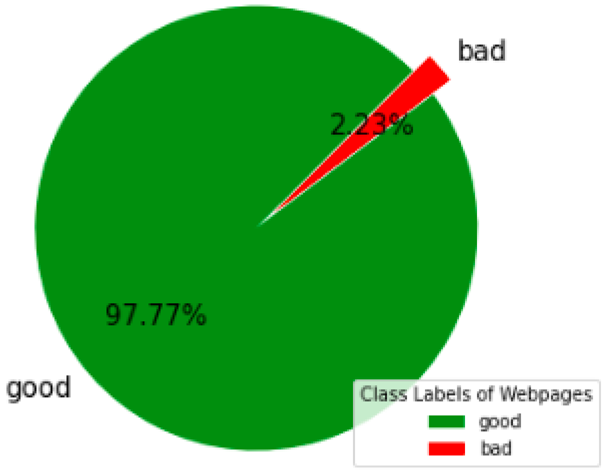


```
1 df.isnull().sum()

Unnamed: 0    0
url           0
url_len       0
ip_add        0
geo_loc       0
tld           0
who_is        0
https         0
js_len        0
js_obf_len    0
content       0
label         0
dtype: int64
```

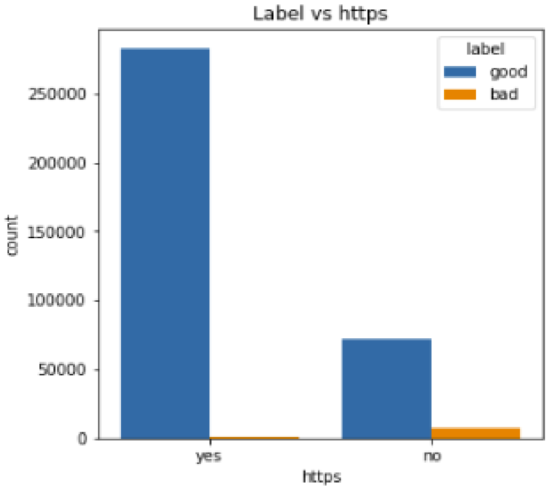
- **Data Visualisation and Pre-processing**

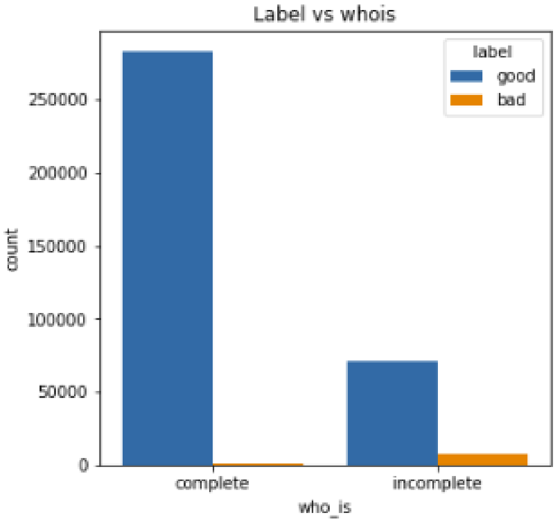
We initially compare the amount of malicious[bad] and benign[good] websites in the dataset. This is visualised as a pie-chart and shown below.



Here we notice that the proportion of good websites in the dataset is larger as compared to bad websites. This imbalance is a reflection of the real internet where the number of good websites are more in proportion to bad websites and hence it can be used for effective analysis.

It is imperative to understand the relationship between the target variable and predictor variable. Plots have been plotted to analyse the relationship of ‘who_is’ and ‘https’ variables with the target variable ‘label’.





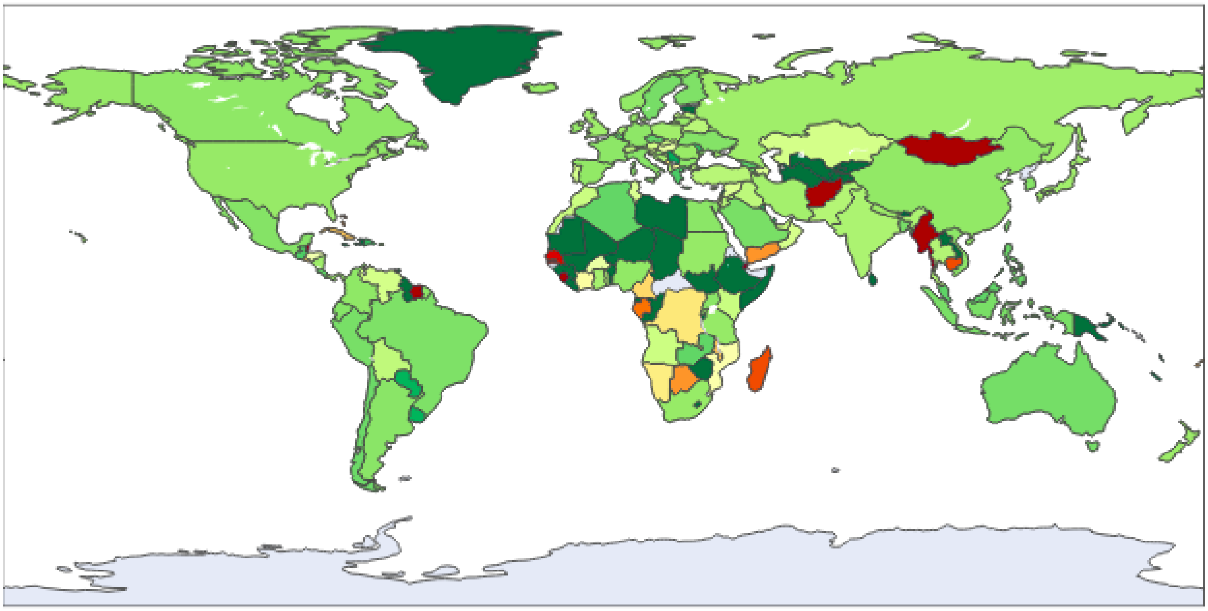
The plot shows the proportion of good and bad websites with respect to the different classes in whois and https columns. Https column implies whether the given website has a secure http or not. Whois column refers to whether the whois lookup returns the complete information regarding a website or not. Whois lookup searches the whois register for details regarding a website such as who or what entity owns or manages that domain name, including their contact information such as name, phone number and address. We see that the proportion of bad websites are more for cases where the websites are not secured using https and also, in cases where the information of who owns a malicious website[who_is] is not readily available. Hence, a website is more likely to be safe if it uses https protocol as compared to those with http and if the whois registration details are complete.

Another factor that may have an impact in the authenticity of a website is the geographical location where the site is registered. As it is not right to compare based on the number of benign or malicious websites alone, we come up with a relation that captures the relative proportion between the two types of websites in a particular country. A plot have been plotted in order to show the safety of websites from a country using the expression

$$\textit{Safety of websites in a country} = \frac{\textit{Number of benign websites - Number of malicious websites}}{\textit{Number of benign websites + Number of malicious websites}}$$

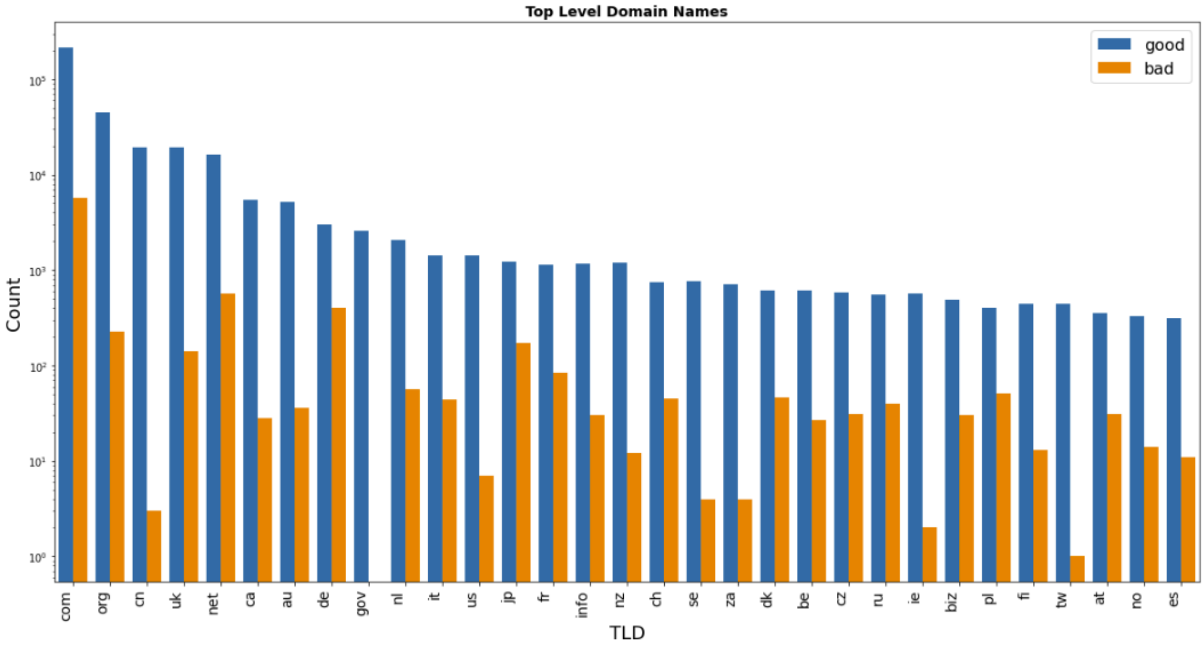
[In the above relation, it is to be noted that number of benign websites and number of malicious websites are the numbers for a particular country that is under consideration.]

The safety factor for websites from each country is calculated and plotted.

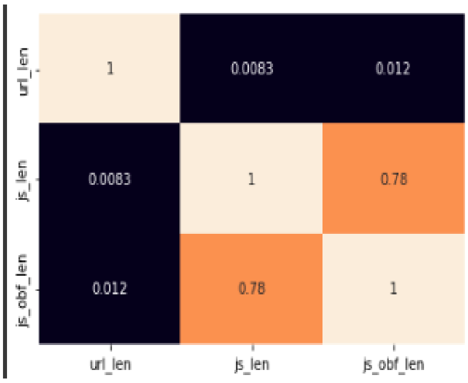


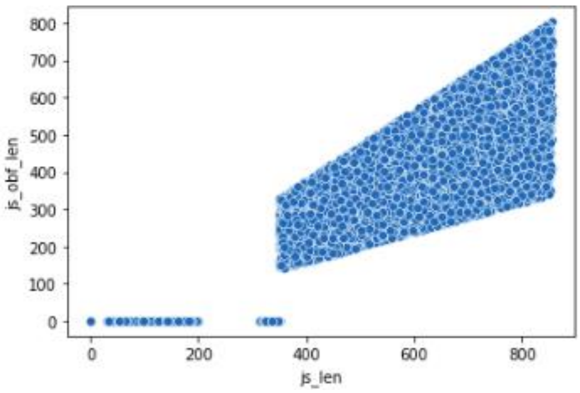
As the safety score for the websites from a country increase, it signifies that the country produces a better proportion of benign websites as compared to malicious ones. In the above graph, countries that are dark green are said to be producing the safest websites. The quality of websites decreases as the colour changes from darker shades of green to lighter shades to yellow to red. We see that Sri Lanka, Greenland, Guayana, Zimbabwe, Somalia, Ethiopia, South Sudan, Congo Republic, Guinea, Mali, Mauritiana, Niger, Chad, Libya, Uzbekistan, Tajikistan, Kazakhstan, Laos and New Guinea makes the safest websites according to the dataset with a safety score of 100%. Also it can be noted that Mongolia, Myanmar, Senegal, Afghanistan, Sierra Leone and Guinea are the countries that have comparatively lower safety scores.

A countplot for top-level-domain[tld] has been plotted. It can be observed that different domains offer different probabilities for a site to be malicious. Out of the various observations, it can be noted that the '.gov' domain used for government websites are completely safe as per the given dataset. Also it can be noted that most of the .tw, .cn and .ie websites are authentic and can be trusted by the user.



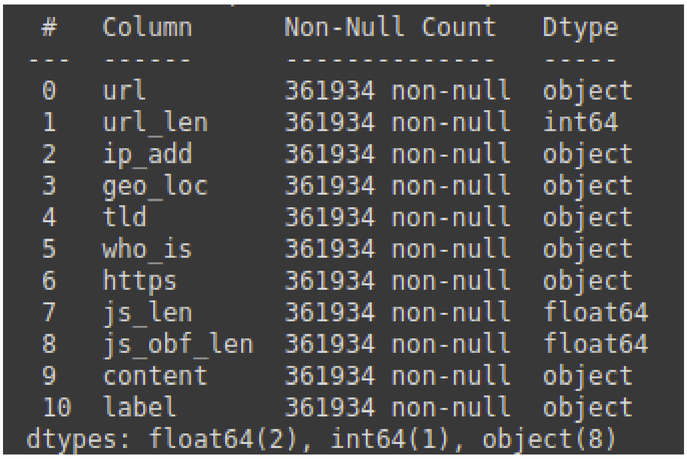
We have java script code length as well as obfuscated java script code length in our dataset. Obfuscated java script code is made from performing a series of transformations on the java script code and it is converted into a form that is hard to understand and reverse engineered. This is done in order to hide the logic of the code from the user as well as to reduce the size of the java script code. Since it is the java script code that is being obfuscated, we find that the columns that give the information regarding both of these parameters are correlated. This can be observed from the correlation plot as well as the scatterplot shown below.





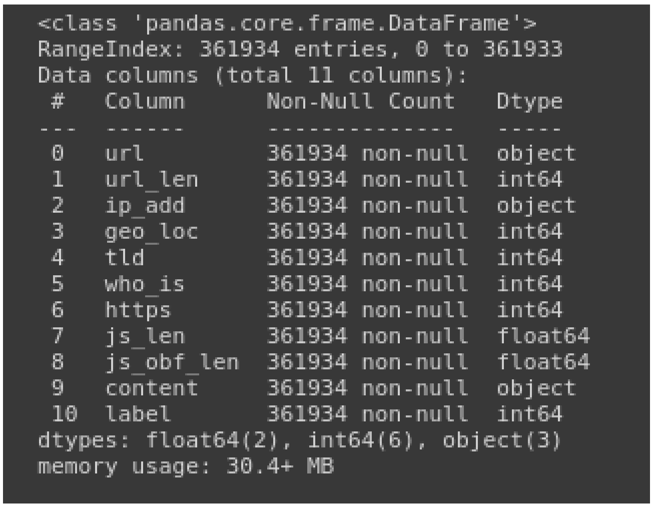
We can observe from the plot that there is a positive correlation of 0.78 between 'js_len' and 'js_obf_len'. Also it can be seen from the scatter plot that the two parameters vary uniformly with each other. Hence for evaluation, it is necessary to use only one of these features. Hence the column js_obf_len is dropped.

Our dataset contains various columns where the data is of type object. This is not suitable for prediction using machine learning algorithms. Therefore, we need to convert these into numerical values.



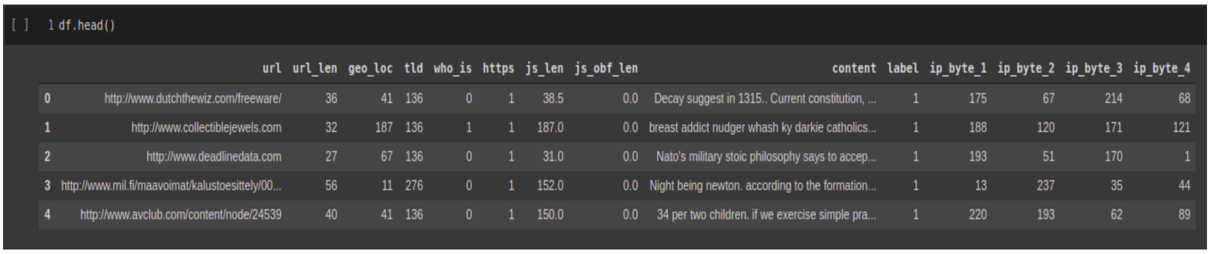
```
 #   Column      Non-Null Count   Dtype
---  ------      --------------   -----
 0   url         361934 non-null  object
 1   url_len     361934 non-null  int64
 2   ip_add      361934 non-null  object
 3   geo_loc     361934 non-null  object
 4   tld         361934 non-null  object
 5   who_is      361934 non-null  object
 6   https       361934 non-null  object
 7   js_len      361934 non-null  float64
 8   js_obf_len  361934 non-null  float64
 9   content     361934 non-null  object
 10  label       361934 non-null  object
dtypes: float64(2), int64(1), object(8)
```

The columns who_is, https, label, geo_loc and tld contain various categorical values of type object. We map each of these to numerical values using label encoder. After this operation, the dataset and its datatypes are provided below



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 361934 entries, 0 to 361933
Data columns (total 11 columns):
 #   Column      Non-Null Count   Dtype
---  ------      --------------   -----
 0   url         361934 non-null  object
 1   url_len     361934 non-null  int64
 2   ip_add      361934 non-null  object
 3   geo_loc     361934 non-null  int64
 4   tld         361934 non-null  int64
 5   who_is      361934 non-null  int64
 6   https       361934 non-null  int64
 7   js_len      361934 non-null  float64
 8   js_obf_len  361934 non-null  float64
 9   content     361934 non-null  object
 10  label       361934 non-null  int64
dtypes: float64(2), int64(6), object(3)
memory usage: 30.4+ MB
```

Now we also observe that the IP-address is also an object. We have IPv4 addresses. We can extract a lot of information from an IP-address, such as the class of the IP-address. We split the ip-address into 4 where each part is a byte of the IP address and these splitted values are converted to numeric datatype and appended back into the dataframe as separate columns for further analysis.



```
df.head()
```

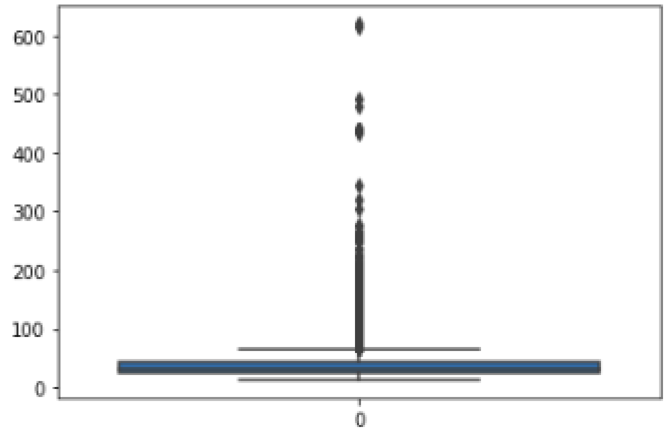
	url	url_len	geo_loc	tld	who_is	https	js_len	js_obf_len	content	label	ip_byte_1	ip_byte_2	ip_byte_3	ip_byte_4
0	http://www.dutchthewiz.com/freeware/	36	41	136	0	1	38.5	0.0	Decay suggest in 1315.. Current constitution, ...	1	175	67	214	68
1	http://www.collectiblejewels.com	32	187	136	1	1	187.0	0.0	breast addict nudger whash ky darkie catholics...	1	188	120	171	121
2	http://www.deadlinedata.com	27	67	136	0	1	31.0	0.0	Nato's military stoic philosophy says to accep...	1	193	51	170	1
3	http://www.mil.fi/maavoimat/kalustoesittely/00...	56	11	276	0	1	152.0	0.0	Night being newton. according to the formation...	1	13	237	35	44
4	http://www.avclub.com/content/node/24539	40	41	136	0	1	150.0	0.0	34 per two children. if we exercise simple pra...	1	220	193	62	89

We now observe that we have two columns, namely url and content whose data type is object. It is imperative that we convert these to numerical values for prediction.

For proper analysis of the content, we first **remove the stopwords** from the content. Stopwords are commonly used words in English such as “is”, “and” etc which do not contribute to the sentiment or validity of the content in a website and hence can be ignored. This was followed by **stemming** the data in the content column. Stemming is the process wherein we reduce words into their root form. In order for the content to be subjected to various classification algorithms, it is required to transform the text data into numerical data. For this, we apply the technique of count vectorization to the data. **Count vectorization** is the process of transforming a text into a vector on the basis of the frequency of each word in the text. We have set the max_feature parameter during count vectorization as 1000. This implies that we are taking the most significant/most occurring 1000 words from the dataset for the purpose of analysis. After this operation, we will have 1000 more columns or parameters for consideration while prediction.

Similarly, the above method of count vectorization has also been applied on URL. This was done after we split the url into individual words after removing the dots[.] and slashes [/] in the URL. Various parameters such as top-level domain [.com, .org etc] as well as words like http, https and www are removed from the url. These are not necessary here as we already have separate columns for http and tld in the dataset.

A box plot is plotted for url_len and it can be observed that there exists some outliers.



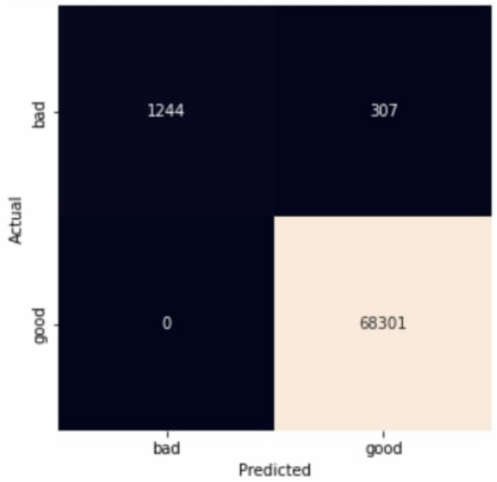
The IQR [Inter quartile range] is calculated and values less than minimum and greater than maximum are dropped, where maximum = third quartile + 1.5 * IQR and minimum = first quartile - 1.5 * IQR.

● CLASSIFICATION

We subject the data to various classification algorithms to determine which is the best algorithm that helps in classifying the website with maximum accuracy. The classification algorithms used are described below.

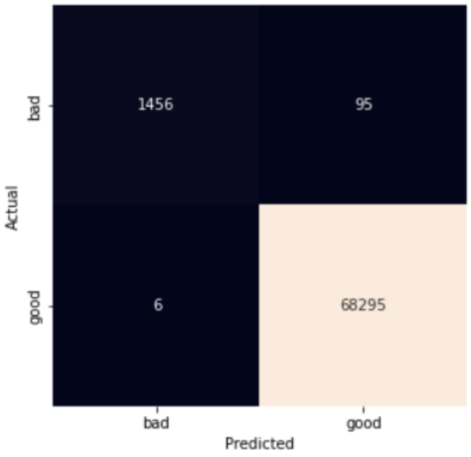
● KNN algorithm

K-nearest neighbours (KNN) is a type of supervised learning algorithm used for both regression and classification. KNN tries to predict the correct class for the test data by calculating the distance between the test data and all the training points. Then select the K number of points which is closest to the test data. There are many methods to measure the distance. Euclidean distance (minkowski distance with p=2) is one of most commonly used distance measurements. KNN classifier determines the class of a data point by majority voting principle. Among these k neighbors, count the number of the data points in each category. Assign the test data to that category for which the number of the neighbours is maximum.
The accuracy using KNN was **99.55%**. The confusion matrix is shown below.



- **Logistic Regression**

Logistic regression is a regression based algorithm for classification purposes. In this method a hypothesis is made using feature variables in such a way that we obtain parameters that give a minimum value for a particular cost function. This hypothesis is then subjected to a data point to get a value which is then subjected to sigmoid function and based on the value obtained, we classify it into different classes.
The accuracy obtained was **99.85%**. The confusion matrix is shown below.



- **Naive Bayes Classifier**

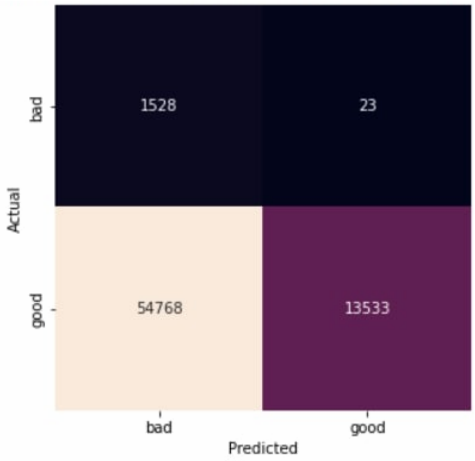
Naive Bayes algorithm is a supervised learning algorithm, which is based on Bayes theorem and used for solving classification problems. Naive Bayes Classifier is one of the simple and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions.

It is a probabilistic classifier, which means it predicts on the basis of the probability of an object. Bayes' theorem is also known as Bayes' Rule or Bayes' law, which is used to determine the probability of a hypothesis with prior knowledge. It depends on the conditional probability.

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Where,
P(A|B) is Posterior probability: Probability of hypothesis A on the observed event B.
P(B|A) is Likelihood probability: Probability of the evidence given that the probability of a hypothesis is true.
P(A) is Prior Probability: Probability of hypothesis before observing the evidence.
P(B) is Marginal Probability: Probability of Evidence.

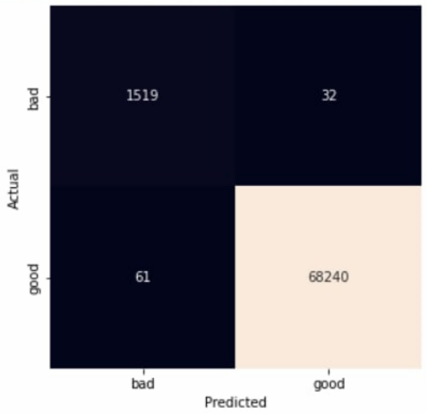
The accuracy obtained is **21.18%.** The confusion matrix for the observation is shown below.



Actual \ Predicted	bad	good
bad	1528	23
good	54768	13533

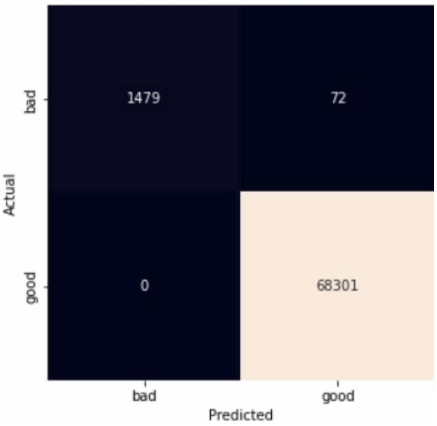
- **Decision Tree**

It is a non-parametric method used for classification . The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. A decision tree is a tree where each node represents a feature(attribute), each link(branch) represents a decision(rule) and each leaf represents an outcome(categorical or continuous value). It progressively divides data sets into smaller data groups based on a descriptive feature, until they reach sets that are small enough to be described by some label. The accuracy obtained is **99.87%** and the confusion matrix shown below**.**



- **Random Forest Classifier**

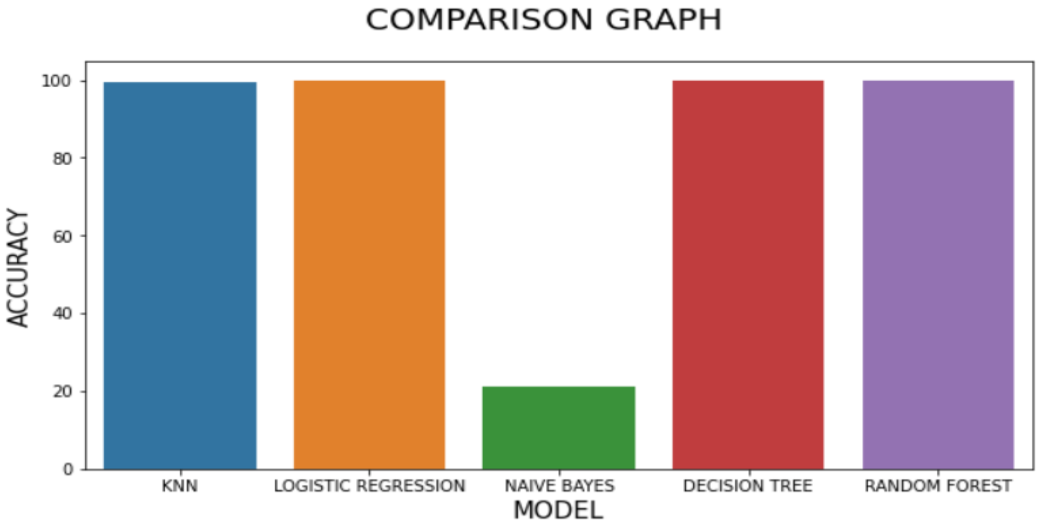
The Random Forest Algorithm is composed of different decision trees, each with the same nodes, but using different data that leads to different leaves. It merges the decisions of multiple decision trees in order to find an answer, which represents the average of all these decision trees which in turn results in more accurate predictions.
The accuracy obtained was **99.90%**. The confusion matrix is shown below.



RESULTS:

The accuracy obtained by using various algorithms are analysed and tabulated in the table below.

Algorithm Used	Accuracy obtained
K Nearest Neighbour Algorithm	99.55%
Logistic Regression	99.85%
Naive Bayes Classifier	21.18%
Decision Tree	99.87%
Random Forest Classifier	99.90%



DISCUSSIONS:

With technology developing day by day, thousands of websites are created on the internet day to day and hackers evolving with newer techniques, makes cyber security a serious threat for web users. Therefore, various studies have been performed to find out malicious and benign websites.

Despite the advancement of artificial intelligence and machine learning, it is still a big challenge to ensure cyber security over the internet. Some of the reasons for this are:

- ➢ Huge amount of data on the internet: In 2012, it was found that above 30 trillion unique URLs were found by Google's search engine. Due to the same reason, it's very hard to train a malicious URL detection machine learning model.

- ➢ Malicious content even under HTTPS protocol: Websites that use 'https' protocol are believed to be very safe as https protocol encrypts the data that is transferred and received over the website. Despite this fact, from our study we can observe that there are few malicious websites which use https protocol .

- ➢ Authenticity of the data collected : Even if we manage to collect data despite its huge amount, the data we collect may not always be true. For example, in our dataset we have a feature called 'geo_loc' which gives the geographic location where the webpage is hosted. But one can easily manipulate this geographic location using a VPN , thereby showing a completely different geographic location which may even be from a different continent.

- ➢ Short span of malicious websites: In the real world all URLs may not always be alive. Many malicious URLs may be short-lived and the same content will be replicated in another URL hence accessing its feature will turn out to be impenetrable. Due to this flagging a website as malicious and maintaining a list of such websites would be pointless.

All these challenges pose a lot of research and development difficulties for collecting features especially for constructing training datasets.

LITERATURE CITED:

1. Dataset of Malicious and Benign Webpages:

   https://www.kaggle.com/aksingh2411/dataset-of-malicious-and-benign-webpages

2. Malicious and Benign Webpages Dataset, A.K. Singh

   https://doi.org/10.1016/j.dib.2020.106304

3. Detecting Malicious Websites Using Machine Learning , Saeed Ahmad Al Tamimi, RIT Dubai, April 20, 2020

https://scholarworks.rit.edu/cgi/viewcontent.cgi?article=11869&context=theses

4. Malicious URL Detection using Machine Learning: A Survey , DOYEN SAHOO, CHENGHAO LIU, STEVEN C.H. HOI, School of Information Systems, Singapore Management University ,arXiv:1701.07179v3 [cs.LG] 21 Aug 2019

   https://arxiv.org/pdf/1701.07179.pdf

CONCLUSION:

We have analyzed the dependence of various features of a website and how it can be used to determine the authenticity of a website. Different machine learning algorithms were applied in order to come up with the algorithm that classifies the website as benign or malignant based on the factors under consideration.

Based on the accuracy score observations, we can observe that Random Forest and Decision tree algorithms make better predictions with an accuracy score of **99.90%** and **99.87%** respectively. KNN and Logistic Regression also performs considerably well with accuracies **99.55%** and **99.85%.**