Machine Learning Assignment
Supervised Learning

Marc Karp (1356562), Shaylin Pillay (1060478), Preshen Goobiah (1355880)

Introduction

Phishing is a cybercrime in which a target is contacted by someone posing as a legitimate institution to lure individuals into providing sensitive data such as personally identifiable information, banking and credit card details, and passwords (What-is-phishing, 2018). Phishing can result in huge personal, institutional and societal loss and hence there is a need for an efficient mechanism to predict if a website is of a phishing nature or not. This report employs supervised learning algorithms, namely Naïve Bayes Classification and Decision Tree ID3 which was coded in python, to model this task and make predictions.

Dataset

The dataset was retrieved from the UCI Machine Learning repository (Phishing Websites Data Set, 2018) and provides useful attributes (see appendix) with a finite amount of distinct discrete values (usually 2 or 3) that can be used the classify each entry.

This dataset was chosen for the following reasons:

- It has many features (30 in total)
- There are many instances (2456 row entries)
- Its relatively current (Donated 26/3/2015, which makes our results valuable and interesting)

Target: Whether it is a phishing website or not (a legitimate website).

Some sample data points and descriptions of all the features are given in the Appendix.

Data Manipulation

Pre-processing

Some entries from the dataset had missing attributes and additional attributes, since these instances could not be interpreted. It was decided to remove these instances from the dataset using a conditional if statement, ensuring that all data used was complete and not incorrectly skewed.

The dataset contained 720 duplicate instances (29.31% of dataset). It was chosen to retain these instances as there may be websites that have the exhibit the exact characteristics in the real world.

The dataset is fairly balanced as 44.54% (1094) instances are Legitimate websites and 55.46% (1362) are Phishing websites.

Structuring

The dataset was read from a text file, into a list which contained each instance.

Normalisation

No normalization was needed as the range of values that the attributes could take on is between 2 and 3.

Splitting of Data

The list of all data was split into training, validation and test data using the *train_validate_test* function.

- Training - 70% (1720 instances)
- Validation - 10% (245 instances)
- Test - 20% (491 instances)

Implementation Details

Decision Tree

Decision trees are useful when working with discrete data with multiple attributes when a distinct target attribute is known. This method was chosen to classify websites as phishing or legitimate since we have 30 features with discrete values and a binary-valued target feature. The ID3 algorithm is used to generate the decision tree from our dataset, since our values are discrete it makes the process less computationally expensive.

Data structure

A Node class was created to act as a decision node. This class contains the following attributes:

- subset - contains the subset of the data that was filtered from its parent's attribute value.
- parent - the parent to this node.
- list of children - all child nodes that are created from this node.
- column_num - feature column number that it is associated with.
- name - feature name that it is associated with.
- subset_value - value from which it was split on (see Figure 1)

This node class was instantiated when building the tree. Each instance of the node represented a single decision point or leaf e.g. root node - SSL Final State (see Figure 1)

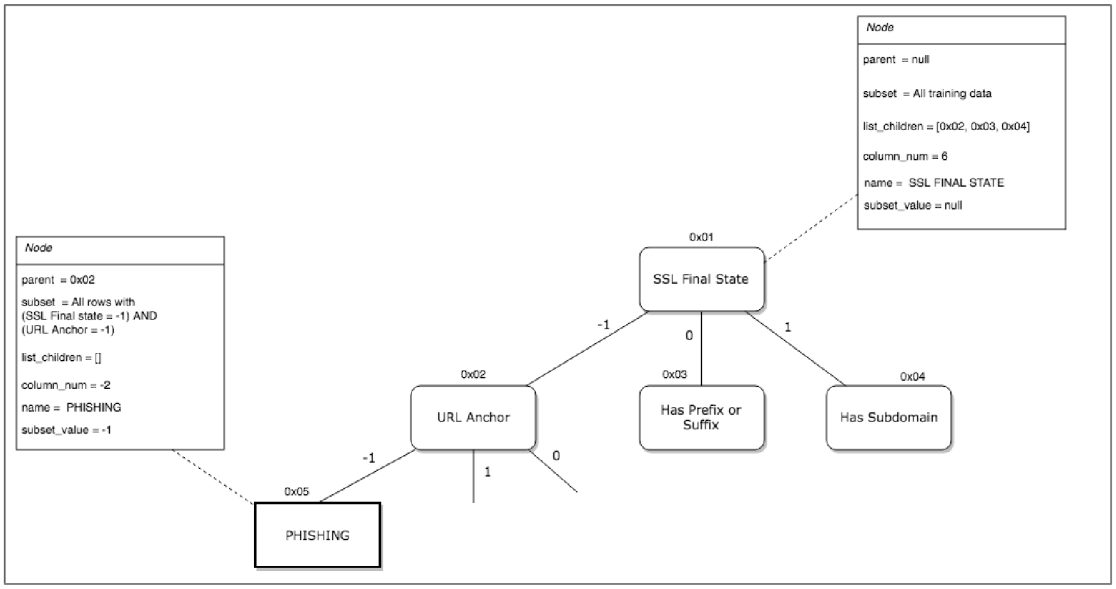

Figure 1. Diagram of Partial Decision Tree in relation to implementation

Building of the tree

The *build_tree* function calls *create_node* which in turn recursively creates nodes in the tree. The recursion is necessary for the tree to be built and it builds in a depth first style. As a node is created, the best feature to act as a decision point is determined. This is done by calculating the information gain on an attribute using the *best_attribute* function. For all values in the domain of the "best feature", the data that was used to identify the best feature is split into subsets. For each value in the domain of the "best feature" a child is created containing the subset, mentioned above. These children are then assigned a best attribute to split on by calling the *create node* function and the process recursively continues.

Design Decisions

The recursion was stopped based on two conditions:

1. If a node's subset did not have instances containing each of the distinct values in the domain of the best attribute found for that node, it is converted into a leaf node by using the *calc_majority* function, in which its name can be "PHISHING" or "LEGITIMATE"
2. If a child contained a subset that is pure (defined by an *entropy threshold*) by using the *pure_subset* function, it is converted into a leaf node by using the *calc_majority* function, in which its name can be "PHISHING" or "LEGITIMATE"

Hyper-parameters

The hyper-parameter, *entropy threshold,* serves as a stopping criterion for the recursion as mentioned above. After building multiple decision trees with entropy thresholds between 0 and 1, where 0 is complete certainty and 1 is complete uncertainty within the subset of data, the accuracy of the predictions on the training and validation set was graphed using the *matplotlib.pyplot* library in order to find the optimal *entropy threshold*. The validation set achieved its highest accuracy when the *entropy threshold* value was between 0.13 and 0.51 (see figure 2) therefore, the value of *entropy threshold* was set to 0.13. This was value was chosen as the trees built with the entropy threshold in the range of [0.13,0.51] will not over fit the training data or generalise too much.

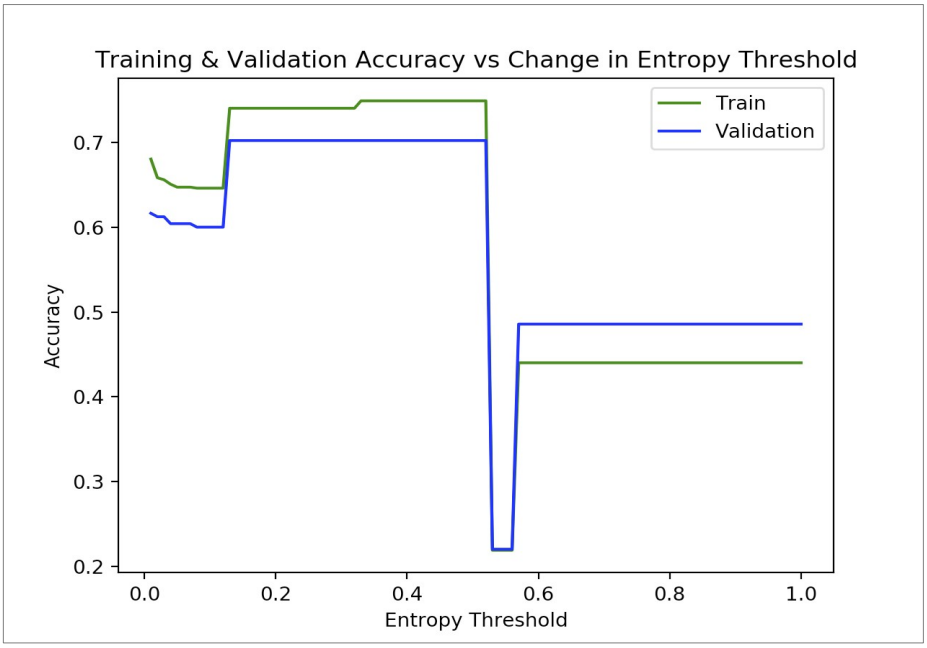

Figure 2. Graph of Training & Validation Accuracy vs Change in Entropy Threshold

Naïve Bayes Classification

The Naive Bayes Classifier technique is based on Bayesian theory and is particularly suited when the dimensionality of the inputs is high. (Naive_Bayes_Classifier, 2018) This technique was chosen largely since the dataset contains 30 features.

Data Structure

An Attribute class was created which serves as a container for a feature and details that are needed for the Naive Bayes classification, which are explained below.

This class contains the following attributes:

- column_number - feature column number that it is associated with.
- distinct_values - all distinct values that this feature can have.
- list_probs - a list of probabilities for legitimate and phishing websites for each distinct value of the domain of a feature i.e. P(feature value | legit) and P(feature value | phishing) (see figure 3)

Each instantiated object of this class was used to store all the probabilities associated with each value of the domain of the feature.

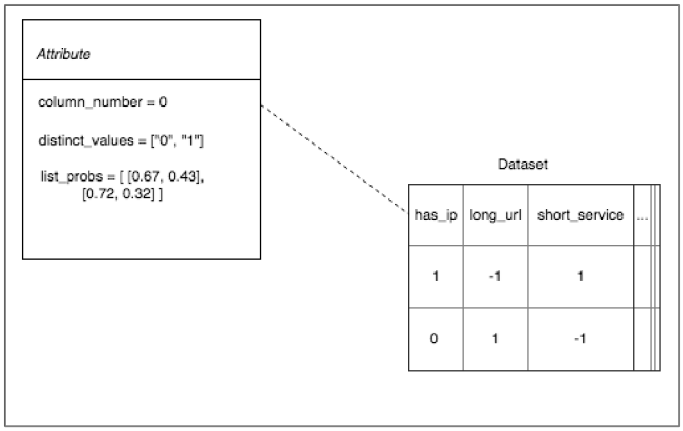

Figure 3. Diagram of Attribute class implementation

Building the classifier

Initially, 30 Attribute objects are created using the *setup_attributes* function. The *learn_probs_priors* function is then called to calculate the prior probabilities and populate the conditional probabilities of each of the distinct values in the domain of a specific feature (Attribute). An instance of the dataset is then parsed into the *naive_bayes_predictor* function and the class which it is most likely to be in, by the MAP solution, is output.

Hyper-parameters

Laplacian smoothing was used to account for cases when a feature with a specific value was not seen during training. After changing the values of the Laplacian smoothing constant, *laplacian_add*, the accuracy of the predictions on the training and validation set was graphed using the *matplotlib.pyplot* library

It was noted that change in Laplacian smoothing value did not affect the accuracy of the model (see figure 4). Thus, the *laplacian_add* value was set to 1.

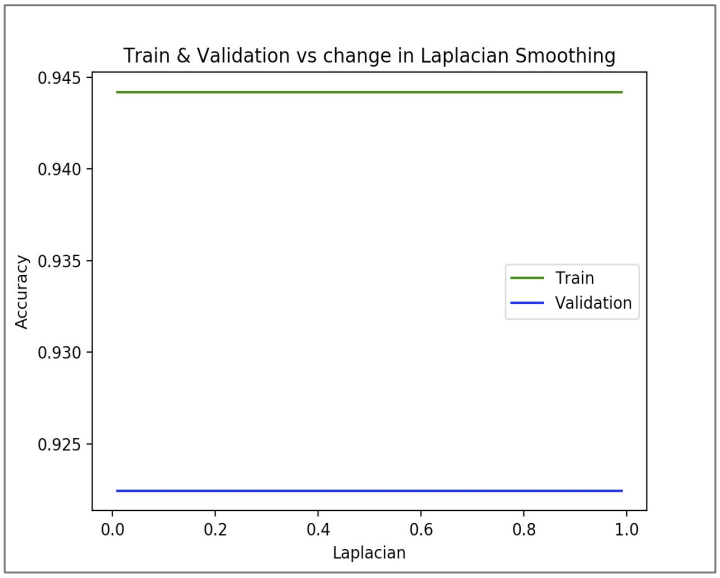

Figure 4. Graph of Training & Validation Accuracy vs Change in Laplacian Smoothing Constant

Results

Repeated Accuracy

The following table depicts the accuracy after randomising the training, validation and testing set 3 times. This was done to decrease the likelihood of the training set being unbalanced.

	Shuffle 1	Shuffle 2	Shuffle 3	Average Accuracy
Decision Tree Accuracy (%)	74.54	61.10	59.67	65.10
Naive Bayes Accuracy (%)	93.89	93.28	93.69	93.62

Note: All results here forth are using dataset on Shuffle 1

Decision Tree

Confusion Matrix

	Actual Class	
Predicted Class	Legit	Phishing
Legit	100	7
Phishing	118	266

Evaluation Metrics

Accuracy	Classification Error	Precision (Legit)	Precision (Phishing)	False Negative Rate	True Positive Rate
74.54%	25.45%	93.46%	30.73%	54.13%	45.87%

It was seen that there are much more decision points in tree to classify a website as legitimate as opposed to phishing.
According to the decision tree a phishing website can be identified by a few features having a certain value. Since some legitimate websites may exhibit these traits, the tree prematurely classifies websites as phishing therefore the False Negative rate is higher than the True Positive rate and the Precision of classifying phishing websites is relatively low. Since there were more decisions made to classify a legitimate website, the Precision is relatively high.

Naive Bayes Classifier

Confusion Matrix

	Actual Class	
Predicted Class	Legit	Phishing
Legit	202	14
Phishing	16	259

Evaluation Metrics

Accuracy	Classification Error	Precision (Legit)	Precision (Phishing)
93.89%	6.11%	93.42%	94.18%

As Naïve Bayes uses probabilistic statistics it is not as susceptible to over fitting to the training data as seen with the Decision Tree therefore the Classification error is relatively low. The Precision on Legit and Phishing are equally high compared to that of the Decision Tree, since the probabilistic output allows us to differentiate better between a phishing and legitimate website.

Possible Improvements

A random forest classifier can be used instead of a single decision tree. This will most likely increase the accuracy of the predictions.

The initial dataset with missing and additional features can be used, by replacing the missing features with features that occur frequently in certain classes and by removing the additional features. This will contribute to better accuracy of the classifiers as there will be more data to train the models on.

Conclusion

The Naïve Bayes classifier had the highest average accuracy, as well as precision for this dataset and was less complicated to implement. Implementing the decision tree was challenging as there were many implementation specific considerations, this model did not generalise as well as the Naïve Bayes model. This can be attributed to Naïve Bayes using probability rather than discriminatory questions to classify.

References

Naive_Bayes_Classifier. (2018, May 20). Retrieved from Statsoft: http://www.statsoft.com/textbook/naivebayes-classifier

What-is-phishing. (2018, April 19). Retrieved from Phishing.org: http://www.phishing.org/what-is-phishing

Phishing Websites Data Set. (2018, May 19). Retrieved from UCI Machine Learning Repository: https://archive.ics.uci.edu/ml/datasets/phishing+websites

Appendix

Feature Description

Adapted from the feature description on the UCI Repository (Phishing Websites Data Set, 2018)

Features:

Feature	Domain Values	Description
Having IP Address	{ 1,0 }	Most phishing websites contain IP address as their domain name. If the domain name in the page address is an IP Address then a value of 0 is given, otherwise the value 1 is given.
Having long URL	{ 1,0,-1 }	An abnormally long URL filled with multiple characters is a sign of a less secure website since phishers can easily hide suspicious wording. Short URLs are represented by -1, medium length by 0 and long by 1.
Uses Shortening Service	{ 0,1 }	URL shorteners are a great tool to share a web address without a lot of typing and hence are often used by phishers. The use of a URL shortener is indicated by 1 and the absence by 0.
Having '@' Symbol	{ 0,1 }	Presence of @ symbol in page address indicates that, all text before @ is comment. Thus, the page URL should not contain @ symbol. If the page URL contains @ symbol, a value of 0 is given. Otherwise a value of 1 is given.
Double slash redirecting	{ 0,1 }	The page address and URL should not contain more number of slashes. If they contain more than five slashes then the url is considered to be a phishing url and a value of 0 is given. If the page address contains less than 5 slashes, a value of 1 is given.
Having Prefix Suffix	{ -1,0,1 }	Phishers tend to add prefixes or suffixes separated by (-) to the domain name so that users feel that they are dealing with a legitimate website. If the URL contains no prefix, suffix or dash then a value of -1 is given. If it contains 1 or 2 of the above, then a value of 0 is given and if it contains all 3 then a value of 1 is given.
Having Sub-Domain	{ -1,0,1 }	If the number of “dots” after (www.) is greater than one, then the URL is given a value of 0 since it has one sub domain. However, if the dots are greater than two, it is given a value of 1 since it will

		have multiple sub domains. Otherwise, if the URL has no subdomains, a value of -1 is given.
SSL final State	{ -1,1,0 }	SSL creates an encrypted connection between the web server and the user's web browser allowing for private information to be transmitted without the problems of eavesdropping. All legitimate websites will have SSL certificate. But phishing websites do not have SSL certificate. If SSL certificate exists and is verified a value of 1 is given. If there is no SSL certificate, then a value -1 is given. If the SSL certificate exists but is not verified, then a value of 0 is given.
Domain registration length	{ 0,1,-1 }	A phishing website lives for a short period of time and it is believe that legitimate websites have their domains payed for years in advance. Hence, if the domain expires in under a year a value of -1 is given. If it expires in 1-2 years, a value of 0 is given and if it expires in over 2 years a value of 1 is given.
Favicon	{ 0,1 }	Graphical browsers show favicon as a visual reminder of the website identity in the address bar. If the favicon is loaded from a domain other than that shown in the address bar, then the webpage is likely to be considered a Phishing attempt. Hence, if the favicon is loaded from an external domain a value of 0 is given. Otherwise, a value of 1 is given.
Is standard Port	{ 0,1 }	Several firewalls will, by default, block all or most of the ports and only open the ones selected. Hence, if a non-standard port is used a value of 0 is given. Otherwise, a value of 1 is given.
Uses HTTPS token	{ 0,1 }	The phishers may add the "HTTPS" token to the domain part of a URL in order to trick users. Hence, if the token is used a value of 0 is given and if not, a value of 1 is given.
Request_URL	{ 1,-1 }	Websites request images, scripts, CSS files from other websites. Phishing websites to imitate the legitimate website request these objects from the same page as legitimate one. <img> and <script>, background attribute of body tag, href attribute of link tag and code base attribute of object and applet tag. If the domain in these URLs is foreign domain then the value is –1 else the value is 1.
Abnormal URL anchor	{ -1,0,1 }	An anchor tag contains href attribute whose value is an url to which the page is linked with. If the domain name in the url is not similar to the domain in page URL then it is called as foreign anchor. A website can contain foreign anchor. But too many

		foreign anchor is a sign of phishing website. So all the tags in the webpage are collected. And they are checked for foreign anchor. If the number of foreign domain exceeds, then a value of -1 is given. If it's equal, a value of 0 is given and if it less a value of 1 is given.
Links in tags	{ 1,-1,0 }	Legitimate websites often use <Meta> tags to offer metadata about the HTML document; <Script> tags to create a client-side script; and <Link> tags to retrieve other web resources. If no tags are used, a value of -1 is given. If only 1 or 2 tags are used, a value of 0 is given and if all 3 tags are used a value of 1 is given.
SFH	{ -1,1 }	Forms are used to pass data to a server. Even though some legitimate websites use third party services and hence contain foreign domain, it is not the case for all the websites. If the feature has its own domain name a value of 1 is given. Otherwise a value of -1 is given.
Submitting to email	{ 1,0 }	A phisher might redirect the user's information to his personal email. Hence, a server-side script language might be used such as "mail()" function in PHP. Thus if mail() is used a value of 0 is given. Otherwise, a value of 1 is given.
Abnormal URL	{ 1,0 }	If the hostname is included in the URL a value of 1 is given. Otherwise, a value of 0 is given.
Redirect	{ 0,1 }	If the website has been redirected more than 1 time, a value of 0 is given. Otherwise, a value of 1 is given.
on mouseover	{ 0,1 }	Phishers may use JavaScript to show a fake URL in the status bar to users. If the mouseover feature is removed, then the URL may change. This is a sign of phishing and hence if the URL changed, a value of 0 is given. Otherwise a value of 1 is given.
Right Click	{ 0,1 }	Phishers use JavaScript to disable the right-click function, so that users cannot view and save the webpage source code. Hence, if this function is removed a value of 0 is given. Otherwise a value of 1 is given.
popUp Window	{ 0,1 }	It is unusual to find a legitimate website asking users to submit their personal information through a pop-up window. Hence, if this is required a value of 0 is given. Otherwise a value of 1 is given.
Iframe	{ 0,1 }	IFrame is an HTML tag used to display an additional webpage into one that is currently shown. Phishers can make use of the "iframe" tag and make it invisible. Hence, if Iframe is used a value of 0 is given. Otherwise a value of 1 is given.

Age of domain	{ -1,0,1 }	Most phishing websites live for a short period of time. Hence, if the domain age is younger than 6 months, a value of 01 is given. If it is older than 6 months but under a year a value of 0 is given. Otherwise, a value of 1 is given.
DNS Record	{ 1,0 }	If the DNS record is empty or not found on WHOIS then the website is given a value of 0. Otherwise a value of 1 is given.
Web traffic	{ -1,0,1 }	If the website ranks less than 100 000 on the Alexa database, a value of 1 is given. If it is ranked but above 100 000 then a value of 0 is given and if it is unranked then a value of -1 is given.
Page Rank	{ -1,0,1 }	PageRank aims to measure how important a webpage is on the Internet. The greater the PageRank value the more important the webpage. If the PageRank of the URL is < 0.2, a value of -1 is given. If it is > 0.8, a value of 1 is given. Otherwise a value of 0 is given.
Google Index	{ 0,1 }	When a site is indexed by Google, it is displayed on search results. Phishing websites are usually short term and hence not indexed by Google. Thus, a value of 0 is given if the site is indexed and 1 if it is.
Links pointing to page	{ 1,0,-1 }	The number of links pointing to the webpage indicates its legitimacy level. Thus, if there are no links present a value of -1 is given. If there are between 0 and 2 links, a value of 0 is given. Otherwise, a value of 1 is given.
Statistical report	{ 1,0 }	Phishing sites are regularly founded and added to an online blacklist. Thus, if the URL is on one or more of these lists a value of 0 is given. Otherwise, a value of 1 is given.
Result	{ 1,-1 }	A value of -1 is given if the website is of a phishing nature and 1 is given if the website is legitimate.

Sample data points

Attribute	Entry 1	Entry 2	Entry 2	Entry 4
has_ip	0	0	0	1
long_url	-1	-1	1	-1
short_service	0	0	0	1
has_at	0	0	0	0
double_slash_redirect	0	0	0	1
pref_suf	0	0	-1	0

has_sub_domain	0	-1	-1	0
ssl_state	1	-1	-1	1
long_domain	0	0	0	0
favicon	0	1	0	0
port	0	1	0	0
https_token	0	0	0	1
req_url	1	1	1	1
url_of_anchor	0	-1	-1	0
tag_links	1	0	-1	0
SFH	-1	-1	1	-1
submit_to_email	0	1	0	0
abnormal_url	0	0	0	1
redirect	0	0	0	1
mouseover	0	1	0	0
right_click	0	0	0	0
popup	0	1	0	0
iframe	0	1	0	0
domain_Age	-1	-1	-1	0
dns_record	1	1	1	1
traffic	1	0	-1	1
page_rank	-1	0	-1	-1
google_index	0	0	0	0
links_to_page	1	1	1	1
stats_report	0	1	0	0
target	-1	1	1	-1