PHISHING WEBSITE CLASSIFIER USING MACHINE LEARNING

PRESENTED BY,
ANIMESH MANDAL
SOUMEN KHATUA

UNDER THE GUIDANCE OF,
DR. SUSHOVON JANA

INTRODUCTION

What people do through out phishing website that they tried to steal information. They will make some dummy website or they will try to replicate a very famous website with a minor change in URL. They will try to steal the clients personal information. Maybe they will try to replicate your bank website and minor difference will be there. You might be getting fooled and you might enter username and password there and then they will get stored somewhere in the attacker side and they can use that information. So this type of things might happen so you have to make such a system that based on certain criteria we can identify a website is phishing or not that will be our goal. So our criteria is present in our data set based on features.

Objective

TO BUILD A CLASSIFICATION METHODOLOGY TO PREDICT WHETHER A WEBSITE IS A PHISHING OR NOT ON THE BASIS OF GIVEN SET OF PREDICTORS.

Approach

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

- Collect dataset containing phishing and legitimate websites from the open source platforms.
- Analyze and preprocess the dataset by using EDA techniques.
- Divide the dataset into training and testing sets.
- Clustering the training datasets.
- Run selected machine learning model like SVM, XGBoost on the different cluster of training datasets and Choose the best model for each cluster by analyzing the performance matrices.

	having_IP_Address	URL_Length	Shortining_Service	having_At_Symbol	double_slash_redirecting	Prefix_Suffix	having_Sub_Domain	SSLfinal_State	Don
0	-1	1	1	1	-1	-1	-1	-1	
1	1	1	1	1	1	-1	0	1	
2	1	0	1	1	1	-1	-1	-1	
3	1	0	1	1	1	-1	-1	-1	
4	1	0	-1	1	1	-1	1	1	
...	...	...	...	...	...	...	...	...	
11050	1	-1	1	-1	1	1	1	1	
11051	-1	1	1	-1	-1	-1	1	-1	
11052	1	-1	1	1	1	-1	1	-1	
11053	-1	-1	1	1	1	-1	-1	-1	
11054	-1	-1	1	1	1	-1	-1	-1	

11055 rows × 31 columns

Dataset Description

- Having IP Address
- URL length
- Shortening Service
- Having @ symbol
- Double Slash Redirection
- Prefix Suffix
- Having Sub Domain
- SSL state
- Domain registration length
- Favicon

Address bar features

- Using Non-Standard Port
- HTTPS token
- Request URL
- URL of Anchor
- Links in Tags
- Server Form Handler
- Submitting Information To E-mail
- Abnormal URL

Abnormal features

- Website Redirect Count
- Status Bar Customization
- Disabling Right Click
- Using Pop-up Window
- Iframe

HTML and JavaScript-based features

- Age of Domain
- DNS Record
- Web Traffic
- Page Rank
- Google Index
- Links Pointing To Page
- Statistical Report

Domain-based features

1. Using the IP Address :

If an IP address is used as an alternative of the domain name in the URL, such as "http://125.98.3.123/fake.html", users can be sure that someone is trying to steal their personal information. Sometimes, the IP address is even transformed into hexadecimal code as shown in the following link "http://0x58.0xCC.0xCA.0x62/2/paypal.ca/index.html".

<u>Rule</u>: IF $\begin{cases} \text{If The Domain Part has an IP Address} \rightarrow \text{Phishing} \\ \text{Otherwise} \rightarrow \text{Legitimate} \end{cases}$

2. Long URL to Hide the Suspicious Part :

Phishers can use long URL to hide the doubtful part in the address bar. For example: http://federmacedoadv.com.br/3f/aze/ab51e2e319e51502f416dbe46b773a5e/?cmd=_home&dispatch=11004d58f5b74f8dc1e7c2e8dd4105e811004d58f5b74f8dc1e7c2e8dd4105e8@phishing.website.html

To ensure accuracy of our study, we calculated the length of URLs in the dataset and produced an average URL length. The results showed that if the length of the URL is greater than or equal 54 characters then the URL classified as phishing. By reviewing our dataset we were able to find 1220 URLs lengths equals to 54 or more which constitute 48.8% of the total dataset size.

Rule: IF $\begin{cases} URL\ length < 54 \rightarrow feature = \text{Legitimate} \\ else\ if\ URL\ length \geq 54\ and\ \leq 75 \rightarrow feature = Suspicious \\ otherwise \rightarrow feature = \text{Phishing} \end{cases}$

3. URL's having "@" Symbol :

Using "@" symbol in the URL leads the browser to ignore everything preceding the "@" symbol and the real address often follows the "@" symbol.

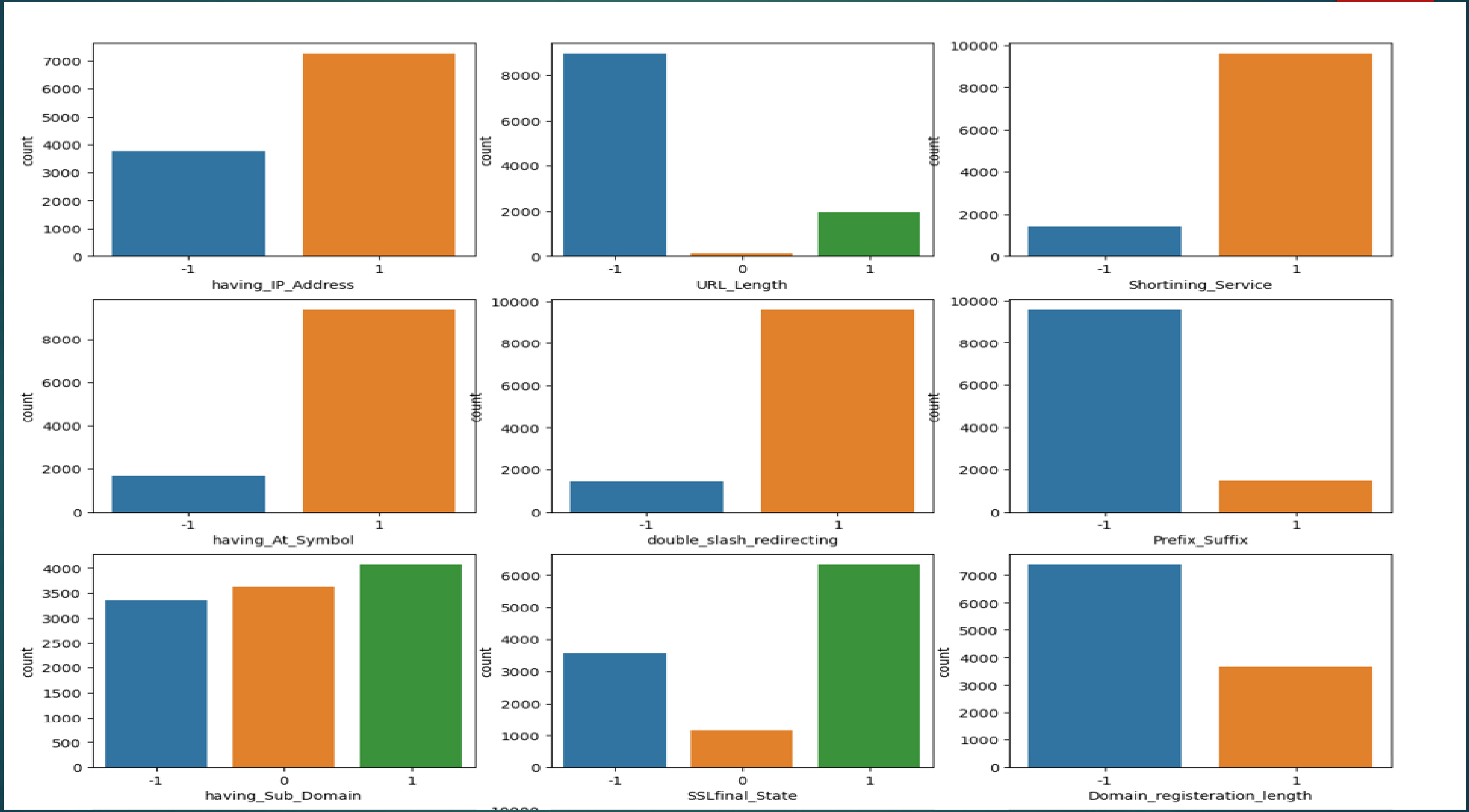
Rule: IF $\begin{cases} \text{Url Having @ Symbol} \rightarrow \text{ Phishing} \\ \quad\quad \text{Otherwise} \rightarrow \text{ Legitimate} \end{cases}$

4. Age of Domain :

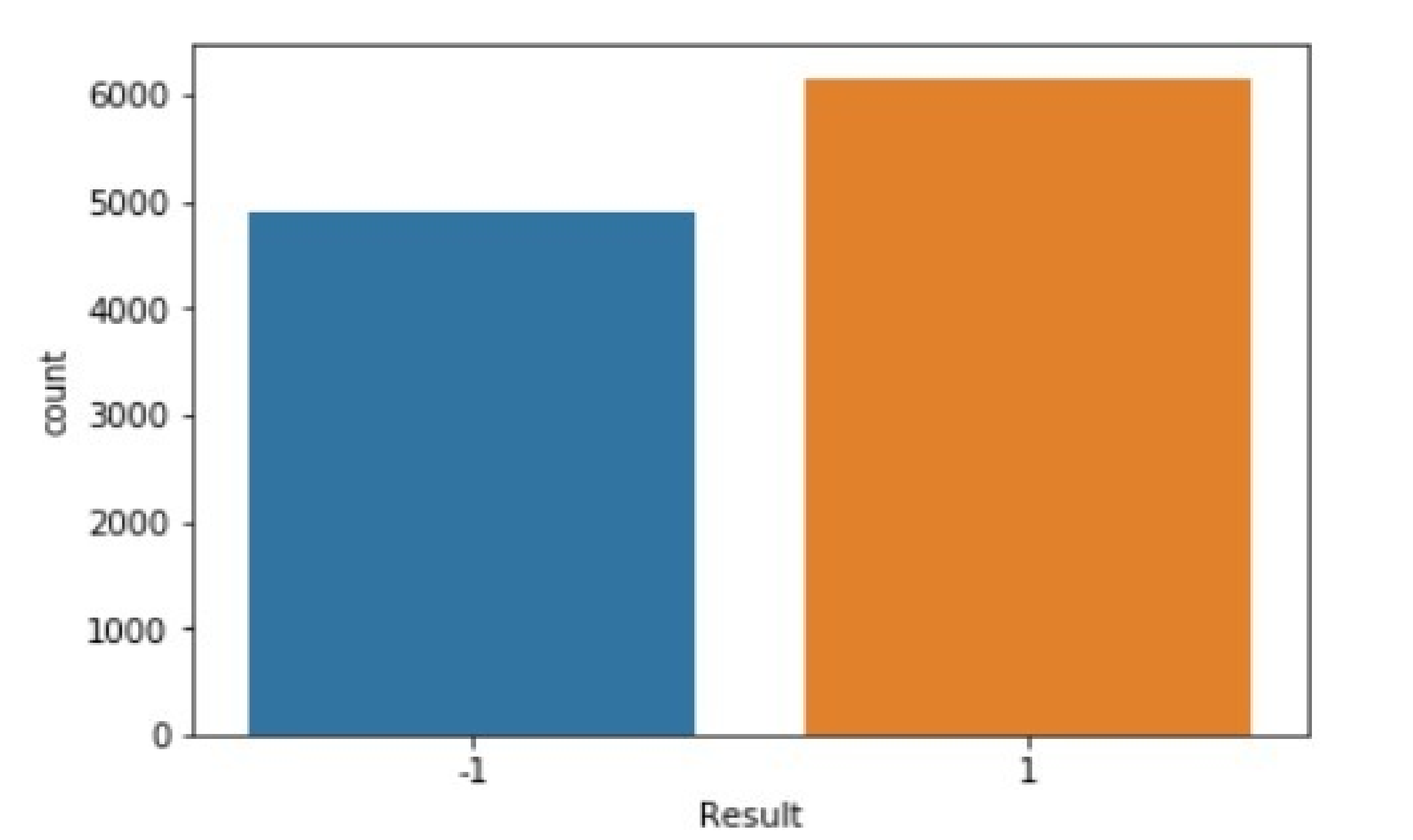
This feature can be extracted from WHOIS database (Whois 2005). Most phishing websites live for a short period of time. By reviewing our dataset, we find that the minimum age of the legitimate domain is 6 months.

Rule: IF $\begin{cases} \text{Age Of Domain} \geq 6 \text{ months } \rightarrow \text{ Legitimate} \\ \quad\quad\quad\quad \text{Otherwise } \rightarrow \text{ Phishing} \end{cases}$

CountPlot of Input Column



CountPlot of Output Column

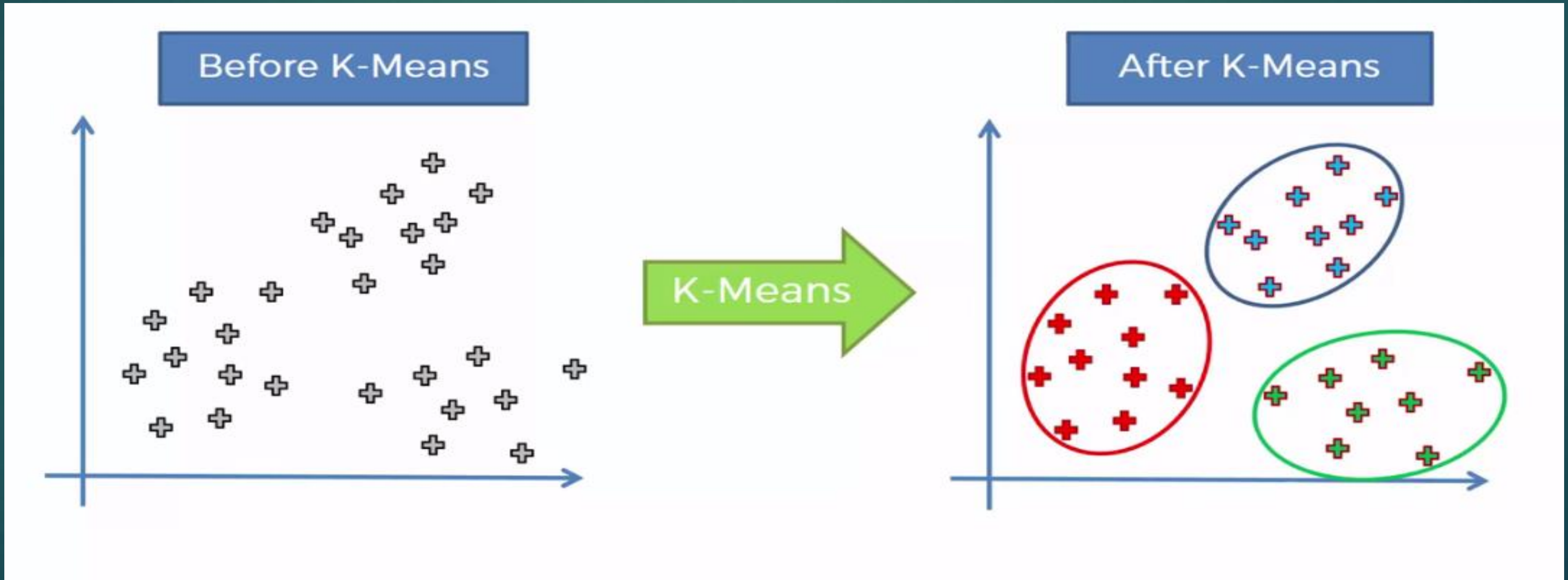


K-means Clustering

What is K means algorithm?

K means algorithm that try to partition the dataset into K predefined distinct non overlapping subgroups(clusters) where each data point belongs to only one group.

The K-means algorithm is used to find groups which have not been explicitly levelled in the data. This can be used to confirm assumption about what data types of group exist or to identify unknown groups in complex data sets.



Steps of K-means Clustering :

Step 1: decide the number of cluster using Elbow method(WCSS

Step 2: Initialize centroids.

Step 3: calculate distance from each data point to centroids.

✓ What type of distance should we use?

- Squared Euclidean distance

Step 4: Assign each object to the closest cluster

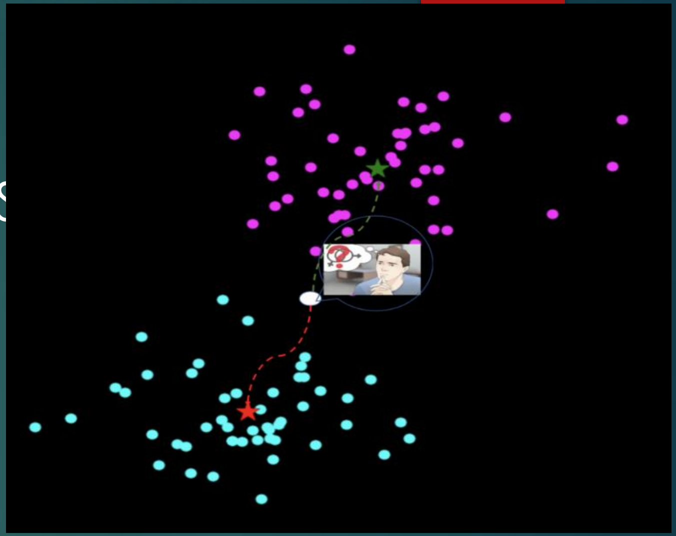
Step 5: Compute the new centroid for each cluster

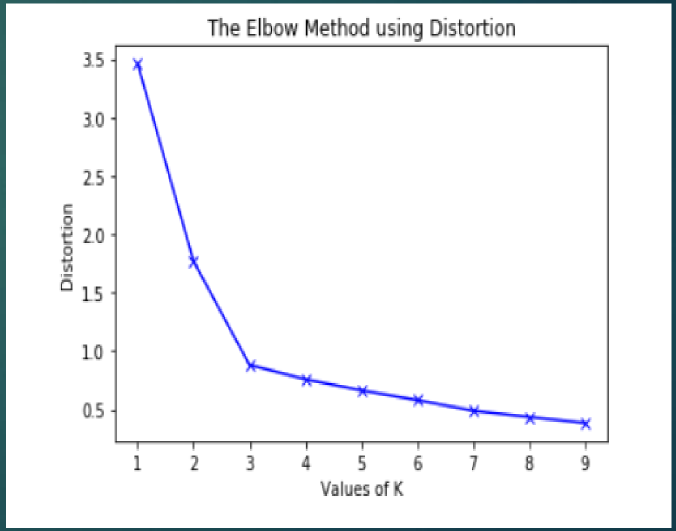
Step 6: Iterate:

- Calculate distance from objects to cluster centroids.
- Assign objects to closest cluster
- Recalculate new centroids

Step 7: Stop based on convergence criteria

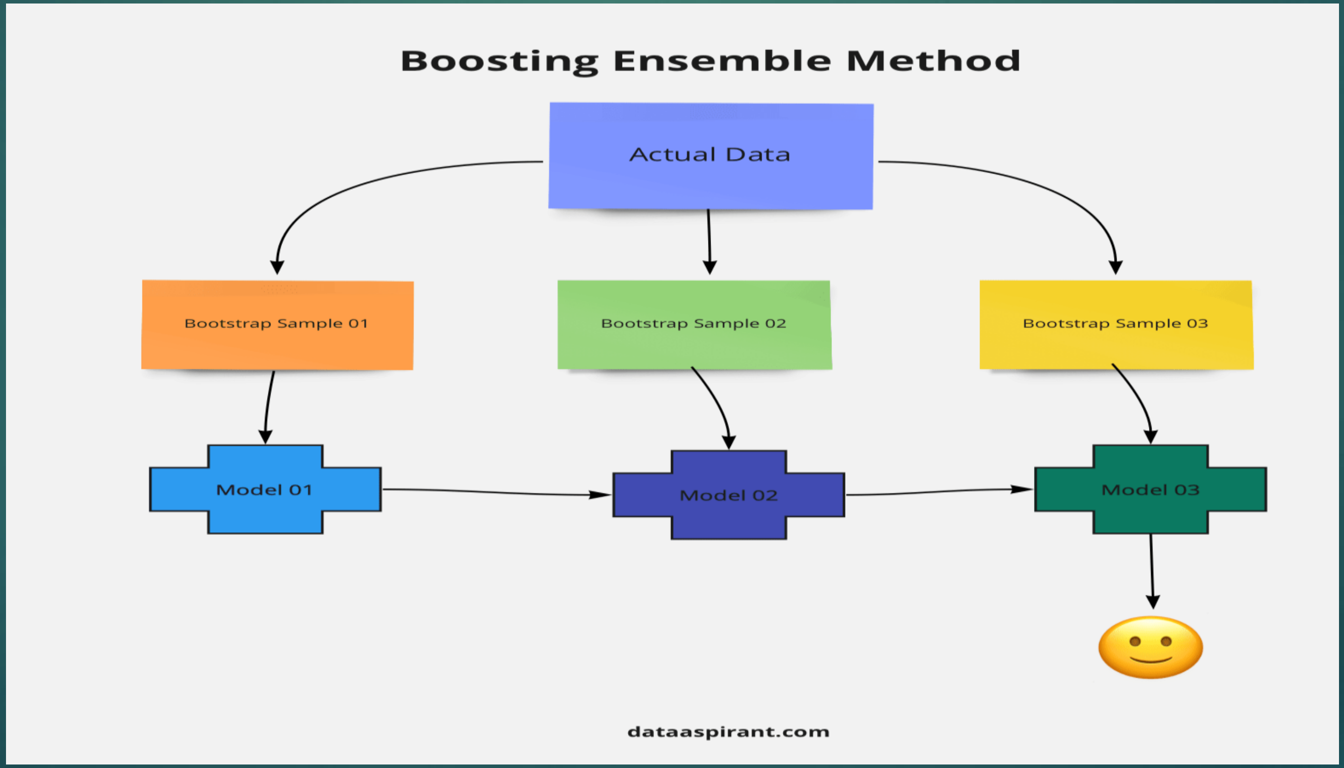
- No change in clusters
- Max iterations





What is Boosting?

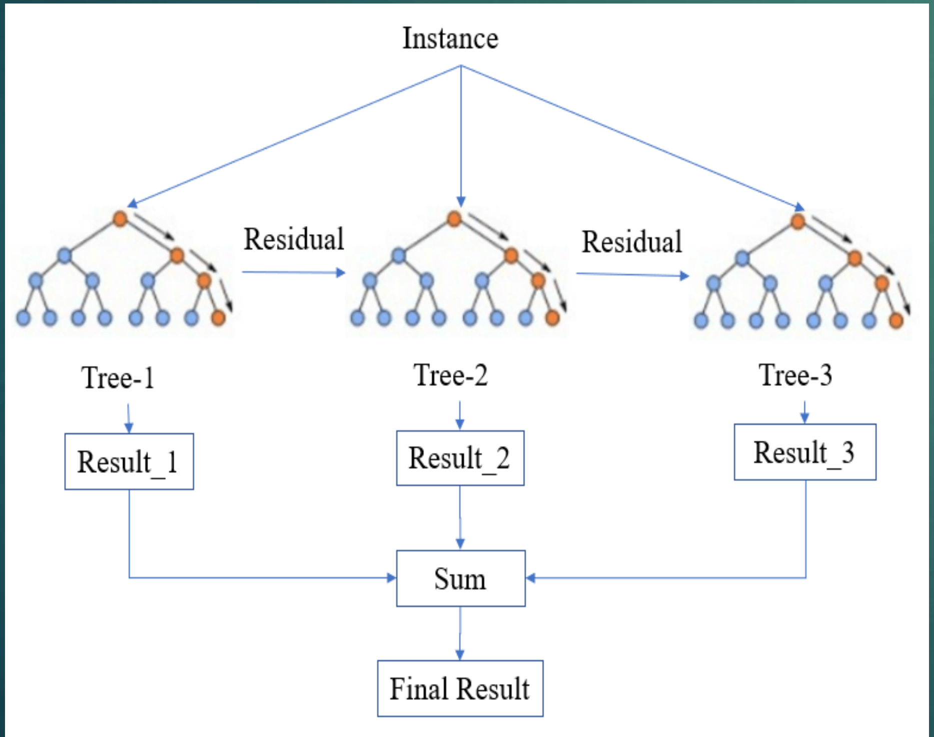
Boosting is an ensemble learning method that combines a set of weak learners into a strong learner to minimize training errors. In boosting, a random sample of data is selected, fitted with a model and then trained sequentially—that is, each model tries to compensate for the weaknesses of its predecessor. With each iteration, the weak rules from each individual classifier are combined to form one, strong prediction rule.

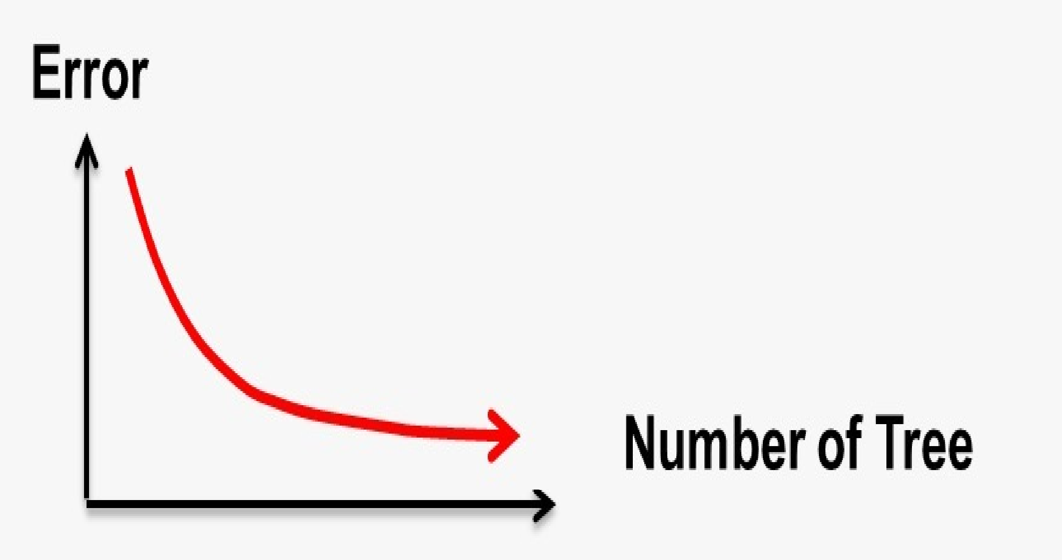


XGBoost

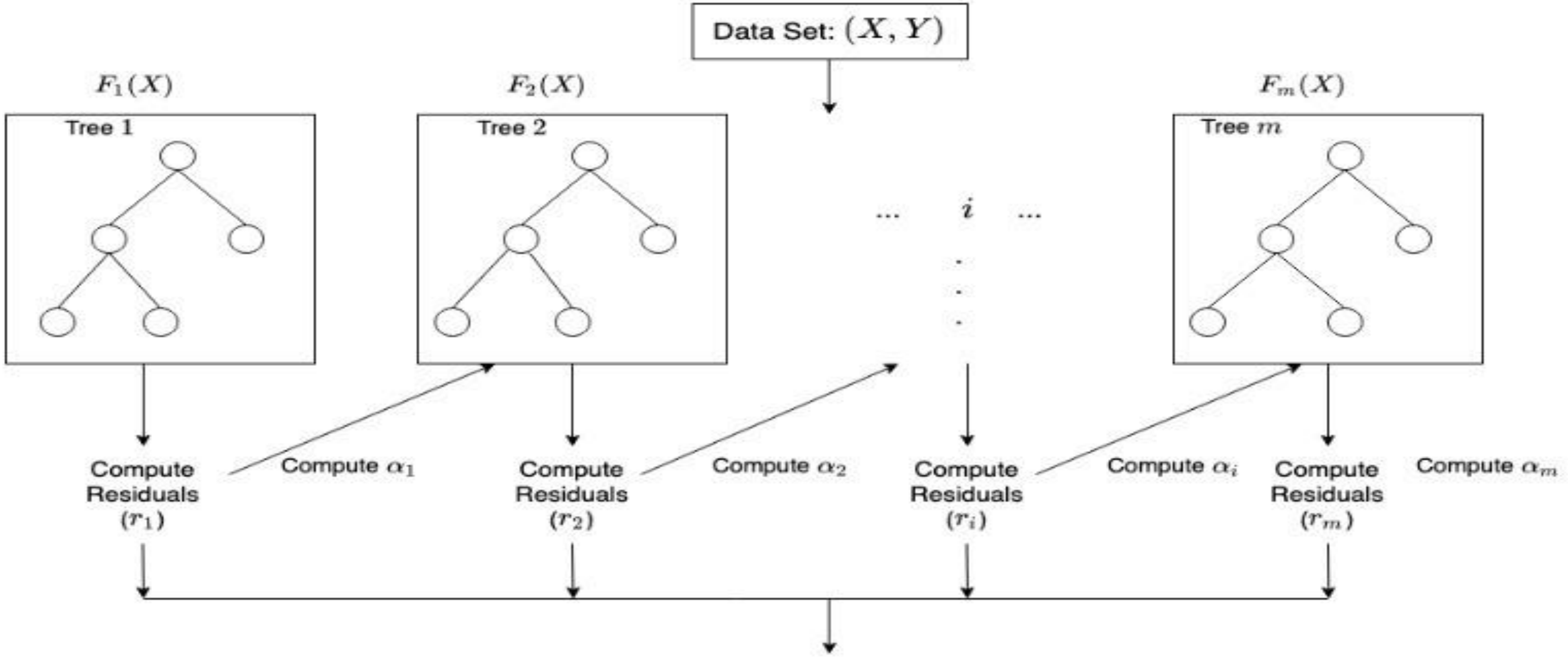
XGBoost stands for extreme gradient boosting it became popular in the recent days and is dominating applied machine learning and kaggle competition for structured data.

XGBoost is an extension to gradient boosted decision trees and specially designed to improve speed and performance. So our problem is based on classification so we have used XGBoost classifier.





How does XGBoost work?



$$F_m(X) = F_{m-1}(X) + \alpha_m h_m(X, r_{m-1}),$$

where α_i, and r_i are the regularization parameters and residuals computed with the i^{th} tree respectfully, and h_i is a function that is trained to predict residuals, r_i using X for the i^{th} tree. To compute α_i we use the residuals computed, r_i and compute the following: $arg \min_{\alpha} = \sum_{i=1}^{m} L(Y_i, F_{i-1}(X_i) + \alpha h_i(X_i, r_{i-1}))$ where $L(Y, F(X))$ is a differentiable loss function.

RESULTS

Clusters	Model	Accuracy	Precision	Recall	f1-score
0	SVM	0.95219	0.946580	0.968815	0.957543
0	XGBoost	0.98323	0.987235	0.981047	0.976802
1	SVM	0.94353	0.935450	0.945440	0.935650
1	XGBoost	0.98569	0.983230	0.987850	0.984120
2	SVM	0.92349	0.936540	0.912320	0.921180
2	XGBoost	0.98453	0.976580	0.986620	0.976570
3	SVM	0.94565	0.946720	0.965570	0.953220
3	XGBoost	0.98878	0.981120	0.984330	0.981490

CONCLUSION & FUTURE WORK

As mentioned, at first we have divided the training data into 4 clusters using K-means algorithm.
After that for every cluster we have perfomed SVM and XGBoost algorithm. On the basis of the performance matrices we have noticed that XGBoost perfoms well for all clusters.
So we have selected XGBoost algorithm for phishing website classifier.

Our future plan for this project is to make an application which can automatically perform the validation and the transformation of a website and extract all the features to perform the chosen Machine Learning model to rectify whether a website is phishing or not.

REFERENCES

- https://archive.ics.uci.edu/ml/machine-learning-databases/00327/
- [1] Abdelhamid N, Thabtah F, Abdel-Jaber H Phishing detection : a recent intelligent machine learning comparison based on models content and features. In Beijing, China: IEEE, 2017.
- [2] Harikrishnan NB, Vinayakumar and Soman KP on “A machine learning approach towards Phishing email detection; 2018.
- [3] V. B.et al, “Study on Phishing attacks,” International Journal of Computer Applications, 2018.
- [4] S. Mishra and D. Soni, “ Smishing detector: A security model to detect smishing through SMS content analysis and URL behaviour analysis,” (in English), future generation computer systems-the International Journal of Escience, Article vol. 108, pp. 803-815, Jul 2020.
- [5] Ollmann, Gunter. "The Phishing Guide: Understanding and Preventing Phishing Attacks". Technical Info. Archived from the original on 2011-01-31. Retrieved 2006-07-10.
- [6] "Spear phishing". Windows IT Pro Center. Retrieved March 4, 2019.
- [7] O'Leary, Daniel E. (2019). "What Phishing E-mails Reveal: An Exploratory Analysis of Phishing Attempts Using Text Analyzes". SSRN Electronic Journal.
- [8] R. Kohavi and F. Provost, "Glossary of terms," Machine Learning, vol. 30, no. 2–3, pp. 271–274, 1998
- [9] Chen, Tianqi; Guestrin, Carlos (2016). "XGBoost: A Scalable Tree Boosting System". In Krishnapuram, Balaji; Shah, Mohak; Smola, Alexander J.; Aggarwal, Charu C.; Shen, Dou; Rastogi, Rajeev (eds.). Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016. ACM. pp. 785–794.
- [10] Gandhi, Rohith (2019-05-24). "Gradient Boosting and XGBoost". Medium. Retrieved 2020-01-04.

THANK YOU