Mini Project Report

On

Detecting Phishing Website – Exploratory Data Analysis

Course: Data Warehousing and Data Mining

Course Code: CAP447

Submitted by

Name: Mohamed jaseem kurikkal
Reg:12318317
Roll No:52
Section: D2339

Submitted to

Ms. Ranjit Kaur
UID: 28632
Assistant Professor
SCA, LPU

Lovely Faculty of Technology & Sciences

Lovely Professional University

Punjab

2023-2024 (Term 1)

Introduction

In today's digital age, the internet has become an indispensable tool for communication, commerce, and information access. However, this interconnected world has also provided fertile ground for malicious actors to exploit unsuspecting users. Phishing websites, meticulously crafted to deceive users into surrendering sensitive personal information, have emerged as a prevalent form of cybercrime. In response to this growing threat, researchers and cybersecurity experts are continuously developing innovative techniques to detect and identify phishing website.

RapidMiner stands as a powerful data science and machine learning platform, offering a comprehensive suite of tools for data analysis, modelling, and prediction. Its intuitive graphical user interface and extensive library of operators make it an ideal platform for constructing phishing website detection models.

The successful development of a robust phishing website detection model using RapidMiner will significantly enhance cybersecurity measures and safeguard internet users from potential financial losses and identity theft. This project not only contributes to the ongoing battle against cybercrime but also serves as a valuable learning experience, providing hands-on exposure to cutting-edge data science and machine learning techniques.

The laboratory specializes in utilizing data warehouse and data mining techniques to advance phishing website detection. Their project focuses on developing robust algorithms and models to analyze diverse online activity datasets, extracting meaningful patterns. Through advanced data mining methodologies, the lab aims to identify subtle indicators inherent to phishing websites, contributing to ongoing cybersecurity efforts. The research includes exploring various data sources, feature engineering, and applying machine learning algorithms to create an efficient phishing detection system. This initiative addresses evolving challenges posed by deceptive online practices, emphasizing the significance of data-driven approaches in enhancing web security.

The significance of this project extends beyond the realm of cybersecurity. Successfully detecting phishing websites not only enhances our ability to protect sensitive information but also contributes valuable insights to the ongoing efforts to understand and counteract cyber threats.

Project Objectives:

1. Data Collection: Collect a comprehensive dataset of URLs, website content, and HTML tags for both legitimate and phishing websites. This dataset will serve as the foundation for training and evaluating the machine learning model.
2. Data Preprocessing: Clean and prepare the collected data by removing irrelevant features, handling missing values, and transforming categorical features into numerical representations suitable for machine learning algorithms.
3. Feature Engineering: Extract meaningful features from the website data that can effectively distinguish between legitimate and phishing websites. This may involve creating new features, such as URL length, presence of suspicious keywords, and HTML tag patterns.
4. Model Selection: Choose an appropriate machine learning algorithm for phishing website detection. Popular choices include decision trees, random forests, support vector machines, and neural networks.
5. Model Training: Train the selected machine learning algorithm on the preprocessed data, iteratively adjusting hyperparameters to optimize the model's performance.
6. Model Evaluation: Evaluate the trained model's performance using metrics such as accuracy, precision, recall, and F1-score. Employ a holdout set or cross-validation to ensure the model generalizes well to unseen data.
7. Feature Importance Analysis: Identify the most important features in the model, providing insights into the characteristics that distinguish legitimate and phishing websites. This information can be used to improve the model's interpretability and effectiveness.
8. Model Deployment: Integrate the trained model into a web application or browser extension to enable real-time phishing website detection.
9. Performance Monitoring: Continuously monitor the model's performance as new phishing websites emerge and attack patterns evolve. Update the model as needed to maintain its effectiveness.
10. Error Analysis: Analyze incorrectly classified websites to understand the model's limitations and identify areas for improvement.
11. Comparative Analysis: Compare the performance of the developed model with other state-of-the-art phishing detection techniques.
12. Visualization: Create visualizations to illustrate the model's behavior, feature importance, and performance metrics. This can enhance the project's understanding and presentation.

One of the key objectives of detecting phishing websites in RapidMiner is to develop a robust and accurate machine learning model that can effectively distinguish between legitimate and phishing websites. This involves carefully selecting and training an appropriate machine learning algorithm, thoroughly evaluating its performance, and continuously monitoring its effectiveness as new phishing websites emerge.

Description of Dataset

Length URL:
This feature measures the character count of the URL, providing insights into potential anomalies or suspicious elongated URLs.

Length Hostname:
The length of the hostname is crucial for identifying unusual patterns that might indicate phishing attempts, as attackers often manipulate hostnames.

IP:
Indicates the presence of an IP address in the URL, which can be a sign of phishing attempts, as legitimate websites typically use domain names.

Random Domain:
Flags the usage of randomly generated domains, a common tactic employed by phishing sites to evade detection.

Random Subdomain:
Similar to random domains, this feature focuses on subdomains, detecting irregularities that could point to malicious intent.

Path Extension:
Examines the extension of the URL path, offering insights into whether the URL structure aligns with typical patterns or exhibits suspicious behaveio.

Shortening Service:
Identifies the use of URL shortening services, a tactic often exploited by attackers to obscure the destination and deceive users.

Phish Hints:
This feature incorporates known phishing indicators or patterns, aiding in the identification of potentially malicious websites.

NB Hyperlinks:
Counts the number of hyperlinks on a page, providing a metric for assessing the complexity of the webpage and potential phishing activity.

Ratio Int Hyperlinks:
Calculates the ratio of internal to external hyperlinks, aiding in distinguishing between genuine websites and those attempting to redirect users.

Page Rank:
Utilizes page rank algorithms to assess the importance and relevance of a webpage, contributing to the evaluation of a site's legitimacy.

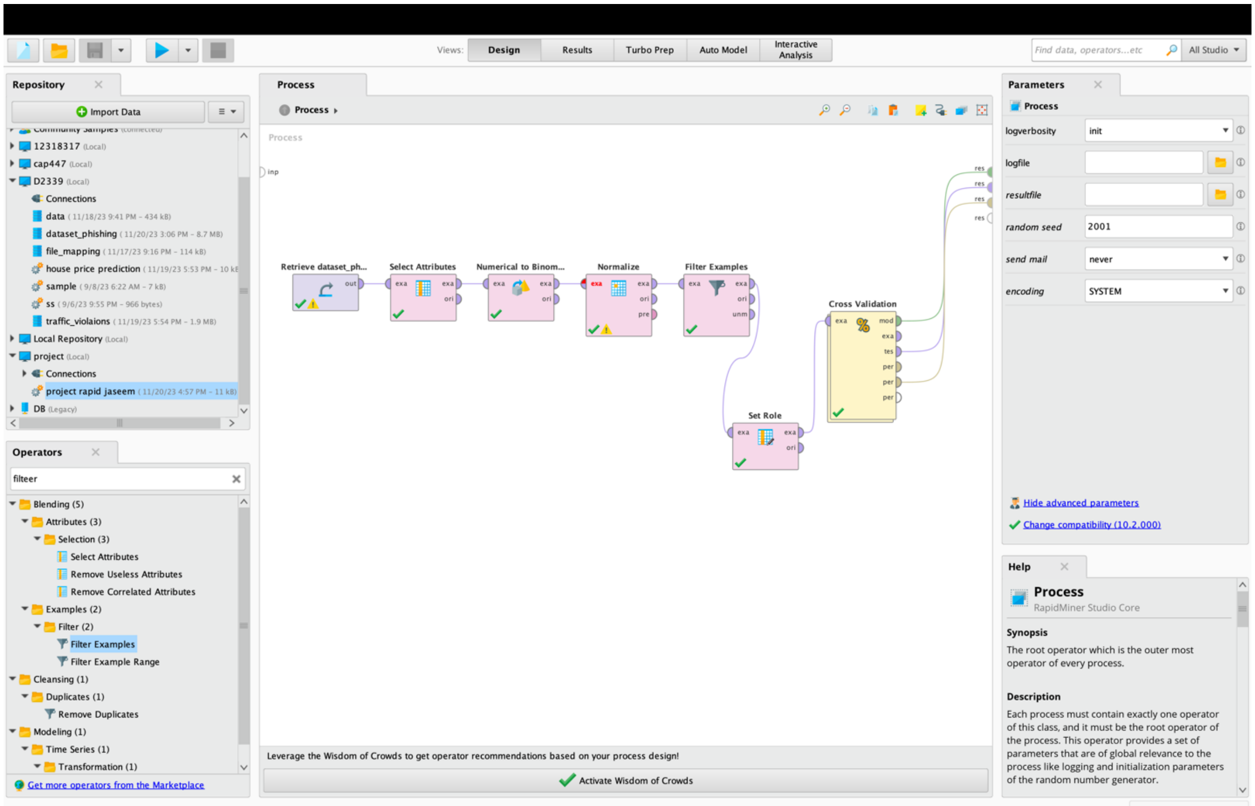
Google Index:
Indicates whether the webpage is indexed by Google, offering insights into its visibility and potential legitimacy.

Web Traffic:
Measures the volume of web traffic to the site, with a focus on identifying anomalies that may indicate phishing attempts.

Domain Registration Length:
Examines the duration of a domain's registration, as shorter durations may be indicative of fraudulent intent.

Domain Age:
Provides the age of the domain, a crucial factor in assessing the credibility of a website, as established domains are less likely to be associated with phishing activities.

Steps for model creation



Steps 1:

- Retrieve dataset phishing
  1. We need to study the data set
  2. We need to filter the data

Step 2:

- Select Attributes
  1. We need to take select attributes from operators
  2. In select attribute we need to select the attributes which are needed.
  3. Then we need to change the parameters.
  4. Select attributes filter type A Subset.
  5. Then select subset select attributes..
  6. After selecting attributes click on Apply.

Step 3:

- Normalize
    1. Select normalize from operator
    2. Add it to the model/design
    3. In parameters method will be Z- transformation

Step 4:

- Filter Example
    1. Select filter example from operators
    2. Add it to the model
    3. In parameters condition class No_missing_attributes

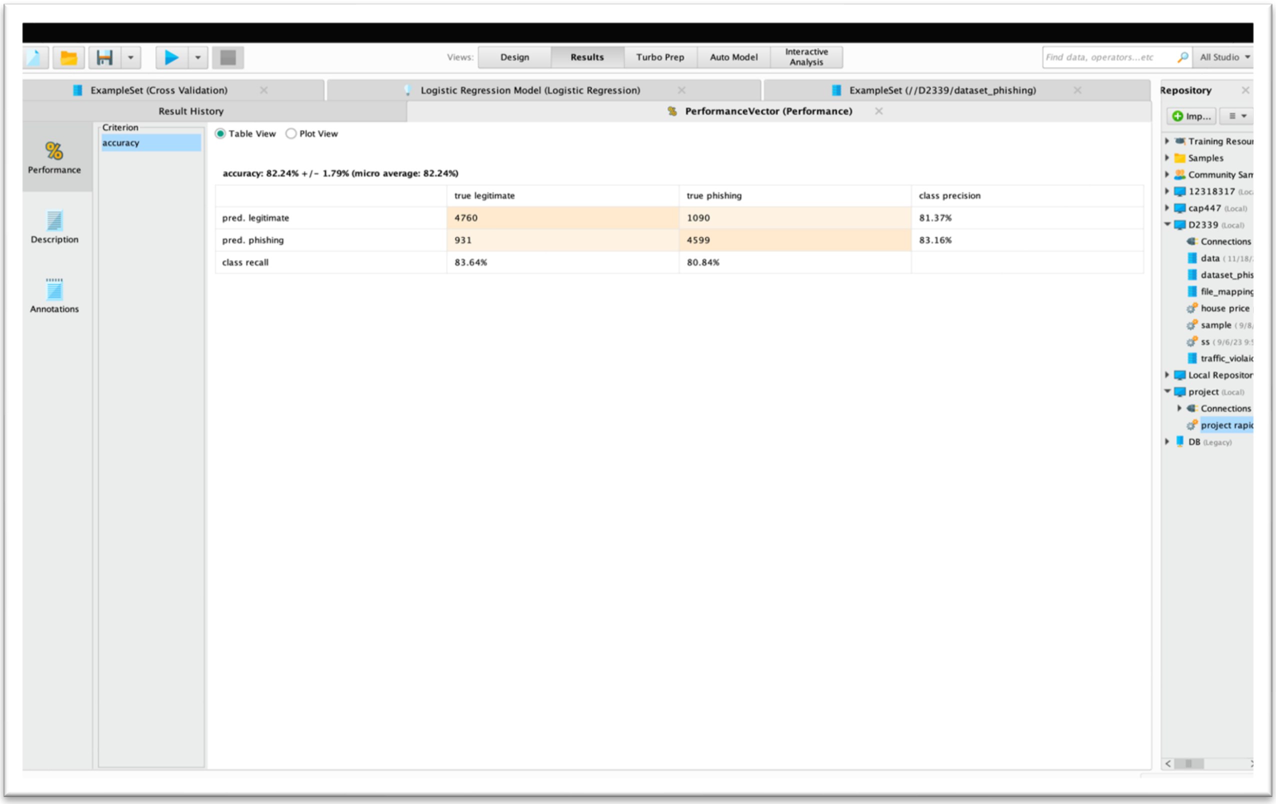
Step 5:

- Set role
    1. Select set role from operators
    2. In set role me need to change the parameter
    3. Attributes name Status target role will be label
    4. Click on apply

Step 6:

- Cross validations
    1. Select cross validation from operators
    2. Add it to the model.
    3. Click on cross validation.
    4. Now in cross validation we need to add operator sample and decision tree in training model.
    5. In testing model we need to select operator like apply model and performance classification

Results and Discussion

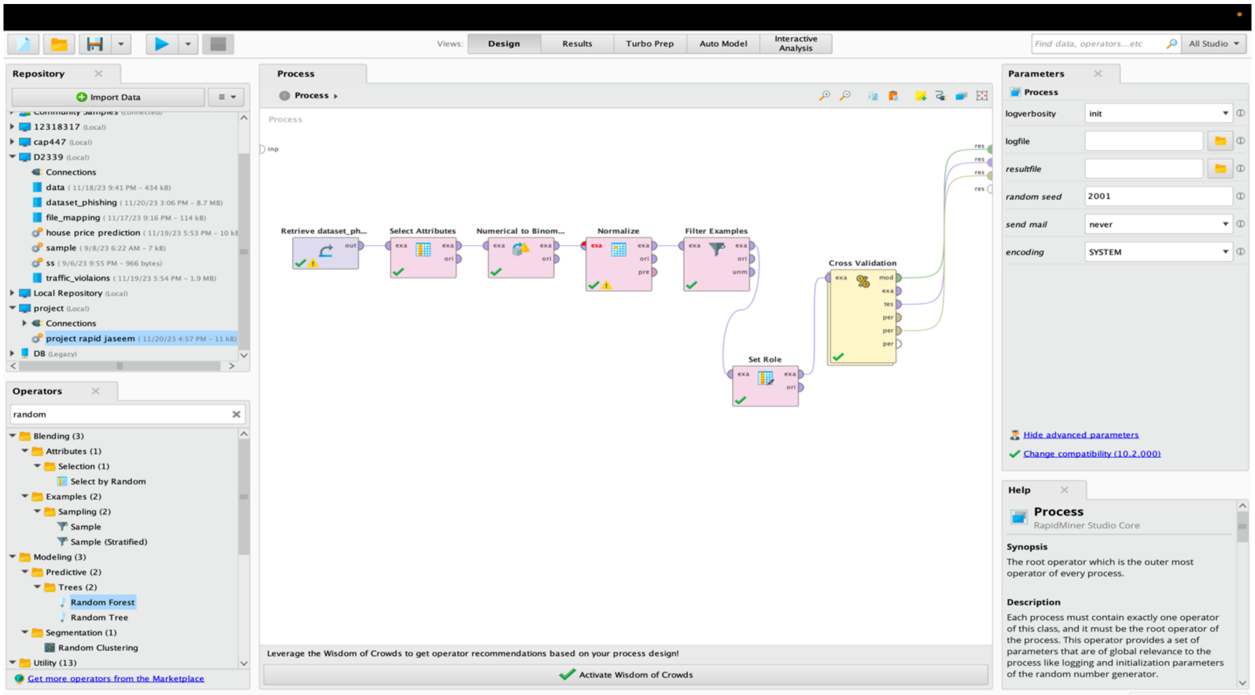


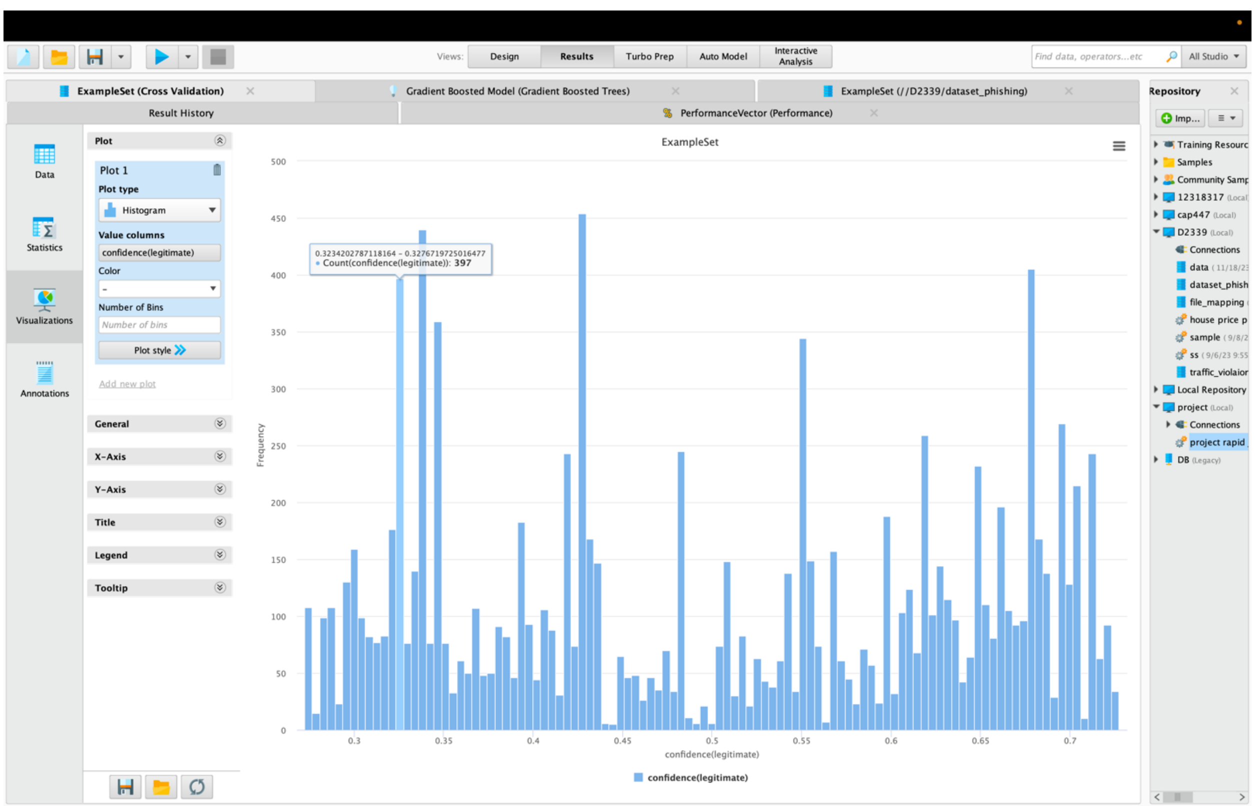
	true legitimate	true phishing	class precision
pred. legitimate	4760	1090	81.37%
pred. phishing	931	4599	83.16%
class recall	83.64%	80.84%	

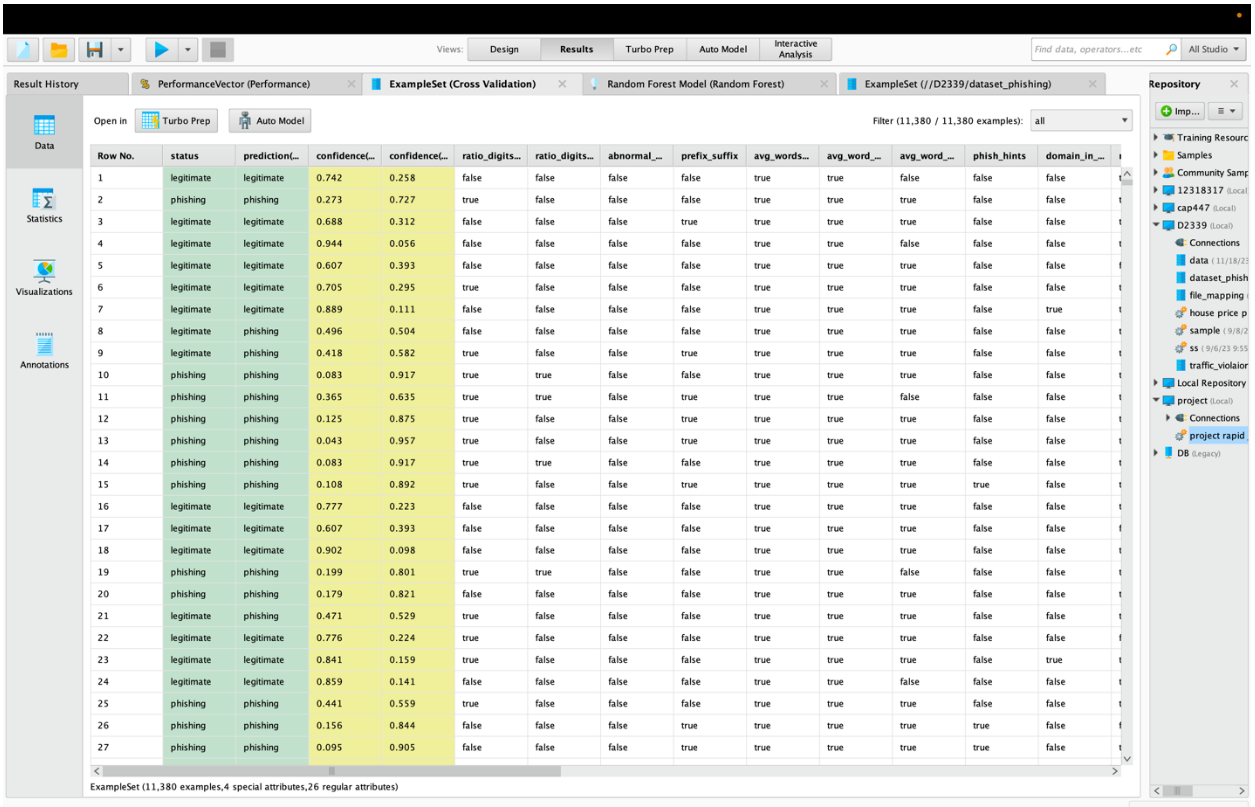
accuracy: 82.24% +/- 1.79% (micro average: 82.24%)

Screenshot of Data

Screenshot of data

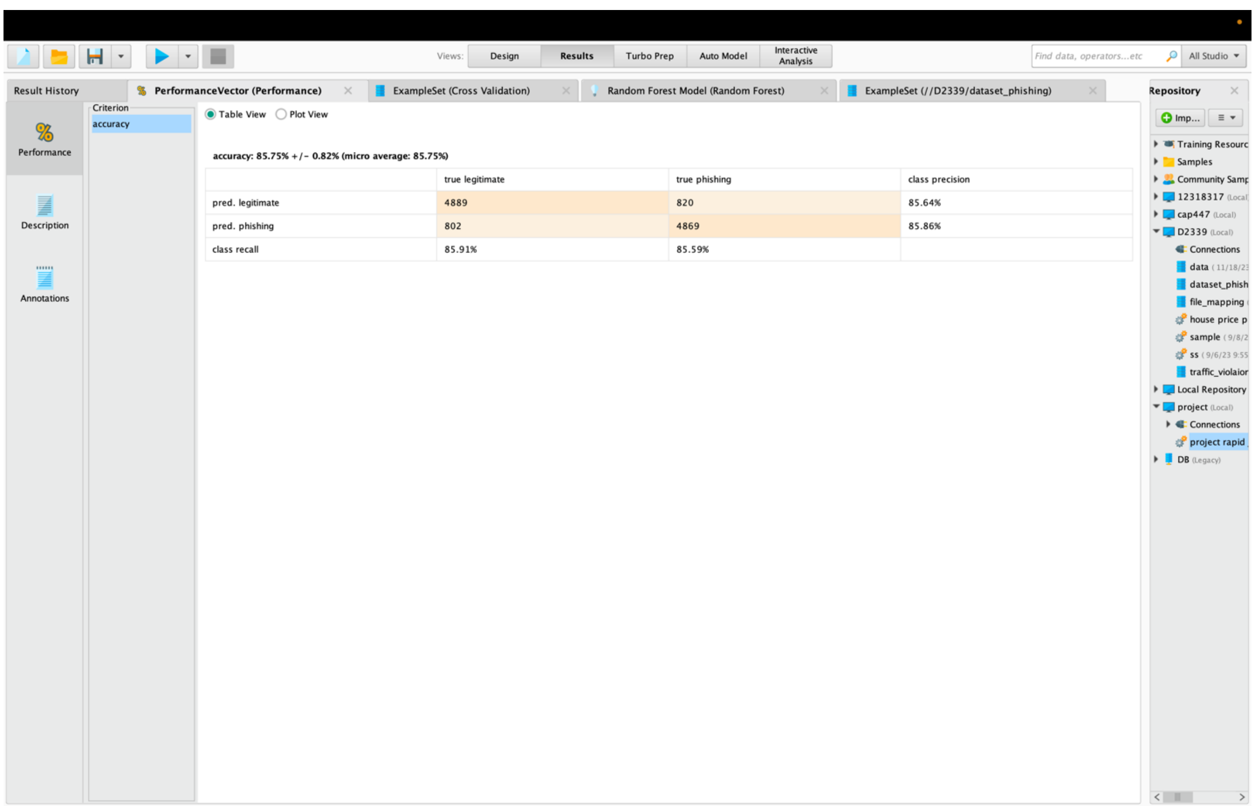






Row No.	status	prediction(...	confidence(...	confidence(...	ratio_digits...	ratio_digits...	abnormal_...	prefix_suffix	avg_words...	avg_word_...	avg_word_...	phish_hints	domain_in_...
1	legitimate	legitimate	0.742	0.258	false	false	false	false	true	true	false	false	false
2	phishing	phishing	0.273	0.727	true	false	false	false	true	true	true	false	false
3	legitimate	legitimate	0.688	0.312	false	false	false	true	true	true	true	false	false
4	legitimate	legitimate	0.944	0.056	false	false	false	false	true	true	false	false	false
5	legitimate	legitimate	0.607	0.393	false	false	false	false	true	true	true	false	false
6	legitimate	legitimate	0.705	0.295	true	false	false	false	true	true	true	false	false
7	legitimate	legitimate	0.889	0.111	false	false	false	false	true	true	true	false	true
8	legitimate	phishing	0.496	0.504	false	false	false	false	true	true	true	false	false
9	legitimate	phishing	0.418	0.582	true	false	false	true	true	true	true	false	false
10	phishing	phishing	0.083	0.917	true	true	false	false	true	true	true	false	false
11	phishing	phishing	0.365	0.635	true	true	false	true	true	true	false	false	false
12	phishing	phishing	0.125	0.875	true	false	false	true	true	true	true	false	false
13	phishing	phishing	0.043	0.957	true	false	false	true	true	true	true	false	false
14	phishing	phishing	0.083	0.917	true	true	false	false	true	true	true	false	false
15	phishing	phishing	0.108	0.892	true	false	false	true	true	true	true	true	false
16	legitimate	legitimate	0.777	0.223	false	false	false	false	true	true	true	false	false
17	legitimate	legitimate	0.607	0.393	false	false	false	false	true	true	true	false	false
18	legitimate	legitimate	0.902	0.098	false	false	false	false	true	true	true	false	false
19	phishing	phishing	0.199	0.801	true	true	false	false	true	true	false	false	false
20	phishing	phishing	0.179	0.821	false	false	false	false	true	true	true	false	false
21	legitimate	phishing	0.471	0.529	true	false	false	false	true	true	true	false	false
22	legitimate	legitimate	0.776	0.224	true	false	false	false	true	true	true	false	false
23	legitimate	legitimate	0.841	0.159	true	false	false	false	true	true	true	false	true
24	legitimate	legitimate	0.859	0.141	false	false	false	false	true	true	false	false	false
25	phishing	phishing	0.441	0.559	true	false	false	false	true	true	true	false	false
26	phishing	phishing	0.156	0.844	false	false	false	true	true	true	true	true	false
27	phishing	phishing	0.095	0.905	false	false	false	true	true	true	true	true	false

ExampleSet (11,380 examples,4 special attributes,26 regular attributes)



accuracy: 85.75% +/- 0.82% (micro average: 85.75%)

	true legitimate	true phishing	class precision
pred. legitimate	4889	820	85.64%
pred. phishing	802	4869	85.86%
class recall	85.91%	85.59%	

ALGORITHMS

Decision Tree: -

A decision tree is a decision support hierarchical model that uses a tree- like model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. It is one way to display an algorithm that only contains conditional control statements.

Random Forest: -

Random Forest is a commonly used machine learning algorithm trademarked by leo breiman and adele cutler, which combines the output of multiple division tree to reach a single result. Its each of use and flexibility have fueled its adoption, as it handles both classification and regression problems.

KNN Algorithm: -

k- nearest neighbors algorithm. The k-nearest neighbors, also known as KNN, is a non-parametric, supervised learning classifier. Which uses proximity to make classification or prediction about the grouping of an individual data point.

Logistic Regression Algorithm: -

Logistic regression is a statistical method that predicts a binary outcome based on prior observations. It uses mathematics to find relationships between two data factors. The model uses this relationship to predict the value of one of those factors based on the other.

Gradient Boosted Model: -

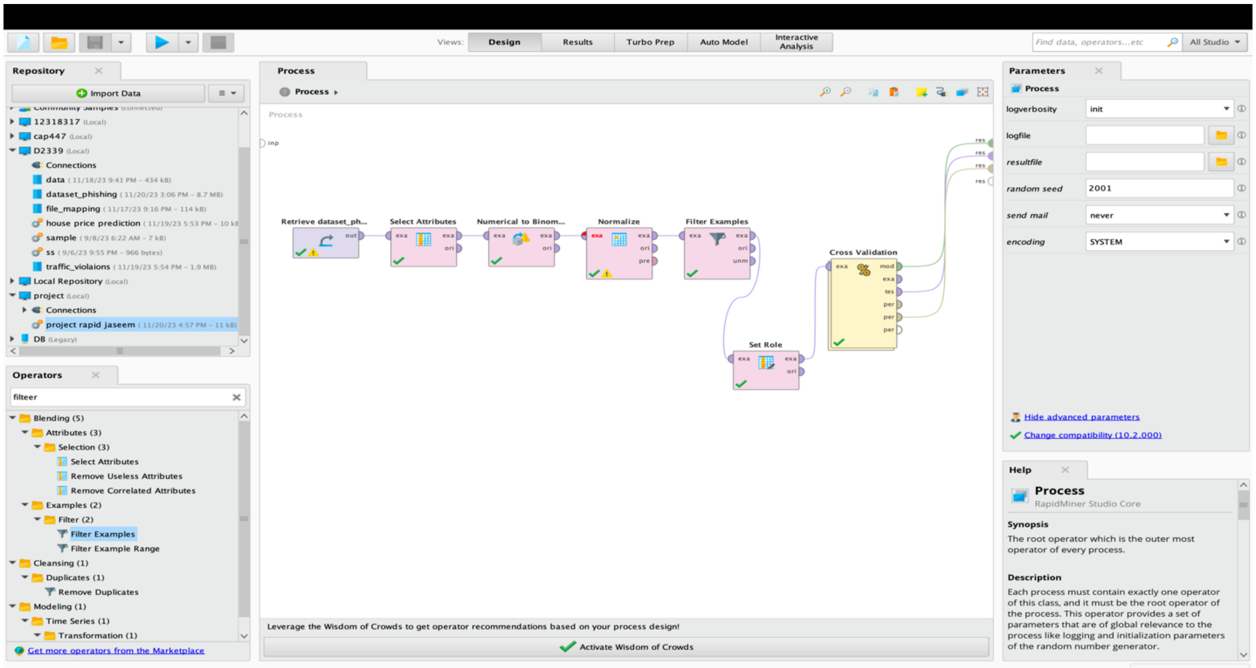
A gradient boosted model is a machine learning technique that uses an ensemble of weak prediction models to produce a more accurate final model. The weak predictions model are typically simple decision trees that make few assumptions about the data.

Simple Distribution: -

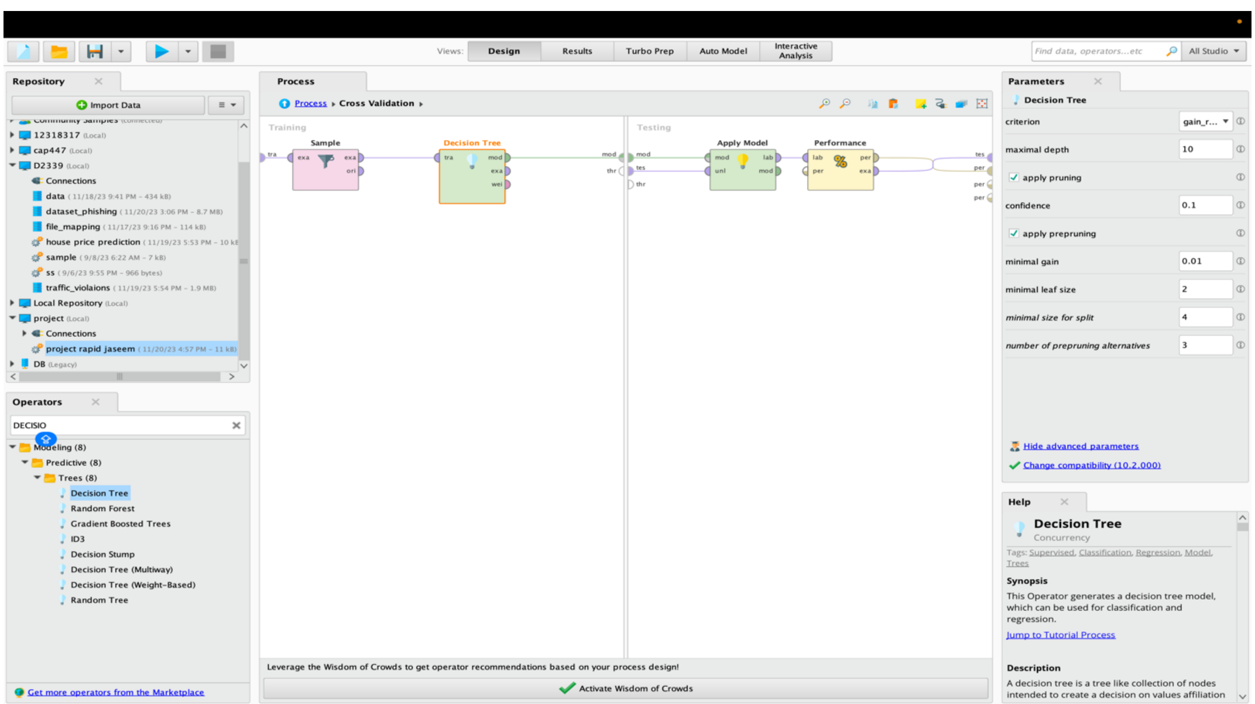
Distribution learning is another classic unsupervised learning task, which includes density estimation and generative modelling. As its name indicates, this task consists of learning the probability distribution of the data.

Machine Learning Model

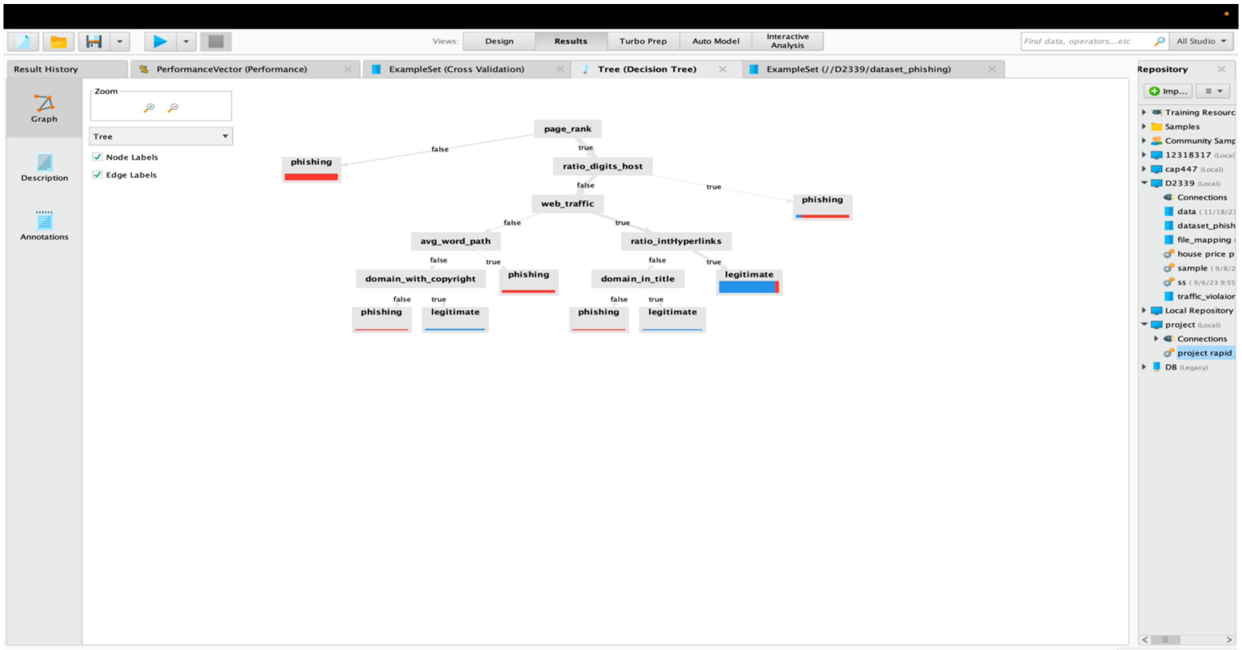
Process: -

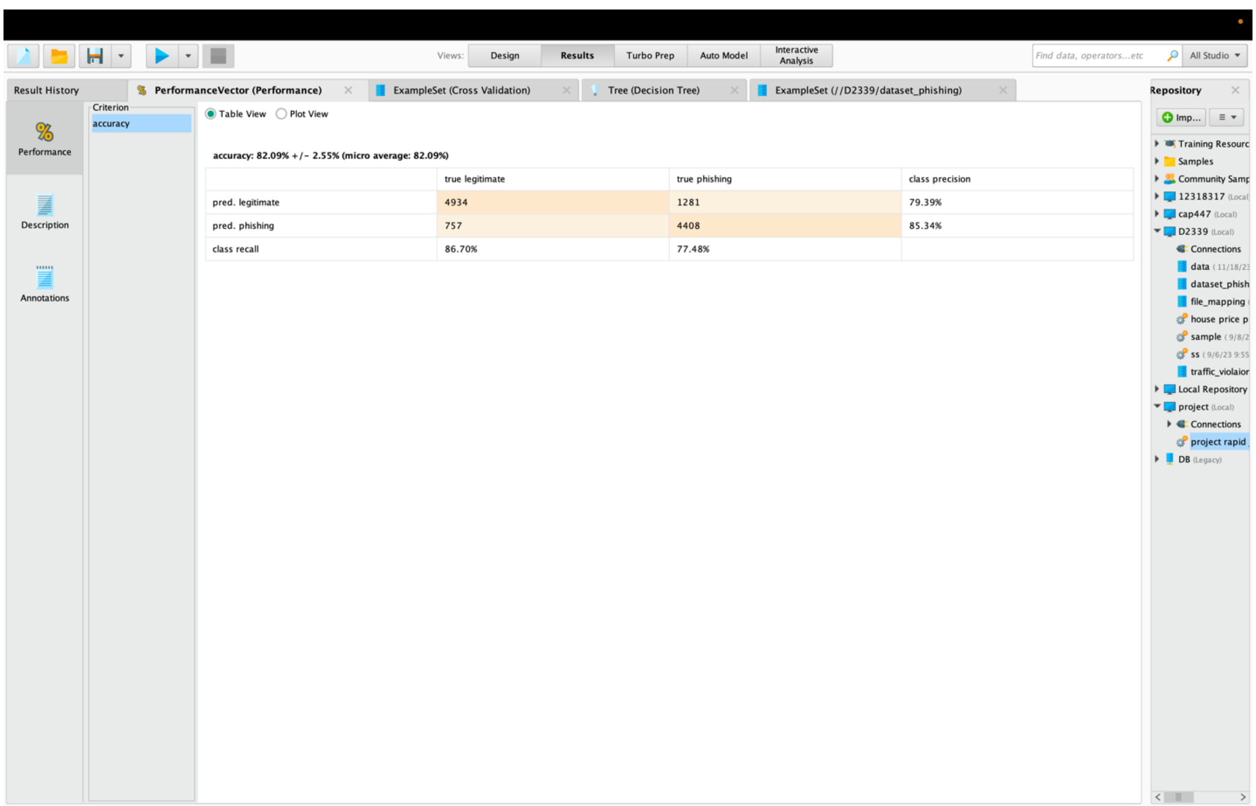


Cross Validation: -



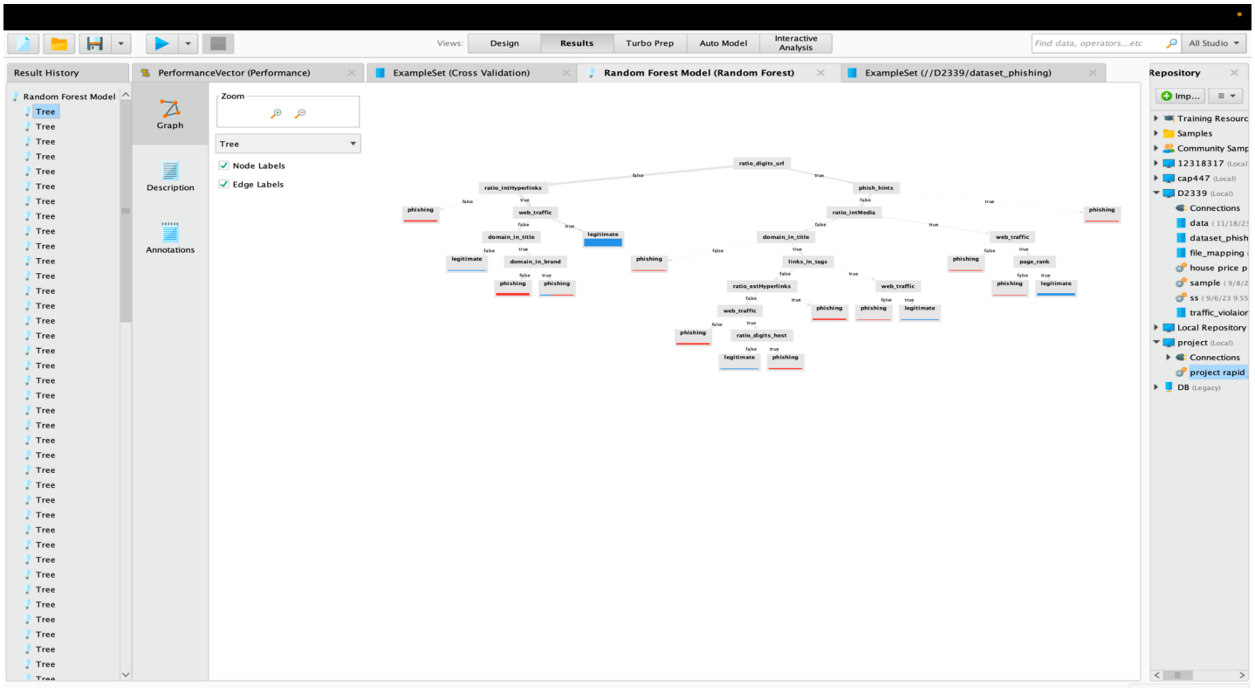
Decision Tree:-

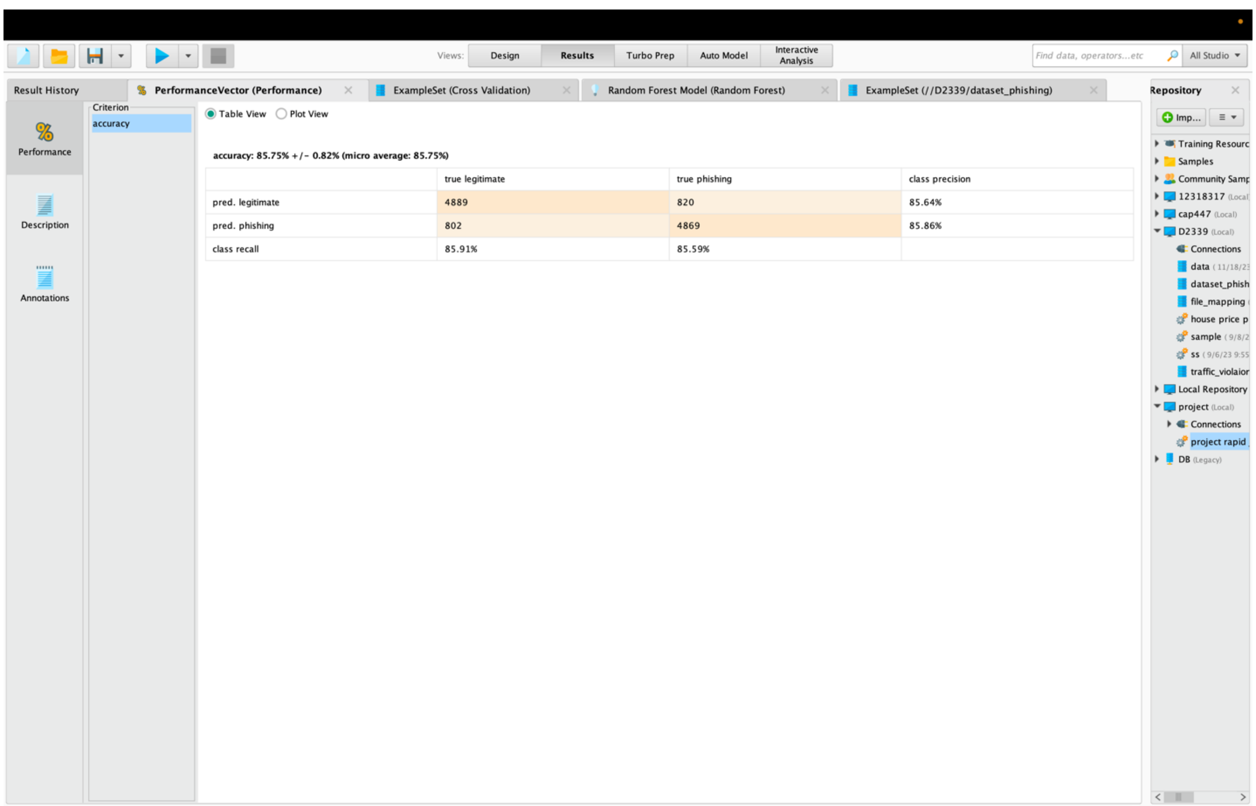




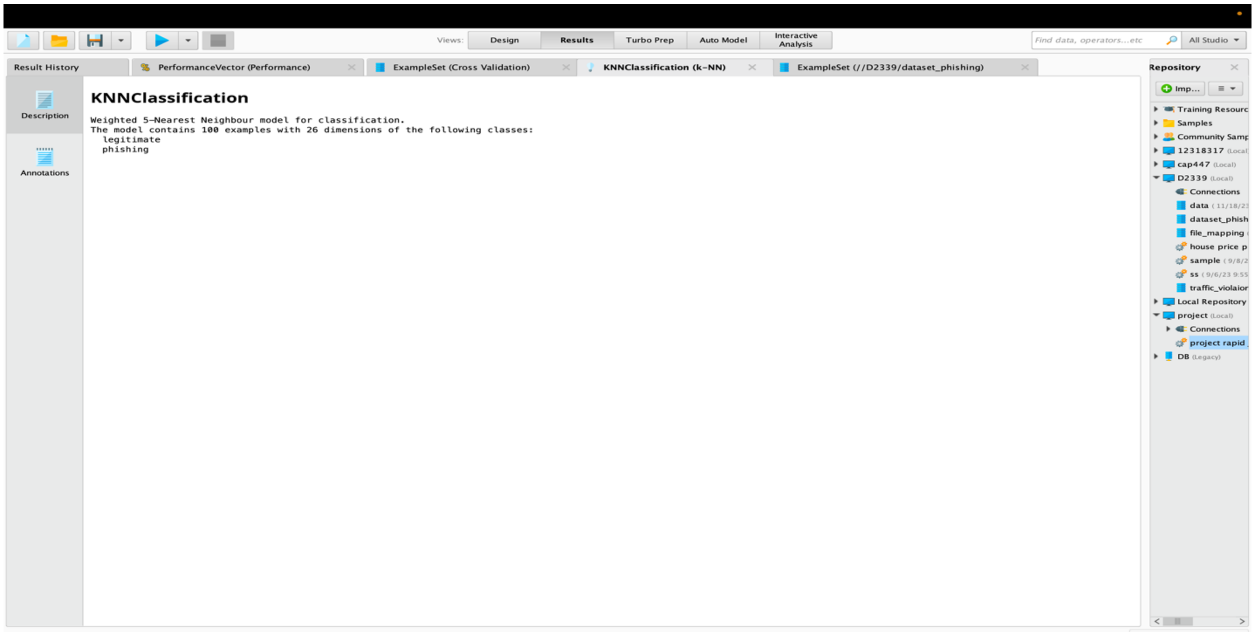
	true legitimate	true phishing	class precision
pred. legitimate	4934	1281	79.39%
pred. phishing	757	4408	85.34%
class recall	86.70%	77.48%	

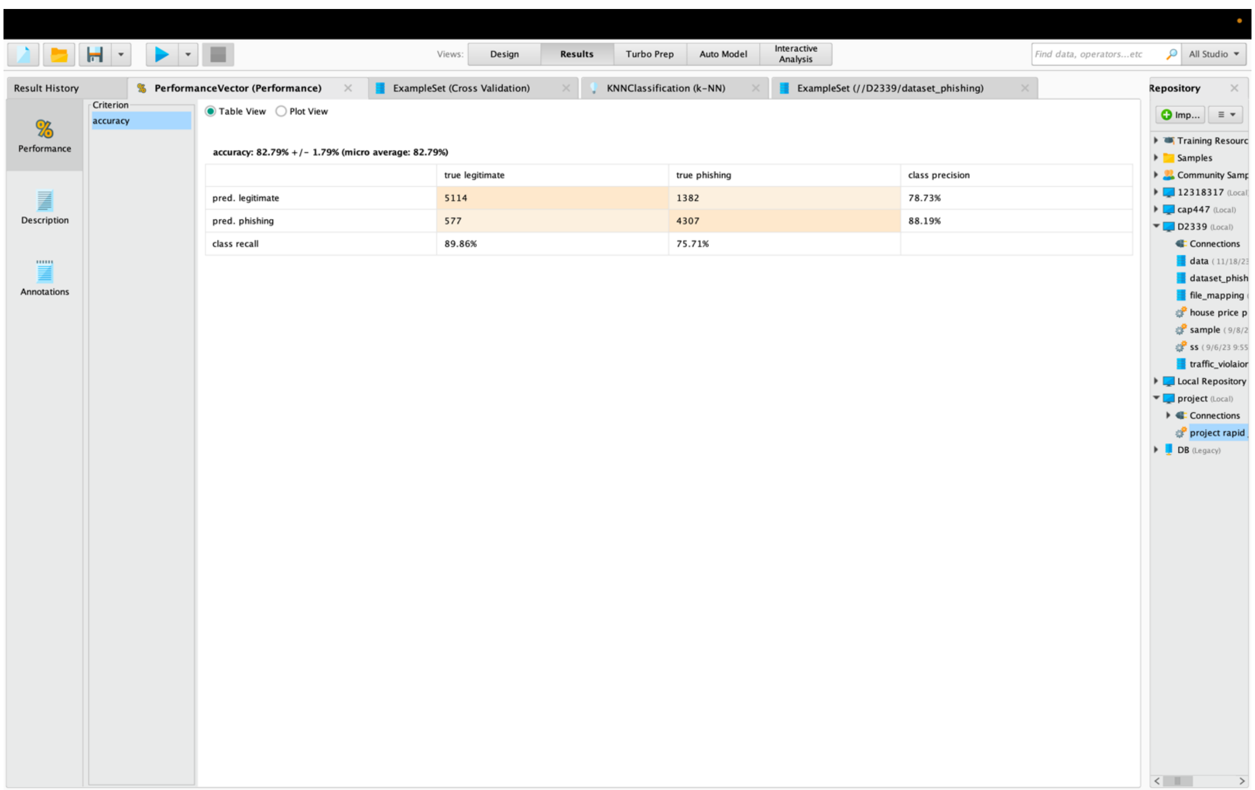
Random forest:-



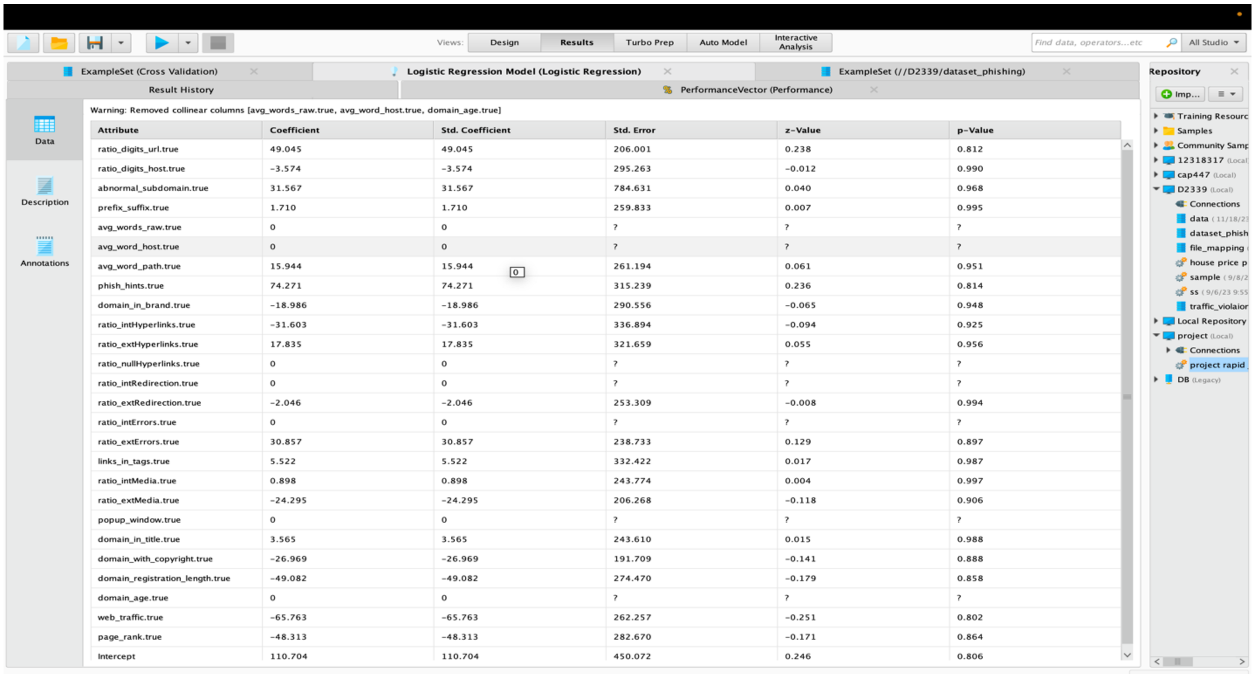


KNN Algorithm:-



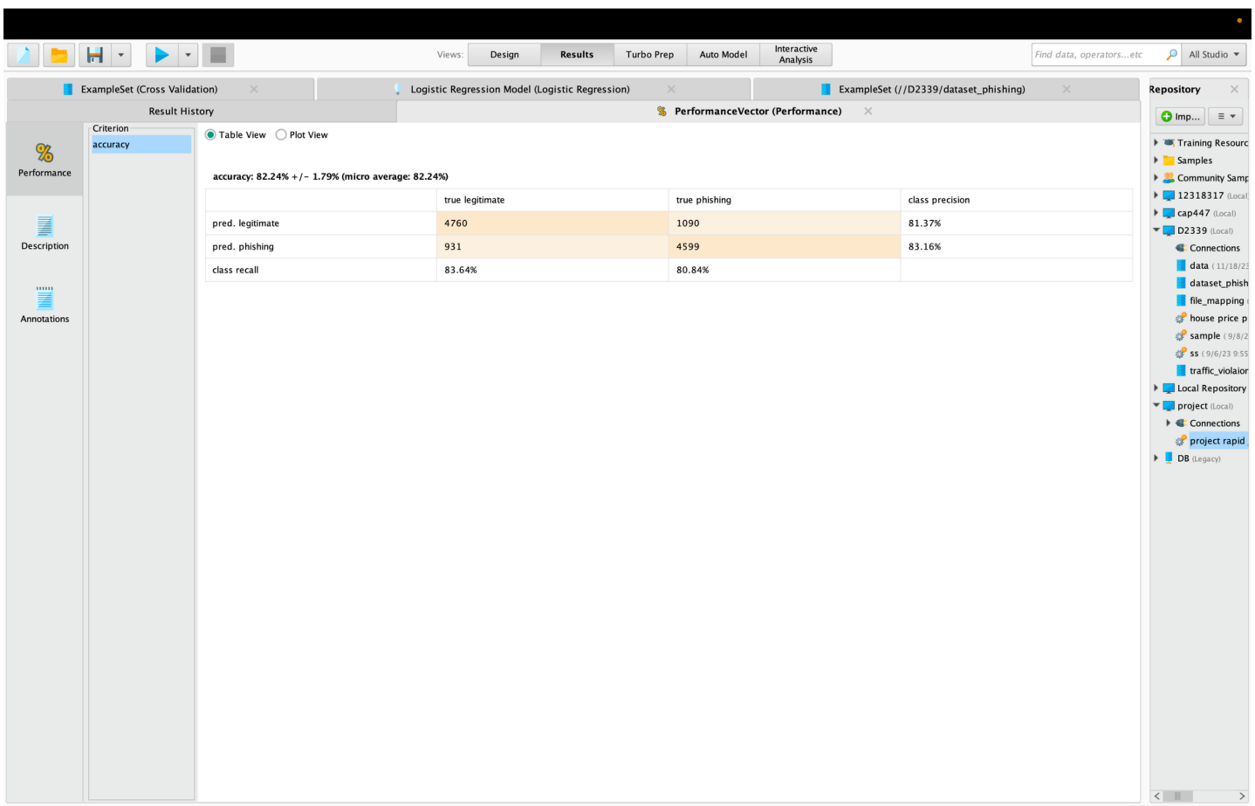


Logistic Regression:-



Warning: Removed collinear columns [avg_words_raw.true, avg_word_host.true, domain_age.true]

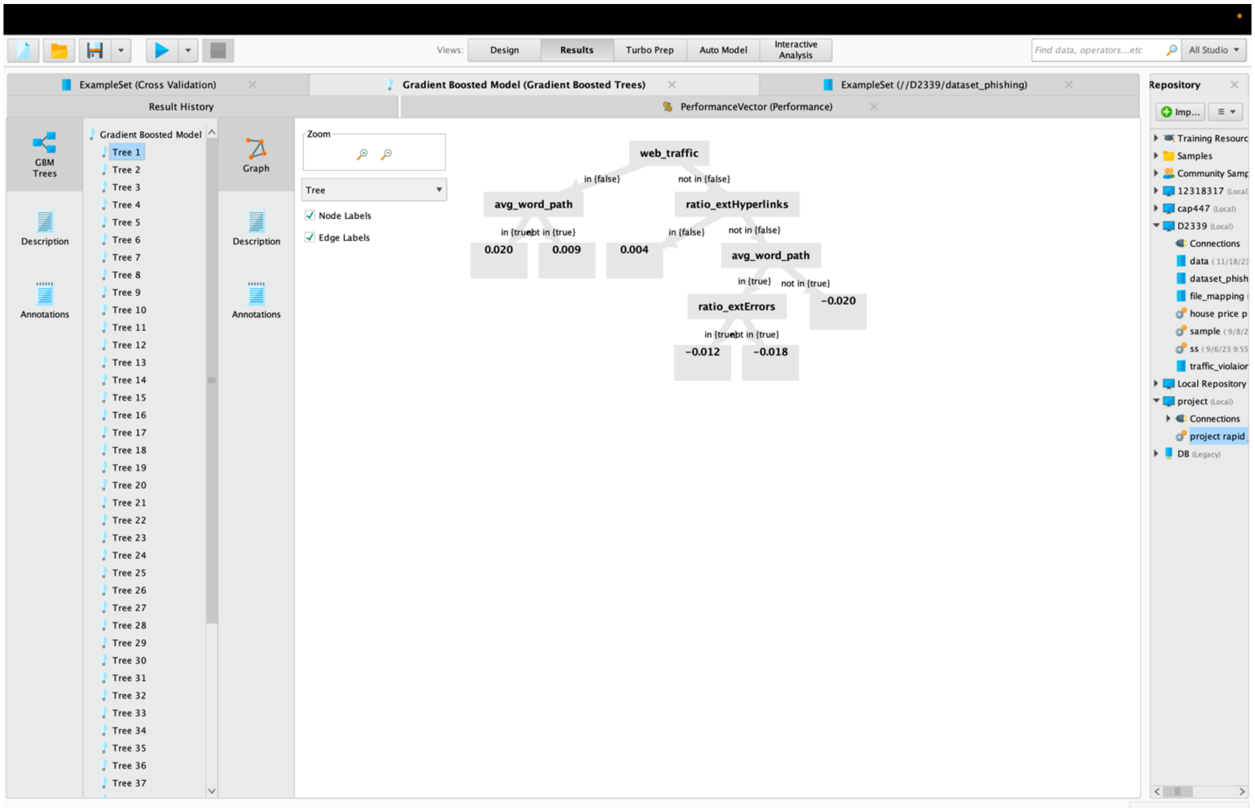
Attribute	Coefficient	Std. Coefficient	Std. Error	z-Value	p-Value
ratio_digits_url.true	49.045	49.045	206.001	0.238	0.812
ratio_digits_host.true	-3.574	-3.574	295.263	-0.012	0.990
abnormal_subdomain.true	31.567	31.567	784.631	0.040	0.968
prefix_suffix.true	1.710	1.710	259.833	0.007	0.995
avg_words_raw.true	0	0	?	?	?
avg_word_host.true	0	0	?	?	?
avg_word_path.true	15.944	15.944	261.194	0.061	0.951
phish_hints.true	74.271	74.271	315.239	0.236	0.814
domain_in_brand.true	-18.986	-18.986	290.556	-0.065	0.948
ratio_intHyperlinks.true	-31.603	-31.603	336.894	-0.094	0.925
ratio_extHyperlinks.true	17.835	17.835	321.659	0.055	0.956
ratio_nullHyperlinks.true	0	0	?	?	?
ratio_intRedirection.true	0	0	?	?	?
ratio_extRedirection.true	-2.046	-2.046	253.309	-0.008	0.994
ratio_intErrors.true	0	0	?	?	?
ratio_extErrors.true	30.857	30.857	238.733	0.129	0.897
links_in_tags.true	5.522	5.522	332.422	0.017	0.987
ratio_intMedia.true	0.898	0.898	243.774	0.004	0.997
ratio_extMedia.true	-24.295	-24.295	206.268	-0.118	0.906
popup_window.true	0	0	?	?	?
domain_in_title.true	3.565	3.565	243.610	0.015	0.988
domain_with_copyright.true	-26.969	-26.969	191.709	-0.141	0.888
domain_registration_length.true	-49.082	-49.082	274.470	-0.179	0.858
domain_age.true	0	0	?	?	?
web_traffic.true	-65.763	-65.763	262.257	-0.251	0.802
page_rank.true	-48.313	-48.313	282.670	-0.171	0.864
Intercept	110.704	110.704	450.072	0.246	0.806

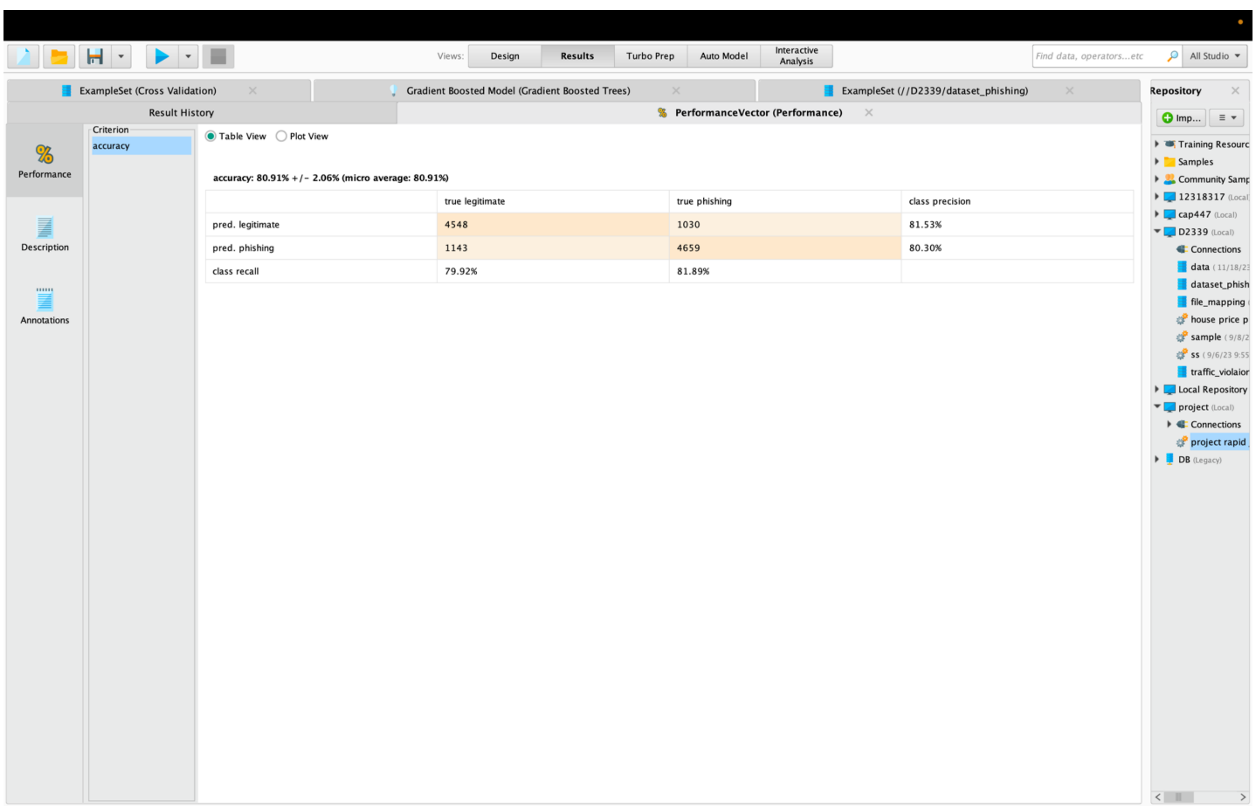


accuracy: 82.24% +/- 1.79% (micro average: 82.24%)

	true legitimate	true phishing	class precision
pred. legitimate	4760	1090	81.37%
pred. phishing	931	4599	83.16%
class recall	83.64%	80.84%	

Gradient Boosted model:-





Conclusion:-

With its intuitive interface and powerful data mining and machine learning capabilities, RapidMiner provides a valuable tool for developing sophisticated phishing website detection models. By carefully selecting and training machine learning algorithms, rigorously evaluating model performance, and continuously monitoring model effectiveness, researchers and cybersecurity experts can effectively combat phishing websites and protect internet users from potential harm.