Data Analysis with Machine Learning
CISA Cybersecurity Dataset 2022
Classification Algorithms

Jacqueline Ocaña

Machine Learning, MSBA 5505

Professor Rao Mikkilineni, PhD

February 25, 2023

Executive Summary

ABC Company has been assigned by CISA to develop a classification algorithm. The algorithm will identify what type of product is being attacked based on the cvss, complexity, severity, and vector. The data provided by CISA had extensive cleaning done which resulted in 37.8% less rows (data points) and 31.2% less columns (independent variables). All the algorithms used to classify scored below 24.33% accuracy in identifying the product being attacked based on those variables alone. The algorithm with the highest accuracy score of 24.33% was the decision tree algorithm. It was recommended to CISA that they continue testing the model with additional data to improve the results before deciding where to direct additional funds to.

Code used to execute this project can be found on GitHub.

Introduction

Problem Statement

ABC Company, a cybersecurity security company specializing in cybersecurity analysis, has been contracted by CISA, the Cybersecurity and Infrastructure Security Agency of the United States of America, to find a way to predict the type of products that will be attacked based on given inputs. This will allow the agency to determine where to direct more funds to.

CISA has provided a dataset set with in-depth exploration of the security vulnerabilities across the United States from the CISA Known Exploited Vulnerabilities catalog for 2022.

The Dataset

The full data set contains 3,984 rows indicating 3,984 instances of cyber attacks were reported. It is important to note that not all security vulnerabilities are reported in this data, therefore only being a sample. The dataset contains 16 columns with the following variables:

- **vendor_project:** The name of the vendor project associated with the vulnerability.
- **product**: The name of the product associated with the vulnerability.
- **vulnerability_name**: The name of the vulnerability that affects a particular product or system.
- **date_added**: The date on which a particular vulnerability was added to CISA's Known Exploited Vulnerabilities catalog.
- **short_description**: A brief description of what makes that specific vulnerability dangerous/risky in nature.
- **required_action**: A clear outline or instruction as to how an organization should address/mitigate that particular vulnerability within its own network/infrastructure setup.
- **due_date**: The date by which required action should be completed at minimum if any mitigation measures are taken against those weaknesses discovered from this analysis report .
- **notes**: A brief text field related additional insights about potential further risks and concerns related each discovered vulnerabilities
- **CVSS Score and Severity assessments:** CISA calculates these two indicators (Common Vulnerability Scoring System (CVSS) score & Severity) based on certain parameters by giving weighted marks in order understand it visibility regarding type & magnitude (likely impact level vislly clearly indicating red amber gn & green status } as per internal computation). Ratings typically range from 0 - 10 indicating descending order i e; 0 being Very Low' 10 - Extremely High severity or criticality

Code used to execute this project can be found on GitHub.

The Path Towards a Solution

Upon inspection of the provided data, the approach that was taken to solve CISA's problem was through applying a classification algorithm using machine learning.

Cleaning the Dataset

The data was processed using Python as the coding language and Juptyer Notebook as the integrated development environment. The packages and libraries used were:

- Scikit-learn
- Pandas
- Numpy
- One hot encoder
- Model selection
- Train_test_split
- DecisionTreeClassifier
- Accuracy_score
- Tree
- Logistic Regression
- KNeighborsClassifier
- RandomForestClassifier
- MultinomialNB

The dataset received was in raw format, indicating that the data needed to be cleaned. The dataset contained multiple null cells. In order to proceed with the analysis, all rows and columns with null variables were dropped from the original dataset. The final, cleaned dataset resulted in 2,480 rows and 11 columns. This is a drop of 1,504 rows and 5 columns resulting in 37.8% less rows and 31.2% less columns.

The data also contained a variety of data types such as strings (words) and floats (numbers). Algorithms are unable to work with strings, so the product column data was converted using one hot encoding. This resulted in all the strings being converted to representative numbers. There were 308 different products identified.

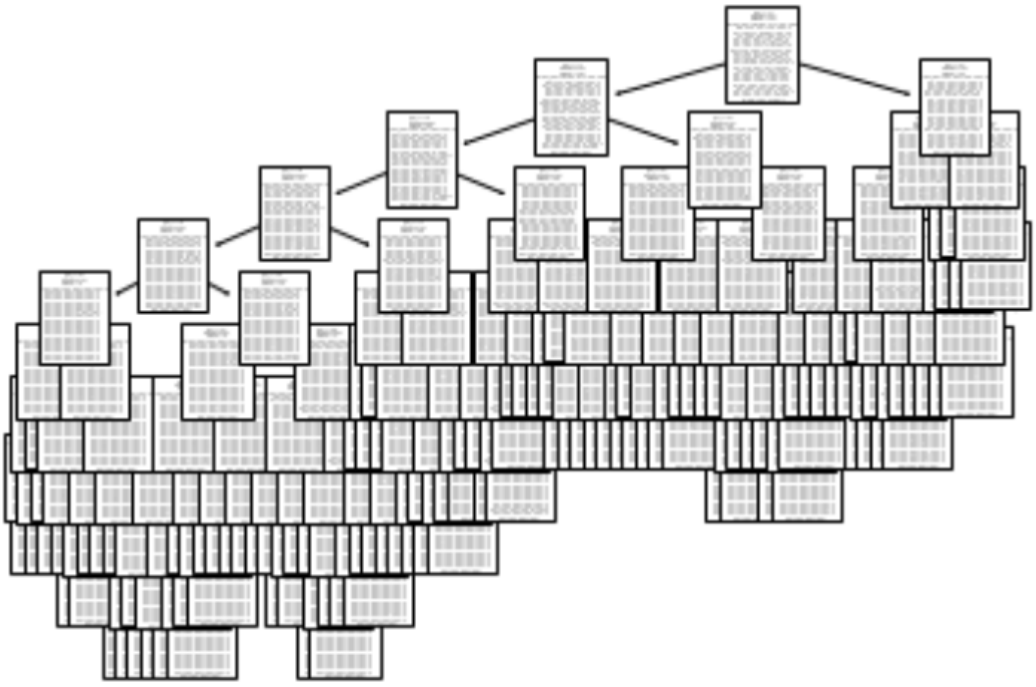
Selecting a Classification Algorithm

As the problem has more than two types of classification types, the linear regression model was ruled out. The following models were tested:

- Decision Tree
- Logistic Regression
- K-Nearest Neighbor
- Random Forest
- Naive Bayes

Code used to execute this project can be found on GitHub.

The Winner: Decision Tree



Decision trees start with the nodes at the top and gradually work all the way down, leaving only one leaf or classification. Decision trees work well when with more than two classifications, but can be costly to run over time. The idea behind a decision tree is to take a set of information and further identify specific features, ultimately leading to the final classification.

In the case for ABC Company, a decision tree model has been applied because there is only one model available. A random forest would require more data and models to be built out.

The image above is a representation of what the decision tree looks like for the CISA data set. This model was accurate in predicting the product classification 24.33% of the time.

While logistic regression, k-nearest neighbor, random forest, naive bayes models were tested, their accuracy was 14.38%, 14.78%, 24.33%, and 11.16% respectively. The random forest model received the same accuracy score as the decision tree, but that is because a random forest takes several decision trees and groups them all together to output the classification. In this case, only one decision tree was provided, therefore giving the same score.

Code used to execute this project can be found on GitHub.

Limitations

Some limitations to this study include:

- Accuracy scores in the tested models were too low to have confident results.
- Need to keep training the model to improve classification accuracy.
- Loss of data upon cleaning the dataset impacted the accuracy scores of the models.

Conclusions

Given the models tested, ABC Company recommends CISA to use the decision tree modeling algorithm. However, because of its low accuracy score of 24.33%, it is recommended that the model is further tested with more data before moving forward with this approach. Upon successful application of these recommendations, the agency can use the algorithm to classify incoming product attacks, track the rate of each type of attack, and direct funds to protect products that are disproportionately affected.

Code used to execute this project can be found on GitHub.

References

Cybersecurity risk (2022 CISA vulnerability) (no date) Kaggle. Available at: https://www.kaggle.com/datasets/thedevastator/exploring-cybersecurity-risk-via-2022-cisa-vulne (Accessed: February 23, 2023).

Galarnyk, M. (2022) Visualizing Decision Trees with python (scikit-learn, Graphviz, matplotlib), Medium. Towards Data Science. Available at: https://towardsdatascience.com/visualizing-decision-trees-with-python-scikit-learn-graphviz-matplotlib-1c50b4aa68dc (Accessed: February 23, 2023).

Introduction to decision trees: Why should you use them? (2021) 365 Data Science. Available at: https://365datascience.com/tutorials/machine-learning-tutorials/decision-trees/ (Accessed: February 23, 2023).

Learn (no date) scikit. Available at: https://scikit-learn.org/stable/index.html (Accessed: February 23, 2023).

What is Random Forest? (no date) IBM. Available at: https://www.ibm.com/topics/random-forest (Accessed: February 23, 2023).

Code used to execute this project can be found on GitHub.