Text Mining and Classification in Defining Cybersecurity Education Requirements

Caitlyn Boyd, Noah Manuel, James Crabb

INTRODUCTION

- What is text mining? Text mining is a type of data mining specifically focused on unstructured text. Using natural language processing, text mining extracts useful information from the unstructured data.
- The goal of text mining is to convert unstructured data into structured data for further analysis.
- How does this apply to cybersecurity? In order to evaluate the effectiveness of cybersecuirty education, we must be able to compare it to real world requirements.
- The problem with comparing cybersecurity course outcomes and job requirements is that they are formatted in unstructured text. Often, course outcomes and job requirements are simple listed statements.
- To order these statements for analysis, we need to be able to classify them into structured data for comparison via text mining and different classifiers.
- Our project focuses on finding the text mining model that most accurately classifies learning outcomes and job requirements in the cybersecurity field with a goal of organizing the data for further analysis.

REVISED BLOOM'S TAXONOMY

- A classification of cognitive skills developed by Benjamin Bloom, Max Englehat, Edward Furst, Walter Hill, and David Krathwohl in 1956.
- The Taxonomy was later revised and updated by Anderson and Krathwohl in 2001
- This revised taxonomy includes six categories:
  - Create: Combining and reorganizing elements into a new pattern through generation, planning or production.
  - Evaluate: Making judgements based on standards by checking and critiquing.
  - Analyze: Determining how components of a subject relate to each other and their overall purpose or structure.
  - Apply: Using a procedure through implementation or execution.
  - Understand: Construct meaning from given materials.
  - Remember: Recall knowledge from memory.

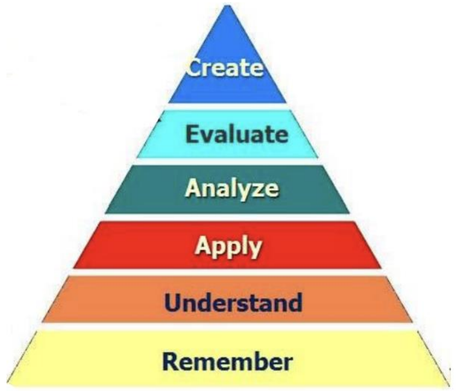

Figure 1: Graphic from “Bloom's Taxonomy Revised." Depicts the pyramid of classifications for cognitive skills as revised by Anderson and Krathwohl (2001).

MACHINE LEARNING ALGORITHMS

- Procedures and techniques that are used to allow computers to learn from a dataset in order to identify patterns, perform tasks, and make predictions without explicit instruction.
- The machine learning models used in this project come from the *scikit-learn* library for machine learning in the programming language Python.
- **Multinomial Naïve Bayes Classifier:** A machine learning algorithm based on Bayes' Theorem.
  - Bayes Theorem calculates the conditional probability of an event based on previous results with similar parameters and their outcomes
- **Random Forest Classifier:** A machine learning model that uses the majority decision of multiple decision trees in the classifier.

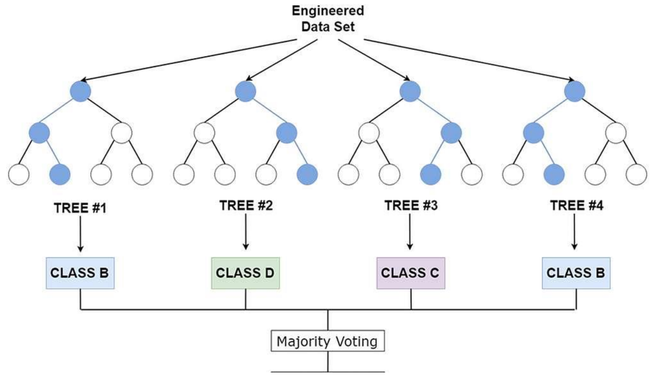

Figure 2: Graphic depicting the structure of a random forest classifier.

EXPERIMENTAL SETUP

- Our experiment started by creating a training dataset of categorized learning outcomes and job requirements in cybersecurity.
- To achieve this, we used Anderson and Krathwohl's revised Blooms Taxonomy labels and applied them to the *Labels* column of a .csv file containing the learning outcomes and job requirements.
- Using this labeled dataset we hyperparameter tuned a Random Forest Classifier and a Multinomial Naïve Bayes Classifier using the grid search methods in the *scikit-learn* library.
- Using the results of the hyperparameter tuning, we trained the machine learning models to classify the unlabeled learning outcomes and used a ROC AUC score to determine which model was more accurate.
- **Hyperparameter**: parameters for which the values control the learning process and parameter values of a machine learning algorithm.
- **Hyperparameter Tuning:** The process of trying different combinations of parameter values and evaluating their results.
- **ROC AUC Score**: Receiver Operating Characteristic Area Under the Curve. The score summarizes the performance of a classifier across all thresholds. A score of 0.5 indicates random classification while a score of 1.0 indicates perfect classification.

RESULTS

- **Summary:**
  - Random Forest Classifier had a better accuracy value compared to Multinomial Naïve Bayes where Random Forest averaged around 0.90 ROC_AUC_SCORE compared to 0.83 for Multinomial Naïve Bayes.
  - Multinomial Naïve Bayes has a faster average compile and classification time than Random Forest with the Bayes taking 0.02 seconds for fitting with its best parameters while the Random Forest Classifier took 0.6 seconds with its best parameters.
- **Multinomial Naïve Bayes** - Naïve Bayes has a shorter compilation time than Random Forest at the cost of its accuracy. The hyperparameters used, 'alpha' and 'fit_prior', changed the behavior on how text was classified. The best settings found was an 'alpha' value of 1.0 and a 'fit_prior' status of true.
- **Random Forest Classifier –** The accuracy of a Random Forest Classifier was found to be the best using the criterion 'gini' and having a large amount of decision trees in the forest. However, with a larger number of trees, the time to classify the text increases as more trees need to analyze the text and make a decision.

REFERENCES

- Hayes, A. (2023, August 10). *Bayes’ Theorem*. Investopedia
- PICRYL - Public Domain Media Search Engine. (2020, December 21). *Random Forest,* PICRYL.
- Wilson, L. O. (2020, July 7). *Bloom’s taxonomy revised*.

ACKNOWLEGEMENTS

This work is supported by funding for the VICEROY Northwest Institute for Cybersecurity Education and Research (CySER) provided by The Office of the Undersecretary of Defense for Research and Engineering, in collaboration with the Air Force Research Laboratory and Griffiss Institute.