Data Analysis Report: CVSS Unsupervised Learning

Course: COMP2420/COMP6420 - Introduction to Data Management, Analysis and Security
Assignment: Assignment 2 (2022)

1. Introduction

This report details the unsupervised learning tasks performed on the Common Vulnerability Scoring System (CVSS) dataset, as part of Assignment 2 for COMP2420/COMP6420. The primary objective is to develop a clustering model that groups software vulnerabilities based on their CVSS metrics, excluding Base Scores, Sub Scores, and Base Severity. This model will assist a cybersecurity procurement team in making risk-based decisions regarding the introduction of new products into their systems.
The analysis is structured into two main sections: data preprocessing and the implementation and justification of the clustering model.

2. Data Preprocessing (1.1)

The CVSS dataset, CVSS_data_complete.csv, was imported for this analysis. The initial dataset contained 9210 entries.

2.1 Missing Value Handling

Upon loading, the dataset was inspected for missing values. All rows containing NaN (Not a Number) values were dropped. After this operation, the dataset retained 9210 entries, indicating that there were no rows with missing values that needed to be handled.

2.2 Feature Engineering and Recoding

To prepare the data for clustering, several categorical CVSSv3.1 metrics were recoded into numerical values based on Section 7.4 (Metric Values) of the CVSSv3.1 specification document. This transformation is crucial for applying numerical clustering algorithms like K-Means. The following new columns were added, using short codes according to CVSS acronyms:

- v3_AV: Attack Vector (PHYSICAL: 0.2, LOCAL: 0.55, ADJACENT_NETWORK: 0.62, NETWORK: 0.85)
- v3_AC: Attack Complexity (LOW: 0.77, HIGH: 0.44)

- v3_UI: User Interaction (NONE: 0.85, REQUIRED: 0.62)
- v3_S: Scope (UNCHANGED: 0, CHANGED: 1)
- v3_PR: Privileges Required (NONE: 0.85, LOW: 0.62 (unchanged scope) / 0.68 (changed scope), HIGH: 0.27 (unchanged scope) / 0.50 (changed scope))
- v3_A: Availability Impact (NONE: 0, LOW: 0.22, HIGH: 0.56)
- v3_I: Integrity Impact (NONE: 0, LOW: 0.22, HIGH: 0.56)
- v3_C: Confidentiality Impact (NONE: 0, LOW: 0.22, HIGH: 0.56)

The unique values for these newly created numerical columns were verified to ensure the recoding was successful and consistent with the CVSS specification.

The columns used for clustering were: v3_AV, v3_AC, v3_UI, v3_S, v3_PR, v3_A, v3_I, and v3_C. These features represent the core characteristics of a vulnerability's exploitability and impact, aligning with the procurement team's request to exclude Base Scores, Sub Scores, and Base Severity.

2.3 Data Scaling

Before applying the K-Means algorithm, the selected features were scaled using StandardScaler. This step is essential to ensure that all features contribute equally to the distance calculations in the clustering algorithm, preventing features with larger numerical ranges from dominating the clustering process.

2.4 Dimensionality Reduction

To facilitate visualization and potentially improve clustering performance by reducing noise, Principal Component Analysis (PCA) was applied to reduce the scaled data to 2 dimensions. This allows for a clear 2D visualization of the clusters.

3. Building a Clustering Model (1.2)

3.1 K-Means Clustering Implementation

K-Means clustering was chosen for this unsupervised learning task. The algorithm aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean (cluster centers or cluster centroid), serving as a prototype of the cluster.

3.2 Determining the Optimal Number of Clusters (k)

To determine the optimal number of clusters (k), the Elbow Method was employed. This method involves plotting the sum of squared distances (inertia) against the number of clusters. The "elbow" point in the plot, where the rate of decrease in inertia sharply changes,

typically indicates the optimal k.
The elbow plot generated for the 2-dimensional reduced data is shown below:
[Insert Elbow Method Plot Here - as provided in the notebook output]

Based on the elbow plot, a value of **k = 5** was selected as the optimal number of clusters. The plot shows a clear "elbow" around 5 clusters, after which the decrease in the sum of squared distances becomes less significant, suggesting diminishing returns for adding more clusters.

3.3 Clustering Results

With k=5, the K-Means model was trained on the 2-dimensional reduced data. The kmeans.cluster_centers_ output provides the coordinates of the centroids for each of the 5 clusters in the reduced PCA space. The kmeans.inertia_ value of 3760.16 indicates the sum of squared distances of samples to their closest cluster center, serving as a measure of the internal coherence of the clusters. A lower inertia generally indicates better clustering.

3.4 2D Visualization of Clusters

To visually represent the identified clusters, a scatter plot of the 2-dimensional reduced data was generated, with each data point colored according to its assigned cluster. The cluster centroids are also marked on the plot.
[Insert 2D Cluster Visualization Plot Here - as provided in the notebook output]

Verbal Justification of Visualization:
The 2D visualization clearly shows the separation of data points into 5 distinct clusters. While there is some overlap, the clusters generally form discernible groups, supporting the choice of k=5. Each cluster represents a different "severity level" or grouping of vulnerabilities based on their CVSS metrics.

- **Cluster 0 (Purple):** Appears to be a more spread-out cluster, potentially representing vulnerabilities with a wider range of CVSS metric values.
- **Cluster 1 (Blue):** A relatively compact cluster, suggesting a group of vulnerabilities with similar CVSS characteristics.
- **Cluster 2 (Green):** Another distinct cluster, indicating a specific profile of vulnerabilities.
- **Cluster 3 (Red):** A well-defined cluster, possibly representing vulnerabilities with high commonality in their CVSS metrics.
- **Cluster 4 (Orange):** This cluster seems to be positioned distinctly from others, indicating a unique set of vulnerability characteristics.

The visual separation confirms that the K-Means algorithm successfully identified natural groupings within the CVSS data based on the chosen metrics. The procurement team can now analyze the characteristics of each cluster (e.g., by examining the mean values of the original CVSS metrics for each cluster) to understand the risk profile associated with each group of software vulnerabilities. This allows them to make informed, risk-based decisions about product introduction.

4. Conclusion

This unsupervised learning analysis successfully identified five distinct clusters of software vulnerabilities within the CVSS dataset, based on their CVSS metrics (excluding Base Scores, Sub Scores, and Base Severity). The preprocessing steps, including handling missing values, recoding categorical features, scaling, and dimensionality reduction, prepared the data effectively for clustering. The Elbow Method provided a data-driven justification for selecting 5 clusters, and the 2D visualization confirmed the meaningful separation of these clusters. The identified clusters can serve as "Severity Levels" for the cybersecurity procurement team, enabling them to categorize and assess the risk associated with different software products based on their underlying vulnerabilities. This approach provides a structured framework for risk-based decision-making in procurement.