EAS 509LEC LK2: Statistical Learning II

PROJECT 1

Land Mine Detection and Classification: A Comprehensive Data-Driven Analysis

Team members:

Tharun Teja Mogili	UB Person Number: 50559877)
Prathibha Vuyyala	(UB Person Number: 50559983)
Manasa Lakshmi Gunampalli	(UB Person Number: 50559593)
Divyakanth Reddy Buchupalle	(UB Person Number: 50559937)

Introduction:

Land mines continue to pose significant risks to human life and infrastructure in post-conflict zones around the world. The detection and classification of buried land mines remain critical challenges due to their varied physical and material properties, as well as the diversity of environmental conditions such as soil types and moisture levels. Traditional methods for land mine detection, while effective in certain scenarios, often fail to achieve the precision required for complete accuracy. This limitation highlights the need for advanced data-driven approaches that can leverage sensor data to enhance detection and classification capabilities.

This project focuses on analyzing and extracting insights from a dataset collected using magnetic anomaly sensors. These sensors measure distortions in the magnetic field caused by metallic objects buried underground, making them a valuable tool for land mine detection. The dataset comprises four key variables:

- **Voltage (V):** Represents the sensor's output due to magnetic disturbances.
- **Height (H):** The distance of the sensor from the ground, influencing its sensitivity to buried objects.

- **Soil Type (S):** A categorical feature indicating six distinct types of soil based on texture and moisture.
- **Mine Type (M):** The target variable representing five commonly encountered classes of land mines.

The primary objectives of this project are:

1. To explore the data and understand the underlying patterns and relationships among features.
2. To identify natural groupings within the data using clustering techniques.
3. To classify mine types accurately using supervised machine learning models.
4. To evaluate the effectiveness of various algorithms and provide actionable insights for improving detection accuracy.

By employing techniques such as exploratory data analysis (EDA), clustering, and classification, this project aims to demonstrate how advanced analytical methods can be applied to real-world challenges in engineering and safety. In addition, principal component analysis (PCA) is utilized to reduce dimensionality and enhance the interpretability of patterns within the dataset.

Through this analysis, we aim to provide a robust framework for leveraging sensor data in land mine detection, contributing to the ongoing efforts to develop safer and more reliable detection systems. The findings and methodologies presented here can also serve as a reference for similar projects in data analysis and classification.

Dataset Overview:

Data set Link: https://archive.ics.uci.edu/datasets?search=Land%20Mines

The dataset used in this project is derived from magnetic anomaly sensors designed to detect buried land mines by measuring distortions in the magnetic field caused by metallic objects. This dataset provides essential features that capture the physical and environmental factors affecting land mine detection. The key characteristics of the dataset are outlined below:

Dataset Summary:

- **Number of Instances:** 338
- **Number of Features:** 4 (3 independent variables and 1 target variable)
- **Feature Types:**

    1. Continuous: Voltage (V), Height (H)
    2. Categorical: Soil Type (S), Mine Type (M)

- **Target Variable:** Mine Type (M) with five distinct classes representing common types of land mines.

Feature Descriptions:

1. **Voltage (V):**
    - A continuous variable representing the output voltage of the sensor due to magnetic distortions caused by buried objects. It reflects the sensor's sensitivity to metallic disturbances.
2. **Height (H):**
    - A continuous variable indicating the height of the sensor from the ground in centimeters. This feature directly impacts the sensor's ability to detect mines at varying depths.
3. **Soil Type (S):**
    - A categorical variable representing six distinct soil types, determined by their texture and moisture levels. Soil types are encoded as follows:
        - 0.0: Dry + Sandy
        - 0.2: Dry + Humus
        - 0.4: Dry + Limey
        - 0.6: Humid + Sandy
        - 0.8: Humid + Humus
        - 1.0: Humid + Limey
4. **Mine Type (M):**
    - The target variable, a categorical feature with five classes representing different types of land mines:
        - Class 1: Null
        - Class 2: Anti-Tank
        - Class 3: Anti-Personnel
        - Class 4: Booby-Trapped Anti-Personnel
        - Class 5: M14 Anti-Personnel

Key Characteristics:

- **No Missing Values:** The dataset is complete, with no null values across all features.
- **No Duplicates:** There are no duplicate rows in the dataset, ensuring data integrity.

- **Balanced Representation of Soil Types:** The six soil types are evenly distributed, ensuring fair modeling.
- **Imbalanced Target Variable:** While soil types are balanced, the mine types show slight imbalance, which may affect model performance.

Purpose of the Dataset:

This dataset was originally created for a Ph.D. thesis focused on developing a hybrid detection model for land mines using magnetic anomaly methods. The data has been preprocessed to handle any missing values, and its multivariate nature makes it suitable for tasks like clustering, classification, and dimensionality reduction.

Data Context:

The dataset highlights the challenges associated with mine detection, such as:

- The variability in sensor output based on environmental factors (soil type, height).
- The complex relationship between physical measurements and mine classification.

By analyzing this dataset, we aim to extract meaningful insights and develop models capable of improving the accuracy of land mine detection systems.

Preprocessing:

Preprocessing is a critical step in any data analysis pipeline. It ensures that the dataset is clean, consistent, and formatted for effective application of machine learning algorithms. In this project, preprocessing involved checking for missing values, eliminating duplicates, scaling features and preparing the data for dimensionality reduction and modeling.

Data Cleaning:

1. **Checking for Missing Values:**

    - The dataset was checked for missing values using the is.na() function in R.
    - **Result:** No missing values were found, confirming that the dataset is complete.

2. **Checking for Duplicates:**

    - Duplicate rows were identified using the duplicated() function.
    - **Result:** No duplicate rows were found, ensuring data integrity.

Data Inspection:

1. **Checking Data Types:**

    - The structure of the dataset was inspected using the str() function to verify the data types of each feature.
    - **Result:**

        - Voltage (V), Height (H), and Soil Type (S) were confirmed as numerical variables.
        - Mine Type (M) was confirmed as an integer representing categorical labels for mine types.

2. **Summary Statistics:**

    - Using the summary() function, the dataset was examined to understand the ranges, mean, median, and variability of each feature.
    - **Insights:**

        Voltage and Height displayed varying ranges, indicating sensor variability.

        Soil Type was evenly distributed across six categories, ensuring no bias in representation.

Feature Scaling:

- **Why Scaling is Necessary:**

    Features like Voltage (V) and Height (H) operate on different scales, which can disproportionately affect distance-based models (e.g., clustering) and machine learning algorithms.

- **Scaling Method:**

    Z-score standardization was applied to numerical features (V, H, and S):

$$Z = \frac{(X - \text{mean})}{\text{standard deviation}}$$

- **Result:**

    After scaling:

- All numerical features were transformed to have a mean of 0 and a standard deviation of 1.
- This ensured equal contribution of features during analysis.

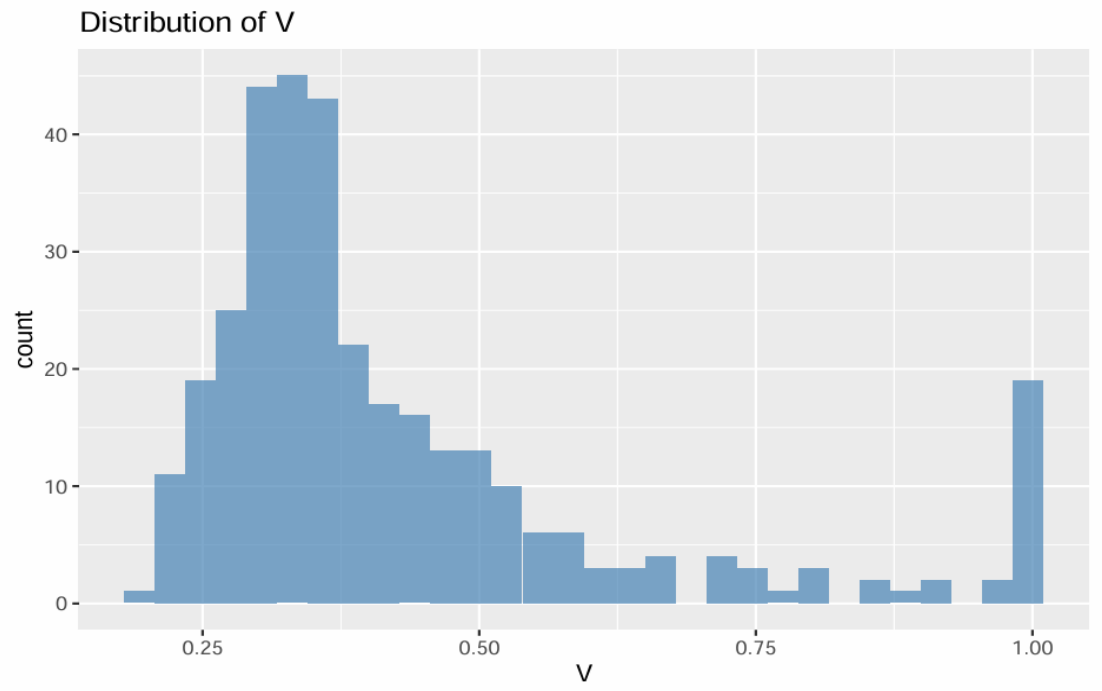
Exploratory Data Analysis (EDA):

Exploratory Data Analysis (EDA) is a crucial step in understanding the structure, patterns, and relationships within the dataset. This section outlines the visualizations and statistical summaries performed to gain insights into the data.

Distribution of Features:

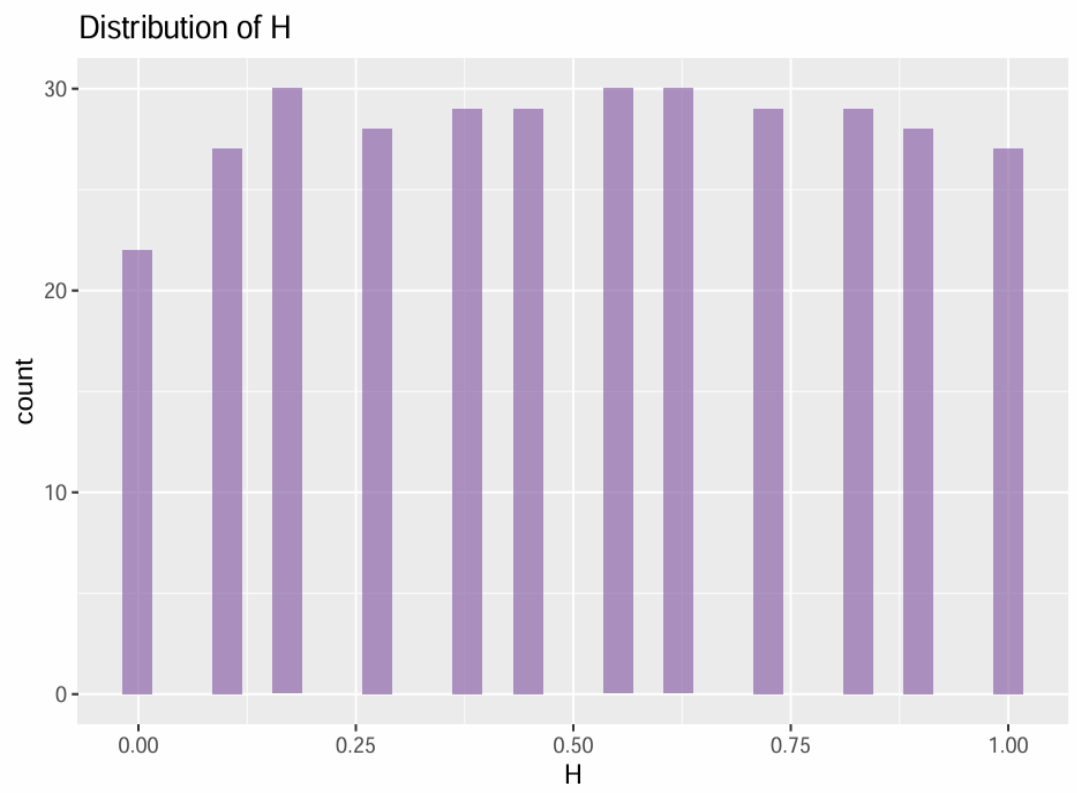
1. **Voltage (V):**
    - **Visualization:** A histogram was plotted for Voltage (V) to analyze its distribution.
    - **Observations:**
        - The distribution is slightly skewed to the right, with most values concentrated between 0.2 and 0.5.
        - This suggests that the majority of sensor outputs are within a consistent range under normal conditions.
    - **Interpretation:** The variability in Voltage indicates that different types of mines or soil conditions may impact the sensor readings.

Visualization:



2. **Height (H):**
    - **Visualization:** A histogram was used to examine the distribution of Height (H).
    - **Observations:**
        - The data showed a uniform-like distribution, with slight peaks at standard operational sensor heights.
    - **Interpretation:** The height variations are well-distributed, ensuring that the dataset captures scenarios from various sensor positions.

Visualization:



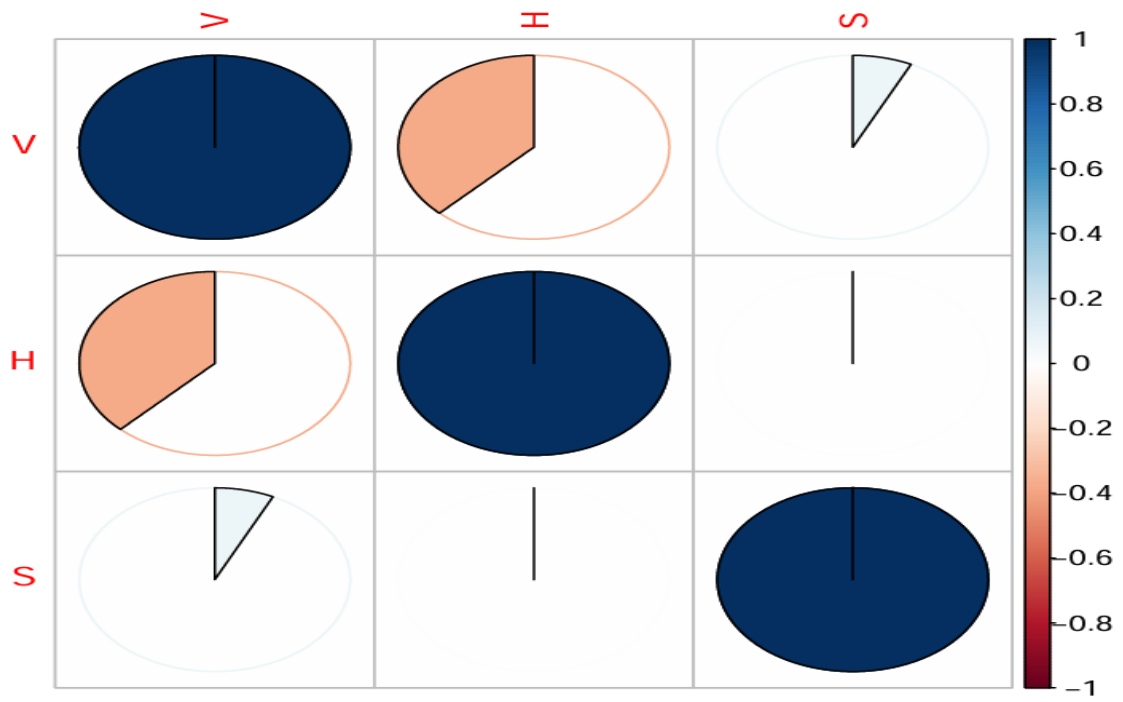
Correlation Analysis:

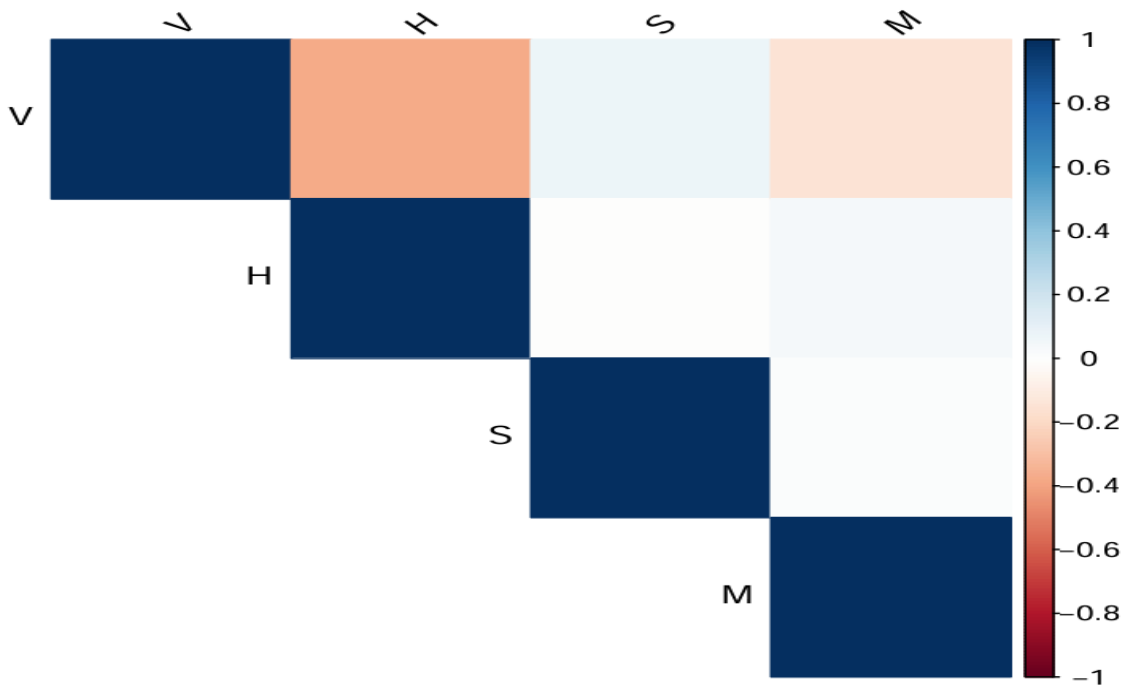
Purpose: To identify relationships between the numerical features (Voltage, Height, and Soil Type).

- **Method:** A correlation matrix was computed, and a corrplot was used to visualize the relationships.
- **Observations:**
    - Weak correlations were observed between the features (V, H, and S).
    - This indicates that the features independently contribute to the detection and classification tasks.

Interpretation: The independence of features suggests that each one carries unique information, which can be useful for clustering and classification.

Visualization:

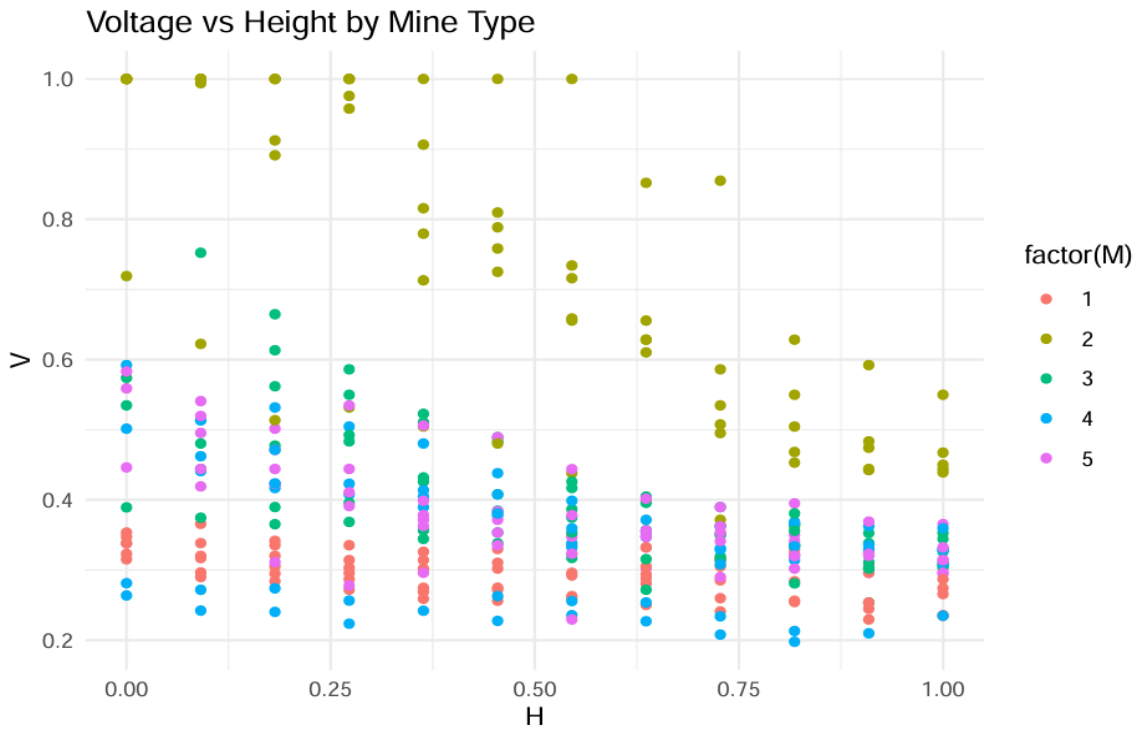




Feature Relationships:

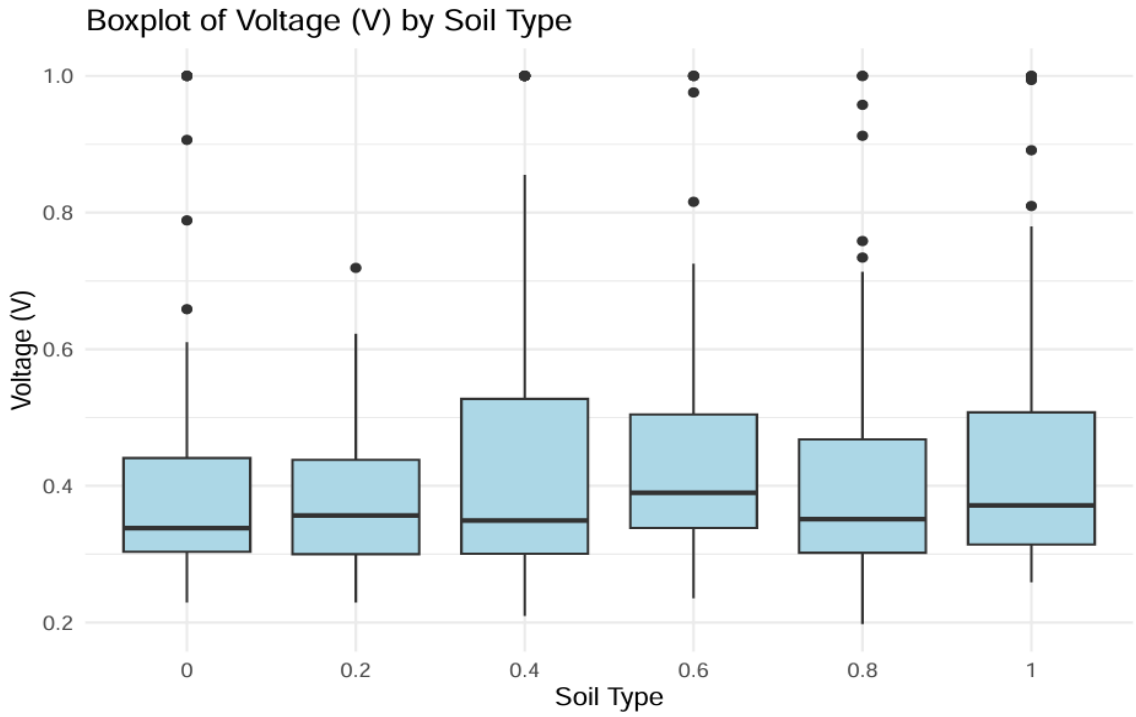
1. **Voltage vs. Height by Mine Type:**
    - **Visualization:** A scatter plot was created, with points colored by Mine Type (M).
    - **Observations:**
        - Data points showed some clustering patterns, potentially aligning with specific mine types or soil conditions.
    - **Interpretation:** The clustering hints at separability between mine types based on Voltage and Height.

    Visualization:



2. **Voltage Distribution by Soil Type:**
    - **Visualization:** A box plot was created to visualize Voltage (V) distributions across Soil Types (S).
    - **Observations:**
        - Voltage distributions varied across soil types, with distinct medians and interquartile ranges for each category.
    - **Interpretation:** Soil type impacts sensor readings, which could be a key feature in clustering and classification.

Visualization:



Principal Component Analysis (PCA):

Principal Component Analysis (PCA) is a dimensionality reduction technique used to project high-dimensional data into a lower-dimensional space while retaining the most important information. In this project, PCA was applied after Exploratory Data Analysis to simplify the dataset, improve computational efficiency, and enhance the visualization of patterns.

Purpose of PCA:

- **Dimensionality Reduction:** Reduce the dataset from three numerical features (Voltage, Height, Soil Type) to fewer components while retaining most of the variance.
- **Pattern Recognition:** Uncover hidden relationships and groupings in the data that may not be apparent in the original feature space.
- **Visualization:** Enable 2D and 3D visualizations of the data to identify potential clusters and separability between mine types and soil types.

Methodology:

1. **Data Standardization:**

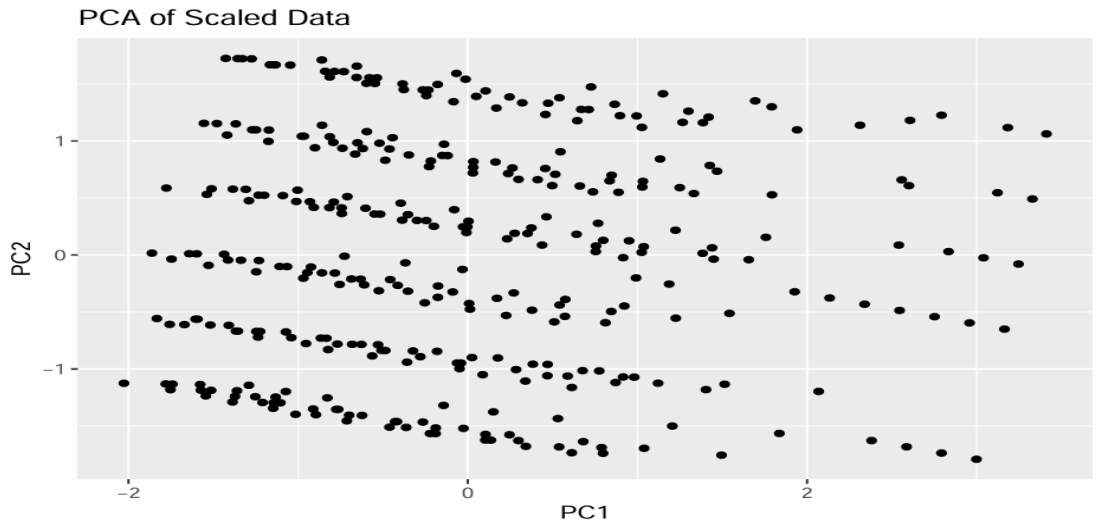
- Before applying PCA, the features were standardized to have a mean of 0 and a standard deviation of 1. This ensures that all features contribute equally to the principal components.

2. **Computation of Principal Components:**
   - PCA was performed, and the eigenvalues and eigenvectors of the covariance matrix were calculated to determine the principal components.
   - The principal components are linear combinations of the original features and are ranked based on the variance they explain.
3. **Variance Explained:**
   - The proportion of variance explained by each principal component was computed.
   - **Result:** The first two principal components explained **79.4%** of the total variance:
     - PC1: 46.2%
     - PC2: 33.2%
     - PC3: 20.6%

Visualizations:

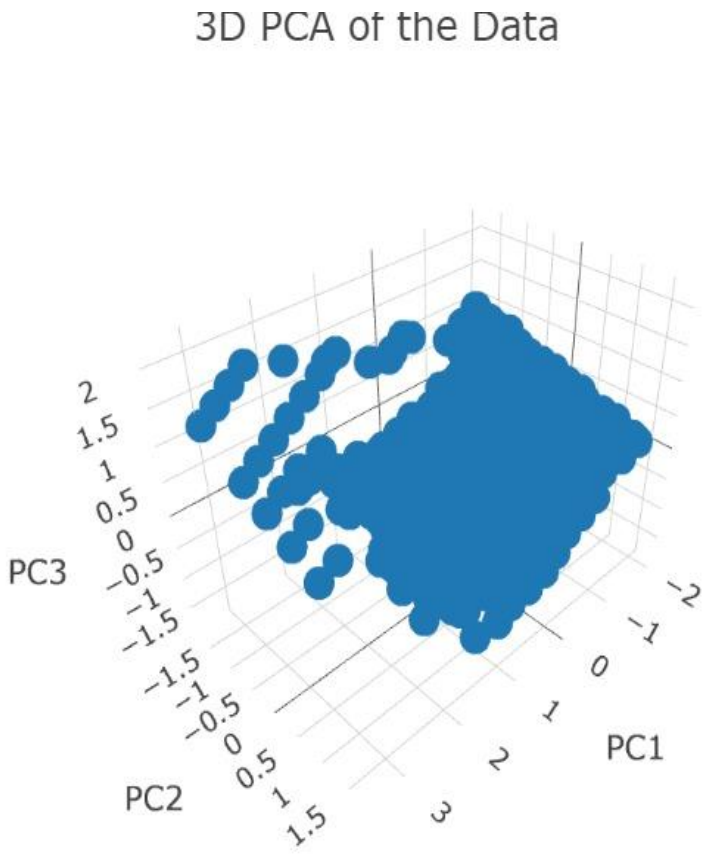
1. **2D PCA Scatter Plot:**
   - A scatter plot of the first two principal components was created to visualize data clusters.
   - **Observations:**
     - Distinct clusters were observed, potentially aligning with mine and soil types.
     - Overlap in some regions suggested that certain mine types may share similar characteristics or be influenced by similar soil conditions.
   - **Interpretation:** PCA in 2D provided a clearer understanding of the separability among classes, which aids in clustering and classification.

Visualization:



2. **3D PCA Scatter Plot:**
    - A 3D scatter plot was created using the first three principal components to capture more variance and provide a deeper view of the data structure.
    - **Observations:**
        - Additional separability was noted in the third dimension, further differentiating certain clusters.
        - This enhanced visualization highlighted outliers and reinforced the clusters observed in 2D.
    - **Interpretation:** The 3D PCA plot added depth to the clustering analysis, making it a valuable tool for exploring data groupings.

Visualization:



Key Insights from PCA:

1. **Dimensionality Reduction Success:**
    - PCA effectively reduced the dataset from three features to two or three components while preserving nearly 80% of the variance.
    - This simplification is computationally efficient and aids in subsequent clustering and modeling steps.

2. **Cluster Insights:**
    - The PCA plots revealed natural groupings, indicating that Voltage, Height, and Soil Type collectively influence the separability of mine and soil types.
    - Clusters were evident, aligning with certain soil types and mine types, but some overlap suggested the need for more advanced modeling to achieve better classification.

Clustering:

Clustering is the cornerstone of this project, as it allows us to uncover hidden groupings in the data and analyze the relationships between features like Voltage, Height, and Soil Type. Clustering is particularly valuable in the context of land mine detection, where the goal is to group similar objects (e.g., mines with shared characteristics) without relying on pre-labeled data. In this project, we used **K-Means Clustering** and **Hierarchical Clustering** to explore the dataset and derive meaningful clusters.

K-Means Clustering:

Concept:
K-Means is a distance-based unsupervised learning algorithm that partitions the dataset into k clusters. It works by:

1. Randomly initializing k centroids.
2. Assigning each data point to the nearest centroid based on Euclidean distance.
3. Recomputing the centroids as the mean of the assigned data points.
4. Repeating steps 2 and 3 until the centroids stabilize (convergence).

Determining the Optimal Number of Clusters (k):

The choice of k is critical to ensure meaningful and interpretable clusters. Three methods were used to identify the optimal k:

1. Elbow Method

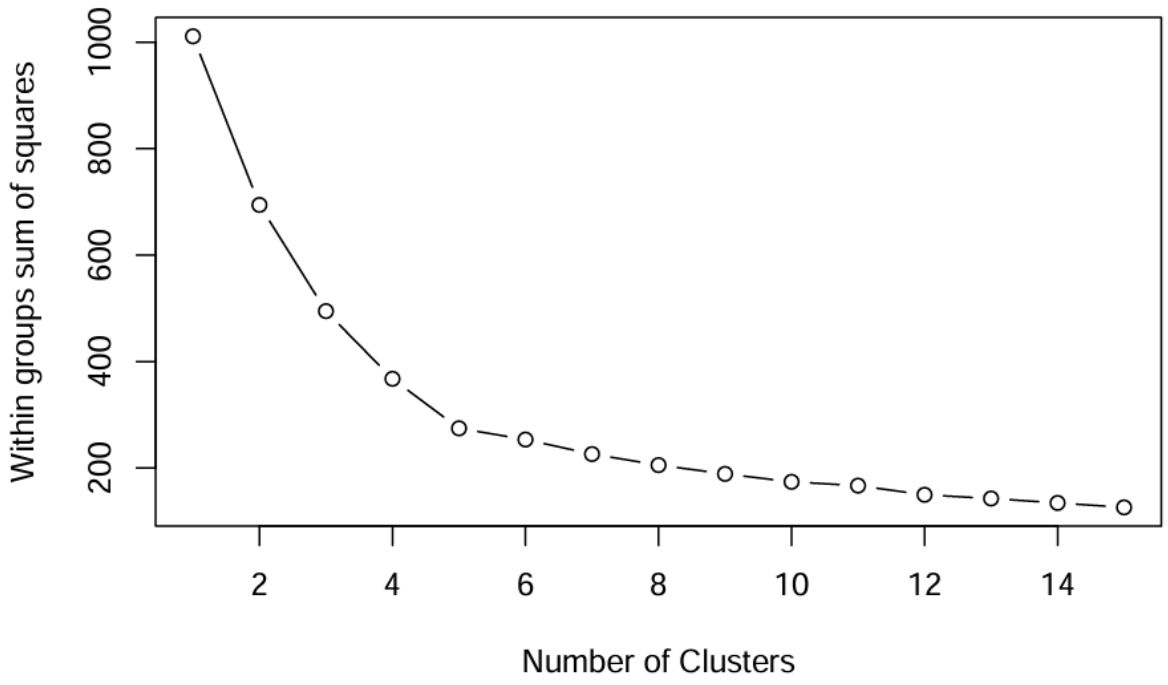
The Elbow Method is a visual technique that evaluates the **Within-Cluster Sum of Squares (WCSS)** for different values of k. WCSS measures the total variance within clusters, where lower values indicate tighter, more compact clusters.

- **How it Works:**
    1. Calculate WCSS for a range of k values (e.g., 1 to 10).
    2. Plot k against WCSS.

3. Identify the "elbow point" where the WCSS curve starts to flatten. This point represents the optimal trade-off between the number of clusters and the compactness of the clustering.

- **Result in This Project:**
  - The elbow point was observed at k=5.
  - Adding more clusters beyond k=5 yielded only marginal reductions in WCSS.

Visualization:



Interpretation: The Elbow Method identified k=5 as the optimal number of clusters, balancing simplicity and compactness.

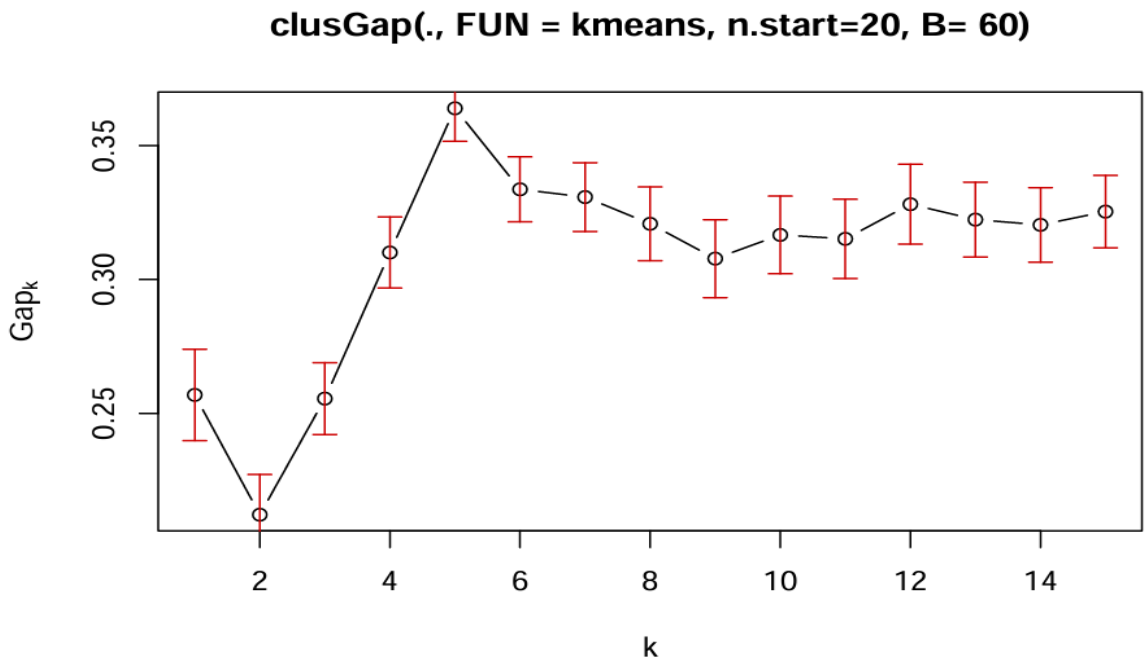
2. Gap Statistic

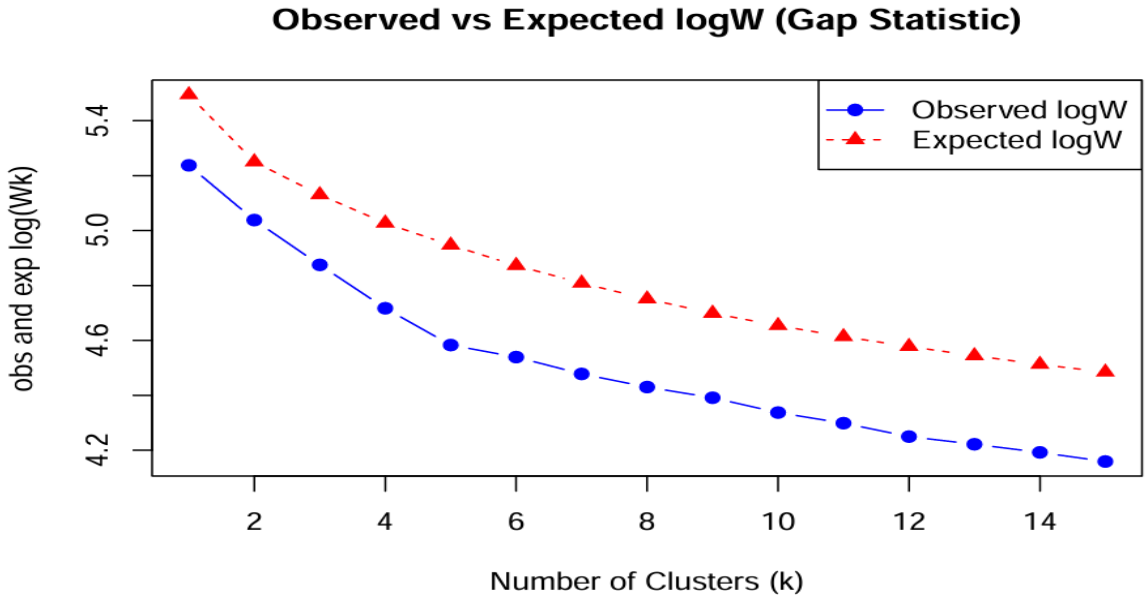
The Gap Statistic compares the WCSS of the observed data to the WCSS of a reference dataset generated with a random uniform distribution. This method evaluates how well-separated the clusters are compared to random noise.

- **How it Works:**
  1. Generate a reference dataset with a uniform random distribution.
  2. Compute WCSS for the observed and reference datasets across a range of k values.
  3. Calculate the gap between the WCSS of the observed and reference datasets. The larger the gap, the more distinct the clustering.

- **Result in This Project:**
  - The Gap Statistic indicated a maximum gap at k=5, confirming that this value of k created well-separated clusters compared to random noise.

Visualization:





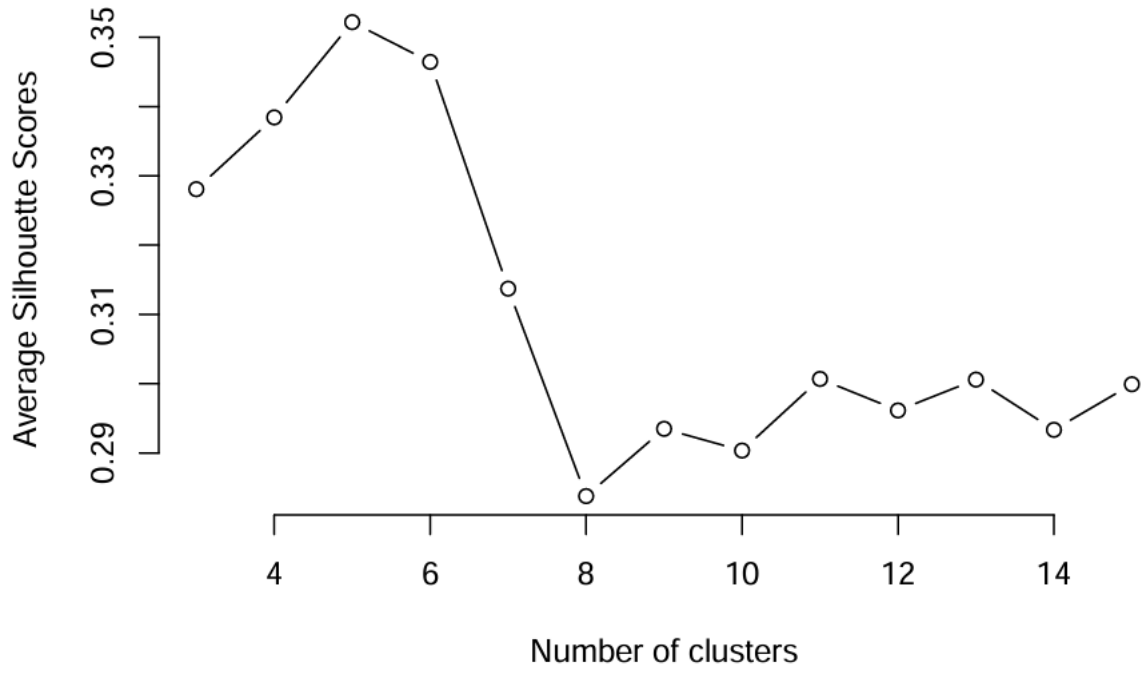
Interpretation: The Gap Statistic reinforced the choice of k=5 as the optimal number of clusters, demonstrating that the observed clusters are significantly better defined than random groupings.

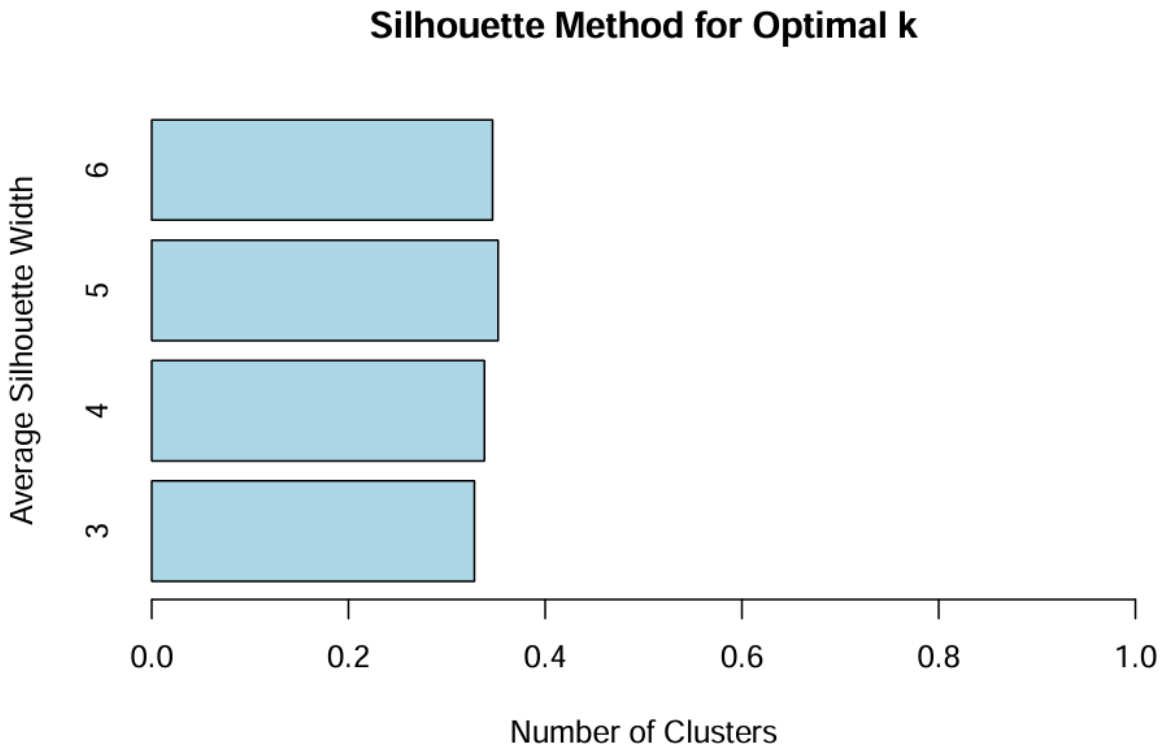
3. Silhouette Method

The Silhouette Method measures the quality of clustering by calculating a **Silhouette Score** for each data point. The score ranges from -1 to 1:

- A score close to **1** indicates that the data point is well-matched to its own cluster and poorly matched to neighboring clusters.
- A score close to **0** indicates that the data point lies on the boundary between two clusters.
- A score close to **-1** indicates that the data point is misclassified.
- **How it Works:**
    1. Compute the average Silhouette Score for all data points for each k value.
    2. Plot k against the average Silhouette Score.
    3. Identify the k with the highest average score, which indicates the best-defined clusters.
- **Result in This Project:**
    - The highest Silhouette Score was observed at k=5, indicating well-defined and compact clusters.

Visualization:





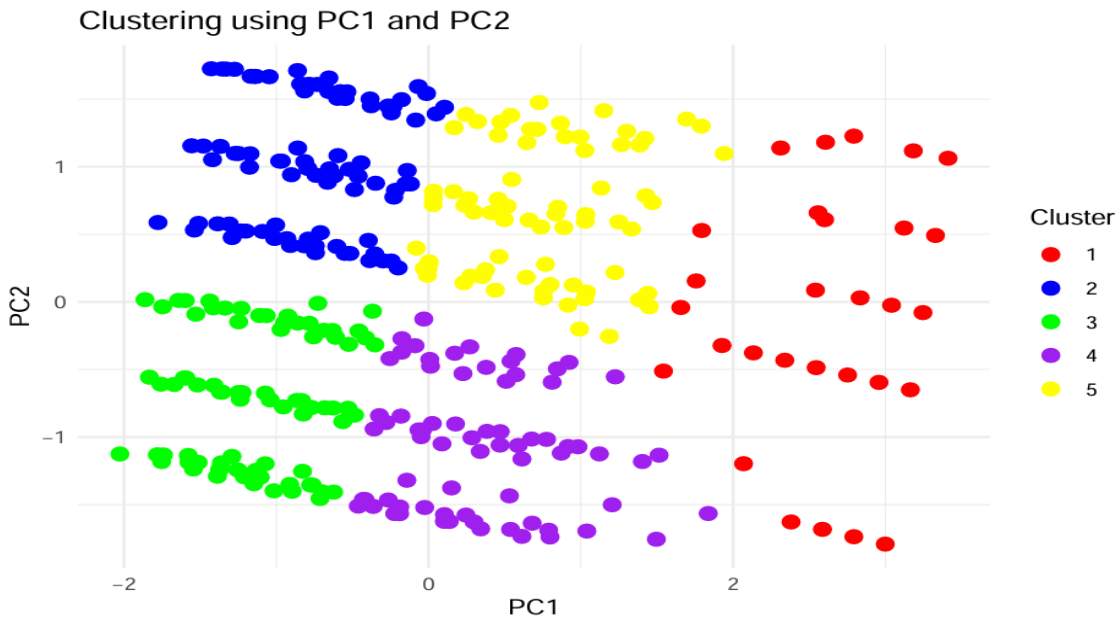
Interpretation: The Silhouette Method confirmed that k=5 provided the most cohesive and well-separated clusters.

Applying K-Means Clustering:

Using k=5, K-Means was applied to the scaled data. The resulting clusters were visualized using the PCA-transformed dataset to enhance interpretability.

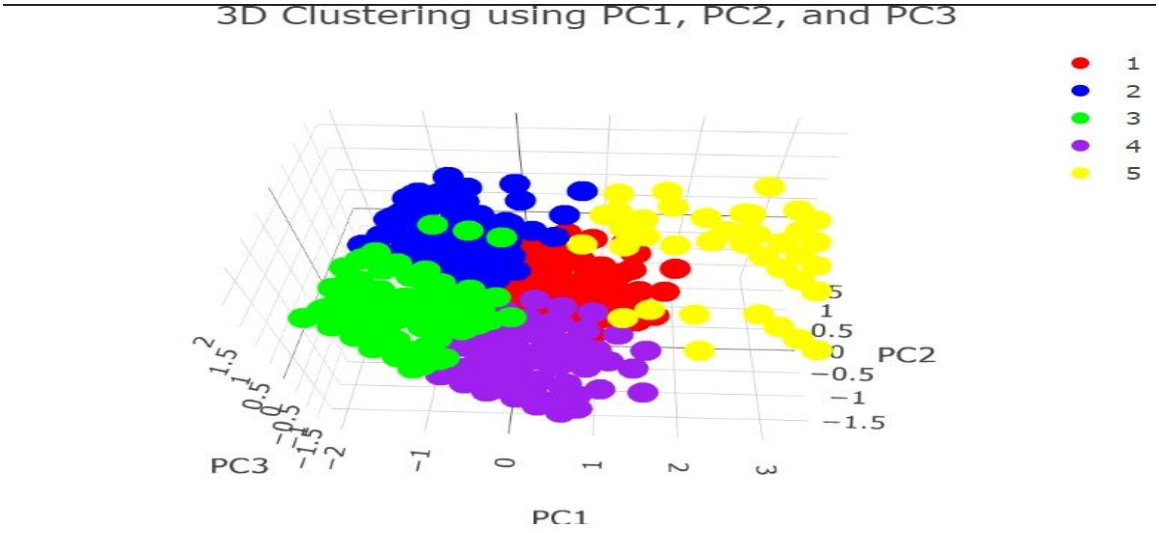
1. **2D Visualization (PC1 vs. PC2):**
    - **Observations:**
        - Five distinct clusters were visible, with some overlap between certain groups.
        - Clusters appeared to align with patterns in Voltage, Height, and Soil Type.
    - **Interpretation:** The separability of clusters suggests that the features collectively influence the grouping of data points.

Visualization:



2. **3D Visualization (PC1, PC2, PC3):**
   - **Observations:**
     - The third principal component added depth to the clustering, further separating certain overlapping groups seen in 2D.
     - The 3D plot provided a clearer perspective on the natural groupings in the data.
   - **Interpretation:** The enhanced visualization confirmed that Voltage, Height, and Soil Type collectively contributed to distinct clustering patterns.

Visualization:



Hierarchical Clustering:

Hierarchical clustering is an unsupervised learning technique that creates a hierarchy of nested clusters, visualized using a dendrogram. Unlike K-Means, it does not require the user to predefine the number of clusters. This method is particularly effective for understanding the hierarchical relationships between data points.

Types of Hierarchical Clustering:

There are two main approaches to hierarchical clustering:

1. **Agglomerative Clustering:**
    - Starts with each data point as its own cluster.
    - Iteratively merges the closest pairs of clusters until all points are in one single cluster.
    - This approach was used in this project.
2. **Divisive Clustering:**
    - Starts with all data points in a single cluster.
    - Iteratively splits clusters into smaller groups.

Agglomerative clustering was chosen for its computational efficiency and its ability to clearly reveal cluster hierarchies.

Distance and Linkage Methods:

Hierarchical clustering requires a distance metric and a linkage criterion to determine how clusters are merged:

1. **Distance Metric:**
    - Euclidean distance was used to measure the similarity between data points. It calculates the straight-line distance between two points in the feature space:

$$d(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

2. **Linkage Methods:**
    - Different linkage methods were explored to decide how the distance between clusters is calculated:
        - **Complete Linkage:** Uses the maximum distance between any two points in different clusters. It tends to create compact clusters.

- **Average Linkage:** Uses the average distance between all pairs of points in different clusters. It provides a balance between compactness and uniformity.
- **Centroid Linkage:** Uses the distance between the centroids of two clusters.

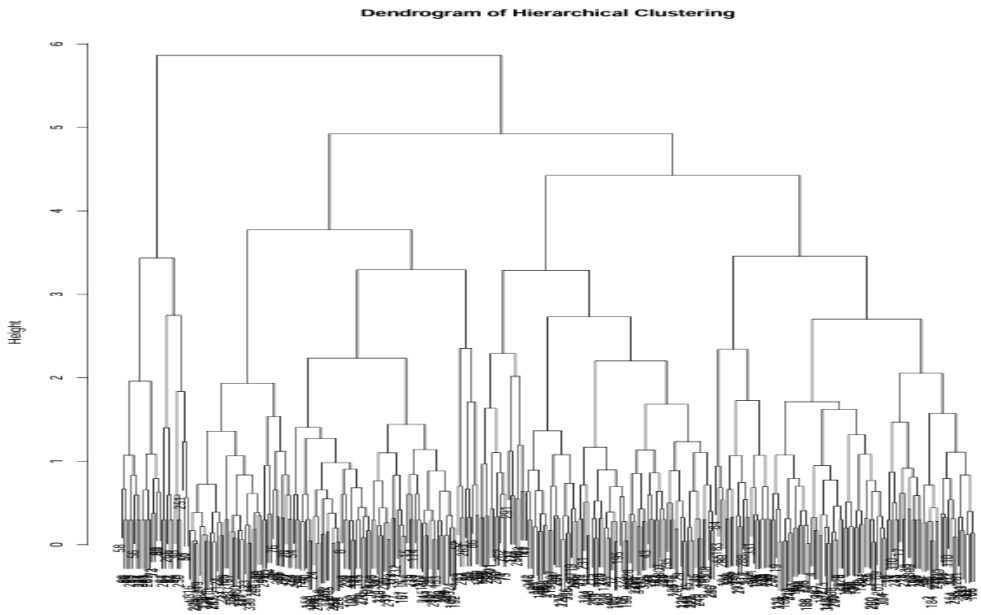
Result in This Project: Complete linkage produced the most distinct clusters and was used for further analysis.

Dendrogram:

A dendrogram is a tree-like diagram that illustrates the process of merging clusters. It helps determine the optimal number of clusters (k) by observing where the vertical lines are longest before merging.

- **Steps to Create the Dendrogram:**
  1. Calculate pairwise distances between data points using the Euclidean metric.
  2. Apply the chosen linkage method to merge clusters iteratively.
  3. Plot the dendrogram to visualize the cluster hierarchy.
- **Interpretation of the Dendrogram:**
  - The height of each branch represents the distance (or dissimilarity) between the merged clusters.
  - To determine k, the dendrogram is "cut" at a level where the clusters are sufficiently distinct.
  - **Result:** Cutting the dendrogram at a reasonable height revealed k=5 clusters, consistent with the results from K-Means.

Visualization:

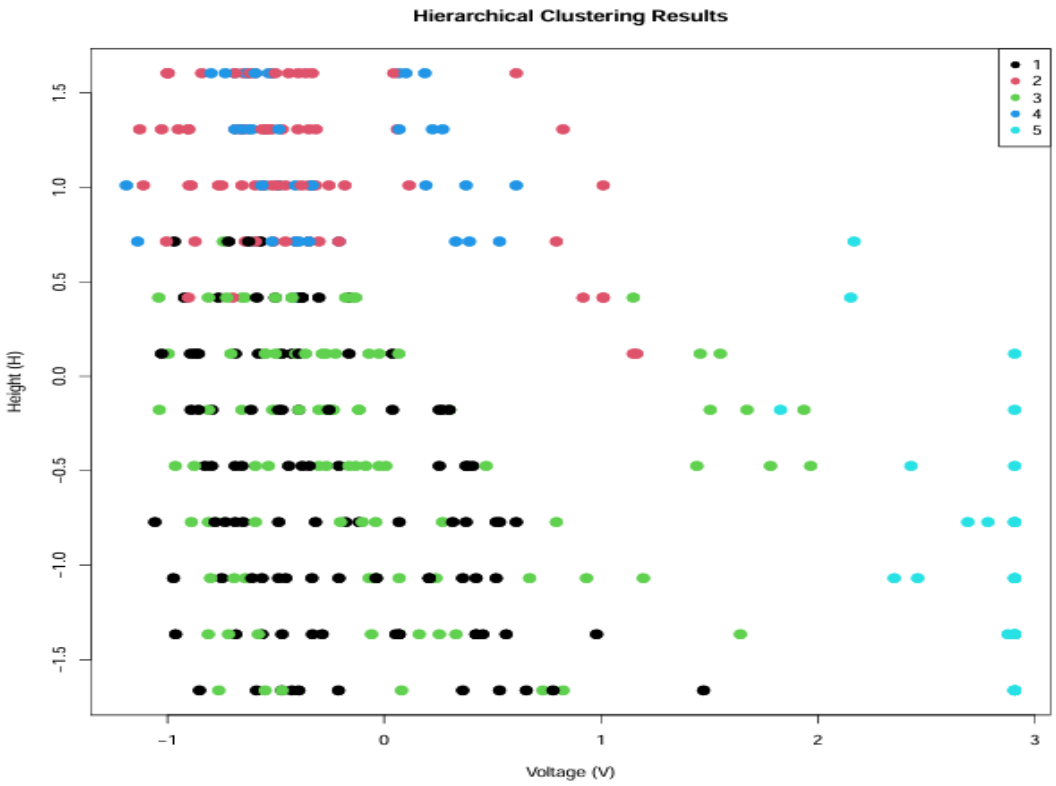


Cluster Visualization:

Clusters from hierarchical clustering were visualized in the original feature space to evaluate their separability:

1. **Voltage vs. Height (2D Plot):**
    - **Observations:** Clusters showed reasonable separability, particularly in areas with distinct Voltage and Height values.
    - **Insights:** Overlaps in some regions suggested shared characteristics among certain mine types or soil types.

Visualization:

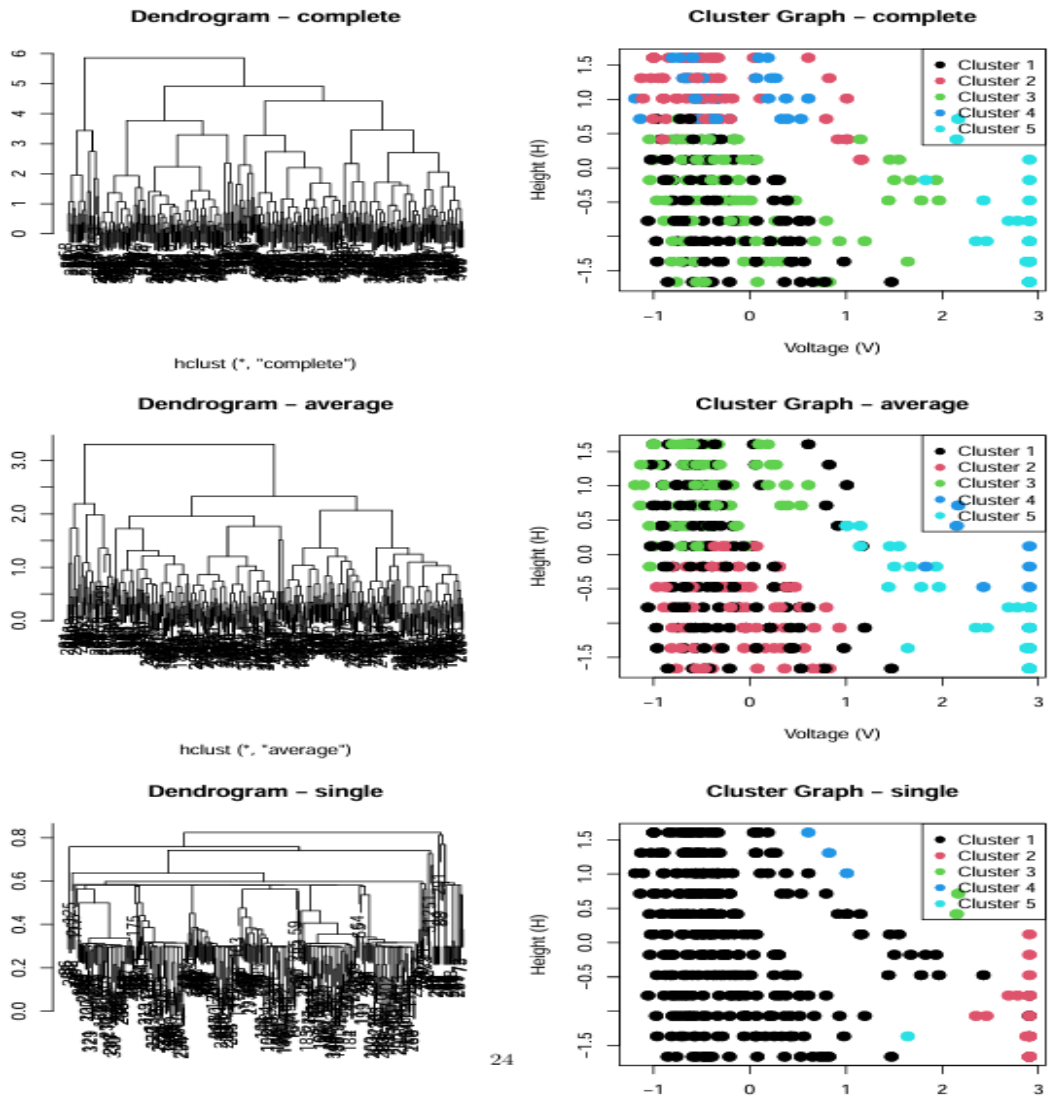


Comparison of Linkage Methods:

A detailed comparison of the linkage methods was performed to evaluate their effectiveness:

Linkage Method	Cluster Characteristics	Observations in the Project
Complete Linkage	Tends to create compact clusters.	Produced distinct, well-separated clusters.
Average Linkage	Balances between compactness and chaining.	Showed reasonable separation, but less distinct than complete linkage.
Single Linkage	Tends to create elongated or "chained" clusters.	Resulted in elongated clusters, which were not meaningful for this dataset.

Visualization:



Choice for Further Analysis:

- **Complete Linkage** was selected for further analysis as it provided the most distinct and meaningful clusters.

Key Advantages of Hierarchical Clustering:

1. **Flexibility in Choosing k:**
   - Unlike K-Means, hierarchical clustering does not require predefining k. This makes it particularly useful for exploratory analysis.
2. **Visualization with Dendrogram:**
   - The dendrogram offers a detailed view of the nested relationships between data points, providing additional insights into the structure of the data.
3. **Capturing Hierarchies:**
   - Hierarchical clustering reveals hierarchical relationships that are not evident in flat clustering methods, such as clusters within clusters.

Comparison with K-Means:

Feature	K-Means	Hierarchical Clustering
Number of Clusters	Predefined (k=5)	Determined from dendrogram (k=5)
Cluster Shape	Compact and spherical	Flexible, dependent on distance and linkage
Computational Efficiency	More efficient for large datasets	Computationally intensive for large datasets
Interpretability	Clear and concise clusters	Reveals nested relationships

Key Insights:

1. **Optimal Number of Clusters:**
   Both hierarchical clustering and K-Means consistently identified k=5k = 5k=5, reinforcing the reliability of this cluster count.
2. **Cluster Relationships:**
   Hierarchical clustering provided a richer understanding of the hierarchical structure within the data, complementing the compact clusters from K-Means.
3. **Applicability to Classification:**
   The insights from hierarchical clustering can inform classification models by highlighting complex relationships between Voltage, Height, and Soil Type.

Kernel Principal Component Analysis (Kernel PCA):

Kernel Principal Component Analysis (Kernel PCA) is an extension of the traditional PCA method, designed to handle datasets with non-linear relationships. By applying a kernel function, Kernel PCA maps the original data into a higher-dimensional space where linear separability can be achieved. This technique is particularly useful when standard PCA fails to capture complex patterns in the data.

Purpose of Kernel PCA:

The primary goals of Kernel PCA in this project were:

1. To uncover non-linear relationships between the features (Voltage, Height, and Soil Type).
2. To enhance cluster visualization in a transformed feature space.
3. To evaluate whether Kernel PCA could improve separability between clusters compared to standard PCA.

Kernel Functions:

Kernel functions determine how the data is transformed into the higher-dimensional space. In this project, the **Radial Basis Function (RBF) Kernel** was used:

- **RBF Kernel:** Also known as the Gaussian Kernel, it maps data points to a higher-dimensional space based on their similarity. The kernel is defined as:

$$K(x, y) = \exp\left(-\frac{\|x - y\|^2}{2\sigma^2}\right)$$

- where $\|x - y\|^2$ is the squared Euclidean distance between two data points, and σ\sigmaσ is the kernel width parameter.

Why RBF Kernel?

- The RBF Kernel is effective for datasets with complex, non-linear relationships, such as this dataset with interactions between Voltage, Height, and Soil Type.

Methodology:

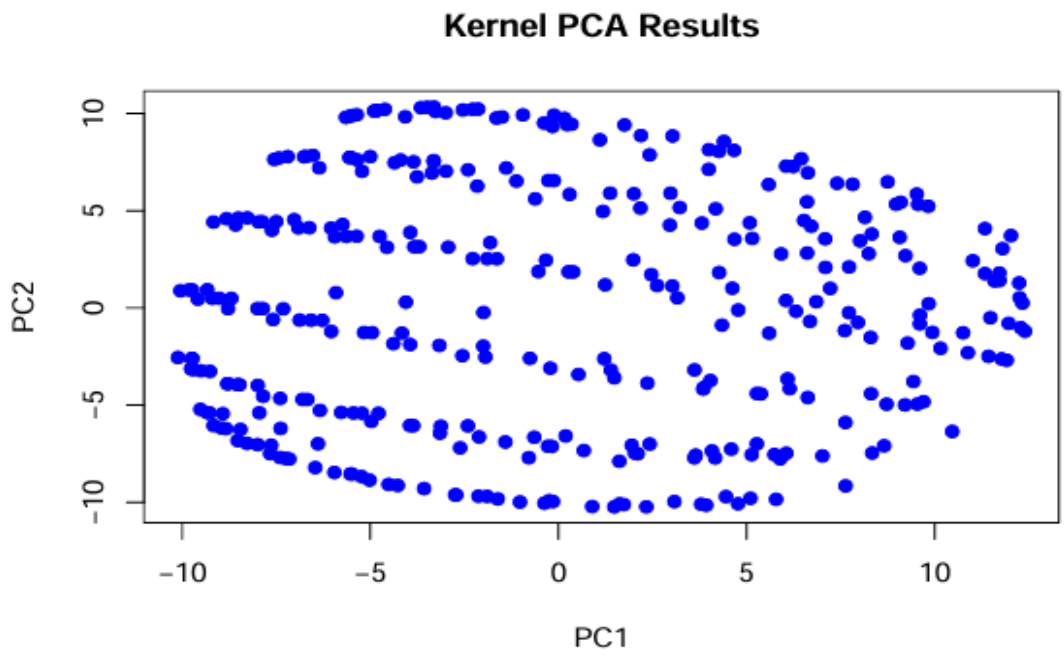
1. **Kernel PCA Transformation:**
    - The RBF Kernel was applied to transform the dataset into a higher-dimensional feature space.
    - The transformed components were ranked based on the variance they explained.

2. **Dimensionality Reduction:**
    - The first two principal components were retained for visualization, capturing most of the variance in the transformed space.
3. **Cluster Visualization:**
    - Clusters were visualized in the Kernel PCA-transformed 2D space to evaluate separability.

Results and Visualizations:

1. **Cluster Visualization in Kernel PCA Space:**
    - **Observations:**
        - The transformed feature space revealed distinct clusters, with improved separability compared to standard PCA.
        - Overlapping regions observed in standard PCA were reduced, particularly for clusters influenced by soil type and mine type.
    - **Interpretation:** Kernel PCA effectively captured non-linear relationships, providing clearer separation between clusters.

    Visualization:



2. **Comparison with Standard PCA:**
    - **Standard PCA:** While effective for visualizing linear separability, it struggled to differentiate overlapping clusters.

- **Kernel PCA:** The RBF Kernel introduced a non-linear transformation, improving cluster separability and highlighting previously hidden relationships.

Key Insights from Kernel PCA:

1. **Enhanced Visualization:** Kernel PCA improved the visual distinction between clusters compared to standard PCA, particularly for clusters influenced by complex feature interactions.
2. **Capturing Non-Linear Relationships:** The RBF Kernel effectively mapped the data into a space where clusters were more distinct.
3. **Complementary Tool:** Kernel PCA complemented standard PCA by addressing its limitations, making it an invaluable tool for cluster analysis in this project.

Classification and Model Evaluation:

After clustering analysis, the next step was to assess the performance of K-Means as a classifier and then apply supervised classification models to predict mine types based on the dataset's features. This section provides details on train-test splitting, K-Means classification evaluation, and the performance of various supervised models.

Train-Test Split:

To evaluate model performance and ensure the reliability of predictions, the dataset was divided into:

- **Training Set (80%):** Used for training the models.
- **Testing Set (20%):** Used for evaluating the models on unseen data.

The split ensures that the models are evaluated on data they have not been exposed to during training, providing an unbiased estimate of their performance.

K-Means Clustering as a Classifier:

Although K-Means is an unsupervised learning algorithm, its cluster assignments can be used as a proxy for classification by mapping each cluster to the most frequent class label within it.

1. **Methodology:**

- K-Means clustering (with k=5) was applied to the training data.
- Cluster assignments were mapped to the most common mine type in each cluster.
- The classifier was then evaluated on the test data by comparing predicted cluster labels to actual mine type labels.

2. **Evaluation Metrics:**
   - **Accuracy:** The percentage of correctly classified instances.
   - **Confusion Matrix:** A breakdown of correctly and incorrectly classified instances for each class.
3. **Results:**
   - **Accuracy:** 26.47%
   - **Confusion Matrix Observations:**
     - The classifier performed poorly due to the unsupervised nature of K-Means, as it does not consider class labels during training.
     - Significant misclassifications were observed, particularly for overlapping clusters.

Interpretation: While K-Means clustering provided meaningful groupings for unsupervised analysis, its performance as a classifier was limited. This result highlighted the need for supervised learning methods.

Supervised Classification Models:

To improve classification accuracy, supervised learning algorithms were applied to the dataset. The models trained and evaluated include:

Random Forest:

1. **Concept:**
   - A Random Forest is an ensemble learning method that aggregates predictions from multiple decision trees to improve accuracy and reduce overfitting.
   - Each tree is trained on a random subset of the data, and the final prediction is based on majority voting.
2. **Results:**
   - **Accuracy:** 42.65%
   - **Strengths:**
     - Random Forest showed better performance compared to K-Means, particularly in handling complex feature interactions.
   - **Limitations:**
     - Misclassifications occurred for minority classes due to class imbalance in the dataset.
3. **Confusion Matrix:**
   - Highlighted better performance for majority classes but lower sensitivity for minority mine types.

Support Vector Machine (SVM):

1. **Concept:**
    - SVM aims to find the hyperplane that best separates the classes in a high-dimensional feature space.
    - A radial basis function (RBF) kernel was used to handle non-linear separability.
2. **Results:**
    - **Accuracy:** 41.18%
    - **Strengths:**
        - SVM performed well for certain mine types and captured non-linear relationships in the dataset.
    - **Limitations:**
        - Similar to Random Forest, class imbalance affected overall performance.
3. **Confusion Matrix:**
    - Showed higher sensitivity for specific mine types but struggled with overlapping clusters.

Comparison of Classification Models:

Model	Accuracy	Strengths	Weaknesses
K-Means (Cluster)	26.47%	Unsupervised, useful for initial cluster analysis	Poor accuracy due to unsupervised nature
Random Forest	42.65%	Captured complex interactions, reduced overfitting	Misclassified minority classes
SVM	41.18%	Good for non-linear relationships	Sensitive to class imbalance and overlapping clusters

Key Insights:

1. **K-Means as a Classifier:** While useful for clustering, K-Means was not effective for classification due to its unsupervised nature.
2. **Supervised Learning Performance:** Random Forest outperformed SVM slightly, but both models showed limitations due to class imbalance and feature overlap.

Conclusion:

This project demonstrated the application of data-driven techniques to the problem of land mine detection using magnetic anomaly sensor data. By leveraging unsupervised and supervised learning methods, we uncovered valuable insights, evaluated clustering patterns, and built classification models for mine type prediction. Below are the key takeaways and outcomes from the project:

Key Findings:

1. **Exploratory Data Analysis:**
    - The features (Voltage, Height, Soil Type) were analyzed for patterns and distributions.
    - Weak correlations between features suggested independent contributions, making them valuable for clustering and classification tasks.
2. **Clustering Analysis:**
    - **K-Means Clustering:** Identified five clusters, aligning well with the dataset's mine types. Visualization in PCA space revealed meaningful groupings, though some overlap was present.
    - **Hierarchical Clustering:** Provided a hierarchical perspective, confirming k=5k = 5k=5 as the optimal number of clusters. Comparison of linkage methods indicated that Complete Linkage yielded the most distinct clusters.
    - **Kernel PCA:** Enhanced cluster separability by capturing non-linear relationships, providing better visual and analytical insights.
3. **Classification Analysis:**
    - **K-Means as a Classifier:** Showed poor performance (accuracy: 26.47%), highlighting its limitations in supervised tasks.
    - **Random Forest and SVM:** Achieved higher accuracies (42.65% and 41.18%, respectively). Random Forest performed slightly better due to its ability to handle complex feature interactions.
    - **Challenges:** Class imbalance and overlapping feature distributions limited the effectiveness of classification models.

Challenges and Limitations:

- **Class Imbalance:** The uneven distribution of mine types impacted model sensitivity, particularly for minority classes.
- **Feature Overlap:** Clusters showed overlaps in certain regions, indicating shared characteristics among mine or soil types.
- **Model Performance:** While Random Forest and SVM improved accuracy, the overall performance suggests room for further optimization.

Future Recommendations:

1. **Addressing Class Imbalance:**
    - Techniques like oversampling, undersampling, or Synthetic Minority Oversampling Technique (SMOTE) could be applied to balance the dataset and improve model performance.
2. **Exploring Advanced Models:**
    - Deep learning approaches, such as neural networks, could capture more intricate patterns and improve classification accuracy.
    - Ensemble methods combining multiple algorithms might enhance robustness.
3. **Feature Engineering:**
    - Additional features (e.g., spatial location, environmental conditions) could provide more context and improve separability.
    - Feature selection methods could be applied to identify the most influential attributes.
4. **Optimization of Hyperparameters:**
    - Fine-tuning model parameters (e.g., number of trees in Random Forest, kernel parameters in SVM) could yield better results.