SMART CONTRACT CLUSTER ANALYSIS

~Prakhyati Bansal
Webacy Externship

INTRODUCTION

- **Objective**: To analyze smart contracts using clustering to identify risk tags and enhance blockchain security.
- **Context**: Blockchain technology introduces risks through autonomous smart contracts. Clustering helps us understand patterns of risk in these contracts, allowing us to improve their security.
- **Approach**:
  Use hierarchical clustering to group contracts with similar risk profiles.
  Identify high-risk factors like exploitation, bad contracts, and external dependencies.

DATA PREPARATION

- **Dataset Overview**:
  Smart contracts dataset tagged with risk factors.
  Features include; exploitation, bad contracts, external dependencies, buy and sell tax, modifiable slippage, and anti-whale mechanisms. 7 selected Risk tags.
- **Feature Selection**:
  One-hot encoding for categorical variables.
  Normalization of numerical features for balanced clustering influence.
- **Data Transformation**:
  Ensures that all features contribute proportionally to the clustering process, avoiding bias from high-magnitude values.

CLUSTERING IMPLEMENTATION

- **Algorithm**: Hierarchical Clustering (HCA)
  HCA creates a tree-like structure, allowing exploration of the data at different levels of granularity.
  It is useful for understanding relationships between contracts in terms of risk.
- **Distance Metric**: Jaccard distance, suitable for binary or sparse data.
- **Linkage Method**: Average linkage for combining clusters based on average similarity.
- **Optimal Number of Clusters**: Determined using dendrogram analysis, cut at 5 clusters for balance between granularity and interpretability.

CLUSTERING ANALYSIS

Cluster Breakdown:

- Cluster 1: Low risk across all key factors. Contracts in this group are relatively safe and stable.
- Cluster 2: High risk of exploitation and bad contracts. Contracts here are highly vulnerable and require immediate attention.
- Cluster 3: High external dependencies. These contracts depend heavily on third-party services, introducing additional risks.
- Clusters 4 & 5: Mixed risks, particularly around modifiable tax structures (buy/sell tax) and features like slippage. These could be contracts with dynamic behavior that can alter risk depending on settings.

VIZUALISATION OF CLUSTERS

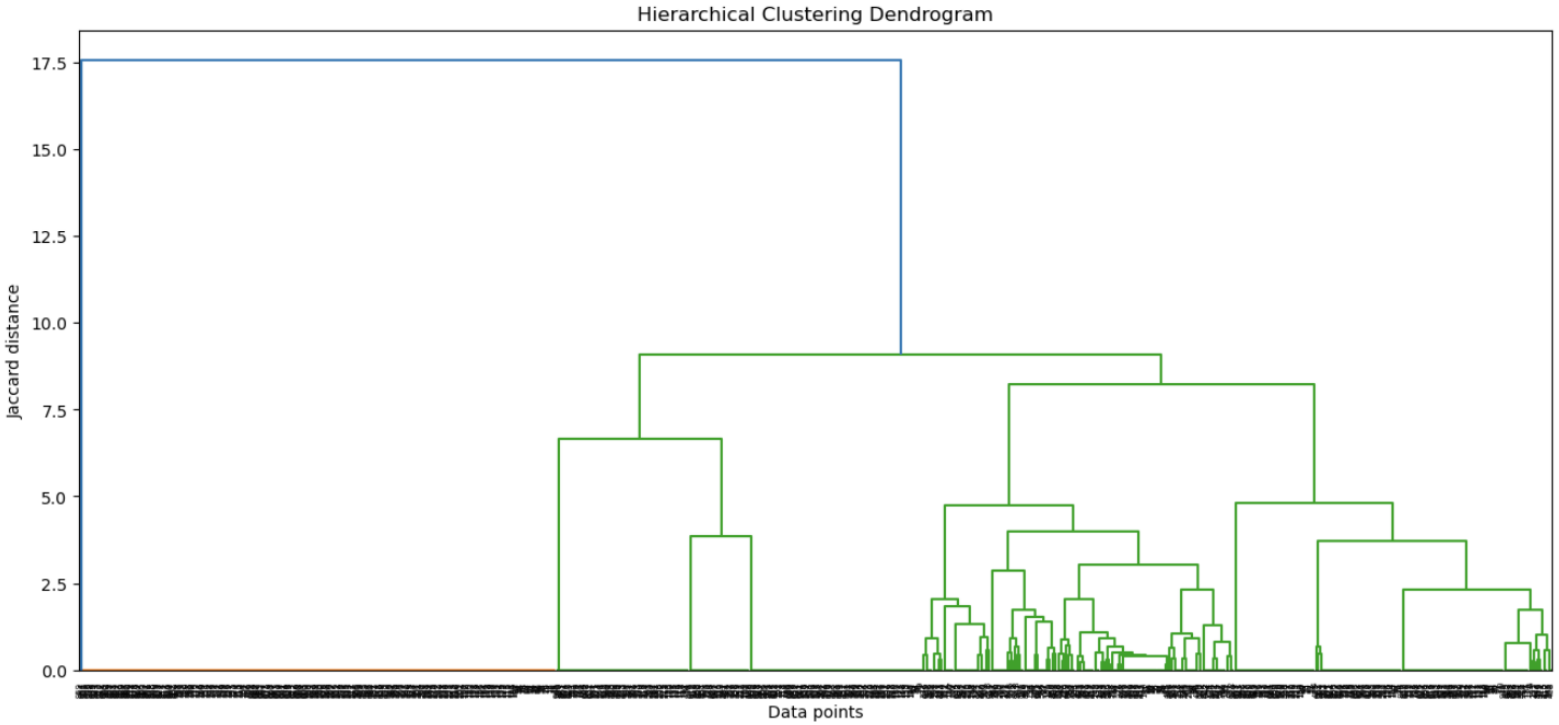
Cluster Centroid Heatmap:

- Red represents high average risk within a cluster for a particular risk factor.
- Cluster 2 shows significant risk of exploitation and bad contracts.
- Cluster 3 is notable for high external dependencies, which increases risk through third-party reliance.

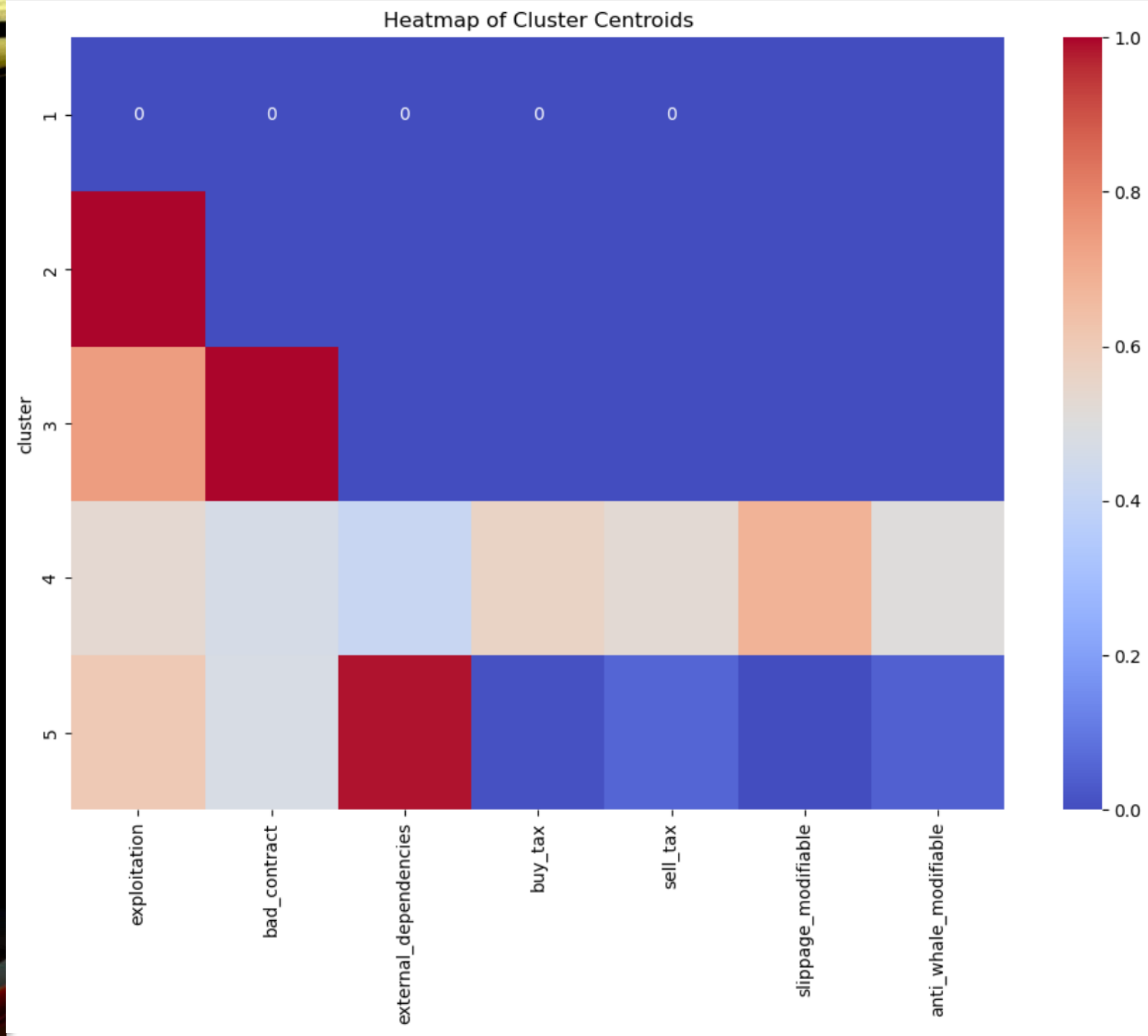
Dendrogram:

- Helps visualize the merging of clusters at different levels.
- Large vertical lines indicate significant differences between clusters.

- **Dendrogram**: Visual representation of hierarchical clustering.
  Shows the merging of clusters at various distance levels.



- **Heatmap**: Correlation heatmap of features within each cluster. Highlights the predominant risk tags in each cluster.



METHODOLOGY & FINDINGS

Methodology:

- Preprocessing: One-hot encoding and normalization.
- Clustering Approach: Hierarchical clustering based on the Jaccard distance metric, which is ideal for binary features.
- Visualizations: Used heatmaps and dendrograms for better interpretability of the clusters.

Findings:

- Cluster 2: A significant number of contracts fall into a high-risk category, specifically related to exploitation vulnerabilities.
- Cluster 3: External dependencies introduce a medium-level risk.
- Impact: Clustering helped prioritize which contracts need immediate auditing for security purposes.

CONCLUSION

Summary:

- Clustering revealed distinct patterns in smart contracts based on their risk profiles.
- Contracts grouped into high-risk categories should be the focus of security audits.

Impact on Blockchain Security:

- Clustering allows for a more structured approach to identifying and mitigating risks.
- Potential for real-time risk monitoring by integrating clustering into security auditing tools.

Future Applications:

- Expanding to larger datasets.
- Exploring more advanced clustering algorithms like K-means.
- Incorporating clustering into blockchain governance tools for better risk management.