AI & ML project

By group 11
D'Aloia Cascone Francesca
Luca Petroccione
Emanuel Antonio Rizzo Rosas

Alien Galaxy and Planets colonizations

An Unsupervised Learning Project

[section 1]

Introduction:

The goal of our project is to analyze and group alien galaxies based on shared characteristics. Using an unsupervised learning approach, we explore the dataset through various clustering algorithms to understand colonization patterns.

The project addresses the need to understand how different galaxies manage resources and develop technologically. It also demonstrates how unsupervised learning can be applied to group unknown entities based on shared characteristics.

[section 2]

Methods:

To approach the project, we divided the work into two main parts. In the first phase, we focused on preprocessing and preparing the dataset to create a reliable foundation for clustering. This involved an initial selection of features based on their logical relevance and definitions. We then normalized the data and handled missing values by testing various approaches, ultimately selecting KNN imputation as the most appropriate method. To further refine our selection, we examined feature distributions to identify additional criteria for inclusion. We also analyzed correlations using a correlation matrix, highlighting the most significant relationships to guide the next steps.

The second phase concentrated on applying and evaluating clustering models. After further refining the features to address potential redundancies and ensure unique and meaningful correlations, we tested three different clustering methods: hierarchical clustering, K-Means, and DBSCAN. Each model was assessed for its effectiveness, enabling us to identify patterns and draw meaningful conclusions from the resulting clusters.

Environment Setup

This project was run in a Conda environment with Python 3.11. Some key libraries used included Pandas, NumPy, Matplotlib, Seaborn, and Scikit-learn (all listed in the first cell of the script).

Possible needed commands to recreate the Environment

```
conda create -n alien_galaxy_env python=3.11

conda activate alien_galaxy_env

conda install pandas numpy matplotlib seaborn scikit-learn
```

Dataset Overview

The dataset contains information on various characteristics of alien galaxies, such as:

- Economic and Trade Features
- Scientific and Technological Metrics
- Environmental Factors
- Planetary Characteristics and Resources
- Population Metrics
- Intergalactic Communication and Contact

However, the presence of missing values was faced, requiring careful handling to avoid data loss and maintain dataset integrity.

Features Selection

Feature selection was based on domain knowledge and correlation analysis. We aimed to retain features that provide meaningful insights while reducing redundancy and noise.

Along the the process of the project there were made three features selections:

First features selection:

One of the first things done, we have selected the features based on their definitions and therefore on the relevance to the objectives of the project and for exclusions of the irrelevant ones.

Exclusions reasons:

Lack of Relevance to Clustering Goals: `Last_Contact_Days`, `Discovery_Date`, and `Colonization_Year` are time-based variables and do not directly contribute to clustering based on environmental or societal characteristics.

Abstract or Generalized Features: `Cultural_Exchange_Programs`, `Species_Expansion_Response`, and `Dominant_Species_Social_Structure` are abstract and do not have clear quantitative impact or direct relationship to clustering environmental or societal traits.

Irrelevant Identifiers: `Planet_ID` and similar identifiers are not useful for clustering because they do not carry meaningful information about relationships or differences between planets.

Second features selection:

The second feature occurred in the middle of the project, after having processed and imputed the data. Using the distribution plotting we have been able to analyze the variance and behaviour of the values of the features. As the distributions appeared, features such as MIlitary_Engagements, Peace_Treaty_Accords, Trade_Agreements_Signed and Galactic_Visits showed minimal variance. These features do not contribute much to distinguishing between clusters because their values are nearly uniform across all samples (e.g. distribution of Military_Engagements shows that the same value has a frequency of almost 2000, which is a lot considering that our entire dataset has 2240 rows).

Third features selection:

The last selection was made after analyzing the correlations shown in the correlation matrix and highlighted by the table where relevant correlations were plotted with an absolute value for correlation set at 0.45 (to avoid redundancy in the features and still chose moderate to strong correlations).

Therefore the final selection of features considers the followings:

['Food_Production_Tons', 'Mineral_Extraction_Tons', 'Resource_Mining_Operations', 'Resource_Allocation_Credits', 'CO2_Concentration', 'Liquid_Energy_Consumption_Terawatts', 'Offspring_Colonies', 'Technological_Advancements', 'Biological_Research_Units', 'Alien_Population_Count', 'Exploration_Missions', 'Alien_Civilization_Level_Encoded']

Reason for Feature Selection

Selecting the right features ensures that the clustering algorithm focuses on relevant data points, leading to more interpretable and actionable clusters. We have prioritized features that:

- Provide unique insights into colonization and resource management.
- Have significant variation across galaxies.
- Are less prone to redundancy based on correlation analysis.

Handling Missing Values

As the presence of Missing Values was detected, with a relevance from 8% to 11% for each feature (reference to quantification of missing values in the script), to avoid noise and the reduction of reliability we chose to impute them. Instead of dropping the rows, which could reduce the size of the dataset, we decided to impute the missing values using the K-Nearest Neighbors (KNN) Imputer.
KNN imputation preserves the structure of a dataset by filling missing values with similar data. It will maintain the important relationships between features, which is crucial in clustering.

Processing the data

All along the project processing data was needed in 3 moment:

Normalization was applied first before imputing the missing values with the StandardScaler() method, then again normalization was needed in the second part of the project after the last selection of features, this time with the Min-Max scaling method as it is more suitable for clustering algorithms;

Furthermore the process of encoding was needed for the `Alien_Civilization_Level` feature, using ordinal mapping to convert qualitative values into numerical ones.

Clustering Algorithms

We experimented with three clustering algorithms:

1. K-Means: Chosen for its simplicity and ability to create well-defined, compact clusters. The Elbow Method and Silhouette Analysis were used to determine the optimal number of clusters.
2. DBSCAN: A density-based clustering algorithm that identifies clusters of varying shapes and sizes. It is less sensitive to outliers but requires careful parameter tuning.
3. Hierarchical Clustering: Builds a hierarchy of clusters and allows visualization through dendrograms. However, it can be computationally expensive for large datasets.

Flowchart of Process

The following steps describe the workflow for the project:

- **Data Loading**: Load the dataset from a CSV file.
- **Data Cleaning and Imputation**: Perform handling of missing values using KNN Imputation.
- **Feature Scaling and Normalization**: Select features used for clustering analysis and apply the Min-Max Scaling.
- **Clustering**: Perform K-Means, DBSCAN, and Hierarchical Clustering.
- **Evaluation and Visualization**: Clustering results must be evaluated using different metrics and the clusters visualized.

[Section 3]

Experimental Design

This section outlines the experiments conducted to validate the clustering models and the evaluation metrics used.

Each experiment involved the running of the clustering algorithm on the preprocessed dataset.

Models

We selected three clustering algorithms as baseline models for comparison:

1. **Hierarchical Clustering**: After finding the optimal number of clusters with the construction of a dendrogram, we then proceeded with plotting the clusters and visualizing them as much as calculating the `Silhouette Score.` The results of the visualizations showed 3 clusters : **Cluster 2** is well-separated, while **Clusters 1 and 3** show some overlap, indicating potential challenges in clear boundary delineation for these clusters. The code proceeds in showing the clusters also separated for more clarity. Overall the representation reflects the not so great refinements in feature selection or clustering parameters, since the result is not a clear separation of groups.

2. **K-Means Clustering**: After determining the optimal number of clusters using the elbow method or silhouette analysis, we applied K-Means clustering. The results were visualized to examine the separation of clusters. The visualizations show 5 clusters, some clusters (e.g., green and blue) show distinct separation, others (e.g., brown and cyan) overlap significantly, indicating the challenges faced in clearly distinguishing groups. Although the method produced distinct centroids for each cluster, the boundaries were not entirely clear due to the overlap.

3. **DBSCAN (Density-Based Spatial Clustering of Applications with Noise)**: Using DBSCAN, we aimed to identify clusters based on density and detect outliers. After setting parameters such as `eps` and `min_samples`, we plotted the clusters for visualization and calculated the silhouette score for evaluation. The results highlighted that DBSCAN effectively identified dense regions corresponding to clusters and marked sparse regions as noise or outliers. The visualization shows results with three categories: **Cluster 0 (maroon)** as the main dense group, **Cluster 1 (yellow)** as a smaller distinct cluster, and **Cluster -1 (black)** representing outliers or noise. DBSCAN effectively identifies dense clusters and noise, though still the overlap between clusters is relevant .

Each algorithm was tested using the same dataset to ensure a fair comparison. The results were compared based on their performance on evaluation metrics.

Evaluation Metrics

To measure the quality of clusters formed by each algorithm, we used two key evaluation metrics:

1. **Silhouette Score**: This metric measures how similar a data point is to its own cluster compared to other clusters. It ranges from -1 to 1, with higher values indicating

better-defined clusters. A Silhouette Score closer to 1 indicates that data points are well-matched to their own cluster and poorly matched to other clusters.

2. **Davies-Bouldin Index**: This metric measures the average similarity ratio between each cluster and its most similar cluster. A lower Davies-Bouldin Index indicates better clustering performance, as it suggests that clusters are more distinct from each other.

Summary of Evaluation Results

Model	Silhouette Score	Davies-Bouldin Index
K-Means	0.228	1.560
DBSCAN	0.346	2.88
Hierarchical	0.228	1.529

The clustering results provide insight into the relative performance and effectiveness of each method based on the calculated metrics:

1. **Hierarchical Clustering**:
   - Silhouette Score: **0.267** (moderate cluster cohesion and separation).
   - Davies-Bouldin Score: **1.529** (moderate cluster compactness and separation).
   - Interpretation: Hierarchical clustering provides reasonably interpretable clusters, making it a good choice for understanding the dataset structure. The moderate scores suggest that the clustering is effective but not optimal, likely due to the dataset's complexity.
2. **K-Means**:
   - Silhouette Score: **0.228** (lower cohesion and separation compared to the others).
   - Davies-Bouldin Score: **1.561** (slightly weaker compactness and separation).
   - Interpretation: K-Means struggles with the dataset, as its assumptions of spherical and evenly-sized clusters do not align with the data distribution. These scores indicate that K-Means is less effective for this specific dataset.
3. **DBSCAN**:
   - Silhouette Score: **0.346** (highest cohesion and separation among methods).

- ○ Davies-Bouldin Score: **2.886** (highest, indicating less compact and less well-separated clusters).

Interpretation:

DBSCAN achieves the best silhouette score, excelling at identifying cohesive clusters while handling noise and irregular cluster shapes. However, the high Davies-Bouldin score suggests that the clusters are less compact and more scattered.

[Section 4 & 5]

Results & Conclusions

The clustering models showed mixed performances, with DBSCAN achieving the best silhouette score (0.346), indicating moderate cluster cohesion, but its Davies-Bouldin Index (2.88) suggests poor separation between clusters. K-Means and Hierarchical clustering had similar silhouette scores (0.228) and better Davies-Bouldin values (1.560 and 1.529, respectively), indicating slightly better inter-cluster separation but less cohesive clusters.

Overall, the scores highlight that the performance was weak, and the clustering efforts were not very successful in achieving clear and well-defined groups. Despite this, the team invested significant time and effort in fine-tuning parameters and exploring various combinations to improve the clustering.

The lack of clear cluster separation might also stem from an inherent limitation in the dataset. It is possible that the alien colonies in the dataset share very similar needs and characteristics, making their differences less detectable through the available features. This uniformity in needs and behaviors could make it inherently difficult to identify distinct groups, regardless of the clustering techniques or parameter optimization applied.

Overall interpretation of the clusters

The overall interpretation of the clusters suggests that the alien colonies exhibit both shared and distinct characteristics, but their separation is not entirely clear due to overlapping needs and limitations in the feature space. The largest clusters, such as those represented by broad distributions (e.g., Clusters 1 and 3 in different models), indicate a majority of "generalist" colonies that balance resource use, population, and technological advancements. These colonies likely operate within a similar developmental range, reflecting common needs across the dataset.

Smaller, more compact clusters (e.g., Clusters 2 and 5) represent "specialized" or "developing" colonies, characterized by specific behaviors, such as lower resource consumption, focused research, or unique environmental adaptations. These colonies may reflect early-stage or niche development.

Outliers and noise, as identified in DBSCAN or through small, distinct clusters (e.g., Cluster 4 in K-Means), highlight colonies with extreme or unique traits, potentially due to rare environmental conditions, advanced technologies, or isolated resource needs. These colonies stand apart from the majority and may represent exceptions to general trends.

The overlap observed in many models suggests that the colonies' fundamental needs, such as food production, energy use, and resource allocation, are inherently similar. This points to a challenge in differentiating colonies based on the available features, as the dataset may not fully capture the subtle nuances or unique aspects of their operations. Overall, while the clusters provide valuable groupings, they reflect the difficulty in achieving strong separation, likely due to the uniformity in colony needs and the limitations of the dataset.