**DATA AND INFORMATION QUALITY PROJECT REPORT**

PROJECT ID: 29

PROJECT NUMBER: 1

ASSIGNED DATASETS: abalone, users

STUDENTS:

Gabriele Ginestroni (10687747)

Tommaso Capacci (10654230)

ASSIGNED TASK: Clustering

1. **SETUP CHOICES**

**Chosen ML algorithms:**

Since the given datasets presented different characteristics we decided to use different algorithms for each of them. Particularly, for the “abalone” dataset we used:

* **K-Prototypes**: the K-Prototypes algorithm is a clustering method used to cluster mixed-type data (i.e. data that contains both categorical and numerical variables). This algorithm combines the K-Means algorithm for numerical variables and the K-Modes algorithm for categorical variables.

It works by first initializing the centroids for each cluster using either random sampling or a user-specified initialization method. Then, for each iteration, it assigns each data point to the cluster with the closest centroid. Next, it calculates the new centroid for each cluster by taking the mean of the numerical variables and the mode of the categorical variables of the data points assigned to that cluster. Finally, it repeats these steps until the centroids no longer change or a maximum number of iterations is reached.

* **DBSCAN (Density-Based Spatial Clustering of Applications with Noise):** DBSCAN is a density-based clustering algorithm that groups together data points that are closely packed together (i.e. dense regions of data points) and separates data points that are sparsely located (i.e. less dense regions of data points).

DBSCAN works by defining a neighborhood around each data point, and then identifying clusters based on the density of data points in these neighborhoods. The algorithm takes as input two parameters: eps (epsilon) and minPts. The eps parameter defines the radius of the neighborhood around each data point, and minPts defines the minimum number of data points required to form a dense region.

DBSCAN works as follows:

1. It starts with an arbitrary data point that has not been visited, and finds all data points within a distance of eps from it. If the number of such points is greater than minPts, a new cluster is created, and all points within a distance of eps from the seed point are added to the cluster.
2. If the point is not a core point (i.e. the number of points within eps is less than minPts), the point is marked as noise.
3. The process is then repeated for each of the newly found points, and points are added to the cluster as long as they are within eps distance of already added points and the cluster size is greater than minPts.
4. When no more points can be added to the current cluster, the process is repeated with a new arbitrary point that has not yet been visited, and new clusters are formed in the same manner.

DBSCAN is able to identify clusters of arbitrary shapes and it is also able to identify noise, which other density-based clustering methods are not capable of doing.

For “users” instead we opted for:

**Chosen ML performance evaluation metrics:**

**Imputation/outlier detection techniques selected:**

2. PIPELINE IMPLEMENTATION

a. Description of the steps you performed

3. RESULTS

a. Description of the main results obtained

b. ML performance comparison between the imputation/outlier detection techniques you have implemented