1.	**Kato Joseph Bwanika**	**Reg:2023-B291-11709**
2.	**Mambo Emmanuel**	**Reg:2022-B291-11357**
3.	**Ssenkaayi Vihan**	**Reg:2023-B291-11360**
4.	**Kutosi Mark**	**Reg:2023-B291-11708**
5.	**Lubangakene Jonathan**	**Reg:2023-B291-13195**

Group2 Practical Assignment Report: Wholesale Customer Segmentation (UCI wholesale Customer dataset)

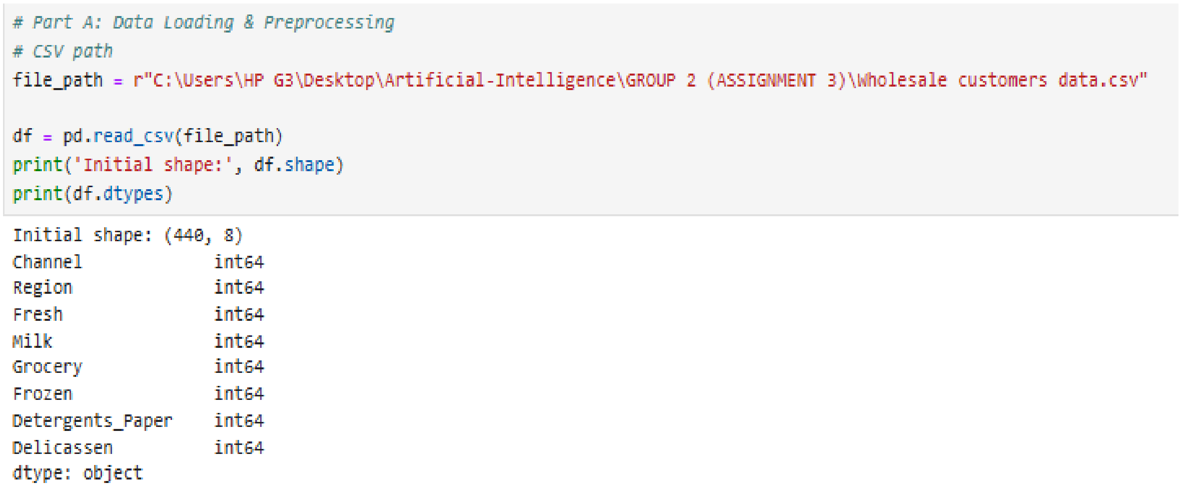
Introduction

This report details the application and comparative analysis of K-means and DBSCAN clustering algorithms on the UCI Wholesale Customers dataset. The goal is to segment wholesale customers based on their annual spending across various product categories (Fresh, Milk, Grocery, Frozen, Detergents_Paper, Delicassen) to provide actionable business recommendations.

Part A: Data Loading & Preprocessing

1. Data Loading and Initial Inspection

The dataset was loaded and inspected to confirm column types and initial dimensions. The primary features are the six annual spending categories along with two categorical identifiers: Channel and Region.



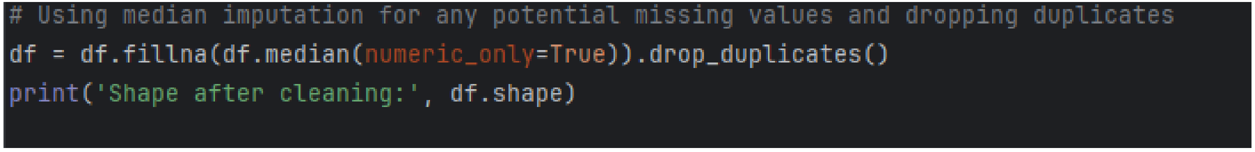
```
# Part A: Data Loading & Preprocessing
# CSV path
file_path = r"C:\Users\HP G3\Desktop\Artificial-Intelligence\GROUP 2 (ASSIGNMENT 3)\Wholesale customers data.csv"

df = pd.read_csv(file_path)
print('Initial shape:', df.shape)
print(df.dtypes)
```

```
Initial shape: (440, 8)
Channel             int64
Region              int64
Fresh               int64
Milk                int64
Grocery             int64
Frozen              int64
Detergents_Paper    int64
Delicassen          int64
dtype: object
```

2. Data Cleaning

- **Missing Data:** A strategy for handling missing data was determined (e.g., imputation using the median or dropping rows).



```
# Using median imputation for any potential missing values and dropping duplicates
df = df.fillna(df.median(numeric_only=True)).drop_duplicates()
print('Shape after cleaning:', df.shape)
```

- **Duplicates:** Exact duplicate rows were identified and removed from the dataset. The resulting dimension changes (number of rows) were reported.

3. **Feature Scaling**

To ensure all spending categories contribute equally to the distance metrics used by K-means and DBSCAN, the numeric features were scaled. **RobustScaler** was selected due to the presence of outliers in the spending data, making it a suitable choice for minimizing the influence of extreme values.

- **Categorical Features:** Channel and Region were retained for interpretation and statistical inference but were **not** used in the distance calculations for the clustering algorithms unless specifically justified.

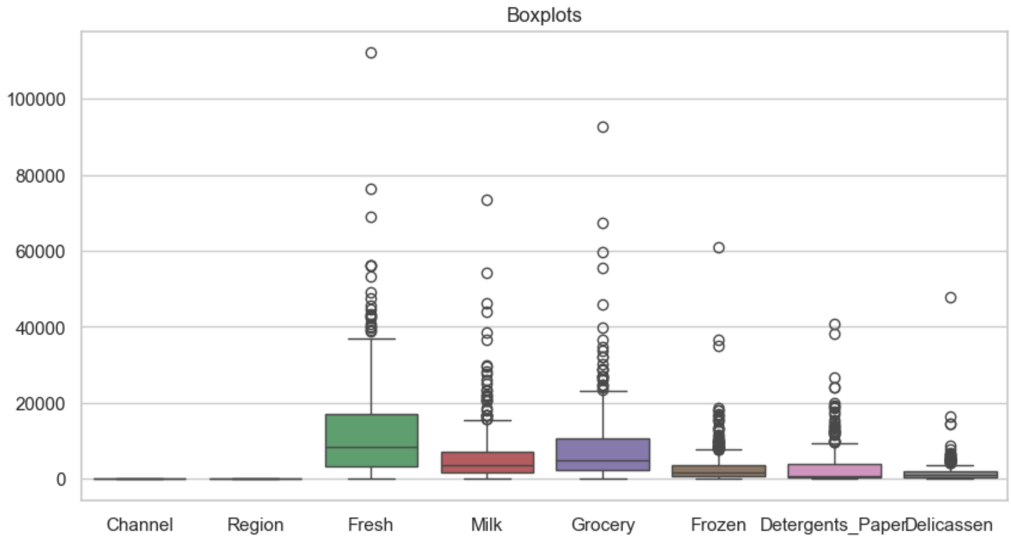
Part B: First Exploratory Data Analysis (EDA)

The goal here was to explore patterns in the spending categories. We generated summary statistics, boxplots, and histograms to check for skewness and identify outliers.

1. **Descriptive Statistics and Outliers**

- **Summary:** Descriptive statistics (mean, median, min, max, quartiles) for all spending categories were tabulated.
- **Boxplots:** Boxplots for the spending categories were generated. These plots highlighted significant right-skewness and the presence of numerous **outliers** across all categories, especially for high annual spending. The counts of these outliers were

explicitly reported.



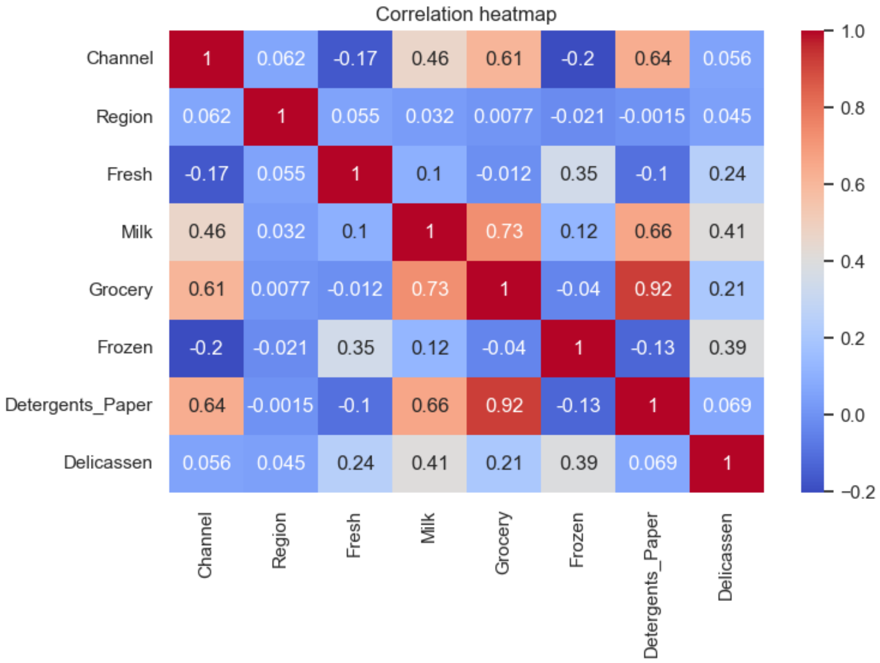
2. **Skewness and Log-Transformation**

Highly skewed spending variables were identified. To normalize the distributions and prepare the data for distance-based algorithms, these variables were **log-transformed**.

- **Visualization:** Histograms showing the distribution of the skewed variables were generated **before and after** the log-transformation to visually demonstrate the effect of normalization.

3. **Correlation Analysis**

A **correlation heatmap** was created for the spending categories. This allowed for the identification of potential co-purchasing patterns and multicollinearity.

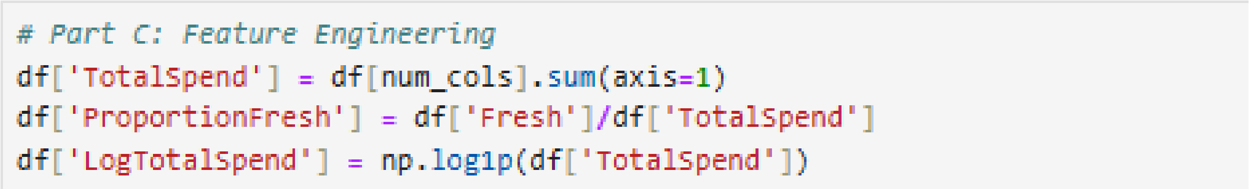


	Channel	Region	Fresh	Milk	Grocery	Frozen	Detergents_Paper	Delicassen
Channel	1	0.062	-0.17	0.46	0.61	-0.2	0.64	0.056
Region	0.062	1	0.055	0.032	0.0077	-0.021	-0.0015	0.045
Fresh	-0.17	0.055	1	0.1	-0.012	0.35	-0.1	0.24
Milk	0.46	0.032	0.1	1	0.73	0.12	0.66	0.41
Grocery	0.61	0.0077	-0.012	0.73	1	-0.04	0.92	0.21
Frozen	-0.2	-0.021	0.35	0.12	-0.04	1	-0.13	0.39
Detergents_Paper	0.64	-0.0015	-0.1	0.66	0.92	-0.13	1	0.069
Delicassen	0.056	0.045	0.24	0.41	0.21	0.39	0.069	1

- *Observation Example:* Strong positive correlations were observed between Grocery, Milk, and Detergents_Paper, suggesting that customers purchasing one often purchase the others, likely representing the Hotel/Restaurant/Café channel or similar businesses.

Part C: Feature Engineering & Aggregation

At least two new features were derived from the original spending data to enhance the clustering process and aid interpretation.



```
# Part C: Feature Engineering
df['TotalSpend'] = df[num_cols].sum(axis=1)
df['ProportionFresh'] = df['Fresh']/df['TotalSpend']
df['LogTotalSpend'] = np.log1p(df['TotalSpend'])
```

I. **TotalSpend:** Calculated as the sum of all six spending categories (Fresh + Milk + Grocery + Frozen + Detergents_Paper + Delicassen).
   - *Justification:* This feature captures the absolute size of the customer, which is a critical business metric for segmentation.

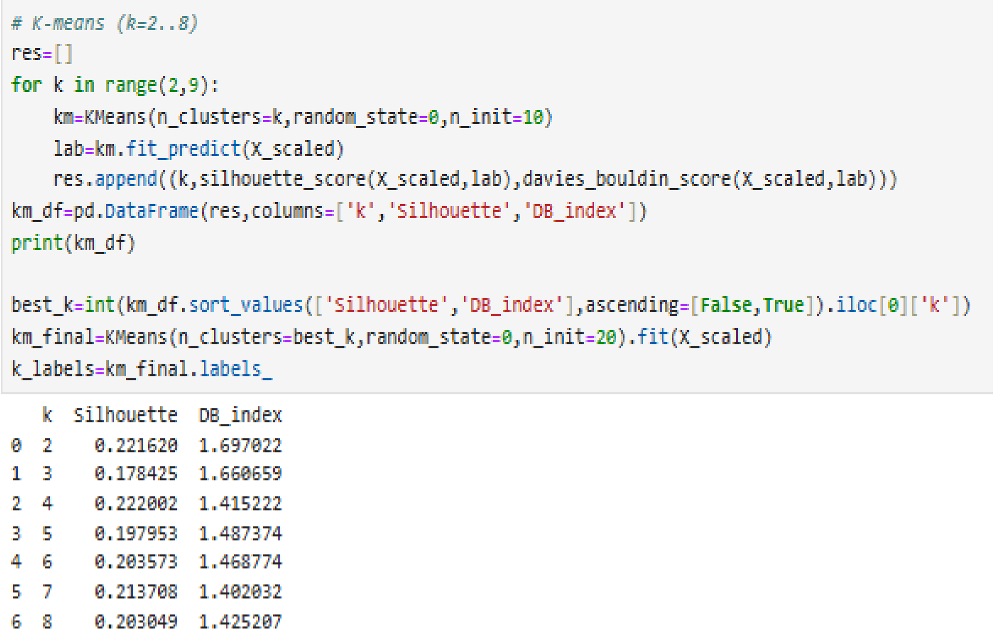
II. **ProportionFresh:** Calculated as Fresh/TotalSpend.
   - *Justification:* This feature captures the relative purchasing behavior, allowing clusters to differentiate between customers who prioritize fresh produce versus

those focused on processed/dry goods, regardless of their total spending volume.

III. **LogTotalSpend:** By using LogTotalSpend instead of TotalSpend in the clustering process, you reduce the influence of extreme outliers, ensuring that the clustering is based on the relative magnitude of spending, rather than being dominated by the largest few customers. This results in more robust and meaningful clusters.

Part D: Clustering Modelling & Parameter Selection

1. K-means Clustering



```
# K-means (k=2..8)
res=[]
for k in range(2,9):
    km=KMeans(n_clusters=k,random_state=0,n_init=10)
    lab=km.fit_predict(X_scaled)
    res.append((k,silhouette_score(X_scaled,lab),davies_bouldin_score(X_scaled,lab)))
km_df=pd.DataFrame(res,columns=['k','Silhouette','DB_index'])
print(km_df)

best_k=int(km_df.sort_values(['Silhouette','DB_index'],ascending=[False,True]).iloc[0]['k'])
km_final=KMeans(n_clusters=best_k,random_state=0,n_init=20).fit(X_scaled)
k_labels=km_final.labels_
```

```
   k  Silhouette  DB_index
0  2    0.221620  1.697022
1  3    0.178425  1.660659
2  4    0.222002  1.415222
3  5    0.197953  1.487374
4  6    0.203573  1.468774
5  7    0.213708  1.402032
6  8    0.203049  1.425207
```

K-means was executed for a range of **k values from 2 to 8**. The optimal *k* was determined using the following internal validation metrics:

Metric	Purpose
Silhouette Index	Measures how similar a point is to its own cluster compared to other clusters (Higher is better).
Davies-Bouldin (DB) Index	Measures the average similarity ratio between clusters (Lower is better).

The chosen optimal *k* value was justified based on the metric trends.

2. DBSCAN Clustering

- **Epsilon (ϵ) Selection:** A k-distance plot (using a value like the 4th Nearest Neighbor, where k=min_samples−1) was generated to visually identify the "elbow" point, which suggests a suitable initial value for the ϵ parameter.

- **Model Runs:** DBSCAN was run for multiple combinations of ϵ and min_samples. The following metrics were recorded for each run:
    1. Total number of clusters found.
    2. The total number of noise points (assigned to cluster -1).

3. **Algorithm Comparison**

   The best K-means model and the best-performing DBSCAN model were compared. The comparison focused on the following characteristics:

   - Total cluster counts.
   - Relative cluster sizes (uniformity).
   - The proportion of noise points identified by DBSCAN.

Part E: Second EDA & Statistical Inference

1. **Cluster Profiling (Centroids/Medoids)**

   The characteristics of the final chosen clusters for **both K-means and DBSCAN** were profiled. A table was generated showing the **cluster centroids (K-means) or medoids (DBSCAN)** for the original unscaled features (annual spending amounts). This step allows for a descriptive label to be assigned to each cluster (e.g., "High Fresh Spend Customer").

Statistical Significance Testing

a) **TotalSpend Difference:** To determine if the clusters represent genuinely different customer segments, a test was performed to see if the groups showed significant differences in the derived **TotalSpend** feature.
   - If the TotalSpend distribution was approximately normal, **ANOVA** was used.
   - Given the highly skewed nature of spending data, the non-parametric **Kruskal-Wallis H-test** was used. The resulting p-value determined the statistical significance.

b) **Categorical Association:** The relationship between the resulting clusters (for either K-means or DBSCAN) and the categorical variables (Region and Channel) was tested using a **Chi-square test**. This determined if a cluster was significantly associated with a specific purchasing channel (e.g., Horeca vs Retail) or region.

Part F: Presentation & Reflection

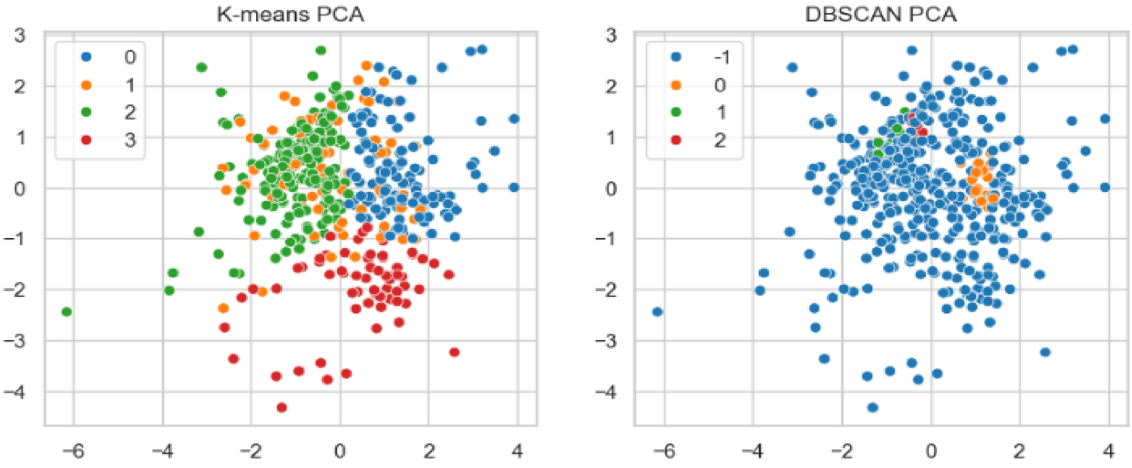
Cluster Visualization

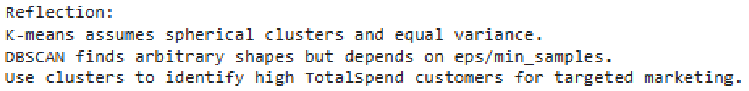
The final clusters were visualized in a two-dimensional plot using **Principal Component Analysis (PCA)**. This 2D projection showed the separation and boundaries of the final clusters for both algorithms. Crucially, the **DBSCAN noise points** were highlighted on this plot.

Reflection on Techniques

A discussion was provided contrasting the two algorithms:

- **K-means:** Discussed its underlying assumption of **spherical clusters** and equal variance, and how this assumption might be violated in real-world customer data.
- **DBSCAN:** Discussed its robustness to arbitrary cluster shapes but highlighted its high sensitivity to the ϵ and min_samples parameters.





```
Reflection:
K-means assumes spherical clusters and equal variance.
DBSCAN finds arbitrary shapes but depends on eps/min_samples.
Use clusters to identify high TotalSpend customers for targeted marketing.
```

Domain Recommendations

Actionable business recommendations were provided based on the cluster profiles (Part E).

- *Example Recommendation:* Recommendations should focus on leveraging the cluster insights, such as developing targeted promotions for a high-spend but low-frequency group or creating tailored services for channel-specific groups