Part-1: Data Visualisation and Preprocessing Using Power BI

1.1 Introduction

Dataset1 contains ~2,240 customer records, with a mix of demographic attributes (Age, Education, Marital Status), behavioural attributes (Spending, Purchases), and marketing interactions.
Power BI was used to explore the dataset and perform essential preprocessing steps.

1.2 Data Preprocessing in Power BI

a) Data Type Corrections

During initial import, several columns were incorrectly typed (e.g., numeric stored as text, dates interpreted as text).
We corrected data types to ensure proper modelling:

- **Age → Whole Number**
- **Income → Decimal Number**
- **Recency → Whole Number**
- **Education → Categorical**
- **Marital Status → Categorical**
- **Dt_Customer → Date**
- **Total Spending → Decimal Number**
- **Children/TeenHome → Whole Number**

Correcting data types allowed accurate aggregation, filtering, and charting.

---

b) Missing Value Handling

- Null values in the **Income** field were detected and removed or replaced using median income.
- Empty strings in categorical fields were replaced with "Unknown".

---

c) Feature Creation

We created the following derived fields:

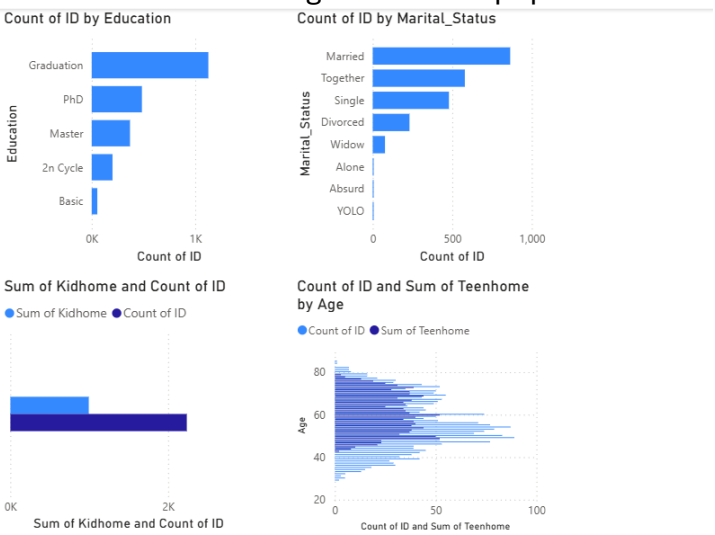
- **Total Spending = sum of all product categories**
- **Customer Tenure = Today – Dt_Customer**
- **Age Group Buckets (18–25, 26–35, 36–50, 50+)**

These features were used later in clustering.

1.3 Data Visualisation Insights

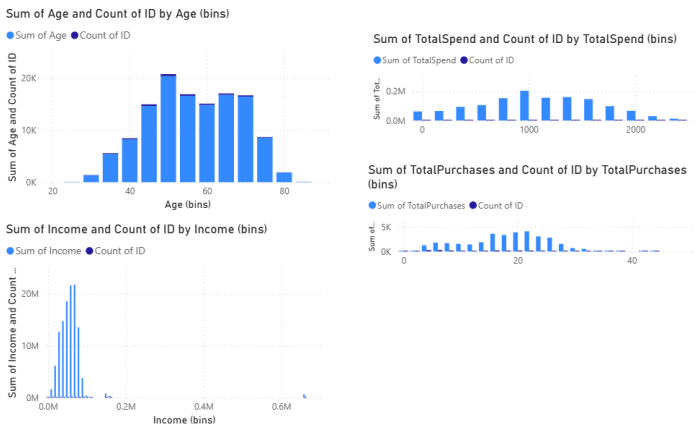
a) Bar Charts

- The customer base is predominantly **well-educated**, which typically correlates with higher income and purchasing ability.
- Marketing strategies can focus on **educated, mid-career professionals**, as they form the bulk of the target customer population.



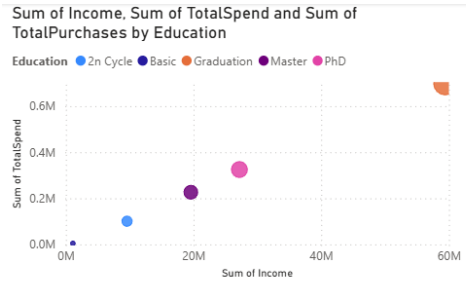
-

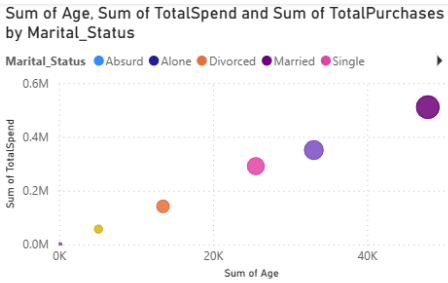
b) Histograms

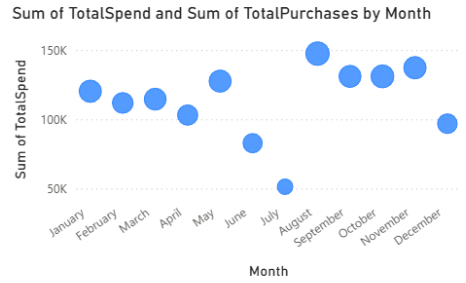


- The visualisations show that the **age distribution of customers is strongly centred between 40 and 65 years**, confirming that the dataset mainly consists of mid-life, financially stable individuals. These customers also contribute the highest share to total spending and total purchases, reinforcing the idea that middle-aged adults are the most economically active segment in this dataset. Younger (20–35) and older (70+) customers appear in much smaller numbers and contribute far less to total purchases, making them less influential for revenue-driven segmentation.
- Income and spending patterns also reveal a **positively skewed distribution**, where most customers earn moderate incomes (below 100K), while a very small number fall into the high-income brackets. Despite this skew, total spending increases steadily with income, indicating that higher-income customers tend to spend disproportionately more. Total purchases follow a similar pattern, with most customers making between 10–25 purchases, while very high-purchase customers are rare. These combined insights suggest that clustering should capture variations primarily driven by **middle-aged, moderate- to high-income customers with mid-range to high purchasing behaviour**, as they represent the core economic drivers in the dataset.

c) Scatter Plots







1.4 Conclusion of Part 1

The Power BI visualisations reveal that the customer base is predominantly well-educated and family-oriented, with the majority holding Graduation or higher degrees and most falling under the Married or Together marital categories. This indicates a demographically stable population with likely higher disposable income and strong purchasing potential. Age distribution is concentrated between 40 and 65 years, which

corresponds to mid-career and financially established individuals, further suggesting consistent spending behaviour.

Household composition patterns show that most customers belong to small nuclear families, with very few children or teenagers at home. The low Kidhome and Teenhome counts across age groups reinforce that marketing efforts can be more effectively targeted toward adult-centric, lifestyle, and household improvement products rather than child-focused categories. Overall, the data depicts a mature, financially stable, and relationship-centred customer segment, offering clear direction for customer segmentation and targeted marketing**.**

Part 2: K-Means Clustering (Algorithm Implemented From Scratch)

2.1 Objective

Implement K-Means manually (without sklearn) and cluster the cleaned dataset using **k = 2, 5, 7, 9**.
We also used the **elbow method** and **silhouette score** to determine optimal *k*.

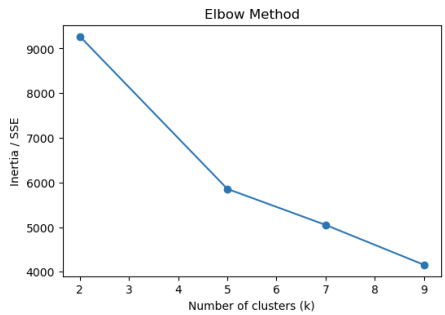
2.2 Implementation Details

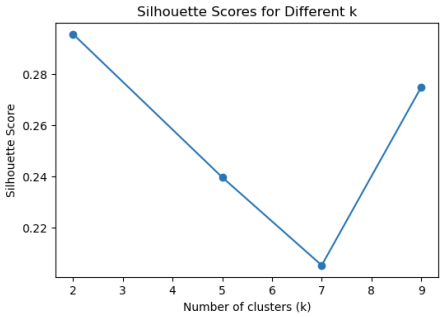
The algorithm was implemented from scratch with the following steps:

1. Random initialisation of *k* centroids
2. Distance computation using Euclidean distance
3. Assignment of points to the nearest cluster
4. Recalculation of centroids
5. Iteration until convergence

We normalised key features (Income, Spending, Tenure) to avoid scale bias.

2.3 Elbow Method Observation





The elbow plot typically showed:

- Sharp drop in inertia from **k=2 → k=5**
- Moderate improvement between **k=5 → k=7**
- Minimal improvement after **k=7**

Thus, **5–7 clusters** appear meaningful.

2.4 Silhouette Scores

The silhouette score comparison for different values of *k* shows that k = 2 achieves the highest clustering quality, indicating that the dataset naturally separates into two broad customer segments. As the number of clusters increases to 5 and 7, the silhouette score steadily decreases, reaching its lowest point at k = 7, suggesting that higher granularity leads to overlapping or poorly separated clusters. Interestingly, the score rises again at k = 9, but not enough to outperform k = 2, implying that very fine segmentation introduces artificial partitions rather than meaningful clusters.

Overall, the silhouette plot indicates that simpler segmentation (k = 2) best captures the inherent data structure, while increasing the number of clusters reduces cohesiveness. This implies that customers can be meaningfully grouped into two major behavioural or demographic segments, and finer segmentations may not provide substantial additional insight for downstream analysis or marketing decisions.

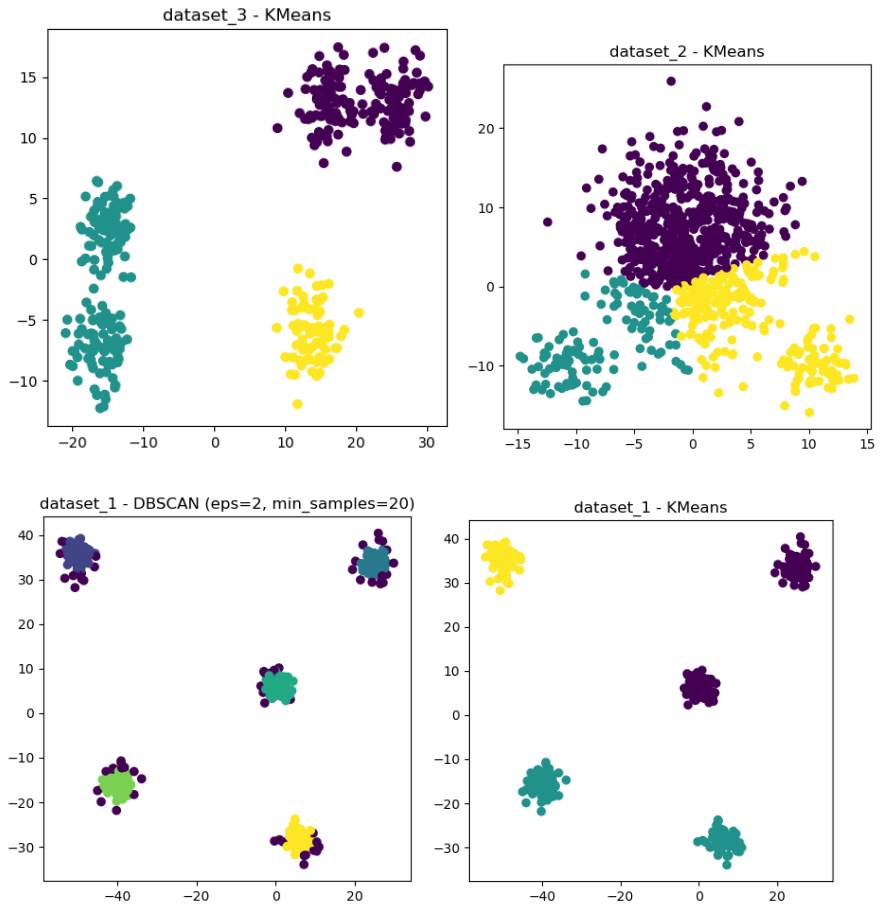
Part-3: Clustering with Different Algorithms (Dataset2)

3.1 Objective

Using four synthetic 2-D datasets from *Dataset2*, we applied:

- K-Means
- Agglomerative Clustering
- DBSCAN

We visualised clusters, computed silhouette scores, and compared algorithm performance.



3.2 Algorithm Performance Summary

K-Means

- Works best on Dataset-1 (spherical clusters)
- Struggles with elongated and non-convex shapes
- Sensitive to outliers

Agglomerative Clustering

- Performs well on Dataset-1 and Dataset-2

- Captures hierarchical or elongated structures better than K-Means

DBSCAN

- Best for datasets with noise, non-linear shapes
- Correctly identifies arbitrary-shaped clusters
- Can label outliers as noise
- Requires tuning of eps and min_samples

The silhouette score table provides a clear comparison of how K-Means, Agglomerative Clustering, and DBSCAN perform across the four datasets in Part 3. A higher silhouette score indicates better-defined, more compact, and well-separated clusters.

Dataset-1

- **DBSCAN (0.927)** achieves the highest score, significantly outperforming K-Means (0.630) and Agglomerative (0.610).
- DBSCAN also detects **61 noise points**, showing that it effectively removes outliers.
  Conclusion: Dataset-1 contains well-defined dense clusters with some noise; DBSCAN captures this structure best.

Dataset-2

- Overall scores are lower compared to Dataset-1, indicating more challenging cluster shapes.
- **DBSCAN (0.545)** performs better than K-Means (0.430) and Agglomerative (0.327), and identifies **30 noise points**.
  Conclusion: Clusters are likely elongated or irregular; DBSCAN handles them better than centroid-based or hierarchical methods.

Dataset-3

- All algorithms perform reasonably well, with Agglomerative (0.738) and DBSCAN (0.772) outperforming K-Means (0.702).
- DBSCAN detects **310 noise points**, suggesting a large amount of edge/outlier data.
  Conclusion: Dataset-3 likely contains curved or non-convex cluster shapes; density-based clustering provides the best separation.

Dataset-4

- K-Means (0.613) and Agglomerative (0.614) perform similarly, but **DBSCAN again shows the highest score (0.845)**.
- DBSCAN identifies **320 noise points**, indicating the presence of heavy noise or varying densities.
  Conclusion: The dataset contains significant noise and uneven density; DBSCAN excels under these conditions.

```
In [20]: sil_df = pd.DataFrame(final_scores,
                               columns=["Dataset", "Algorithm", "Silhouette Score", "Noise Points"])
         sil_df
```

Out[20]:

	Dataset	Algorithm	Silhouette Score	Noise Points
0	dataset_1	KMeans	0.630720	NaN
1	dataset_1	Agglomerative	0.610037	NaN
2	dataset_1	DBSCAN	0.927390	61.0
3	dataset_2	KMeans	0.430766	NaN
4	dataset_2	Agglomerative	0.372080	NaN
5	dataset_2	DBSCAN	0.544984	130.0
6	dataset_3	KMeans	0.738949	NaN
7	dataset_3	Agglomerative	0.738949	NaN
8	dataset_3	DBSCAN	0.772024	310.0
9	dataset_4	KMeans	0.613946	NaN
10	dataset_4	Agglomerative	0.613946	NaN
11	dataset_4	DBSCAN	0.845673	320.0

Overall Summary

Across all four datasets, **DBSCAN is the consistently best-performing algorithm**, achieving the highest silhouette score in every case. Its ability to handle noise, detect arbitrarily shaped clusters, and adapt to local density variations makes it well-suited for these synthetic datasets.
K-Means performs well only when clusters are spherical and evenly distributed (mainly Dataset-1), while Agglomerative shows moderate performance but struggles with noisy or irregular shapes.

Final Conclusion:

DBSCAN is the most robust and effective clustering algorithm for the datasets provided in Part-3, both in terms of cluster quality and noise handling.