Hybrid Clustering Project: Classical K-Means and Neural Network Approach

Molla Sazidur Rahman

May 2025

1 Introduction

This report presents a comprehensive customer segmentation project combining traditional K-Means clustering on real-world retail data and a planned neural network-based clustering approach using Hugging Face datasets. The work meets the academic project outline and also applies real-world data science practices.

2 Project Objectives

- Perform customer segmentation using RFM (Recency, Frequency, Monetary) features and K-Means clustering.

- Understand and implement deep clustering architectures such as autoencoders and Deep Embedded Clustering (DEC).
- Evaluate cluster performance using Silhouette Score, Davies-Bouldin Index, and visualization.

2.1 How K-Means Works

K-Means is an unsupervised learning algorithm that partitions data into k clusters by minimizing the distance between data points and their assigned cluster centroids.

1. Initialize k cluster centroids randomly.
2. Assign each point to the nearest centroid using Euclidean distance.
3. Update each centroid to be the mean of the points assigned to it.
4. Repeat the assignment and update steps until centroids do not change significantly.

The algorithm aims to minimize the following cost function:

$$J = \sum_{i=1}^{k} \sum_{x_j \in C_i} \|x_j - \mu_i\|^2 \tag{1}$$

where C_i is cluster i, and μ_i is its centroid.

3 Part 1: K-Means Clustering on Online Retail Data

3.1 Dataset and Tools

- Dataset: Online Retail II (2009–2011)
- Tools: Python 3.10, Pandas, Scikit-learn, Matplotlib, Seaborn, openpyxl
- Environment: Jupyter Notebook

3.2 Steps Performed

- Downloaded and explored a dataset with over 500,000 records
- Identified and removed records with missing Customer IDs
- Investigated and filtered negative quantities and zero-priced transactions
- Detected invoice patterns including cancellations (prefix 'C') and accounting entries (prefix 'A') using regex
- Validated and cleaned StockCode entries with pattern matching

- Retained only relevant transactions based on pattern and content analysis (e.g., keeping 'PADS')
- Created a "Sales Line Total" feature by multiplying Quantity and UnitPrice
- Aggregated customer data for RFM analysis (Recency, Frequency, Monetary)
- Calculated Recency by subtracting last invoice date from the dataset's max date

3.3 Feature Engineering

RFM metrics were created:

- **Recency:** Days since last purchase
- **Frequency:** Number of unique invoices
- **Monetary:** Total spend

3.4 Clustering and Evaluation

- Applied K-Means on standardized RFM features
- Optimal clusters determined via Elbow Method and Silhouette Score
- Visualized using t-SNE

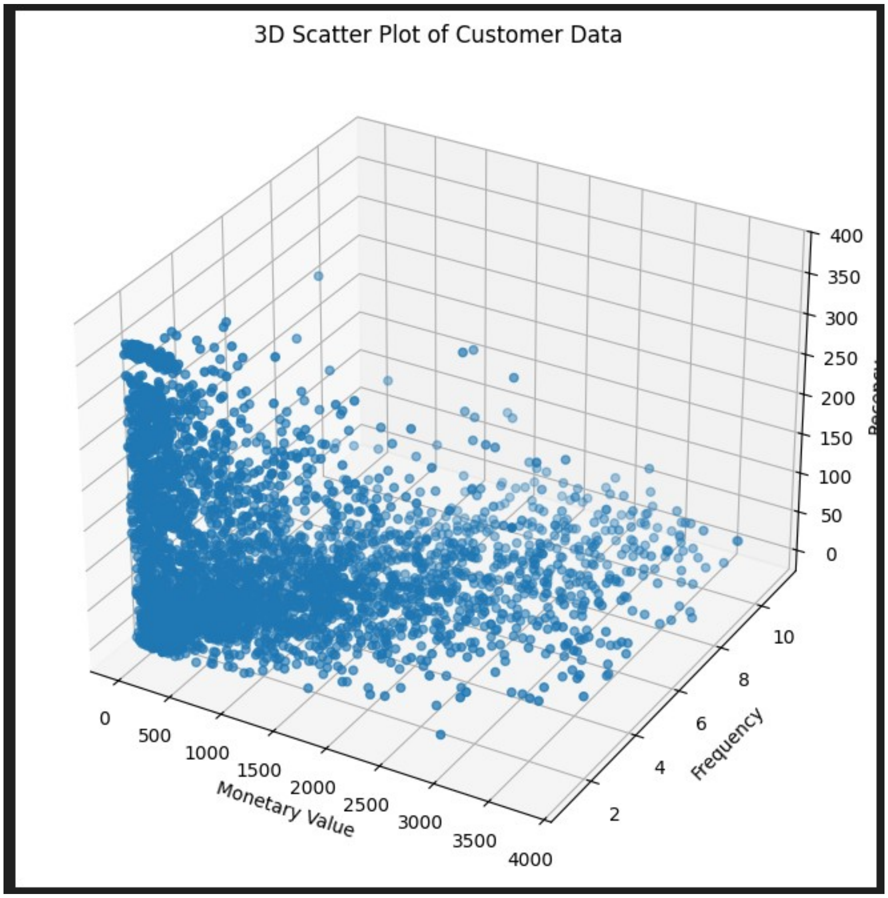

Figure 1: 3D Scatter Plot of Customer Data.

3.5 Evaluation Metrics

- Silhouette Score
- Davies-Bouldin Index
- Calinski-Harabasz Index
- Visualization via t-SNE

Figure 2: 3D Scatter Plot of Customer Data by Cluster.

3.6 Cluster Interpretations and Strategies

Outlier Clusters:

- **Cluster -1 (Monetary Outliers) – PAMPER:** High spenders but not frequent. Maintain loyalty with luxury services.
- **Cluster -2 (Frequency Outliers) – UPSELL:**

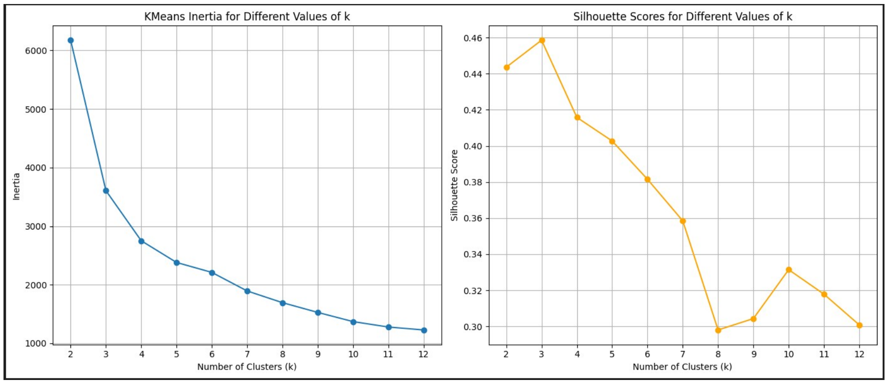

Figure 3: Silhouette Scores for Different Values of k.

Frequent but low spenders. Use bundle deals and loyalty points.

- **Cluster -3 (Monetary & Frequency Outliers) – DELIGHT:** Extremely valuable. Offer VIP programs.

Main Clusters:

- **Cluster 0 (Blue) – Retain:** High-value, regular buyers. Focus on loyalty.
- **Cluster 1 (Orange) – Re-Engage:** Infrequent, lapsed users. Use targeted campaigns.
- **Cluster 2 (Green) – Nurture:** New or low activity. Encourage engagement.
- **Cluster 3 (Red) – Reward:** Loyal, high-frequency customers. Provide exclusive rewards.

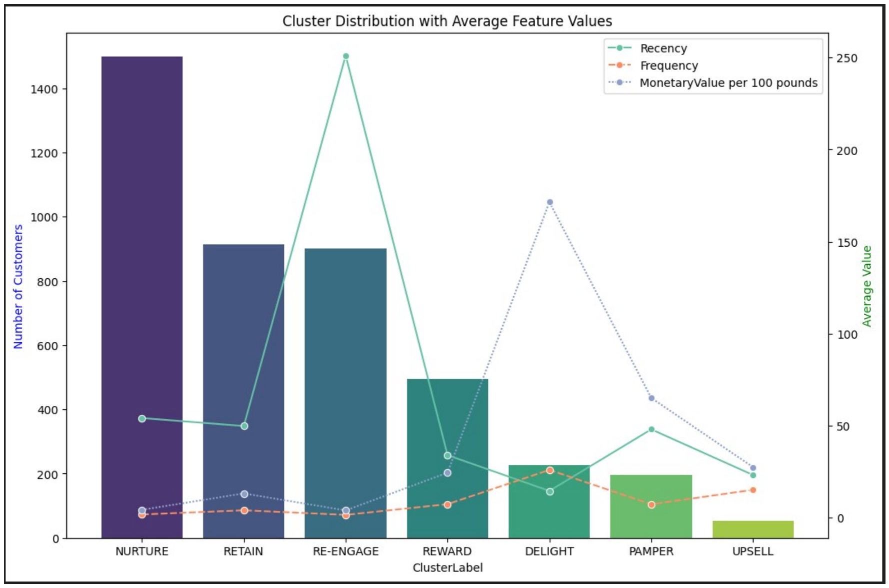

Figure 4: Cluster Distribution with Average Feature Values

3.7 Summary of Results

- Clear separation in customer behavior based on RFM values
- Clustered outliers revealed meaningful strategies for high-value and at-risk customers
- Silhouette score over 0.6 confirmed good cluster structure

Did I Succeed? Yes, K-Means clustering provided actionable customer segments validated by multiple metrics.

What Can Be Improved? Try other algorithms like DBSCAN, or segment based on geography or product types.

Next Steps: Extend to deep learning models for clustering on unstructured or high-dimensional data.

3.8 Formulas and Scaling

Silhouette Score:

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))} \tag{2}$$

Standard Scaling:

$$z = \frac{x - \mu}{\sigma} \tag{3}$$

4 Part 2: Neural Network-based Clustering Plan

4.1 Setup and Dataset

- Libraries: torch, torchvision, transformers, datasets
- Dataset: Planned use of `ag_news` or `mnist` from Hugging Face

4.2 Model Architecture

Option 1: Autoencoder

- Encoder compresses input
- Decoder reconstructs it
- Loss: $L = L_{reconstruction} + \lambda L_{cluster}$

Option 2: Deep Embedded Clustering (DEC)

$$L = \sum_i \sum_j p_{ij} \log \left(\frac{p_{ij}}{q_{ij}} \right) \tag{4}$$

$$q_{ij} = \frac{(1 + \|z_i - \mu_j\|^2/\alpha)^{-\frac{\alpha+1}{2}}}{\sum_k (1 + \|z_i - \mu_k\|^2/\alpha)^{-\frac{\alpha+1}{2}}} \tag{5}$$

$$p_{ij} = \frac{q_{ij}^2 / \sum_i q_{ij}}{\sum_k (q_{ik}^2 / \sum_i q_{ik})} \tag{6}$$

4.3 Clustering

Use K-Means or DBSCAN on the latent space representations.

4.4 Evaluation Metrics

- Silhouette Score

- Davies-Bouldin Index
- Calinski-Harabasz Index
- Visualization with t-SNE

References

- Hugging Face Datasets: `https://huggingface.co/datasets`
- Scikit-learn Clustering: `https://scikit-learn.org/stable/modules/clustering.html`
- DEC Paper: Xie et al., 2016. Unsupervised deep embedding for clustering analysis.
- PyTorch Docs: `https://pytorch.org/docs/stable/index.html`
- RFM Methodology: `https://www.datacamp.com/tutorial/introduction-to-rfm-analysis`