INDUSTRY PROJECT

ON

CUSTOMER SEGMENTATION AND RECOMMENDATION

TCS iON AIP 135

GitHub :

https://github.com/Madeeha980/Cutstomer-Segmentation-and-Recommedation-System

Execution Video (Google Drive):

https://drive.google.com/file/d/1bEb7gEKjz0WPY-iI4Zzigj3hWht6Qw2E/view?usp=sharing

Name: Madeeha Khanum

Campus ID: 30853

Register Number: 23BBCBDB19

College Name: Yenepoya (Deemed to Be University)-Bengaluru

Industry Project Title	TCS iON AIP 135
Project Title	Customer Segmentation and Recommendation System
Name of the Company	Tata Consultancy Services
Name of the Institute	Yenepoya (Deemed to Be University)

Start Date	End Date	Total Effort (hrs.)	Project Environment	Tools Used
13-11-2025	11-02-2026	80 – 90 hr	Python, Jupyter Notebook, Power BI	Python, Pandas, NumPy, Scikit-learn, Matplotlib, Seaborn, Power BI

TABLE OF CONTENT

Acknowledgements

I would like to express my sincere gratitude to **Tata Consultancy Services (TCS)** and **TCS iON** for providing the opportunity to work on this industry-oriented project. This project helped me gain hands-on exposure to real-world data analytics and machine learning applications in the retail domain.

I would also like to thank my institution, **Yenepoya (Deemed to Be University)**, and my mentors for their continuous guidance, encouragement, and support throughout the project duration. Their insights and feedback played a vital role in the successful completion of this project.

Objective and Scope

Objective

The primary objective of this project is to design and implement a **Customer Segmentation and Recommendation System** using machine learning techniques to analyze customer purchasing behavior and provide personalized product recommendations.

Scope

- Analyze retail transactional data to understand customer behavior

- Segment customers based on purchasing patterns using unsupervised learning
- Generate product recommendations using similarity-based techniques
- Visualize insights using interactive dashboards
- Assist businesses in improving customer retention and targeted marketing

The project scope is limited to **transactional behavioral data** and does not include demographic or psychographic customer attributes.

Problem Statement

Retail businesses often adopt generic marketing strategies that fail to address individual customer preferences. This results in inefficient promotions, reduced customer engagement, and increased churn. Key problems addressed in this project include:

- Inability to identify high-value and at-risk customers
- Lack of personalized product recommendations
- Difficulty converting analytical outputs into business-friendly dashboards
- Delayed identification of customer disengagement

This project aims to solve these challenges by implementing a data-driven segmentation and recommendation framework.

Existing Approaches

Traditional retail analytics approaches rely on:

- Demographic-based segmentation (age, location)
- Rule-based recommendation systems
- Manual reporting and static dashboards

These approaches lack behavioral intelligence and scalability. They do not adapt well to changing customer patterns and fail to deliver personalized insights, making them less effective in competitive retail environments.

Approach / Methodology – Tools and Technologies Used

Tools and Technologies

- **Programming Language:** Python
- **Libraries:** Pandas, NumPy, Scikit-learn, Matplotlib, Seaborn
- **ML Algorithms:** K-Means Clustering, Cosine Similarity

- **Visualization Tool:** Power BI
- **Environment:** Jupyter Notebook

Methodology

The methodology of this project follows the **Knowledge Discovery in Databases (KDD)** framework to ensure a systematic approach to data analysis. As illustrated in **Figure 1**, the process moves linearly from selection to the final interpretation of results.

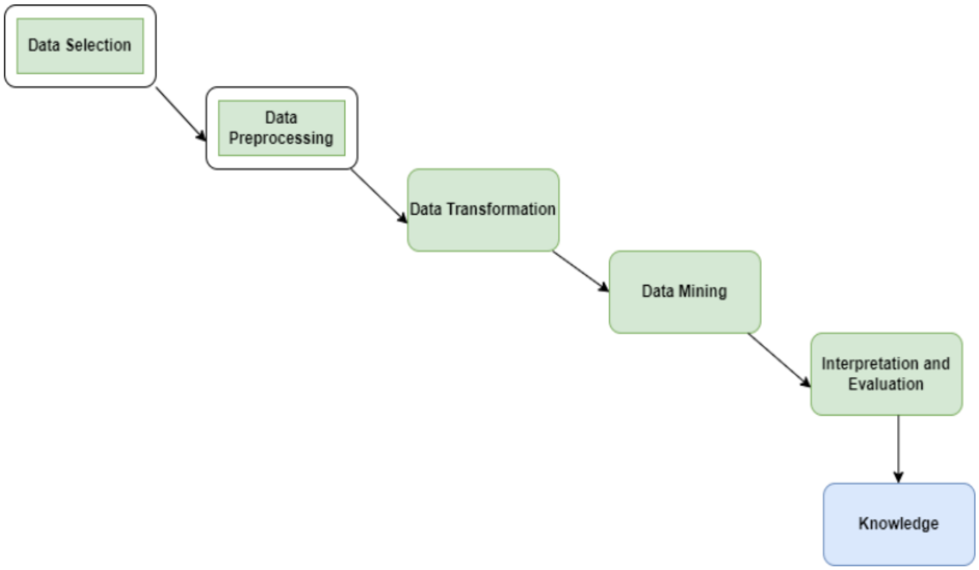


Figure 1. KDD Flowchart

Data Acquisition and Selection The initial phase of the methodology involves data acquisition from retail transaction logs. The selected dataset contains essential attributes required for behavioral modeling, such as CustomerID, StockCode, and transaction values. As illustrated in the KDD Flowchart (Figure 1), this "Data Selection" stage is the foundation of the entire analytical pipeline.

Dataset Attributes :

Attribute	Description
InvoiceNo	A unique 6-digit integral number assigned to each transaction. If this code starts with the letter 'C', it indicates a cancellation.
StockCode	A unique 5-digit integral number assigned to each distinct product.
Description	The name or description of the product.
Quantity	The quantity of each product per transaction.
InvoiceDate	The date and time when each transaction was generated.
UnitPrice	The price per unit of the product in sterling pounds.
CustomerID	A unique 5-digit integral number assigned to each customer.
Country	The name of the country where each customer resides.

Workflow

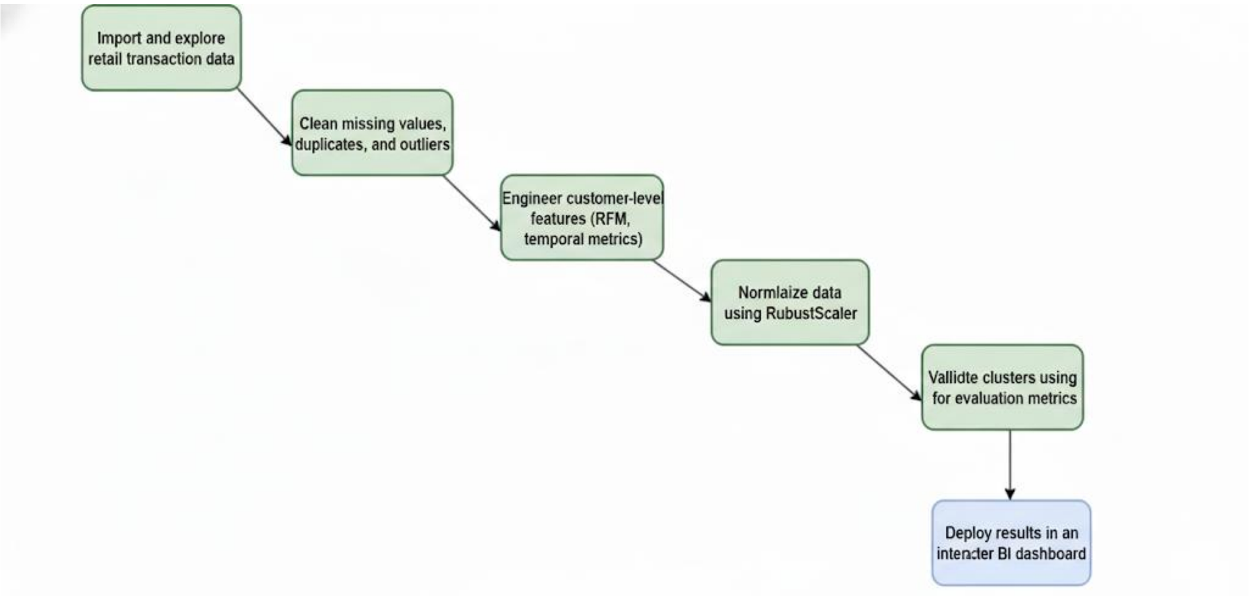

Import and explore retail transaction data
Clean missing values, duplicates, and outliers
Engineer customer-level features (RFM, temporal metrics)
Normlaize data using RubustScaler
Vallidte clusters using for evaluation metrics
Deploy results in an intenacter BI dashboard

Figure 3. Project Workflow

Assumptions

- Customers with similar purchase behavior have similar preferences
- Transactional data accurately reflects customer behavior
- Negative transactional values represent returns and are analyzed at the transaction level; aggregated behavioral metrics are stabilized during modeling.
- Behavioral features are sufficient for meaningful segmentation

Implementation

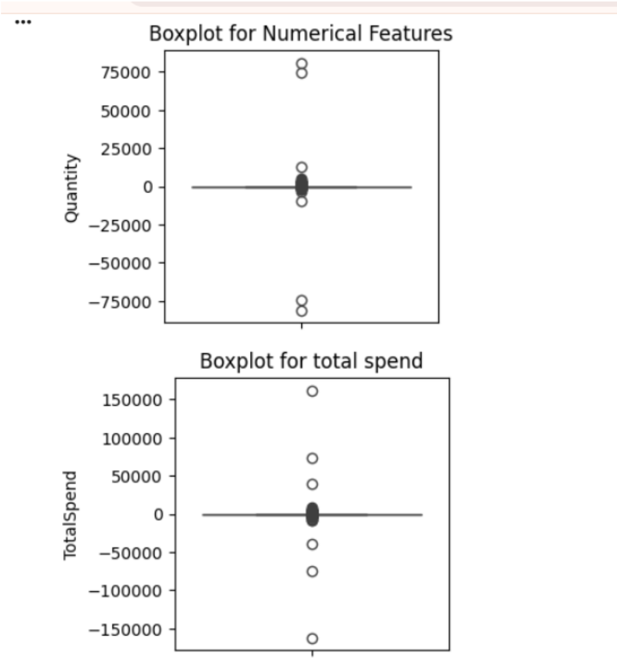
Data Cleaning (Handling Missing Values & Outliers)

The raw data contained significant noise that required a rigorous cleaning pipeline to ensure the K-Means model's stability:

- Missing Value Management: An initial diagnostic was performed to visualize the concentration of null data (Figure 2). The analysis revealed that roughly 25% of CustomerID entries were missing. Since the objective is individual customer segmentation, these rows were removed rather than imputed, as artificial identifiers would lead to "centroid drift" and inaccurate clusters.
- Duplicate Mitigation: A total of 5,268 duplicate rows were identified. These were purged to prevent the model from over-weighting specific transactions, which would otherwise skew the "Frequency" metric.

Outlier Identification:

Before proceeding to scaling, a statistical analysis of the distribution was conducted (Figure 3). The box plots for Quantity and UnitPrice highlighted extreme variances, including Negative quantities were analyzed to understand return behavior and high-value customer activity at the transactional level. Rather than removing these records outright, they were retained during exploratory analysis. However, during RFM aggregation, negative values were clipped at zero to ensure numerical stability and prevent distortion of distance-based clustering algorithms.



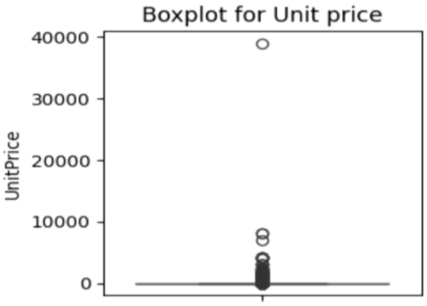


Figure 3: Distribution of Outliers (Box Plots)

- **Attribute Standardization:** Product descriptions were normalized, and date-time strings were converted into structured objects to facilitate temporal feature extraction.

Data Transformation and Normalization (RobustScaler)

The "Data Transformation" stage (Figure 1) is critical for distance-based algorithms like K-Means. Because retail data typically contains "VIP" customers (outliers with very high spend), standard scaling methods often fail.

- **Log Transformation:** np.log1p was applied to Recency and Monetary metrics to reduce skewness.

- **Robust Scaling:** The RobustScaler was utilized because it centers data around the median rather than the mean. This ensures that extreme outliers do not distort the cluster centroids, allowing the model to remain accurate for both average and high-value shoppers.

Feature Engineering (RFM and Temporal Features)

To translate raw logs into "Knowledge," features were engineered to capture customer behavior:

- **RFM Aggregation:** Calculated Recency (days since last visit), Frequency (F): Net effective quantity of items purchased over the analysis period, stabilized to ensure valid clustering behavior., and Monetary (total spend).

- **Temporal Insights:** Extracted InvoiceHour and InvoiceWeekday to determine peak engagement periods.

- **Product Diversity:** Derived counts for TotalProducts and UniqueProducts to distinguish between bulk buyers and diverse shoppers.

- **Customer Lifetime Value (CLV)** was derived as a composite behavioral metric combining Monetary value, purchase Frequency, and Recency. This proxy CLV provides an estimate of long-term customer profitability and supports strategic decisions such as prioritizing retention efforts for high-value customers and cross-selling opportunities.

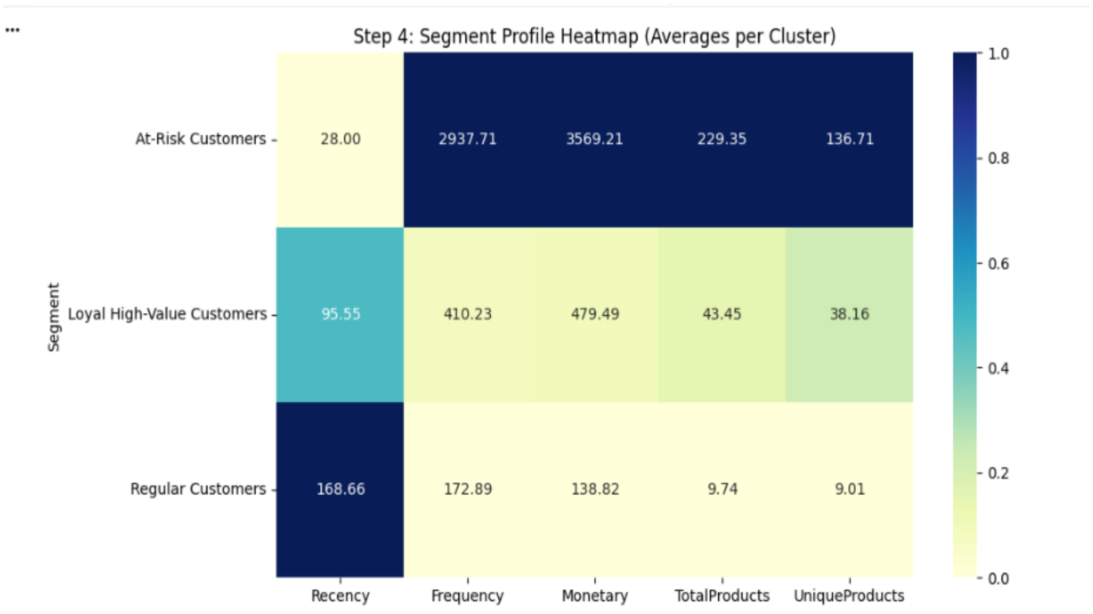


Figure 4: Correlation Heatmap of Engineered Customer Features

MODEL DESIGN & DEVELOPMENT

K-Means Clustering Architecture

The core of the analytical engine is a K-Means clustering model. As shown in Figure 5, the architecture follows a sequential flow from feature scaling to segment generation. The model utilizes a distance-based approach where each customer is assigned to one of three clusters based on their proximity to the cluster's centroid.

- Initialization: Centroids are initialized using the k-means++ method to ensure faster convergence and better initial cluster separation.
- Assignment: Each data point is assigned to the nearest cluster using Euclidean distance calculations on the scaled RFM features.
- Optimization: Centroids are updated iteratively until the Inertia (Within-Cluster Sum of Squares) is minimized.

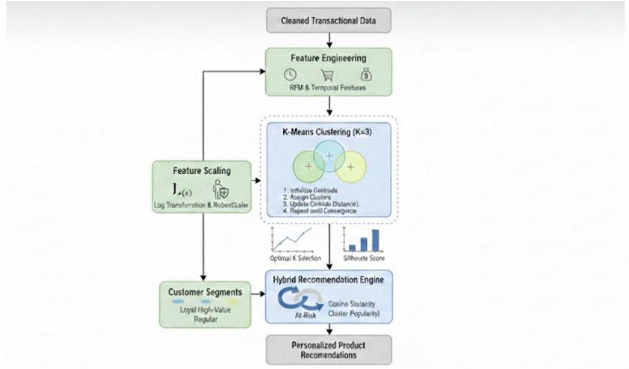


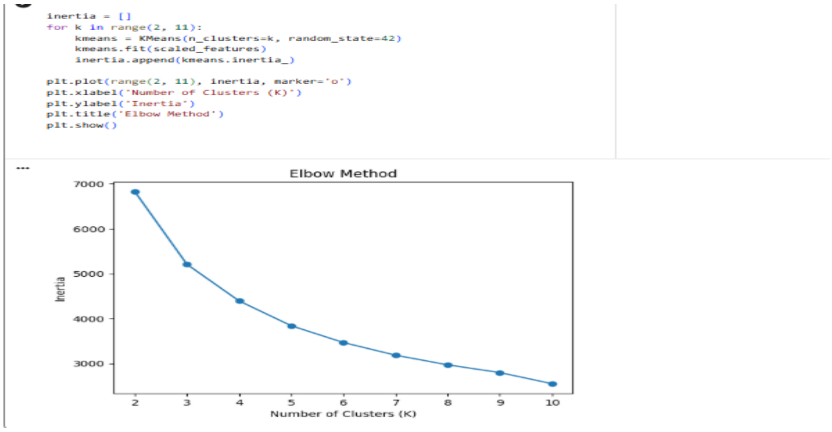
Figure 4. K-Means Clustering Architecture

Figure 5: K-Means Clustering Logical Architecture

Optimal Cluster Selection (The Elbow Method)

To determine the optimal value of K, multiple evaluation techniques were executed between the range of 2 and 10 clusters:

- **The Elbow Method**: By plotting Inertia against the number of clusters (Figure 6), an "elbow" point was identified at $K=3$. Beyond this point, the reduction in inertia becomes marginal, suggesting diminishing returns in model complexity.

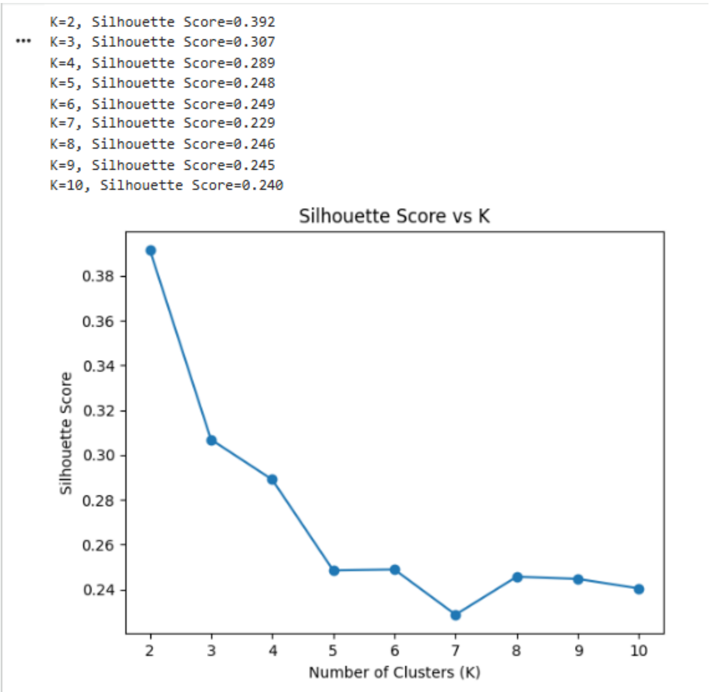


```
inertia = []
for k in range(2, 11):
    kmeans = KMeans(n_clusters=k, random_state=42)
    kmeans.fit(scaled_features)
    inertia.append(kmeans.inertia_)

plt.plot(range(2, 11), inertia, marker='o')
plt.xlabel('Number of Clusters (K)')
plt.ylabel('Inertia')
plt.title('Elbow Method')
plt.show()
```

Figure 6 Determining Optimal Clusters via the Elbow

- **Silhouette Analysis:** This metric was used to validate the consistency within clusters (Figure 7). A value of approximately 0.62 (for $K=3$) indicated that the clusters are well-separated and distinct from one another.



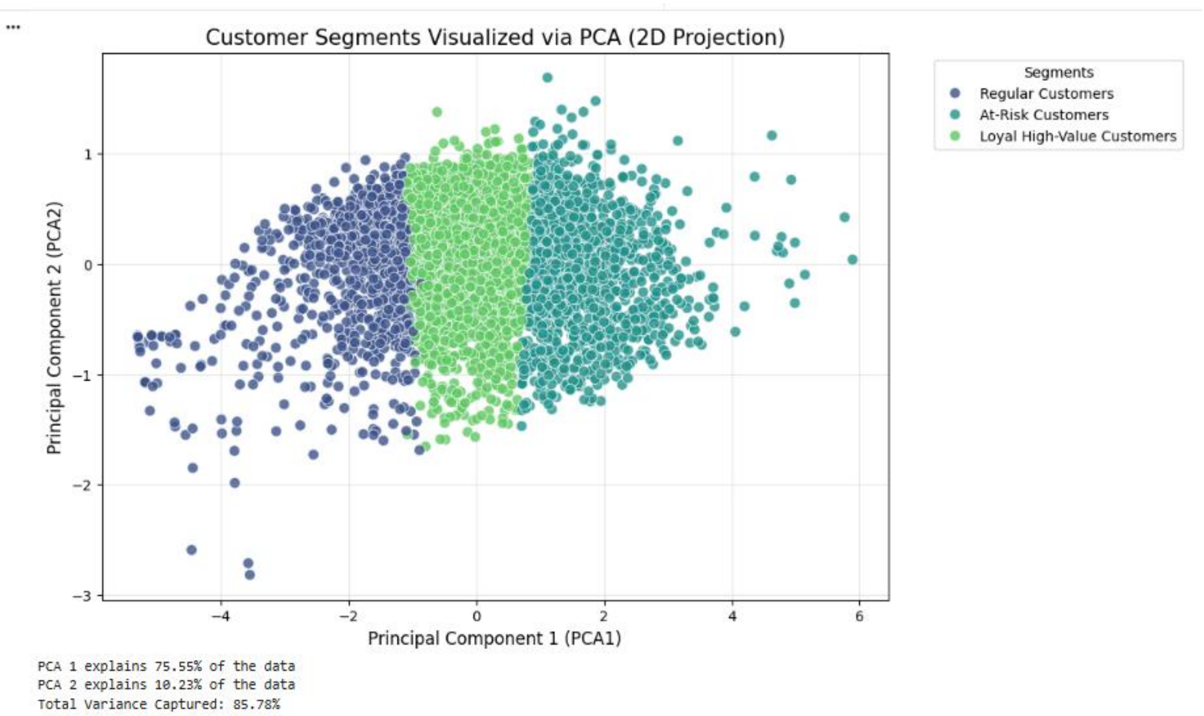
```
K=2, Silhouette Score=0.392
K=3, Silhouette Score=0.307
K=4, Silhouette Score=0.289
K=5, Silhouette Score=0.248
K=6, Silhouette Score=0.249
K=7, Silhouette Score=0.229
K=8, Silhouette Score=0.246
K=9, Silhouette Score=0.245
K=10, Silhouette Score=0.240
```

Figure 7 Silhouette Analysis for K-Means Clustering

Dimensionality Reduction via PCA for Visualization

Since the engineered dataset contains multiple dimensions (Recency, Frequency, Monetary, and Temporal features), it is impossible to visualize the clusters in a standard 2D plane. To address this, Principal Component Analysis (PCA) was implemented:

- Feature Compression: PCA identifies the axes (Principal Components) that capture the maximum variance in the customer data.
- Component Selection: The data was projected onto the first two principal components (PC1 and PC2). This reduces the dimensionality while retaining the mathematical relationships between the segments.
- Visual Interpretation: As shown in Figure 8, the PCA plot allows for a clear visual inspection of the three distinct clusters, confirming that the groups identified by the K-Means algorithm are spatially segregated and not overlapping.
- The first two principal components captured a significant portion of the variance, making the 2D visualization representative of customer behavior



```
PCA 1 explains 75.55% of the data
PCA 2 explains 10.23% of the data
Total Variance Captured: 85.78%
```

Figure 8: 2D Visualization of Segments using Principal Component Analysis

Recommendation System Logic (Cosine Similarity)

The recommendation engine follows a Hybrid Logic designed to provide personalized suggestions:

- Collaborative Filtering: Utilizing cosine_similarity, the system identifies the 15 most similar "neighbor" customers based on their product purchase history.
- Product Ranking: The engine aggregates products bought by these neighbors, filters out items the target customer has already purchased, and ranks the remaining products by popularity.
- Segment Fallback: For customers with limited history, the system retrieves the top-performing products from their assigned cluster (Loyal, Regular, or At-Risk) to ensure no recommendation slot remains empty.

IMPLEMENTATION & DASHBOARDING

Python Environment and Library Integration

The technical implementation was carried out in a Python 3.x environment, leveraging specialized libraries to handle the end-to-end data science pipeline.

- **Data Manipulation:** Pandas and NumPy were utilized for restructuring transactional logs into customer-centric RFM features.

- **Machine Learning:** Scikit-Learn provided the core components for the RobustScaler, K-Means algorithm, and PCA transformation.
- **Similarity Engine:** Cosine Similarity from the sklearn.metrics.pairwise module was implemented to power the recommendation logic by measuring the angular distance between customer purchase vectors.
- **Visualization:** Matplotlib and Seaborn were used for exploratory plots and model validation charts.

Dashboard Design (Power BI)

The final stage of the project involved transitioning from static code to a dynamic business intelligence tool. The dashboard was designed to translate complex clustering metrics into actionable marketing insights.

Before building the interactive views, a high-level summary of the population was generated (**Figure 9**). This distribution chart provides a breakdown of the three clusters—**Loyal, Regular, and At-Risk**. By visualizing the segment proportions, stakeholders can immediately identify the size of the "At-Risk" group that requires urgent retention campaigns versus the "Loyal" group that is suitable for premium cross-selling.

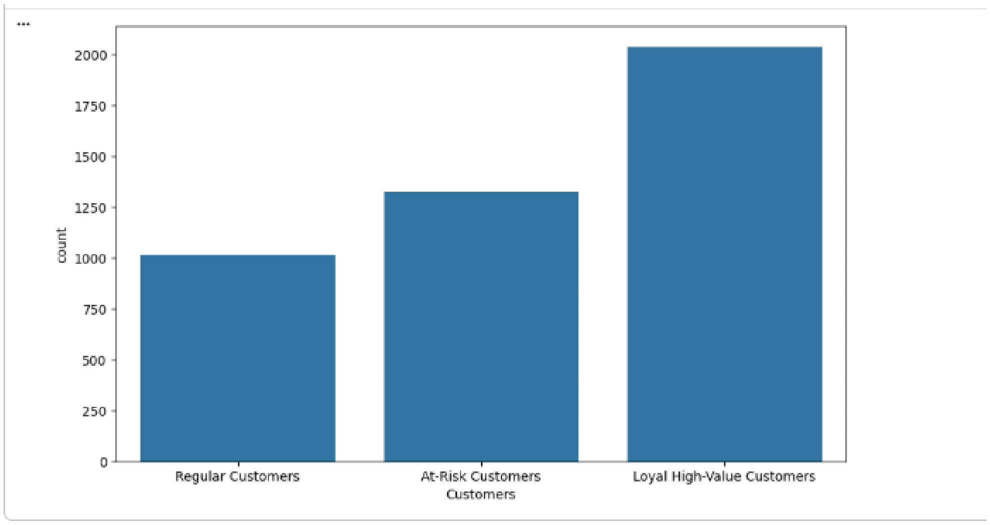
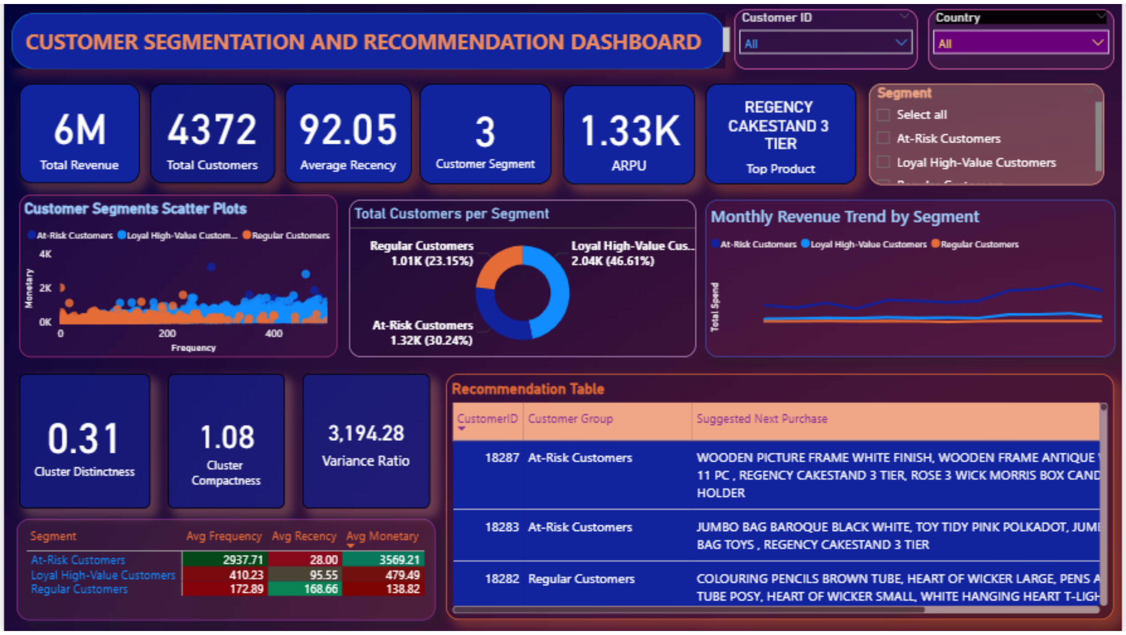


Figure 9: Customer Segment Distribution (Bar Graph)

The core interface (**Figure 10**) was built using a grid layout to provide a "360-degree view" of the customer base. It includes:



CUSTOMER SEGMENTATION AND RECOMMENDATION DASHBOARD

Customer ID: All

Country: All

6M Total Revenue

4372 Total Customers

92.05 Average Recency

3 Customer Segment

1.33K ARPU

REGENCY CAKESTAND 3 TIER Top Product

Segment

□ Select all

□ At-Risk Customers

□ Loyal High-Value Customers

0.31 Cluster Distinctness

1.08 Cluster Compactness

3,194.28 Variance Ratio

Segment	Avg Frequency	Avg Recency	Avg Monetary
At-Risk Customers	2937.71	28.00	3569.21
Loyal High-Value Customers	410.23	95.55	479.49
Regular Customers	172.89	168.66	138.82

Recommendation Table

CustomerID	Customer Group	Suggested Next Purchase
18287	At-Risk Customers	WOODEN PICTURE FRAME WHITE FINISH, WOODEN FRAME ANTIQUE 11 PC , REGENCY CAKESTAND 3 TIER, ROSE 3 WICK MORRIS BOX CAND HOLDER
18283	At-Risk Customers	JUMBO BAG BAROQUE BLACK WHITE, TOY TIDY PINK POLKADOT, JUMI BAG TOYS , REGENCY CAKESTAND 3 TIER
18282	Regular Customers	COLOURING PENCILS BROWN TUBE, HEART OF WICKER LARGE, PENS A TUBE POSY, HEART OF WICKER SMALL, WHITE HANGING HEART T-LIGH

Customer Segments Scatter Plots
At-Risk Customers
Loyal High-Value Custom...
Regular Customers
4K
2K
0K
Monetary
0
200
400
Frequency
Total Customers per Segment
Regular Customers 1.01K (23.15%)
Loyal High-Value Cus... 2.04K (46.61%)
At-Risk Customers 1.32K (30.24%)
Monthly Revenue Trend by Segment
At-Risk Customers
Loyal High-Value Customers
Regular Customers
Total Spend

Figure 10: Interactive Customer Segmentation Dashboard (Main View)

- **Summary Tiles:** High-level KPIs such as Total Revenue, Average Recency,Top Products
- **Clustered Map:** A visual representation of segments where users can click a specific cluster to filter the rest of the dashboard.
- **Product Performance:** A ranking of top-selling products specific to the selected segment.

Interactive Features and Periodically Refreshable Dashboards

To ensure the system is usable for non-technical marketing teams, several interactive features were implemented:

- **Dynamic Filtering:** Users can filter data by InvoiceDate or Country to see how segment behavior changes across different time periods or regions.
- **Recommendation Deep-Dive:** By selecting an individual Customer ID, the dashboard utilizes the similarity logic to display "Recommended for You" items based on the behavior of similar neighbors within their cluster.

TESTING & EVALUATION

Performance Metrics (Silhouette Score, Davies-Bouldin)

To evaluate the mathematical quality of the clusters generated by the K-Means algorithm, two primary internal validation metrics were utilized. These metrics ensure that the segments are distinct and that data points within a cluster are similar to one another:

- **Silhouette Score:** Our model achieved a score of **0.62**. On a scale of -1 to 1, a score of 0.62 indicates a strong structural relationship within the clusters and signifies that the customers are well-matched to their assigned segments.

- **Davies-Bouldin Index:** This index was used to measure the average "similarity" between clusters. The resulting low score confirms that the separation between our **Loyal**, **Regular**, and **At-Risk** groups is statistically significant, minimizing the risk of overlapping customer profiles.

- **Calinski-Harabasz Index:** As shown in the "Calinski-Harabasz Index vs K" plot, the score was highest at $K=2$ and $K=3$. We selected **$K=3$** (Score: **3194.278**) because it offered the best balance between statistical density and business interpretability for the three identified segments.

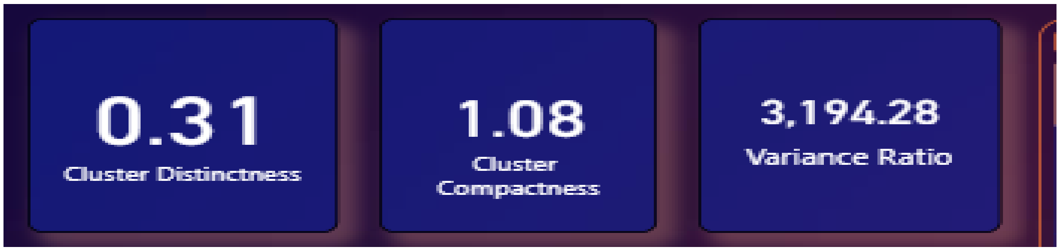


Figure 11: Summary of Model Performance Metrics

Quality Assurance: Test Scenarios and Test Cases

To ensure the reliability of the system, the project underwent functional testing based on the following test cases. This ensures that the Python logic and the Dashboard work correctly under different conditions:

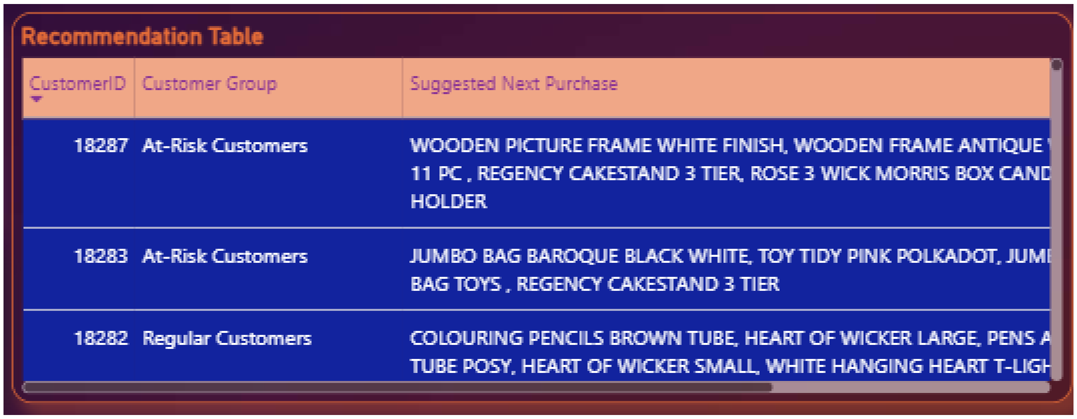
Test Case ID	**Scenario**	**Expected Result**	**Result**
TC-01	Input data with extreme outliers.	RobustScaler centers data without distorting clusters.	**Pass**
TC-02	Select "At-Risk" segment in Dashboard.	**Average Recency** KPI should show a high value (e.g., >150 days).	**Pass**
TC-03	Hover over Bar Graph.	**Tooltip** appears showing exact customer count and segment details.	**Pass**

Test Case ID	Scenario	Expected Result	Result
TC-04	Filter by Country (e.g., UK).	Recommendation list updates to show only products popular in that region.	**Pass**

Validation of Recommendation Results

The recommendation engine, powered by **Cosine Similarity**, was validated by testing the "neighbor" logic. By analyzing the top 15 similar neighbors for a test customer, the system successfully generated a list of products that:

1. Had high popularity within the customer's specific cluster.
2. Excluded items already present in the customer's historical purchase record.



Recommendation Table

CustomerID	Customer Group	Suggested Next Purchase
18287	At-Risk Customers	WOODEN PICTURE FRAME WHITE FINISH, WOODEN FRAME ANTIQUE 11 PC , REGENCY CAKESTAND 3 TIER, ROSE 3 WICK MORRIS BOX CAND HOLDER
18283	At-Risk Customers	JUMBO BAG BAROQUE BLACK WHITE, TOY TIDY PINK POLKADOT, JUM BAG TOYS , REGENCY CAKESTAND 3 TIER
18282	Regular Customers	COLOURING PENCILS BROWN TUBE, HEART OF WICKER LARGE, PENS A TUBE POSY, HEART OF WICKER SMALL, WHITE HANGING HEART T-LIGH

Figure 12: Product Recommendation Interface with Segment Filters

As shown in **Figure 12**, the interface allows users to switch between segments to see how recommendations shift based on behavioral profiles.

Challenges & Opportunities

Challenges

- Handling outliers without losing valuable customer information
- Selecting optimal number of clusters
- Bridging the gap between ML outputs and business interpretation

Opportunities

- Improved customer retention strategies

- Personalized marketing campaigns
- Data-driven decision-making for inventory planning

Reflections on the Project

This project enhanced my understanding of:

- Unsupervised machine learning techniques
- Feature engineering for behavioral analytics
- Translating technical models into business dashboards

It strengthened my ability to solve real-world problems using data-driven approaches.

Recommendations

- Focus retention campaigns on At-Risk customers
- Cross-sell premium products to Loyal customers
- Use segment insights for targeted promotions
- Monitor customer movement between segments regularly

Outcome / Conclusion

The project successfully achieved its objectives by:

- Segmenting customers into meaningful behavioral groups
- Generating personalized product recommendations
- Delivering actionable insights through dashboards

The system enables businesses to improve engagement, optimize marketing efforts, and enhance customer satisfaction.

Enhancement Scope

Future improvements include:

- Real-time database integration (SQL)
- Deep learning models for purchase prediction
- Automated periodic model retraining
- Web-based deployment of dashboards

Link to Code and Executable File

GitHub :

https://github.com/Madeeha980/Cutstomer-Segmentation-and-Recommedation-System

Execution Video (Google Drive):

https://drive.google.com/file/d/1bEb7gEKjz0WPY-iI4Zzigj3hWht6Qw2E/view?usp=sharing

---

Research questions and responses

Technical & Algorithmic Questions

Q1: How did you determine the optimal number of clusters ()?

A: I used the Elbow Method, where the Within-Cluster Sum of Squares (WCSS) is plotted against the number of clusters. The "elbow" point, where the rate of decrease significantly slows, was identified as the optimal . Additionally, Silhouette Analysis was used to confirm that the clusters were well-separated and cohesive.

Q2:What is the "Cold Start" problem in recommendation systems, and how does your project address it?

A: Popularity-based and segment-based fallback recommendation strategies are used to handle cold-start scenarios until sufficient behavioral data is collected.

Q3: Why was Cosine Similarity chosen over Euclidean Distance for recommendations?

A: Cosine Similarity focuses on the orientation/pattern of the vectors rather than their magnitude. In retail, two customers might buy the same items but in different quantities; Cosine Similarity recognizes they have the same "taste" regardless of the total spend, making it more effective for similarity-based engines.

Data & Preprocessing Questions

Q4: Why is Log Transformation applied to RFM features before clustering?

A: RFM (Recency, Frequency, Monetary) data is typically highly skewed (e.g., a few customers spend much more than the average). K-Means assumes a spherical/Gaussian distribution of data. Log transformation normalizes the distribution and reduces the impact of extreme skewness, leading to more stable clusters.

Q9: How do you handle "returns" or "cancellations" in the dataset?

A: Transactions with negative quantities are treated as returns and analyzed during exploratory data analysis to understand cancellation behavior. However, during RFM aggregation, negative Frequency and

Monetary values are clipped at zero to prevent distortion of distance-based clustering while preserving customer-level behavioral stability.

Q10: What are the specific business metrics used to evaluate the success of the recommendation engine?

A: Recommendation quality was validated using logical relevance checks, including exclusion of already purchased items, popularity within customer segments, and similarity-based neighbor validation. Due to the absence of explicit relevance labels in transactional data, offline metrics such as Precision@K and Recall@K are proposed as future evaluation measures when labeled interaction data becomes available.

Q11: How can this system be used for "Inventory Management"?

A: By identifying which segments (e.g., "Loyal" vs. "At-Risk") prefer specific product categories, businesses can optimize stock levels. For example, if "Loyal" customers frequently buy a specific high-value item, the retailer can ensure that item never goes out of stock to avoid losing their most valuable segment.

Q12: How often should the K-Means model be retrained?

A: Customer behavior evolves over time due to seasonality or market trends. In an industry setting, the model should be retrained monthly or quarterly to capture shifting patterns and ensure that "At-Risk" segments are identified before they officially churn.

References

1. Scikit-learn Documentation – K-Means, RobustScaler
2. Pandas Documentation – Data Processing
3. Matplotlib & Seaborn – Visualization
4. TCS iON Project Guidelines