DHIRUBHAI AMBANI INSTITUTE OF INFORMATION AND COMMUNICATION TECHNOLOGY,
GANDHINAGAR , GUJARAT, INDIA

Market Segmentation in SBI life Insurance

Prof. Arpit Rana

202418026 - Krisha Sompura **202418035 - Milan Nagvadiya**

202418041- Palak Jain **202418051- Sheetal Jain**

Overview :

This project aims to develop a customer segmentation model for SBI Life Insurance. The goal is to provide recommendations for savings plans, loans, wealth management, and other financial services tailored to specific customer groups.

The dataset summarizes the behavioral patterns of around 9,000 active credit card holders over six months. It contains 18 behavioral variables at a customer level, sourced from Kaggle.

Data Set : https://www.kaggle.com/arjunbhasin2013/ccdata

Problem Statement :

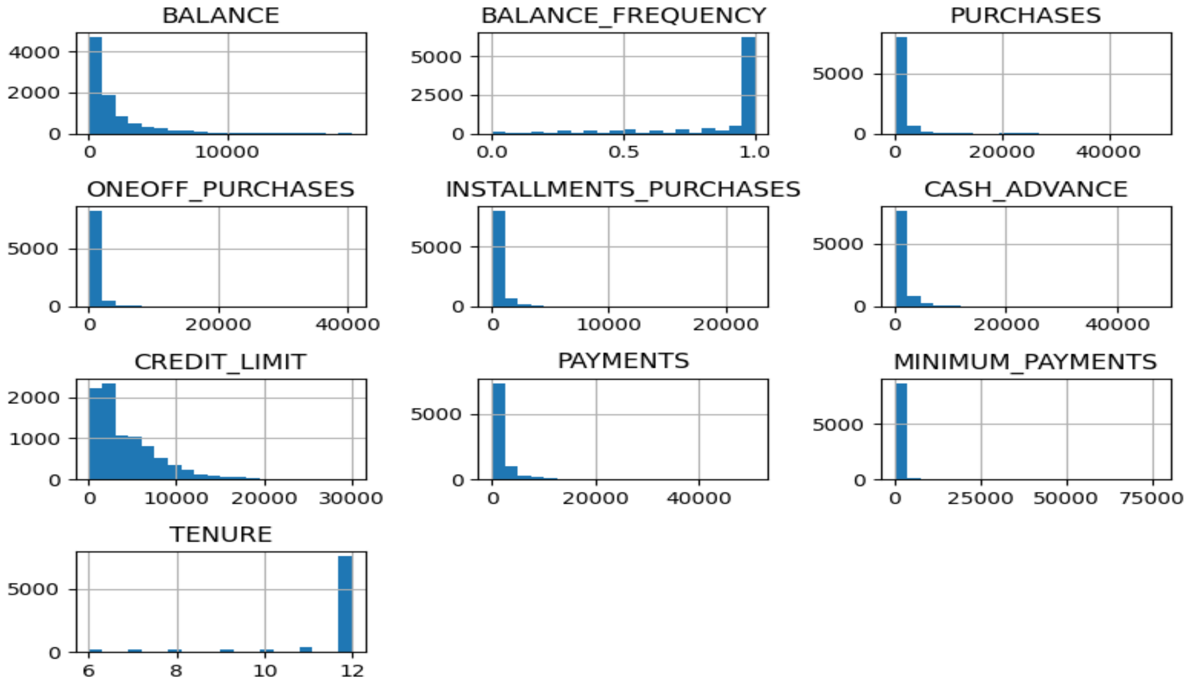
Accurate customer segmentation allows for better targeting of financial products and services. By identifying distinct customer clusters, SBI Life Insurance can optimize marketing strategies and improve customer engagement.

Data Collection :

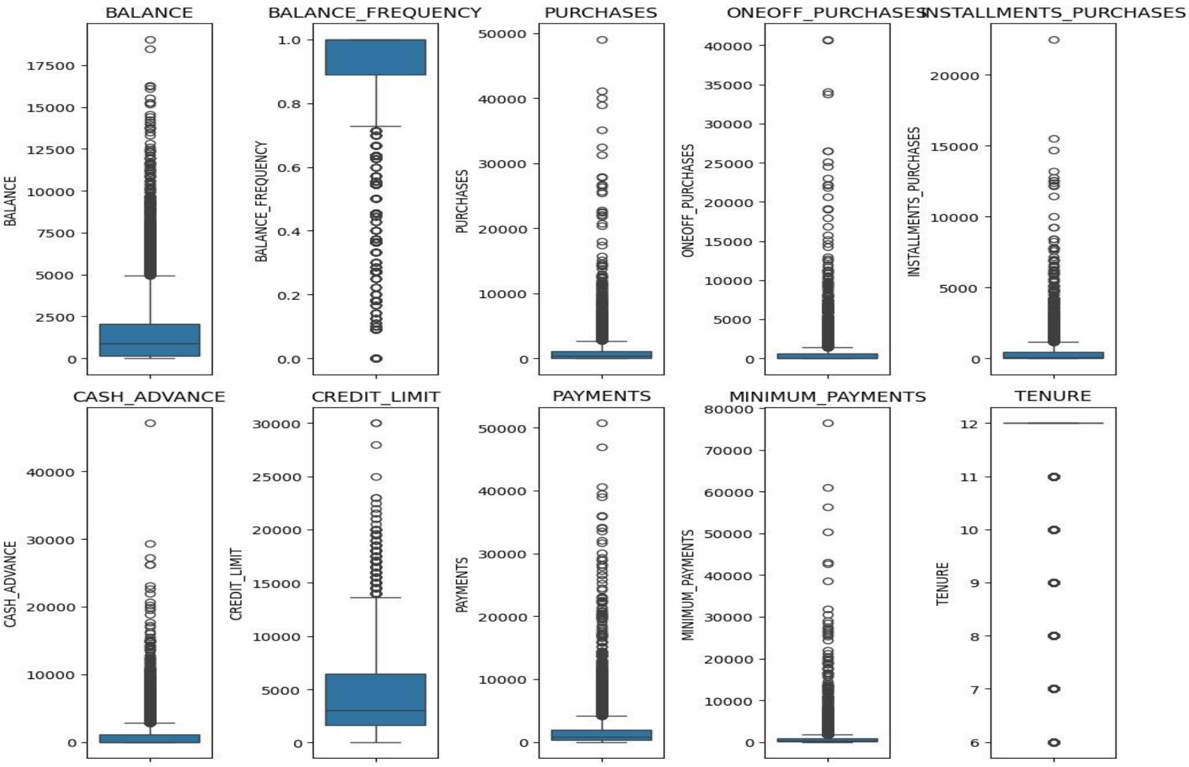
We will be taking the dataset from the datasets available on Kaggle. The dataset will be taken in the wave and csv format

Exploratory Data Analysis :

Distribution of all individual columns:



Outlier Analysis:

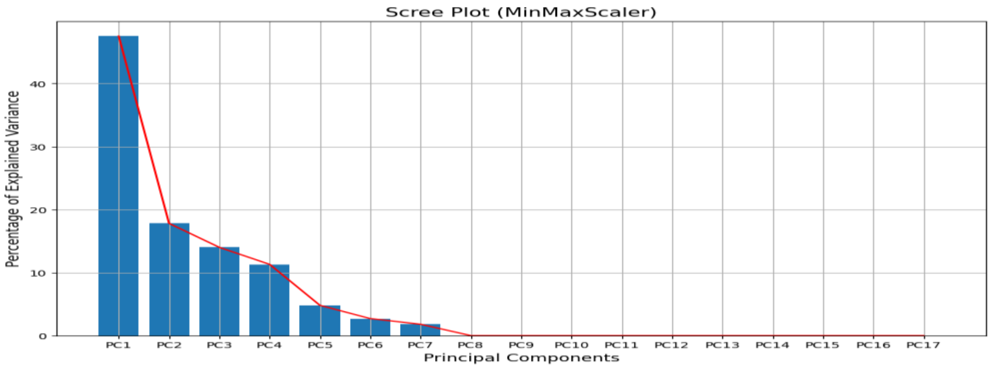


Data Preprocessing :

□ **Normalize Skewed Data:** Log scaling is commonly used for data that has a long tail or is heavily skewed. Large values are compressed more than small ones, making the distribution more balanced.

□ **MinMaxScaler :** Rescales the data set in such a way that all features values are in the range [0, 1]. This is done by feature-wise in an independent way.

□ **PCA:** Reduced dimensionality using Principal Component Analysis to simplify visualization and improve clustering performance.



Data Splitting:

Why Data Splitting Is Not Typically Used in Clustering

1. **No Labels**:
   - Clustering models group data without using pre-existing labels, so there is no "training" and "testing" phase as in supervised learning.
2. **Entire Dataset**:
   - The model works on the entire dataset to find inherent patterns and relationships. Splitting the data could cause the model to miss critical information.

Model Evaluation:

- Calculated **Silhouette Scores** for each clustering algorithm to determine the effectiveness of segmentation.
- Visualized the results in 3D scatter plots using PCA-reduced data

K-Means Clustering:

- Applied with different values of k to find the optimal number of clusters.
- Evaluated using Silhouette Scores.

Agglomerative Clustering:

- Hierarchical clustering approach with dendrogram visualization.

DBSCAN:

- Density-based clustering for detecting irregular cluster shapes.

Silhoutte Score :

K-Means	0.6865277503916535
Hierarchichal	0.6862802533330135
DBSCAN	0.6429220048286987

Best Model: DBSCAN

Reasons for Choosing DBSCAN as the Best Model:

Handles Noise and Outliers:

Identifies irregular data points (e.g., Cluster -1) as noise for better analysis.

No Need for Predefined Cluster Count:

Automatically determines clusters based on data density.

Identifies Arbitrarily Shaped Clusters:

Works well for irregular credit card usage patterns.

Works with Dense and Sparse Regions:

Groups dense customer behaviors while isolating sparse or niche patterns.

Handles Varying Cluster Sizes:

Identifies both large clusters (e.g., Cluster 1) and small ones (e.g., Cluster 7).

Feature Scaling Robustness:

Less sensitive to feature scaling compared to K-Means.

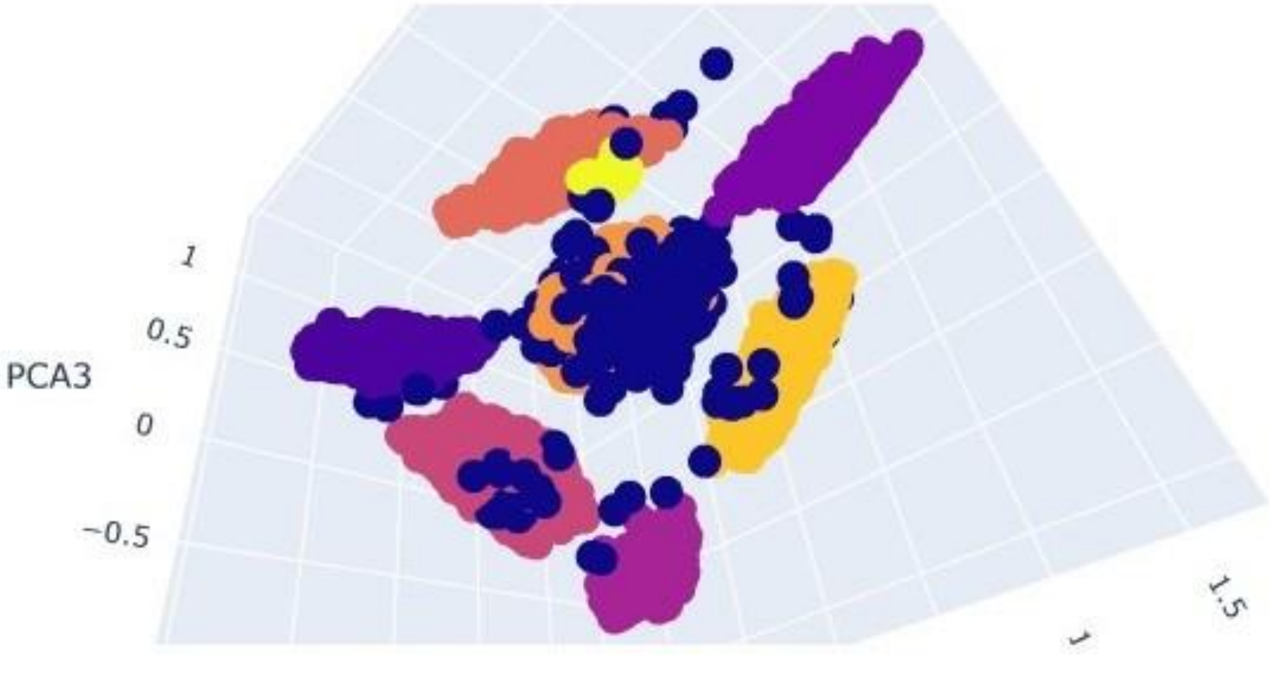
Parameter Flexibility:

Customizable eps and min_samples for domain-specific clustering.

Real-World Suitability:

Perfect for credit card data with noise, irregular patterns, and varying behaviors.

3D Clustering using DBSCAN Algorithm:



Key Insights from Clusters:

Cluster 0: Installment Purchasers

- Behavior: Prefer installments; low balances and activity outside installments.
- Recommendation: Promote installment offers, reward timely payments**.**

Cluster 1: Cash Advance Users

- Behavior: Rely on cash advances; high balances.
- Recommendation: Offer better terms on cash advances, promote diverse usage.

Cluster 2: One-Off Purchasers

- Behavior: Small, single transactions; limited usage.
- Recommendation: Provide cashback/rewards, increase credit limits.

Cluster 3: Low Spenders (No Installments)

- Behavior: Minimal spending, no installments.
- Recommendation: Incentivize installments, promote higher spending.

Cluster 4: Big Spenders (No Cash Advances)

- Behavior: High spending; avoid cash advances.
- Recommendation: Retain with loyalty programs, premium benefits.

Cluster 5: Low Financial Usage

- Behavior: Rarely use cards; minimal transactions.
- Recommendation: Target with trial promotions.

Cluster 6: High Credit Limit, Diverse Usage

- Behavior: Use all features; high-value customers.
- Recommendation: Reward with points, discounts, and personalized perks.

Cluster 7: Niche Segment

- Behavior: Specific, small usage patterns.
- Recommendation: Provide tailored offers based on preferences.

Number of customers in each cluster:

Cluster 0 (1797 customers): Moderate-sized group.

Cluster 1 (2038 customers): Largest group, significant focus needed here.

Cluster 2 (1065 customers): Mid-sized group, room for targeted strategies.

Cluster 3 (1735 customers): Large group, potential for optimization.

Cluster 4 (432 customers): Smaller group but high-value.

Cluster 5 (866 customers): Moderate group with low engagement.

Cluster 6 (783 customers): Premium customers.

Cluster 7 (16 customers): Niche segment, very small.

Pseudocode for DBSCAN:

Input:

- Dataset DDD
- Neighborhood radius ϵ\epsilonϵ
- Minimum points minPts\text{minPts}minPts

Output:

- Clusters
- Noise points

---

1. **Initialize:**
    - Mark all points in DDD as unvisited.
    - Set cluster label C=0C = 0C=0.
2. **Iterate over each point:**
    - For each unvisited point ppp:
        - Mark ppp as visited.
        - Find N(p)N(p)N(p), the set of points within distance ϵ\epsilonϵ of ppp.
3. **Check core point condition:**
    - If ∣N(p)∣≥minPts|N(p)| \geq \text{minPts}∣N(p)∣≥minPts:
        - Increment cluster label CCC.
        - Expand the cluster:
            - Add ppp to cluster CCC.
            - For each point qqq in N(p)N(p)N(p):
                - If qqq is unvisited:
                    - Mark qqq as visited.
                    - Find N(q)N(q)N(q).
                    - If ∣N(q)∣≥minPts|N(q)| \geq \text{minPts}∣N(q)∣≥minPts:
                        - Add all points in N(q)N(q)N(q) to N(p)N(p)N(p).
                - If qqq is not yet assigned to any cluster:

            - Add qqq to cluster CCC.

4. **Mark noise points:**
    - If |N(p)|<minPts|N(p)| < \text{minPts}|N(p)|<minPts:
        - Mark ppp as noise.
5. **Repeat until all points are processed.**

- ## Future Scope:

- Enhanced Models: Explore advanced clustering techniques like Deep Learning-based clustering.

- Customer Insights: Integrate additional data points like transaction history for richer segmentation.

- Business Applications: Use the segmentation to develop personalized marketing campaigns and product recommendations.

REFERENCE :

- https://www.kaggle.com/code/brsdincer/heartbeat-sounds-classification-analysis/input