CUSTOMER SEGMENTATION AND CROSS-SELLING RECOMMENDATION ENGINE FOR THE TELECOM INDUSTRY: A MACHINE LEARNING APPROACH

Janadhi Dilmini Nanayakkara

219445C

Master of Science in Telecommunications

Department of Electronics & Telecommunications Engineering
Faculty of Engineering

University of Moratuwa
Sri Lanka

July 2025

CUSTOMER SEGMENTATION AND CROSS-SELLING RECOMMENDATION ENGINE FOR THE TELECOM INDUSTRY: A MACHINE LEARNING APPROACH

Janadhi Dilmini Nanayakkara

219445C

Dissertation submitted in partial fulfillment of the requirements for the degree
Master of Science in Telecommunications

Department of Electronics & Telecommunications Engineering
Faculty of Engineering

University of Moratuwa
Sri Lanka

July 2025

DECLARATION

I declare that this is my own work and this Dissertation does not incorporate without acknowledgement any material previously submitted for a Degree or Diploma in any other University or Institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text. I retain the right to use this content in whole or part in future works (such as articles or books).

Signature: Date: 18/07/2025

The supervisor should certify the Dissertation with the following declaration.

The above candidate has carried out research for the Master of Science in Telecommunications Dissertation under my supervision. I confirm that the declaration made above by the student is true and correct.

Name of Supervisor: Prof. (Mrs) Dileeka Dias

Signature of the Supervisor: Date:

DEDICATION

To my parents, for their endless sacrifices and unwavering belief in me, and to all those who stood by me with patience and encouragement - this achievement belongs as much to you as it does to me.

ACKNOWLEDGEMENT

This research would not have been a success without the kind support and help of many individuals and organizations. I would like to extend my sincere thanks to all of them.

First of all, I wish to express my heartfelt gratitude to my academic supervisor, Prof. (Mrs.) Dileeka Dias, for her continuous guidance, insightful feedback, and invaluable encouragement throughout the course of this research. Her guidance and dedication were crucial for this study.

I would also like to extend my special thanks to Dr. Kasun Hemachandra, the course coordinator of the PG Diploma/ MSc in Telecommunications program, for his expert advice and dedication towards mentoring students and inspiring them to achieve their full potential.

A special note of appreciation goes to all individuals who offered their assistance in gathering the required datasets and insights which contributed significantly to the successful completion of this research.

To my friends and colleagues, thank you for your encouragement, thoughtful discussions, and moral support throughout this journey.

Finally, and most importantly, I am thankful to my family for the moral support, encouragement and guidance extended in many ways at all stages of my life. This accomplishment would not have been possible without their constant support.

ABSTRACT

In the telecommunication industry, personalized service delivery and effective cross-selling strategies are vital for enhancing customer engagement and driving revenue growth. This research presents a data-driven approach to customer segmentation and personalized recommendation, aiming to promote the transition from Double-Play (Voice and Internet) to Triple-Play (Voice, Internet and IPTV) service adoption.

The study applies unsupervised machine learning techniques, namely Gaussian Mixture Models (GMM) and K-Means clustering to segment customers based on demographic, billing, and service usage attributes. Multiple internal evaluation metrics, including the Silhouette Score, Calinski-Harabasz Index, Davies-Bouldin Index and the Elbow method are employed to determine the optimal clustering structure and validate the segmentation quality.

To generate personalized service recommendations, a collaborative filtering approach based on Singular Value Decomposition (SVD) is utilized. The model predicts the most suitable IPTV package for Double-Play customers by learning from existing customer interaction patterns. The performance of the recommendation system is evaluated using Root Mean Square Error (RMSE) and Mean Absolute Error (MAE).

The implementation leverages Python-based tools and libraries for data preprocessing, modeling and visualization. The research offers a practical and scalable framework for targeted marketing and intelligent service personalization, contributing to the advancement of data-driven strategies in the telecommunications domain.

Keywords: Machine Learning, Telecommunication, Customer segmentation, Recommendation Systems

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Abbreviation	Description
AIC	Akaike Information Criterion
BIC	Bayesian Information Criterion
CH	Calinski-Harabasz
DBI	Davies-Bouldin Index
DP	Double-Play
DQ	Data Quality
GMM	Gaussian Mixture Model
LTV	Lifetime Value
ML	Machine Learning
RFM	Recency, Frequency, Monetary
RMSE	Root Mean Squared Error
SVD	Singular Value Decomposition
TP	Triple-Play
VAS	Value-Added Services

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

In the modern telecommunications industry, customer retention and revenue maximization are key business objectives. Service providers continuously seek ways to enhance customer experience and increase average revenue per user (ARPU) by promoting value-added services. A common strategy is cross-selling, which involves encouraging existing customers to adopt additional services beyond their current subscription. This research aims to design a comprehensive framework that combines customer segmentation and product recommendation system to enhance cross-selling strategies in the telecom industry.

1.1 Background

The telecommunications industry is rapidly evolving and service providers seek innovative ways to enhance customer experience and maximize revenue. One such approach is through customer segmentation, which enables telecom operators to classify their customers into meaningful groups based on demographics, service usage patterns, billing behavior and service preferences. By understanding customer segments, businesses can develop personalized marketing strategies, improve customer retention, and optimize cross-selling opportunities.

Telecommunication companies offer a variety of services to their customers. Among them *"Double-Play"* and *"Triple-Play"* are widely used terminologies which are used to refer bundled service offerings in the field of telecommunications industry.

- **Double-Play** typically includes following two services.
    1. Voice (Landline)
    2. Internet (Broadband)
- **Triple-Play** includes three services bundled together.
    1. Voice (Landline)
    2. Internet (Broadband)
    3. Television services (IPTV)

A major opportunity in the telecom sector is to transition customers from double-play services to triple-play services. However, not all customers are equally likely to upgrade. Identifying the right customers for targeted recommendations can significantly improve adoption rates and marketing efficiency.

By analyzing historical customer data, including demographics, billing patterns, service usage and payment history, predictive models can identify patterns and behaviors that indicate a higher probability of adoption. These insights enable personalized marketing strategies, ensuring that promotional efforts are directed toward customers with a higher propensity to upgrade.

Traditional segmentation techniques may fail to capture complex customer behaviors, necessitating the use of advanced machine learning models.

1.2 Problem Statement

Traditional marketing strategies for service upgrades often use generic promotions, leading to inefficient targeting and lower conversion rates. Without a data-driven approach, telecom providers face challenges in;

- Identifying which double-play customers have a high likelihood of upgrading to triple-play.
- Understanding the behavioral and demographic factors that influence service expansion.
- Optimizing marketing campaigns for better engagement and conversion.

A customer segmentation-based approach can help to address these challenges by clustering customers into meaningful groups and developing a predictive cross-selling recommendation model to identify potential adopters of triple-play services. This study aims to bridge the gap between customer segmentation and predictive modeling to enhance service adoption strategies.

This research addresses the problem by implementing a customer segmentation model based on Gaussian Mixture Model (GMM) to group customers into meaningful clusters. Furthermore, a colloborative filtering approach based on Singular Value Decomposition (SVD) will be used to predict the likelihood of double-play customers upgrading to triple-play service packages.

1.3 Motivation and Importance

By leveraging machine learning-based segmentation and recommendation models, this research aims to enhance the ability to cross-sell IPTV services to existing broadband customers who are currently double-play adopters. The findings will help optimize marketing efforts, improve customer satisfaction and drive business growth by effectively transitioning double-play users to triple-play services.

1.4 Objectives and Scope

1.4.1 Objectives

- To bridge the gap between double-play to triple-play service adoption among customers of a telecommunications service provider through data-driven insights.
- To analyze customer data and identify distinct market segments using clustering techniques for effective cross-selling.
- To identify the most suitable IPTV package for targeted customers through a personalized recommendation engine.

1.4.2 Scope

The scope of this research includes the following.

- The study utilizes anonymized customer data including demographics, billing history, payment details and service usage details from a selected sample of customers of a telecommunications service provider for analysis.
- Preprocessing of customer data to ensure the dataset is suitable for machine learning models.
- Application of unsupervised clustering algorithms to identify meaningful customer segments for cross-selling.
- Evaluation of clustering quality by using internal validation metrics.
- Implementation of a collaborative filtering–based recommendation engine to suggest the most relevant IPTV package to targeted customers.

1.5 Originality and Novelty

This study presents a unique approach to bridge customer segmentation and cross-selling recommendations in the telecom industry by integrating unsupervised learning algorithms (GMM clustering) and a collaborative filtering-based recommendation technique. The key contributions of this research include,

- Combining GMM-based clustering and colloborative filtering-based recommendation models to develop an effective cross-selling recommendation engine.
- Optimizing cluster selection through cluster validation indices to ensure high quality segmentation.

- Applying predictive modeling specifically for triple-play adoption, addressing a gap in the existing literature.
- Providing actionable insights that can directly influence marketing strategies and customer engagement.

Using a data-driven approach, this research enhances the efficiency of cross-selling strategies, helping to personalize offers and maximize customer lifetime value.

1.6 Thesis Outline

This thesis is organized into five chapters:

- **Chapter 1** (Introduction) provides background, objectives and scope of the research.
- **Chapter 2** (Literature Review) examines existing methods and models for data preprocessing, customer segmentation and recommendation systems.
- **Chapter 3** (Methodology) details the proposed framework for customer segmentation and cross-selling recommendations, including data preprocessing, clustering techniques, and collaborative filtering.
- **Chapter 4** (Results & Discussion) presents the empirical results of the applied methodology, with analysis of segmentation accuracy and recommendation performance metrics.
- **Chapter 5** (Conclusion & Future Work) summarizes key findings, limitations, and potential extensions of the model.

CHAPTER 2

LITERATURE REVIEW

This chapter presents a brief review on existing literature related to data preprocessing, customer segmentation, recommendation systems, evaluation metrics, and implementation tools. It also identifies research gaps and details the contributions of this study.

2.1 Data Preprocessing

Data preprocessing is considered as the foremost step in data analytics. It involves transforming raw data to well-formed datasets so that data analytics can be applied. Data cleansing, integration, transformation and reduction are the most important steps of preprocessing. The main purposes of data preprocessing are to remove all irrelevant data and ensure consistency in the data representations [1]. Data preprocessing is a complex phase because its composition is determined by many factors including the processing algorithm, the data and the application. Thus, generalizing or standardizing data preprocessing was deemed unpractical [2]. Existing research presents diverse data preprocessing methodologies, as outlined in the following section.

Generally, the following major steps are used in data preprocessing.

- Data Cleaning
- Data Integration
- Data Transformation
- Data Reduction

The authors in [1] categorized data preprocessing techniques, as illustrated in Figure 2.1. In their work, a comparison is made between various data preprocessing methods. The issue of noisy data is addressed during the data cleansing phase. Furthermore, when data is collected from multiple sources, the data integration step is applied. The data transformation and data reduction phases are then used to convert data into a suitable format and to reduce its volume, respectively.

The authors in [4], discussed the four phases of data preprocessing, including data cleansing, data integration, data reduction, and data transformation along with different approaches for a variety of purposes. They have also explored several application areas where data preprocessing plays a foundational role, such as image processing, data quality management, pattern recognition, and natural language processing.

The authors in [5] proposed an effective preprocessing technique for financial data. Their approach consists of five preprocessing phases, each tailored to suit the specific characteristics of the selected dataset. The first phase is Data Cleaning, which

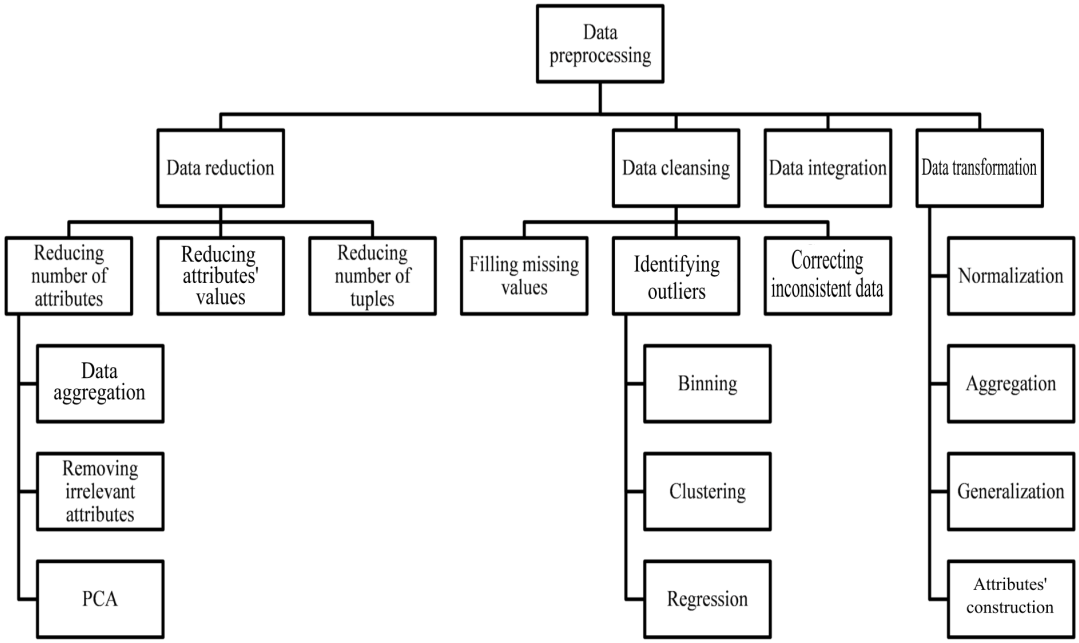

Fig. 2.1: Categorization of Data Preprocessing Techniques
(adapted from [1])

addresses inconsistencies and fills in missing values. The second phase, Data Integration, combines data from different sources and representations. The third phase, Data Transformation, involves normalization, aggregation, and generalization of the data. The fourth phase is Data Reduction, which reduces the volume of data while preserving its integrity. Finally, the fifth phase is Data Discretization, where the number of values for continuous attributes is reduced by dividing the attribute range into intervals, as continuous data is often not directly usable in certain data analysis tasks.

Furthur, authors in [2] identified multiple categories and sub-categories under which specific data preprocessing techniques are classified. They introduced a data preprocessing taxonomy based on the following extended definition of data preprocessing.

> **Definition:** *Data preprocessing includes any operation performed on the data, prior to information and value extraction and after data ingestion, to improve the quality of the data and prepare it for the consuming algorithms.*

The authors categorized these operations as highlighted by the taxonomy in Figure 2.2 to achieve many data quality characteristics including accuracy, unbiased, completeness, and traceability.

The authors in [6] propose an innovative data evaluation step to be performed prior to the actual data preprocessing process. In this step, various Data Quality (DQ)

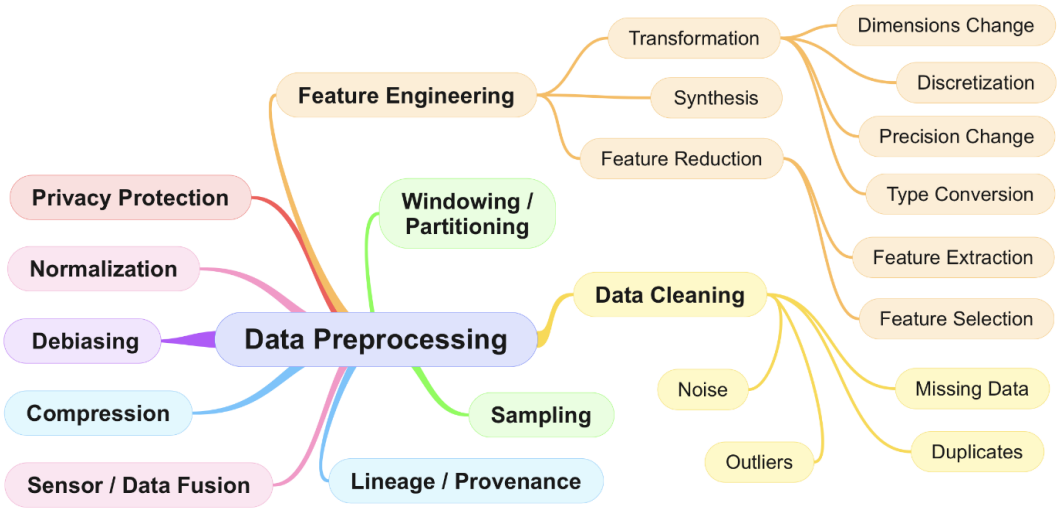

Fig. 2.2: Data Preprocessing Taxonomy (adapted from [2])

Note. The Colors Distinguish the Different Categories (Sub-categories Inclusive) for Improved Readability

checks and evaluation tests are conducted to assess the overall condition of the data. Based on the outcomes of these DQ checks, a hybrid data preprocessing approach is defined dynamically. This approach selects the most suitable preprocessing techniques according to the data's characteristics and DQ results. Each DQ check leads to the selection of a different hybrid preprocessing method, which combines multiple preprocessing techniques tailored to the specific needs of the dataset. A set of possible DQ checks and preprocessing mappings for a data set are summaried in Table 2.1.

The authors in [3] identified the absence of a unified library for data preprocessing, noting that data science practitioners often need to import and use multiple libraries to perform various tasks. To address this gap, they proposed a Python-based data preprocessing library called PrePy, which consolidates essential preprocessing functions. The steps incorporated in the library are illustrated in Figure 2.3.

From the literature it is clear that the data preprocessing plays a foundational role in the success of machine learning applications and data analytics. Techniques such as data cleaning, feature transformation, dimensionality reduction, and handling class imbalances have been widely explored in previous studies. Also, from the existing literature, it is obvious that all of the data preprocessing steps are not always necessary to improve the quality of the data. Hence, to choose the appropriate steps of data preprocessing for a dataset, it is important to understand the nature of the dataset and the problem to be solved.

TABLE 2.1: POSSIBLE DQ CHECKS AND THEIR PREPROCESSING MAPPINGS

Data Quality Check	**Pre-Processing Step**
Number of Columns	Remove unnecessary and unwanted columns if the number of columns > 100
Data Skewness	If skewness < 5% ignore, else remove skewed records from the dataset
Data Completeness	Missing values for an attribute should not be more than 25%, remove attributes with missing values/NaN> 25%
Data Volume	Number of records should always be from 1000 to 10000 for better analysis – correct data accordingly, generate records for the low count, aggregate records for high count
Data Distribution	Analyse Data distribution and normalize data
Data Randomness	Adjust data randomness to maintain data uniformity
Data Uniqueness	Remove Duplicate records

Note. Extracted from [6]

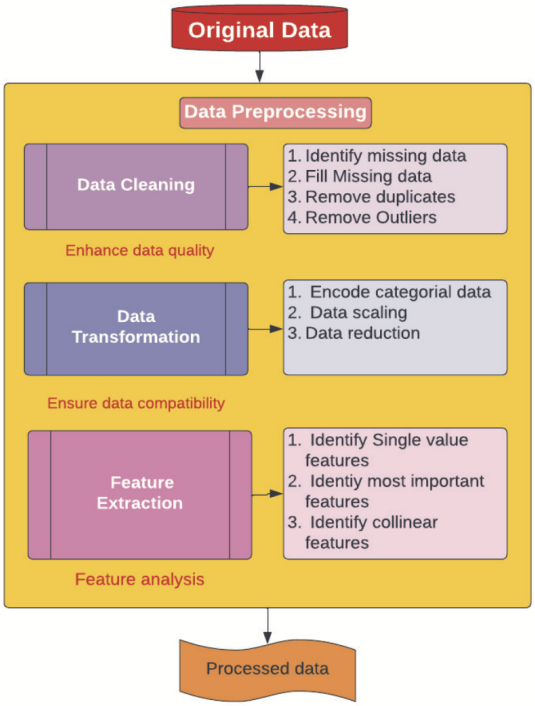

Fig. 2.3: Architecture of Data Preprocessing "PrePy" Package (adapted from [3])

2.2 Customer Segmentation

Customer segmentation is very important in order to enhance business and revenue. It involves dividing a customer base into groups of individuals that share similar characteristics [7]. Effective segmentation allows companies to tailor marketing strategies, optimize resource allocation and enhance customer satisfaction.

The traditional customer segmentation approaches such as using rule-based segmentation or a simple statistical methods to classify the customers, might not be practical when dealing with large sets of information. Therefore, machine learning based segmentation approaches such as clustering and probabilistic modeling have been found to yield better insights.

2.2.1 Overview on Clustering Techniques

Clustering is a widely used technique for customer segmentation, allowing the identification of natural groupings within customer datasets. Several algorithms have been proposed in literature. Figure 2.4 shows a high-level categorization of different clustering techniques.

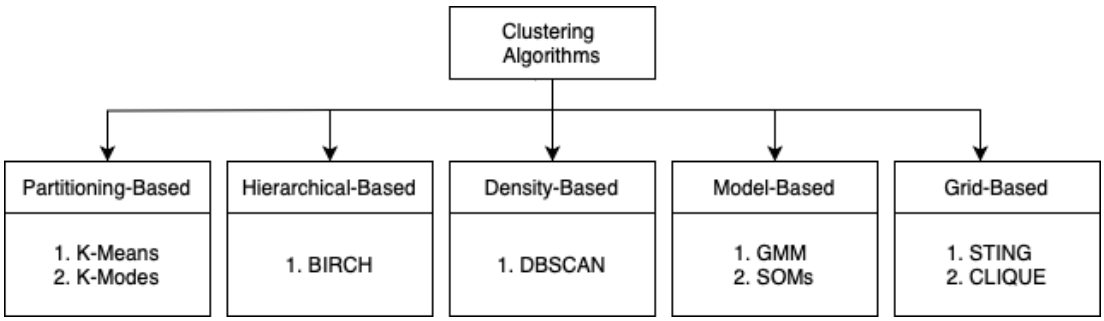

Fig. 2.4: Different Clustering Techniques

Out of the clustering techniques illustrated in Figure 2.4, two prominent unsupervised learning algorithms identified in the literature, K-Means and Gaussian Mixture Models (GMM) have been widely adopted in customer segmentation tasks across various domains.

K-Means Clustering

A common clustering algorithm, K-means was proposed by MacQueen that based on the error squares in 1967. The algorithm is convenient, fast and able to deal effectively with large databases [8]. It is a widely employed technique that groups data points into clusters based on similarity. In the field of customer segmentation, K-means has been used to identify distinct customer groups. However, it assumes spherical clusters and requires specifying the number of clusters beforehand [9].

- **Algorithm**: Uses the Expectation-Maximization (EM) algorithm to estimate cluster parameters probabilistically.
- **Optimal number of cluster selection**: Elbow method, Silhouette Score, Calinski-Harabasz (CH) Index, Davies-Bouldin Index (DBI)

Gaussian Mixture Models

GMMs learn their models by estimating the mean, variance, and weight coefficient for each cluster of data in the overall dataset. This is done by finding the maximum likelihood of the probability that a particular data point belongs to each cluster given the current mean, variance, and weight coefficient of that cluster [10].

- **Algorithm**: Uses the Expectation-Maximization (EM) algorithm to estimate cluster parameters probabilistically.
- **Advantages Over K-Means**: Unlike K-Means, GMM does not assume spherical clusters and is more flexible in handling complex datasets. Also, it is a soft clustering method.
- **Optimal number of cluster selection**: Akaike Information Criterion (AIC) , Bayesian Information Criterion (BIC) , Calinski-Harabasz (CH) Index, Davies-Bouldin Index (DBI)

2.2.2 Clustering Validation Indices

Silhouette Score

The Silhouette score measures the quality of clusters by computing the cohesion within clusters and the separation between clusters. Higher silhouette scores indicate better defined and well-separated clusters [11].

Elbow Method

The number of clusters(k) that determined using k-means, can be used as the input for this elbow method. Then measure the square number of intra-clusters. Intra-cluster distance is the distance between each data point to the corresponding centroid in each cluster center. The square number of intra-cluster for every number of clusters(k) that is determined in the beginning as input [12]. After measuring all values can be a plot that values according to the k values. The point that has the curvature in this plot, represents the optimum number of clusters [13].

Gap Statistic Method

The accumulation of the intra-cluster variance for the number of clusters(k), compares within their expected value under the null reference distribution. The optimal number of clusters(k) are displayed in the point that has a maximum gap statistic [12]. For any

clustering algorithm, this method can be used. [13]

2.2.3 Existing Literature

The authors in [9] have explored the utilization of hierarchical clustering techniques for mall customer segmentation. This technique systematically groups customers into clusters, revealing distinct segments within the mall's customer base. A comprehensive analysis of these clusters unveils profound insights into customer tendencies, preferences, and purchasing habits. These insights form a solid foundation for tailored marketing campaigns, personalized recommendations, and resource allocation within the mall.

The authors in [14] proposed a novel customer segmentation method that integrates Recency, Frequency, Monetary (RFM) , demographic, and Lifetime Value (LTV) data. The method followed a two-phase approach. In the first phase, customers were segmented using K-means clustering based on their RFM attributes. In the second phase, each of these initial clusters were further partitioned using demographic information. Finally, LTV is used to generate a customer profile for each segment, enhancing the understanding of customer value. This approach was applied to a dataset from an Iranian bank and yielded actionable managerial insights and recommendations.

The authors in [13] have experimentally demonstrated the application of K-Means clustering for customer segmentation. To determine the optimal number of clusters, they employed the Elbow Method, Silhouette Score and Gap Statistic. The approach presented in this paper offers a reliable and adaptable segmentation framework that can be applied effectively to subsets of datasets across various sectors.

The authors in [15] provide valuable insights into the segmentation and preferences of mobile internet users. Using data collected from an online questionnaire completed by over 500 respondents, they applied K-modes clustering and multiclass logistic regression analysis to develop customer segmentation and preference models. Since the K-means clustering algorithm is not suitable for categorical data due to its reliance on numeric calculations (e.g., Euclidean distance), the authors adopted the K-modes algorithm.

In [16], the authors address the challenge of clustering chemical compounds with similar antibacterial activity using unsupervised machine learning techniques. They applied K-means, Gaussian Mixture Models and mixtures of multivariate t-distributions to antibacterial activity datasets. To evaluate clustering performance and determine the optimal number of clusters, they emploed various Clustering Validation Indices such as within sum square, connectivity, Silhouette Width and the Dunn Index.

Table 2.2 provides a summary of customer segmentation approaches identified in the literature.

TABLE 2.2: SUMMARY ON THE LITERATURE ON CUSTOMER SEGMENTATION

Paper	Usecase	Segmentation Approach	Evaluation Metrics
[7]	e-commerce	K-means Clustering	Silhouette score WCSS
[11]	Retail	Agglomerative Clustering K-Means Clustering DBSCAN	Silhouette score
[8]	Banking	SOM-K-means	Not discussed
[9]	Mall	Hierarchical Clustering	Silhouette Score
[14]	Banking	K-Means Clustering	Not discussed
[13]	Retail	K-Means Clustering	Elbow Method Silhouette Method Gap Statistic Method
[15]	Telco	K-modes Clustering	Silhouette score
[16]	Pharmaceutical	K-Means clustering GMM Mixture of multivariate t-distribution	WSS Silhouette score Connectivity Dunn Index
[17]	Mall	K-Means clustering GMM BIRCH Gaussian mixture model Mixture of multivariate t-distribution	Davies-Bouldin Score

Customer segmentation is a vital component of telecom analytics, allowing service providers to identify key customer groups and tailor their offerings accordingly. Various clustering techniques, including K-Means, GMM, and hierarchical clustering, have been explored in literature, each with its own strengths and limitations. Hybrid and deep learning-based segmentation approaches offer promising avenues for further research, providing more accurate and dynamic customer insights.

2.3 Recommendation Systems

Recommendation systems have become an essential tool for telecom service providers to enhance customer experience, increase revenue, and optimize service offerings. By analyzing customer behavior, preferences, and historical data, recommendation models can suggest personalized telecom plans, add-ons, and bundled services. Traditional rule-based approaches have been replaced by advanced machine learning-based recommendation techniques, which improve the accuracy and relevance of recommendations.

2.3.1 Overview on Recommendation Systems

Usually, recommender systems can be classified into three types [18].

- Content-based recommendations: The user will be recommended items similar to the ones the user preferred in the past
- Collaborative-filtering recommendations: The user will be recommended items that people with similar tastes and preferences liked in the past. Collaborative filtering can be categorized into;
  - User-Based CF: Identifies similar users and suggests services based on their preferences
  - Item-Based CF: Computes similarities between service plans and recommends items similar to those a customer has previously chose
- Hybrid approaches: These methods combine collaborative and content-based methods.

Before designing the recommendation component of this study, it was essential to review existing literature on recommendation systems to understand commonly used techniques, their effectiveness, and their applicability to different domains. Table 2.3 gives a brief summary on existing work.

TABLE 2.3: SUMMARY ON THE LITERATURE ON RECOMMENDATION SYSTEMS

Paper	**Usecase**	**Recommendation Approach**	**Evaluation Metrics**
[19]	e-commerce	Ensemble-based	AUC MAE
[20]	Movie Recommendation	SVD Collaborative	RMSE MAE
[21]	e-commerce	SVD Collaborative	MAE

The authors in [19] proposed a voting ensemble-based recommendation framework that integrates four distinct algorithms: user-based collaborative filtering, the LightFM hybrid algorithm, KNNWithMeans, and a neural network model trained on historical sales data. By aggregating the outputs of these models, the final recommendations for each customer were determined using a voting ensemble learning strategy, aiming to enhance accuracy and robustness in product recommendations.

The collaborative filtering algorithm is one of the most commonly used algorithms in movie recommendation systems. In [20], authors analyzed data sparsity and cold start problems in the collaborative filtering recommendation algorithm and compared

and analyzed the accuracy of the Item-CF, User-CF and SVD recommendation algorithms. It was concluded that the SVD worked best in the MovieLens ML-100k dataset.

The authors in [21] applied Singular Value Decomposition (SVD) to decompose the user-item rating matrix and reconstruct it for computing user-to-user similarities. These similarities were then leveraged in a collaborative filtering framework to identify nearest neighbors and predict item ratings for recommendation. Their experimental results demonstrated that this approach not only mitigates the challenges of the "cold start" and "data sparsity" problems but also enhances the overall efficiency and scalability of the recommendation system.

Matrix factorization techniques such as Singular Value Decomposition (SVD) and Non-Negative Matrix Factorization (NMF) are widely used to address sparsity and improve recommendation quality.

Recommendation systems have evolved significantly, from rule-based approaches to advanced deep learning-based models. Collaborative filtering, content-based filtering, and hybrid methods play a vital role in telecom service recommendations. However, challenges such as cold start problems, data sparsity, and privacy concerns still need to be addressed. Emerging technologies such as reinforcement learning, federated learning, and explainable AI offer promising solutions for next-generation telecom recommendation systems.

2.4 Existing Research Gaps

Despite advancements in recommendation systems, several research gaps persist:

- Current literature exhibits a paucity of studies effectively combining advanced clustering techniques with personalized recommendation algorithms in telecommunications contexts, particularly for optimizing service adoption strategies.
- While Singular Value Decomposition has proven effective in e-commerce and media recommendations, its specific application and adaptation for telecom service bundling remains underexplored in existing research.
- Few studies have developed tailored machine learning approaches for service cross-selling in developing telecom markets, where unique consumer behavior patterns and infrastructure constraints require specialized solutions.

2.5 Literature-Guided Selection of Clustering and Recommendation Techniques

The selection of clustering and recommendation techniques in this research was guided not only by their success in prior literature, but also through critical evaluation against

existing state-of-the-art models in terms of performance, interpretability and practical feasibility for real-world telecom applications.

In the domain of customer segmentation, K-Means was chosen as a baseline due to its computational efficiency and interpretability, making it suitable for large-scale datasets. While more advanced clustering algorithms such as DBSCAN or Spectral Clustering offer capabilities like detecting arbitrary-shaped clusters or capturing global structure, these methods tend to be sensitive to hyperparameters and are computationally intensive. DBSCAN, for instance, struggles with varying cluster densities common in telecom datasets and Spectral Clustering does not scale well to datasets with thousands of customers.

To complement the baseline K-Means algorithm, Gaussian Mixture Models (GMM) were included due to their probabilistic clustering framework, which enables soft membership assignments and can model non-spherical, overlapping clusters. This is particularly valuable in telecom datasets, where customer behavior often exhibits nuanced, overlapping patterns rather than distinct boundaries. While GMM has not been as extensively applied in customer segmentation literature compared to K-Means, several studies have demonstrated its effectiveness in capturing complex behavioral structures. Moreover, unlike hierarchical or density-based clustering methods, GMM offers greater flexibility in handling real-valued input features and scales well for moderate-sized datasets. The ability of GMM to reflect real-world ambiguity, allowing a customer to belong to multiple segments with varying probabilities, makes it a strong candidate for this study. Therefore, it was incorporated to evaluate its performance against the more traditional K-Means approach, and to offer a richer interpretation of customer segmentation in the telecom domain.

For the recommendation engine, Singular Value Decomposition (SVD) was selected as a robust model-based collaborative filtering technique. While more recent neural approaches such as Neural Collaborative Filtering (NCF) and Autoencoder-based recommenders exist, these often require dense interaction data and extensive computational resources. Given the sparse nature of the telecom user-package matrix and the absence of explicit feedback, matrix factorization techniques such as SVD are well-suited. SVD is also explainable, easy to tune, and has been empirically shown to perform well in implicit-feedback scenarios like service subscription modeling.

Thus, the selected models represent a balance between theoretical rigor, practical relevance, and dataset-specific suitability. GMM was further selected as the final segmentation model due to its superior performance over K-Means in clustering evaluation metrics and its ability to uncover nuanced customer segments.

2.6 Mathematical Background and Parameter Analysis of Selected Models

2.6.1 K-Means Clustering

Mathematical Background

The K-Means algorithm aims to partition a dataset into k clusters by minimizing the within-cluster sum of squares (WCSS).

The objective function is defined as:

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

where:

- k is the number of clusters,
- C_i is the set of data points in cluster i,
- μ_i is the centroid of cluster C_i,
- $\|x - \mu_i\|^2$ is the squared Euclidean distance between point x and cluster centroid μ_i.

The algorithm iteratively updates the centroids and reassigns points until convergence.

Impact of Parameters

- Number of clusters (*k*): The choice of *k* significantly affects the clustering outcome which directly influences the granularity of segmentation. A poor choice can lead to underfitting (too few clusters) or overfitting (too many clusters). The elbow method and silhouette score analysis serve as essential quantitative criteria for objectively determining the optimal cluster count that balances model complexity with meaningful pattern recognition.
- Distance metric: Typically Euclidean, but alternative metrics can change the cluster shapes and sizes.
- Initialization Method: The initial placement of centroids can influence convergence speed and final clustering. Common methods include random initialization and the KMeans++ algorithm, which selects initial centroids more strategically.
- Max iterations: Affects convergence time; more iterations can improve accuracy but increase computation.

Key Parameters

- `n_clusters` – The number of clusters to form.
- `init` – Method for initialization. Options include:
  - `k-means++` : Selects initial cluster centers in a smart way to speed up convergence
  - `random` : Randomly chooses initial cluster centers
- `n_iter` - The number of times the algorithm will be run with different centroid seeds. The final results will be the best output of these runs.
- `max_iter` – Maximum iterations for convergence.
- `tol` - Tolerance for convergence. If the change in the inertia is less than this value, the algorithm will stop.
- `random_state` - Controls the randomness of the initialization. Setting this to a fixed number ensures reproducibility.

Initialization
Centroids μ_i are typically initialized using the K-Means++ method to improve convergence. Here, the first centroid is selected randomly, then subsequent centroids are chosen based on their distance from existing centroids, ensuring a more spread-out selection.

2.6.2 Gaussian Mixture Model (GMM)

Mathematical Background
GMM is a probabilistic model that assumes data points are generated from a mixture of several Gaussian distributions.
The probability density function is given by:

$$p(x) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x \mid \mu_k, \Sigma_k)$$

where:

- K is the number of Gaussian components,
- π_k is the mixing coefficient for the k^{th} component, where $\sum_k \pi_k = 1$,
- $\mathcal{N}(x \mid \mu_k, \Sigma_k)$ is the Gaussian distribution with mean μ_k and covariance Σ_k

$$\mathcal{N}(x \mid \mu, \Sigma) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right)$$

Impact of Parameters

- Number of components (K): Determines model complexity; too high can cause overfitting, too low can under-represent the data.
- Covariance type: (full, diagonal, spherical, tied) affects the flexibility of cluster shape modeling.
- Convergence threshold & max iterations: Impact training time and stability.

Key Parameters

- `n_components` – The number of mixture components (clusters).
- `covariance_type` – Type of covariance parameters to use. This parameter specifies the structure of the covariance matrices for the Gaussian components. The choice of covariance type can significantly affect the model's performance and the shape of the clusters. Options include:
  - `full` : Each component has its own general covariance matrix.
  - `tied` : All components share the same covariance matrix.
  - `diag` : Each component has its own diagonal covariance matrix.
  - `spherical` : Each component has its own single variance.
- `init_params` – Method for initialization of the parameters. Options include:
  - `kmeans` : Initialize the means using KMeans.
  - `random` : Initialize the means randomly.
- `n_init` - The number of initializations to perform. The best result is selected based on the lowest Bayesian Information Criterion (BIC).
- `max_iter` - Maximum number of iterations for the EM algorithm.
- `random_state` - Controls the randomness of the initialization.

Initialization

Parameters π_k, μ_k, Σ_k are initialized via the Expectation-Maximization (EM) algorithm, often using K-Means as a starting point.

2.6.3 Singular Value Decomposition (SVD)

Mathematical Background
SVD is a matrix factorization technique used in model-based collaborative filtering. It decomposes the user-item interaction matrix R into latent factors:

$$R \approx U\Sigma V^T$$

where:

- is an orthogonal matrix containing the left singular vectors,
- Σ is a diagonal matrix with singular values,
- ^{T} is the transpose of an orthogonal matrix containing the right singular vectors.

In recommender systems, SVD is typically applied using a regularized matrix factorization approach, predicting the rating $\hat{r}_{ui}$ by:

$$\hat{r}_{ui} = \mu + b_u + b_i + q_i^T p_u$$

where:

- $\hat{r}_{ui}$ is the predicted rating of user u for item i,
- μ is the global average rating,
- b_u, b_i are user and item biases respectively,
- p_u, q_i are the latent feature vectors for user u and item i.

To train the model, a regularized squared error loss function is minimized:

$$\min_{p_*, q_*, b_*} \sum_{(u,i) \in K} (r_{ui} - \hat{r}_{ui})^2 + \lambda \left(\|p_u\|^2 + \|q_i\|^2 + b_u^2 + b_i^2 \right)$$

Impact of Parameters

- Number of Singular Values: The choice of how many singular values to retain affects the approximation quality. Retaining too few can lead to loss of important information (underfitting), while retaining too many can introduce noise (overfitting).
- Regularization term: Prevents overfitting by penalizing large weights.
- Learning rate: Affects convergence; too high causes instability, too low slows training.

- Epochs: More epochs lead to better convergence but increase training time.
- Data Preprocessing: The performance of SVD can be influenced by how the data is normalized or centered before decomposition.

Key Parameters

- `n_factors` – The number of latent factors to use in the model. More factors can capture more complex patterns but may lead to overfitting.
- `n_epochs` - The number of iterations to run the training process. More epochs can lead to better convergence but will increase computation time.
- `lr_all` : The learning rate for all parameters. A smaller learning rate may lead to more stable convergence but will require more epochs.
- `reg_all` - The regularization term applied to all parameters to prevent overfitting. Adjusting this value can help balance bias and variance.
- `init_mean` - The mean value used for initializing the latent factors. This can help in starting the optimization process from a reasonable point.
- `init_std_dev` - The standard deviation used for initializing the latent factors. This controls the spread of the initial values.
- `biased` - A boolean indicating whether to use the biased version of SVD, which includes user and item biases in the model.

Initialization

Latent vectors p_u, q_i and biases are usually initialized with small random values (e.g., small Gaussian noise) or zeros. The Surprise library handles this internally with sensible defaults but allows manual tuning

CHAPTER 3

METHODOLOGY

This chapter outlines the methodology for developing a Machine Learning (ML) driven customer segmentation and cross-selling recommendation engine specifically defined for telecom industry. The study aims to identify potential customers for Triple-play service adoption by analyzing behavioral and transactional data. It details the dataset, preprocessing techniques, segmentation strategy, recommendation engine framework, evaluation metrics and implementation tools used for the implementation. By leveraging data-driven ML techniques, the approach integrates customer segmentation through clustering and targeted recommendations via model-based collaborative filtering to enhance IPTV service adoption. The framework is validated using key performance metrics, ensuring its effectiveness in supporting data-driven marketing strategies.

3.1 Research Design Architecture

The overall architecture used in this work, which is depicted in Figure 3.1, entails data collection, data preprocessing, customer segmentation and cross-selling recommendation engine.

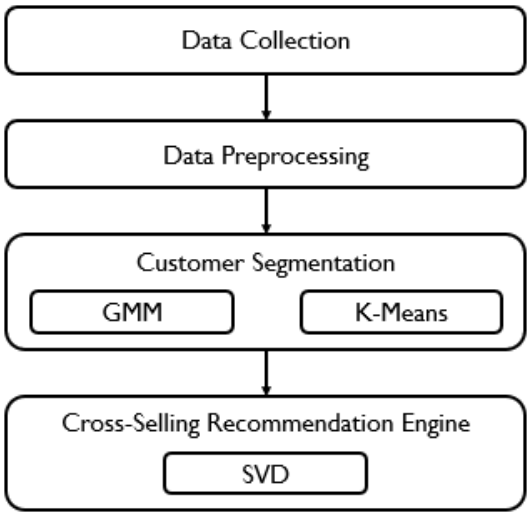

Fig. 3.1: Architecture

The research follows a data-driven ML based approach to analyze telecom customer data. The methodology comprises three key phases.

- **Data Collection and Preprocessing** – Gathering and cleaning customer data for segmentation and recommendation.

- **Customer Segmentation** – Implementing clustering techniques to group customers based on usage patterns.
- **Cross-Selling Recommendation Engine** – Designing and evaluating a recommendation model to suggest IPTV packages to double-play customers.

3.2 Design Implementation

3.2.1 Implementation Tools

The implementation of customer segmentation and recommendation engine involved several tools and libraries that facilitated data preprocessing, clustering, model development, evaluation, and visualization. The selection of these tools was based on their efficiency, ease of integration and compatibility with machine learning techniques. Table 3.1 shows the tools and libraries used for the implementation.

TABLE 3.1: IMPLEMENTATION TOOLS & TECHNOLOGIES

Category	**Tools/Libraries**	**Purpose**
Development Environment	Jupyter Notebook Python	Code development, debugging, and visualization
Data Preprocessing & EDA	NumPy Pandas	Data manipulation, handling missing values, numerical computations
	Matplotlib Seaborn	Data visualization and exploratory analysis
	Scikit-learn (StandardScaler)	Feature scaling
Customer Segmentation (Clustering)	Scikit-learn (K-Means, GMM)	Clustering algorithms for customer segmentation
Clustering Evaluation Metrics	Silhouette Score Davies Bouldin Index Calinski-Harabasz Index Elbow Method	Evaluating clustering quality and determining optimal cluster count
Recommendation System	Surprise Library (SVD)	Implementing collaborative filtering-based recommendation model
Evaluation Metrics for Recommendation	RMSE, MAE, Precision@K, Recall@K	Assessing recommendation accuracy and ranking performance
Visualization & Reporting	Matplotlib, Seaborn	Visualizing customer segmentation and recommendation insights

3.2.1.1 Development Environment

Jupyter Notebook
Used as the primary development environment for coding, debugging and visualizing data. Jupyter provides an interactive interface that enables efficient execution of machine learning workflows.

Google Colab
Due to compatibility issues with the Surprise library on local machine, Google Colab was utilized for running certain parts of the code. Colab provided a cloud-based Python environment with the necessary dependencies to seamlessly execute the machine learning models.

Local Development Environment
For local development, the following system configuration was used:

- Operating System: macOS 15.3.2 (24D81)
- Processor: Intel i5, 8GB RAM
- Python Version: 3.12
- Key Libraries: Pandas 2.2.3, Numpy 2.2.3 and Scikit-learn 1.6.1

Compatibility issues with the Surprise library on this environment led to the use of Google Colab for specific tasks.

Note on Compatibility
While attempting to use the Surprise library on local machine, a compatibility issues was encountered due to the recent release of NumPy 2.x, which is incompatible with certain precompiled modules. The Surprise library, compiled with NumPy 1.x, failed to run on local system (NumPy 2.2.3) and downgrading NumPy conflicted with other dependencies. As a result, Google Colab was used for parts of the development that relied on Surprise.

- Python Version: 3.11.11
- Key Libraries: Numpy 1.23.5, scikit-surprise 1.1.4

3.2.2 Data Collection & Preprocessing

3.2.2.1 Data Collection

To build an effective customer segmentation and cross-selling recommendation system, this study utilizes real-world telecom datasets consisting of anonymized customer

records obtained from a leading telecommunications provider in Sri Lanka. The use of actual operational data ensures both robustness and practical relevance for real-world applications. The dataset comprises four structured tables, including:

- Customer Data: `customer_data.csv`
- Bill Summary: `bill_data.csv`
- Payment History: `payment_data.csv`
- Service Usage: `usage_data.csv`

The dataset represents a statistically sampled subset of the provider's customer base, maintaining proportional representation of FTTH Triple-Play and Double-Play service subscribers through random selection. The summary of the collected datasets is given in Table 3.2

TABLE 3.2: SUMMARY OF COLLECTED DATASETS BEFORE PREPROCESSING

Dataset	Records	Features	Description
customer_data.csv	361,694	58	Contains customer profiles including demographics, account information, and service plan subscriptions
bill_data.csv	486,531	11	Contains five-month historical billing statement details
payment_data.csv	427,731	4	Contains records of customer payments over five months.
usage_data.csv	1,455,073	38	Contains six-months of detailed service utilization, including metrics on voice call durations, internet consumption volumes and IPTV usage for each customer account.

3.2.2.2 Individual Dataset Preprocessing

Robust preprocessing of the collected datasets was essential to ensure reliable segmentation and recommendation outcomes.

Customer Dataset

The dataset contains a mix of demographic (age, gender, location), behavioral (payment history, credit score), and service-related (product subscriptions, tariff plans) features.

The dataset underwent a rigorous feature selection process to ensure optimal model performance and interpretability. Initial analysis revealed several columns requiring removal due to redundancy, low business relevance and data quality concerns. Surrogate keys were eliminated as they duplicated the functionality of natural identifiers, while operational metadata fields were excluded due to their lack of predictive value. Temporal columns demonstrated systemic data integrity issues, with pervasive null values, illogical date sequences, and default placeholders that rendered them unreliable for analysis. High-null features and low-variance columns were similarly removed to prevent analytical bias. All these removals were validated through domain expertise, ensuring the retained features captured the most behaviorally significant dimensions for customer segmentation.

The dataset was further refined by removing records marked for internal testing purposes, as these entries did not reflect genuine customer interactions. The filtering was applied to ensure only valid operational data was retained for analysis.

The dataset was then aggregated to enable customer-promotion level analysis. Using composite keys of account and promotion identifiers, numerical features were summarized via arithmetic means, while categorical variables retained their modal values. This transformation consolidated multiple records into unique customer-promotion pairs, reducing dimensionality while preserving critical financial and operational patterns. The resulting final dataset maintained analytical fidelity by representing each customer-promotion combination as a single observation, facilitating efficient analysis without loss of essential information. The aggregated customer-promotion dataset was enriched with additional customer attributes (e.g., demographics, location) from the original records. To avoid duplication, non-aggregated fields were extracted by retaining the first occurrence per account identifier, resulting in a comprehensive analysis-ready dataset.

Bill Summary Dataset

The original bill summary dataset, comprised of 486,524 transaction records with 11 attributes. The dataset captures comprehensive billing information across financial (balance forward, invoice amounts, tax adjustments), temporal (billing date, payment due date), and transactional (billing type, handling code) dimensions.

During preprocessing, several quality issues and optimization opportunities were addressed by removing non-essential columns to focus the analysis. The resulting feature provides an essential unified metric for subsequent customer-level analysis while maintaining the original dataset's financial precision.

The billing dataset was then transformed from transaction-level records to customer-level features through systematic aggregation. Each customer's multiple billing records were consolidated by computing the arithmetic mean of key financial metrics, providing a stable representation of their typical billing patterns.

Payment Dataset

The payment dataset was systematically processed to derive meaningful customer behavioral features while addressing data privacy and analytical requirements. Initial preprocessing involved the removal of confidential fields and redundant temporal fields to comply with data protection standards and eliminate unnecessary columns.

The dataset was then transformed into a account-level analytical asset through a targeted aggregation process. After removing non-essential columns, transactions were consolidated by the primary account identifier, with payment behaviors represented through the arithmetic mean of transaction values. The aggregation yielded a streamlined dataset with two key variables: the persistent account identifier and the derived metric, which captures each accounts's average payment amount across all transactions.

Usage Dataset

The raw usage dataset, comprising 1,455,073 records, was systematically processed to derive meaningful behavioral features. Initial cleaning removed nine redundant columns including duplicate call metrics and system identifiers, while critical fields related to data usage were converted from bytes to gigabytes for interpretability. Temporal aggregation consolidated usage metrics to account-level averages. Categorical variables were engineered via one-hot encoding to create binary service flags (Voice, Internet and IPTV), with package assignments determined through last-observation-carried-forward methodology. The final merged structure combines aggregated usage patterns (mean call durations, data consumption), current service attributes, and identifier fields. This processed dataset enables robust behavioral analysis through its 22 features while maintaining interoperability with other customer data assets.

The summary of the collected datasets after undergoing preprocessing steps is given in Table 3.3

3.2.2.3 Merging Datasets into a Unified Structure

The individually preprocessed datasets were merged to create a unified dataset, consolidating customer, billing, payment, and usage information into a single structured frame. This integration was essential for capturing the complete relationship, ensuring that all relevant attributes were available for analysis.

Two merging strategies were explored as **left join** and **inner join**. The optimal merged dataframe which was the result gained from the inner join approach was selected based on its suitability for subsequent machine learning tasks, balancing data availability and analytical value.

TABLE 3.3: SUMMARY OF DATASETS AFTER PREPROCESSING

Dataset	Records	Features	Description
customer_data_final.csv	100,559	30	Cleaned customer dataset with removal of irrelevant columns, aggregation of duplicate records by unique identifiers to ensure consistency; and creation of one summary record per customer-promotion. Preserved key metrics through statistical consolidation (mean for financials, mode for categories) while reducing dataset size
bill_data_final.csv	98,721	6	Aggregated billing history showing average balances, invoice amounts, and adjustments per customer. Corrupted date fields removed and monetary values scaled.
payment_data_final.csv	98,334	2	Consolidated payment records containing: customer ID and average payment amounts. Duplicate transactions removed and temporal fields simplified.
usage_data_final.csv	99,074	22	Refined usage metrics featuring: voice/data/TV consumption patterns. Null records dropped and data aggregation was done.

The selected dataframe underwent further analysis to identify abnormalities and refine the relevancy of records through additional cleaning steps. The final inner joined merged cleaned dataset served as the foundational dataset for the segmentation and recommendation models. A comprehensive analysis of the merging results is provided in Section 4.1.1 of this thesis.

3.2.2.4 Data Preparation for Customer Segmentation

Once the inner-joined dataset was finalized, additional preprocessing was necessary to enhance its quality and usability for customer segmentation. This step was crucial for ensuring the dataset's readiness for machine learning by performing a comprehensive assessment of feature distributions, introducing newly engineered features, detecting and handling missing and infinite values, and identifying any irregularities that could potentially affect model performance.

To prepare the dataset ready for segmentation models, several key preprocessing

steps were carried out.

Handling Missing Values :
Missing values are common in telecom datasets due to incomplete records, billing discrepancies, or system-generated gaps. Addressing these missing values appropriately was crucial to prevent data distortions. The handling strategy varied based on the type of feature:

- **Categorical Features**: Converted to string format using `astype(str)` to ensure consistent data representation. Missing values were replaced with 'None' to maintain uniformity without introducing biases in the machine learning model. This approach also prevented errors during encoding.
- **Numerical Features**: Missing numerical values were imputed with `0` using `fillna(0, inplace=True)`, under the assumption that an absent value often indicated a lack of activity rather than a data collection error. This strategy ensured that models could still learn patterns from the available data without artificially inflating or skewing numerical trends.

Feature Engineering :
To improve the dataset's predictive power and capture meaningful customer behaviors, new features were engineered based on service usage metrics. These derived features aimed to quantify service engagement levels, broadband consumption patterns, and overall usage behavior, which are crucial for effective customer segmentation. The following key engineered features were introduced:

- **Service Usage Ratio** (`SERVICE_USAGE_AVG`):
  This metric captures a customer's voice usage in relation to their total data consumption. A higher ratio suggests that a customer relies more on voice services than data, while a lower ratio indicates a preference for data-heavy services.
- **Broadband Consumption Ratio** (`BB_USAGE_REMAINING_RATIO`):
  This metric evaluates a customer's broadband data consumption relative to their remaining data volume. A higher ratio indicates that a customer has utilized most of their allocated data, potentially signaling a need for a higher-tier broadband package.
- **Customer Engagement Score** (`ENGAGEMENT_SCORE`):
  To quantify customer interaction with different services, a composite engagement score was designed by weighting the usage of voice, broadband and IPTV services. To normalize the score and ensure it remained within a consistent range (0 to 1) for better comparability across customers, min-max scaling was applied. This scaling prevents bias from extreme values and allows for fair comparisons

across different customers. A higher value would indicate a customer who is highly engaged across all services, using them frequently and actively.

Handling Missing and Infinite Values in Engineered Features:
Since new features were created during the feature engineering process, an additional validation step was performed to ensure no missing or infinite values were introduced.

- Any missing values in engineered features were handled using the same strategies as before: categorical values were replaced with 'None', and numerical values were filled with `0` where appropriate.
- Infinity values that arose from mathematical transformations (such as division operations in ratio-based features) were identified and replaced with `0`. This replacement ensured that any infinite values, which could otherwise distort the machine learning model, were converted to a neutral value (0), preventing computational errors while maintaining the dataset's usability.

Encoding Categorical Variables
Machine learning models typically require numerical inputs, so categorical variables were transformed into numerical values using Label Encoding. Each unique category was assigned a numerical representation, allowing the model to process categorical attributes efficiently. By converting categorical data into numerical form, the dataset became more structured and model-ready, ensuring that the segmentation algorithm could leverage these features effectively.

By addressing these aspects, the dataset was refined to improve data consistency, enhance interpretability, and minimize errors, ultimately optimizing it for effective segmentation analysis.

3.2.3 Customer Segmentation

This study employs clustering-based customer segmentation using Gaussian Mixture Models (GMM) and K-Means to identify distinct customer groups. The optimal number of clusters was determined using various evaluation metrics, followed by model training and performance assessment.

- Gaussian Mixture Models (GMM)
- K-Means Clustering

3.2.3.1 Data Preprocessing

The input dataset was first preprocessed to remove unique identifiers that do not contribute to meaningful clustering. This ensures that the segmentation process is based

on relevant attributes rather than arbitrary distinctions.

Feature Scaling
Since clustering algorithms perform better with normalized data, all numerical features were standardized using `StandardScaler` from sklearn.preprocessing. The standardization process involved fitting the scaler to the dataset and transforming it into a scaled format.

3.2.3.2 Clustering with Gaussian Mixture Model

The GMM approach was applied to cluster customers into segments with soft assignments, capturing underlying probabilistic distributions.

1. **Finding Optimal Clusters**

    - Bayesian Information Criterion (BIC) and Akaike Information Criterion (AIC) were calculated for different N components. The optimal cluster count was determined as N=8 based on the lowest BIC and AIC values.
    - The Silhouette Score was computed for different N components, identifying N=2 as the best based on cohesion and separation metrics.

2. **Model Evaluation Metrics**
   Log-Likelihood, Silhouette Score, Davies-Bouldin Index, and Calinski-Harabasz Index were calculated to assess clustering performance.

Given the contrasting results, GMM models were trained for both N=2 and N=8, and cluster labels were assigned to the dataset.

3.2.3.3 K-Means Clustering

The K-Means algorithm was applied to segment customers into clusters with hard assignments.

1. **Determining Optimal K (Clusters)**

    - The Elbow Method was used to visualize the inertia value for different cluster counts. The KElbowVisualizer tool from yellowbrick was also used for validation which indicated K=9 as the optimal value.
    - The Silhouette Score was calculated, indicating K=6 as the optimal cluster count.

2. **Model Training and Evaluation**
   To determine the most appropriate number of clusters, the K-Means model was trained with K values ranging from 2 to 9, and the Silhouette Score, Davies-Bouldin Index and Calinski-Harabasz Index metrics were computed to find the optimal K value.

3.2.3.4 Model Selection for Customer Segmentation

To determine the most suitable clustering model for customer segmentation, a comparative analysis was conducted between the Gaussian Mixture Model (GMM) with 8 clusters and the K-Means clustering approach with 6 clusters. The evaluation considered multiple clustering performance metrics, including Silhouette Score, Davies-Bouldin Index, and Calinski-Harabasz Index. A detailed discussion of the comparative results is provided in Section 4.2.3.

Based on the results, it was identified that GMM with 8 clusters provided the most optimal segmentation, offering better-defined clusters with probabilistic membership assignments.

With the selected model, further cluster analysis was performed to identify meaningful patterns within the customer segments, which is crucial for developing the recommendation engine.

3.2.3.5 Cluster Analysis

Customer segmentation was performed using Gaussian Mixture Model (GMM) clustering with eight components. Each cluster was analyzed for its composition of Double-Play (DP) and Triple Play (TP) customers, with particular focus on identifying DP customers in high-TP adoption clusters as potential upgrade targets. Analysis revealed three distinct segment types.

1. TP-dominant clusters (2,3,5 with 100% TP adoption)

2. High-potential mixed clusters (1,4,6 with 62-71% TP penetration)

3. DP-dominant clusters (0,7).

Cluster analytics revealed High-Potential hybrid clusters as optimal targets for TP upgrades, where DP customers demonstrate a greater upgrade readiness based on neighborhood adoption patterns. A detailed view on this analysis is provided in Section 4.2.3.1

3.2.4 Cross-Selling Recommendation Engine

After performing customer segmentation using Gaussian Mixture Model (GMM) clustering, the next step was to develop a recommendation system to suggest the best suited IPTV packages to Double-Play (DP) customers within high TP clusters.

3.2.4.1 Collerberative Filtering Approach

The recommendation system was designed using collaborative filtering, leveraging Singular Value Decomposition (SVD) for predicting package preferences.

Data Preparation
The GMM clustering output with 8 cluster labels was used as the base dataset. The primary focus was on Triple-Play (TP) customers, as they had already adopted IPTV services, making their subscription patterns useful for training the recommendation model. To understand package adoption patterns, all TP customers were extracted from the dataset. Analysing this dataset helped to identify the most popular IPTV packages among TP customers.

Building the Customer-Package Matrix
To construct a customer-package matrix (i.e., user-item), an ENGAGEMENT_SCORE column was created, assuming all TP customers had fully engaged with their subscribed packages.

Model Training & Evaluation
To build the recommendation engine, Singular Value Decomposition (SVD) was chosen from the Surprise library (scikit-surprise). The dataset was split into training (75%) and test (25%) sets to evaluate the model's performance on unseen data while maintaining sufficient training samples for effective learning. This standard practice helps prevent overfitting and provides an unbiased assessment of the model's predictive capability before deployment to real-world scenarios.

To assess the performance of the SVD based recommendation model, a comprehensive evaluation was conducted using both holdout validation and cross-validation techniques. A detailed result analysis is given in Section 4.3.1.

- **Holdout validation**: The dataset was partitioned into a training set (75%) and a test set (25%) to evaluate the model's predictive accuracy on unseen data. The Root Mean Squared Error (RMSE) was computed on the test set.
- **Cross-Validation** : A 5-fold cross-validation was performed to ensure robustness and generalizability. The results (Figure 4.11) demonstrated consistent performance across all folds.

Generating Recommendations

After training and validating the model, recommendations were generated for DP customers in high TP clusters. For each customer, engagement scores were predicted across all available IPTV packages. The resulting customer-package predictions were analyzed to identify optimal offerings. Analysis revealed a singular package recommendation pattern across all users with key insights presented in Section 4.3.1.

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Data Collection & Preprocessing

Summary on dimensions and a nature of the dataframes used through out the study are given in Table 3.2, Table 3.3 and Table 4.2, while key features of the created dataframes are presented in Table 4.1. These tables provide an overview of both the original and processed datasets used throughout the analysis.

4.1.1 Dataset Merging Strategy and Selection

Four raw CSV files were initially processed for dataset preparation. Redundant attributes were removed, and missing values were partially addressed in each dataset. More representative features were created through the aggregation of key variables. Summaries of both collected and individually processed datasets are provided in Tables 3.2 and Table 3.3 respectively and the null percentage count across each dataset is summarized in Figure 4.1.

	Null Percentage
customer_data	4.342941
bill_data	0.000000
payment_data	0.000000
usage_data	58.267006
customer_data_final	1.117851
bill_data_final	0.000000
payment_data_final	0.000000
usage_data_final	20.618840
merged_left	9.402929
merged_inner	8.641227

Fig. 4.1: Null percentage comparison across datasets

To integrate all relevant attributes into a single structured dataset, two merging strategies were applied.

- **Left Join Results**:
  Retained all customers but introduced a significant number of missing values in billing, payment, and usage fields. It preserved the maximum possible dataset size, allowing for imputation strategies to handle missing values.

- **Inner Join Results**:
  Retained only customers with complete records across all datasets, potentially reducing dataset size but ensuring completeness. The datset size was reduced

TABLE 4.1: SOME KEY FEATURES ON THE DATASETS USED

Dataframe	Records	Unique Customers	Unique Accounts	Unique Promotion
customer_data	361694	98838	100117	100698
bill_data	486531	-	98721	-
payment_data	427731	-	98334	-
usage_data	1455073	97411	98647	-
customer_data_final	100559	98829	100096	100559
bill_data_final	98721	-	98721	-
payment_data_final	98334	-	98334	-
usage_data_final	99074	97403	98628	99074
merged_left	100559	98829	100096	100559
merged_inner	97863	96242	97427	97863
merged_inner_final	97863	96242	97427	97863
preprocessed_inner_final_O	97863	96242	97427	97863
preprocessed_inner_final_L	97863	96242	97427	97863
pre_scaled_inner_final_L	97863	-	-	-
post_scaled_inner_final_L	97863	-	-	-
gmm_input	97863	-	-	-
gmm_output	97863	-	-	-

by approximately 3%, removing customers with incomplete billing, payment, or usage data. This ensured a cleaner dataset with fewer missing values, making it directly usable for machine learning.

Ultimately, the inner join strategy was selected as it provided the best balance between data retention and analytical quality. The comparison is given in Table 4.3. This final merged dataset serves as the foundation for customer segmentation and the recommendation engine.

4.2 Customer Segmentation

This section presents the results of the customer segmentation analysis, comparing the performance of GMM and K-Means clustering.

4.2.1 Gaussian Mixture Model

4.2.1.1 Optimal Cluster Selection

Bayesian Information Criterion and Akaike Information Criterion Analysis
The lowest BIC and AIC values were obtained for N=8, indicating that this configuration best balances model complexity and fit. Figure 4.2 shows how BIC and AIC vary with different cluster sizes, with N=8 being optimal.

TABLE 4.2: SUMMARY OF DATAFRAMES USED

Dataset	Records	Features	Description
merged_left	100,559	55	Output dataframe received from the left-join approach.
merged_inner	97,863	55	Output dataframe received from the inner-join approach.
merged_inner_final	97,863	52	merged_inner dataset after some validations.
preprocessed_inner_final_O	97,863	55	The merged_inner_final dataset was preprocessed by replacing missing values in categorical fields with 'None' and numerical fields with 0, assuming no activity rather than data errors. Three new engineered features were added, and their missing values were handled similarly. Additionally, any infinite values in the engineered fields were replaced with 0 to ensure numerical stability.
preprocessed_inner_final_L	97,863	55	The categorical fields in the preprocessed_inner_final_O dataframe that can be labeled were label-encoded to convert them into numerical values, making the data suitable for modeling. This preprocessing step ensures compatibility with machine learning algorithms that require numerical inputs.
pre_scaled_inner_final_L	97,863	51	The categorical columns were removed from preprocessed_inner_final_L to prepare the dataframe for feature scaling.
post_scaled_inner_final_L	97,863	51	Feature scaling was done to the pre_scaled_inner_final_L dataframe using StandardScaler.
gmm_input	97,863	51	post_scaled_inner_final_L dataframe was used as the input for the GMM model.
gmm_output	97,863	52	Output of the GMM model with assigned cluster lables.

Silhouette Score Analysis

The Silhouette Score favored N=2, suggesting strong cohesion for fewer clusters. Fig-

TABLE 4.3: COMPARISON ON THE DATASET MERGING STRATEGY

Feature	Left Join Strategy	Inner Join Strategy
Dimension	(100559, 55)	(97863, 55)
Null value percentage	9.4%	8.6%
Unique Accounts	100096	97427
Unique Customers	98829	96242
Unique Promotions	100559	97863

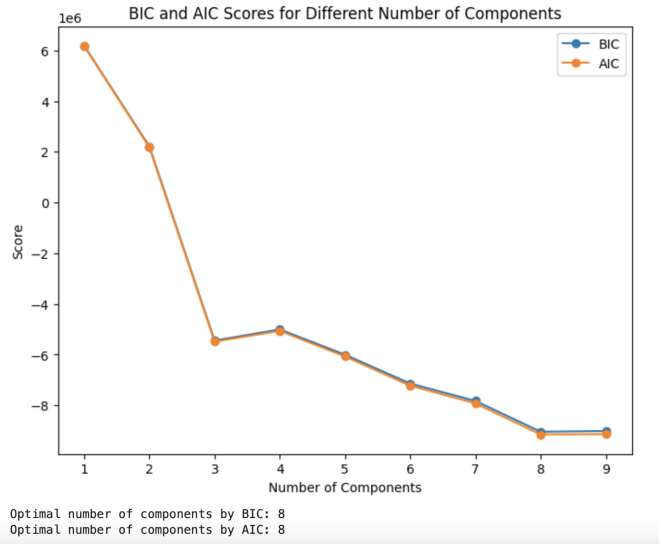

Fig. 4.2: BIC and AIC score variation with different cluster numbers for GMM

ure 4.3 presents the Silhouette Scores for different N values, with N=2 yielding the highest score.

Final Selection
Given these results, both N=2 and N=8 were evaluated further using additional metrics.

4.2.1.2 Model Training and Evaluation

GMM models were trained for N=2 and N=8, and cluster labels were assigned. Figure 4.4 shows the customer distribution across GMM clusters for N=8 while Figure 4.5 shows the customer distribution across GMM clusters for N=2.

4.2.2 K-Means Clustering

4.2.2.1 Optimal Cluster Selection

The Elbow Method and KElbowVisualizer were used to determine the optimal cluster count. Figure 4.6 shows that the elbow occurs at K=9, indicating the best trade-off

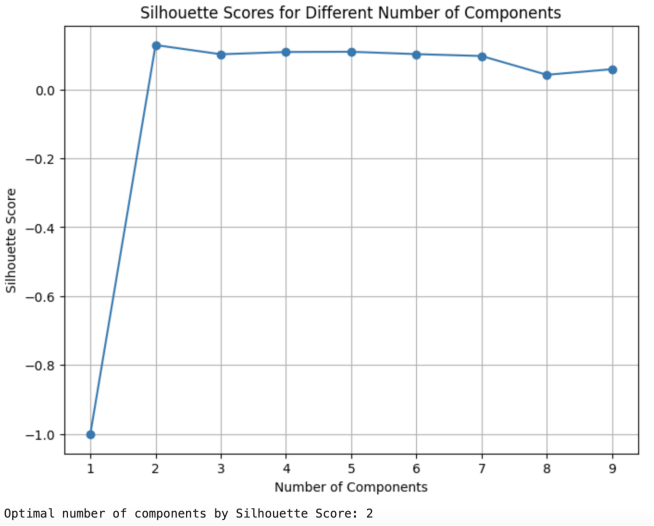

Fig. 4.3: Silhouette Score variation with the number of clusters for GMM

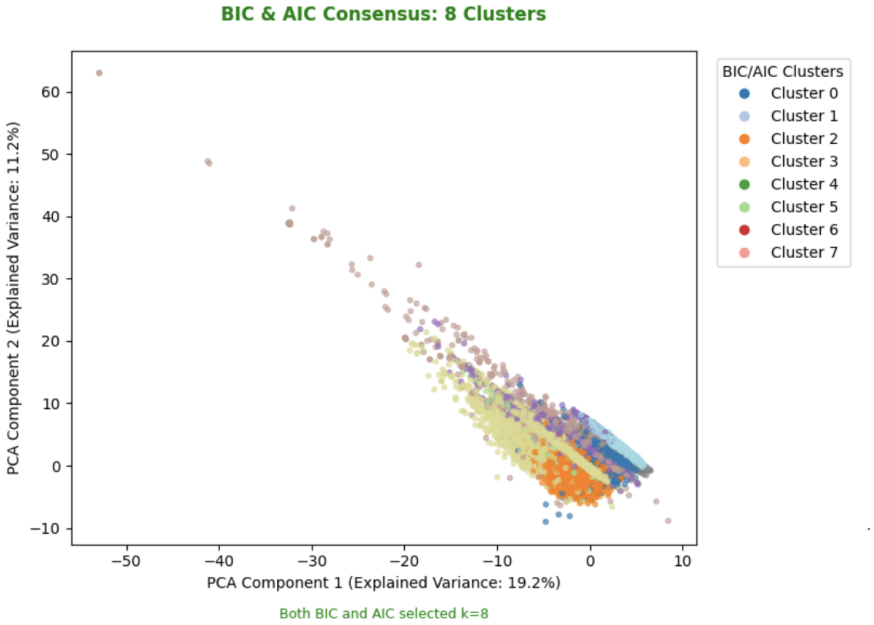

Fig. 4.4: GMM Cluster distribution for N=8

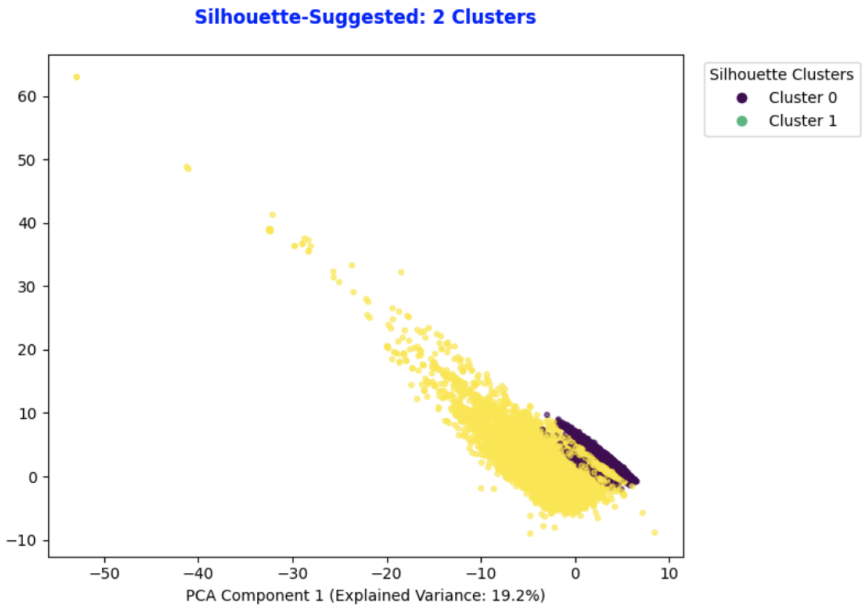

Fig. 4.5: GMM Cluster distribution for N=2

between inertia and model complexity. The corresponding distortion score at K=9 is noted.

Silhouette Score analysis (Figure 4.7) was conducted for different values of K. The best silhouette score was observed at K=6, with a value of 0.1535.

4.2.2.2 Model Training and Evaluation

As the elbow method suggested K=9 and Silhoute sugested K=6, the model was trained for different K values and calculated the Silhouette score, Davies-Bouldin Index, Calinski-Harabasz Index for all the cases and analyzed to find the best K. Table 4.4 presents the clustering performance evaluation for different values of K.

From Table 4.4, K=6 had the highest Silhouette Score (0.1535) and the lowest Davies-Bouldin Index (1.616), making it the most well-separated clustering configuration. Although K=2 had the highest Calinski-Harabasz Index (12010.4), its Silhouette Score and Davies-Bouldin Index were inferior.

From Table 4.4, it is evident that K=9 (optimal cluster value from Elbow method) resulted in a much lower Silhouette Score (0.0756), indicating poor cluster separation. It also had a higher Davies-Bouldin Index (2.461), suggesting that the clusters were not well-defined. The Calinski-Harabasz Index for (5617.0) was lower than for K=6, further confirming weaker cluster structure as the Elbow method only considers the point where inertia (distortion score) stops decreasing significantly. The Elbow method does not evaluate cluster cohesion or separation and it is a heuristic approach that often requires validation from additional metrics.

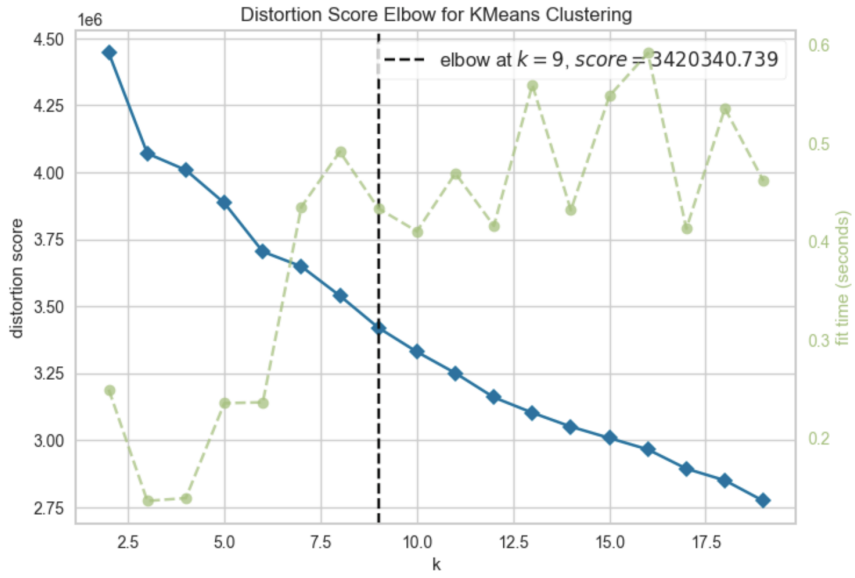

Fig. 4.6: Elbow plot showing the optimal K=9 for K-Means clustering

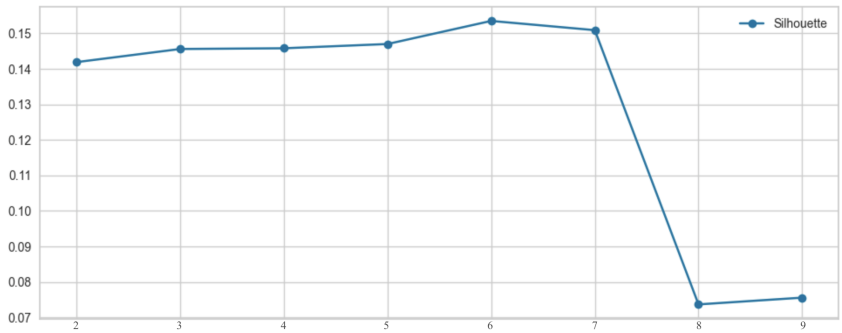

Fig. 4.7: Silhouette Score variation with the number of clusters for KMeans

TABLE 4.4: KMEANS CLUSTERING PERFORMANCE EVALUATION FOR DIFFERENT K VALUES

K	**Silhouette** (Higher Better)	**Davies-Bouldin** (Lower Better)	**Calinski-Harabasz** (Higher Better)
2	0.1419	2.393	**12010.4**
3	0.1456	2.142	11062.2
4	0.1458	1.728	7992.7
5	0.1470	1.700	6959.1
6	**0.1535**	**1.616**	6799.6
7	0.1509	1.920	5998.2
8	0.0737	2.592	5732.2
9	0.0756	2.461	5617.0

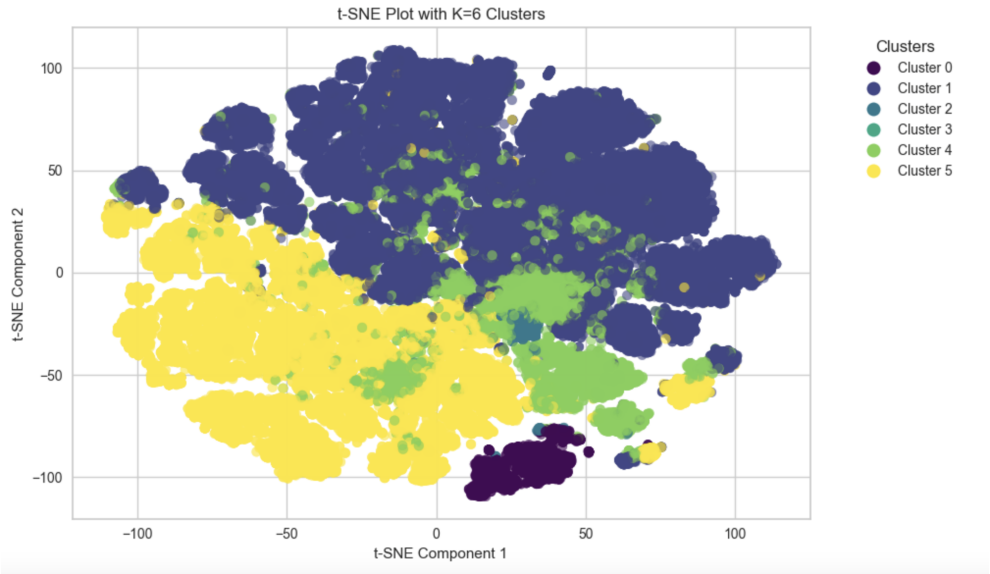

Fig. 4.8: K-Means cluster distribution for K=6

TABLE 4.5: CHOOSING THE BEST CLUSTERING MODELS

Clustering Algorithm	**Clusters**	**Silhouette Score**	**Davies Bouldin Index**	**Calinski Harabasz Index**
GMM	8	0.1798	3.423	4843.8
KMeans	6	0.1535	1.616	6799.6

Since K=6 had the highest Silhouette Score (0.1535) and the lowest Davies-Bouldin Index (1.616), it was selected as the optimal number of clusters for K-Means segmentation. Figure 4.8 shows the customer distribution across KMeans clusters for K=6.

4.2.3 Model Selection for Customer Segmentation

After evaluating both GMM and K-Means clustering techniques, the final model selection was based on multiple clustering performance metrics, as presented in Table 4.5.

Based on the comparison, Gaussian Mixture Model (GMM) the was chosen as the final clustering technique, and the decision was justified by the following factors.

1. The Silhouette Score measures how well-separated the clusters are, where higher values indicate better-defined clusters. GMM outperformed K-Means with a Silhouette score of 0.1798, confirming that its clusters have better cohesion and separation.

2. Unlike K-Means, which assumes clusters are spherical and equal in size, GMM provides a probabilistic clustering approach that models clusters as elliptical

Gaussian distributions. This is particularly useful in real-world datasets, where customer behaviors often do not form strict spherical clusters.

3. GMM provides probability-based membership, where a customer can belong to multiple clusters with varying probabilities. This allows for a more flexible and meaningful segmentation, particularly useful for targeted marketing and personalized recommendations. In K-Means, each data point is assigned to only one cluster, which may not be optimal if some customers exhibit characteristics belonging to multiple segments.

4. The Davies-Bouldin Index (DBI) for K-Means is lower (1.616) compared to GMM (3.423), which might suggest that K-Means has better-defined clusters. However, DBI penalizes cluster overlap, which is not necessarily a drawback in real-world telecom customer segmentation, where customers may naturally exhibit overlapping behaviors across segments.

5. The Calinski-Harabasz (CH) Index is higher for K-Means (6799.6 vs. 4843.8 for GMM), but this metric favors algorithms that create well-separated clusters. Since telecom customers often exhibit continuous behavioral variations, GMM's ability to model soft clusters makes it a better choice for segmentation.

While K-Means achieved a slightly lower DBI and a higher CH index, GMM was ultimately selected due to its ability to model complex customer behaviors, its higher Silhouette score, and its ability to handle overlapping clusters. These advantages make GMM the superior choice for customer segmentation in telecom, where customers often exhibit mixed behaviors across multiple segments.

4.2.3.1 Cluster Analysis

The Gaussian Mixture Model (GMM) with 8 clusters was employed for customer segmentation, chosen for its ability to handle overlapping distributions through probabilistic assignments. Each cluster was analyzed to identify patterns among Double Play (DP) and Triple Play (TP) customers, focusing on identifying DP customers with high TP conversion potential. Results highlight priority clusters where targeted cross-selling maximizes upgrade efficiency.

Cluster Characterization

Distribution of customers across 8 clusters is given in Figure 4.9. The analysis revealed distinct customer segments based on service adoption patterns. Key findings (Table 4.6) identify three distinct customer groupings.

- **TP-Dominant Clusters (2,3,5)**
  Clusters 2, 3, and 5 contain only TP customers, with no DP customers present.

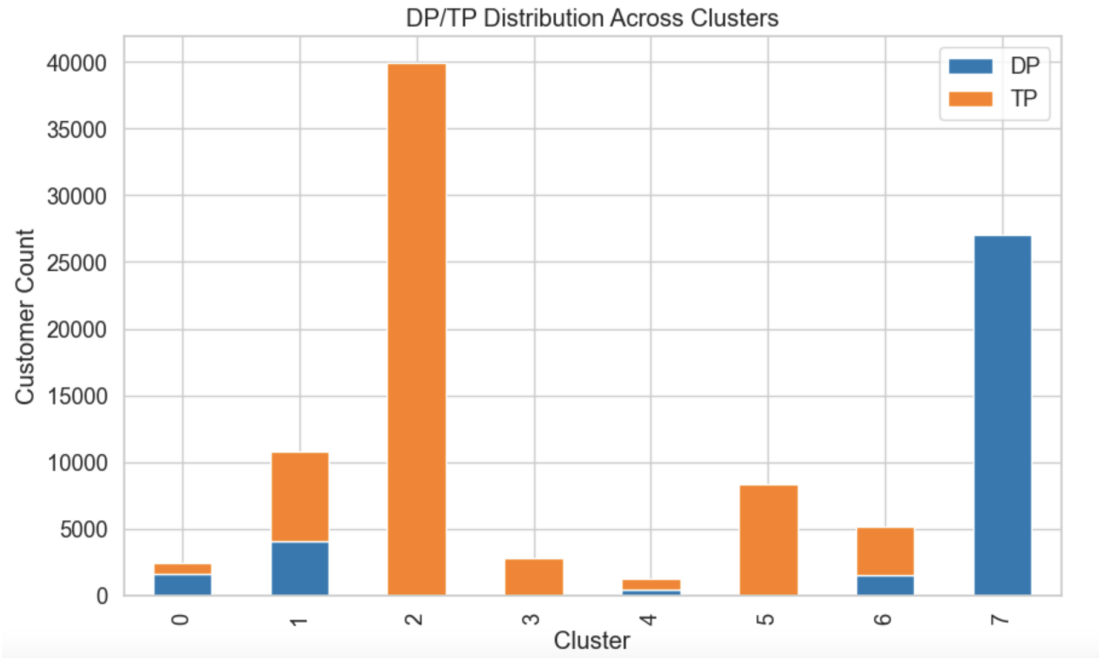

Fig. 4.9: Distribution of customers across 8 clusters

These clusters represent existing TP adopters, meaning these customers have already upgraded their services while these clusters are not relevant for cross-selling TP services, as they are fully saturated with TP users.

- **High-TP Mix Clusters (1,4,6)**
  Clusters 1, 4, and 6 contain a significant mix of DP and TP customers, with TP users forming a large proportion. These clusters represent DP customers who exist in environments where TP adoption is already strong, increasing the likelihood of an upgrade. So, the DP customers in these clusters can be targeted with personalized upgrade offers.

- **DP-Dominant Clusters (0,7)**
  Clusters 0 has a low number of TP customers compared to DP customers while Cluster 7 is particularly noteworthy, as it contains 100% DP customers and no TP customers. This shows that customers in Cluster 7 may require more aggressive marketing efforts (e.g., better pricing, exclusive deals, bundled offers) since TP adoption is currently absent. Cluster 0 may still have potential, but a more detailed behavioral analysis is needed to confirm whether these DP users are truly interested in an upgrade.

Cluster Selection for Cross-Selling Targeting

Based on the above insights, the best target clusters for TP cross-selling can be identified as Clusters 1, 4, and 6. These clusters contain a high proportion of TP customers,

TABLE 4.6: CUSTOMER DISTRIBUTION AND SEGMENT ANALYSIS ACROSS CLUSTERS

Cluster	DP Count	TP Count	TP %	Segment Type
0	1,614	852	34.6%	Low-TP
1	4,042	6,779	62.7%	**High-TP (Target)**
2	0	39,927	100.0%	TP-Only
3	0	2,820	100.0%	TP-Only
4	472	757	61.6%	**High-TP (Target)**
5	0	8,375	100.0%	TP-Only
6	1,510	3,654	70.8%	**High-TP (Target)**
7	27,061	0	0.0%	DP-Only

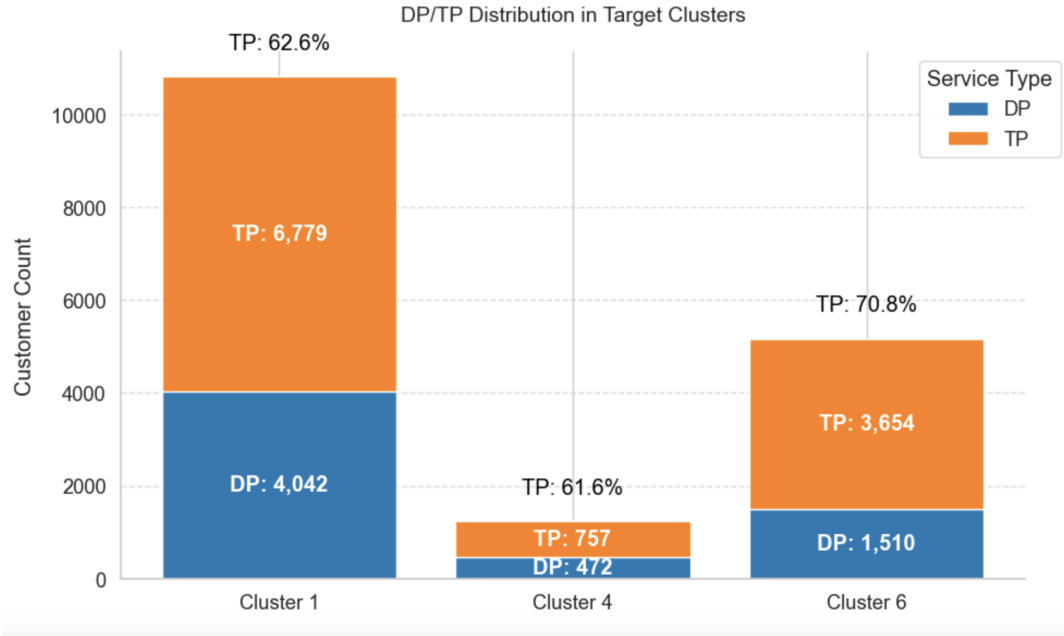

Fig. 4.10: DP/TP distribution in target clusters

making the DP customers in these clusters ideal targets for cross-selling TP services. DP/TP distribution in target clusters is shown in Figure 4.10

Clusters 1, 4, and 6 collectively contain 6,024 DP customers (17.4% of all DP customers) coexisting with 11,190 TP adopters. These target segments represent 44.7% of the total customer base, demonstrating their strategic importance for revenue growth through service upgrades.

4.3 Cross-Selling Recommendation Engine

A collaborative filtering model based on Singular Value Decomposition (SVD) was applied to predict engagement scores for DP customers in high TP clusters across all available IPTV packages. The model was trained on engagement data from existing

TP customers, leveraging latent interactions between users and packages to infer preferences for DP customers who currently do not subscribe to IPTV.

The results obtained from the colloborative filtering recommendation approach followed by a comparative analysis of their performance is presented here.

4.3.1 Collaborative Filtering Approach

Algorithm Selection

Singular Value Decomposition (SVD) was chosen over other collaborative filtering methods due to its ability to effectively reduce dimensionality and extract latent factors, improving prediction accuracy and addressing data sparsity issues.

Model Evaluation

The SVD-based recommender system was evaluated using both a standard hold-out method and 5-fold cross-validation. On the test set, the model achieved a Root Mean Squared Error (RMSE) of 0.0797 and a Mean Absolute Error (MAE) of 0.0612, indicating strong predictive performance when estimating user engagement scores.

Additionally, 5-fold cross-validation was conducted to ensure model stability and generalizability. The results were consistent across folds, with a mean RMSE of 0.0793 ± 0.0003 and a mean MAE of 0.0607 ± 0.0003, as summarized in Table 4.7 and visualized in Figure 4.11. These findings confirm that the SVD model reliably captures patterns in the training data and generalizes well to unseen data points.

Beyond traditional prediction accuracy metrics such as RMSE and MAE, the recommender system's performance was further assessed using ranking-based and system-level metrics to provide a comprehensive evaluation of recommendation quality and practical utility.

The ranking metrics included Recall@5 and Normalized Discounted Cumulative Gain (NDCG@5), which assess the model's ability to correctly prioritize relevant items within the top recommendations. A simulated ground truth, based on the most popular packages among Triple Play users, was employed due to the absence of explicit ground truth engagement data for Double Play customers. The model achieved a **Simulated Recall@5** of 0.2339 and a **Simulated NDCG@5** of 0.8183, indicating that nearly 23.4% of relevant items were captured within the top 5 recommendations and that these relevant items were effectively ranked near the top positions.

To understand the breadth and personalization of the recommendation system, two system-level metrics were calculated:

- **Coverage (0.2000)** quantifies the proportion of all available IPTV packages recommended across the target customer base. The observed value of 20% indi-

```
# Use cross-validation with 5 folds
cross_validate(algo, data, measures=['RMSE', 'MAE'], cv=5, verbose=True)
```

```
Evaluating RMSE, MAE of algorithm SVD on 5 split(s).

                  Fold 1  Fold 2  Fold 3  Fold 4  Fold 5  Mean    Std
RMSE (testset)    0.0787  0.0797  0.0792  0.0795  0.0791  0.0793  0.0003
MAE (testset)     0.0602  0.0610  0.0607  0.0606  0.0608  0.0607  0.0003
Fit time          0.59    0.57    0.62    0.62    0.88    0.66    0.12
Test time         0.15    0.04    0.04    0.03    0.06    0.06    0.05
{'test_rmse': array([0.07873049, 0.07972556, 0.07916188, 0.07950385, 0.07914239]),
 'test_mae': array([0.06017997, 0.06101847, 0.06073334, 0.06056132, 0.06079036]),
 'fit_time': (0.5861196517944336,
  0.5680277347564697,
  0.6231615543365479,
  0.6218581199645996,
  0.884352445602417),
 'test_time': (0.1546025276184082,
  0.03527569770812988,
  0.03615450859069824,
  0.03393268585205078,
  0.05914616584777832)}
```

Fig. 4.11: Output for 5-fold cross validation

cates that the recommendations concentrated on a subset of offerings rather than spanning the entire catalog.

- **Diversity (0.9739)** measures the average dissimilarity between recommendation lists for different users, reflecting the degree of personalized variation. The high diversity score suggests that despite limited coverage, the recommendations varied substantially across customers, supporting a degree of personalization within the constraints of available data.

These metrics together present that the model maintains a focused yet diverse set of recommendations, optimizing for both relevance and personalization despite sparse telecom data.

TABLE 4.7: 5-FOLD CROSS-VALIDATION PERFORMANCE

Metric	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Mean	Std Dev
RMSE	0.0787	0.0797	0.0792	0.0795	0.0791	0.0793	0.0003
MAE	0.0602	0.0610	0.0607	0.0606	0.0608	0.0607	0.0003
Fit Time (s)	0.59	0.57	0.62	0.62	0.88	0.66	0.12
Test Time (s)	0.15	0.04	0.04	0.03	0.06	0.06	0.05

Recommendation Output

The recommendation system demonstrated a pronounced concentration effect, predominantly recommending a single IPTV package, Package-23 for most Double Play (DP) customers within the high-value Triple Play (TP) clusters (Clusters 1, 4, and 6). This dominant pattern emerged clearly in the top-1 recommendation results, where Package-23 was suggested to the vast majority of customers (Figure 4.13 and Figure

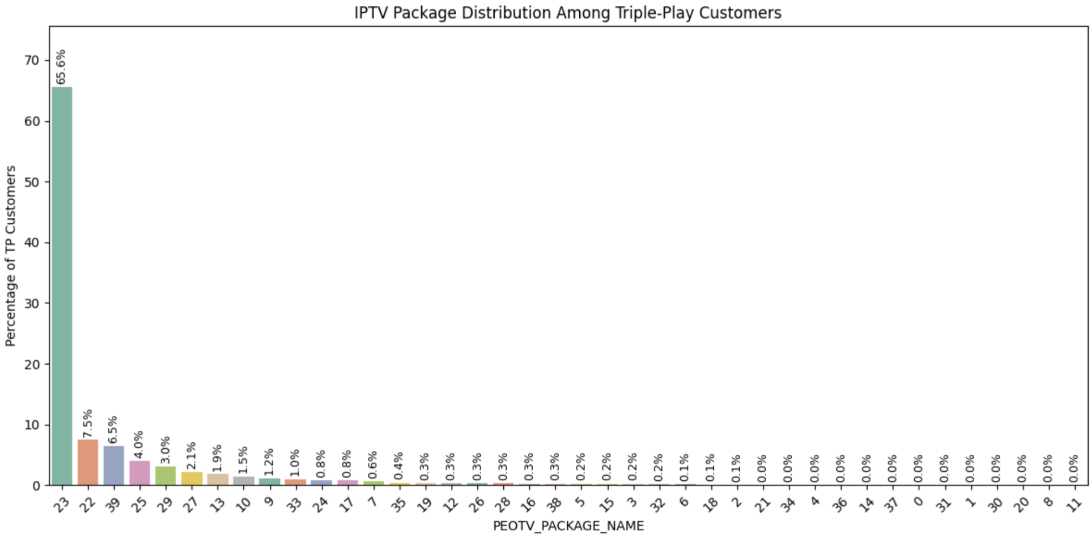

Fig. 4.12: IPTV Package distribution across Triple-Play customers

4.14). Despite initial concerns about limited output variation, this concentration was found to reflect actual user behavior as shown in Figure 4.15.

Further supporting this interpretation, the customer-package interaction matrix exhibited a high sparsity of 98.74%, meaning the majority of potential user–package interactions were unobserved in the dataset. In such sparse environments, collaborative filtering algorithms like SVD tend to recommend highly popular items with strong signal-to-noise ratios, especially when personalization data is limited.

Nevertheless, the recommendation system maintained a relatively high diversity score of 0.9739, indicating some level of personalized variation in predicted scores despite the dominance of a single top choice. However, the coverage—the proportion of distinct items recommended—was only 20%, reinforcing the system's focus on a narrow set of packages.

Ranking-based evaluation further validated the system's behavior. The Simulated Recall@5 was 0.2339, indicating that the model was moderately successful in including a relevant package among its top five predictions. Additionally, the Simulated NDCG@5 score of 0.8183 demonstrated that the model effectively prioritized higher-relevance items near the top of its ranked lists.

These findings highlight both the effectiveness and limitations of the collaborative filtering approach in the given telecom context. The system accurately identified dominant customer preferences and demonstrated strong alignment with subscription behavior, but exhibited constrained recommendation diversity due to inherent data sparsity. These insights suggest that future improvements may benefit from hybrid recommendation models or incorporation of additional contextual and behavioral features.

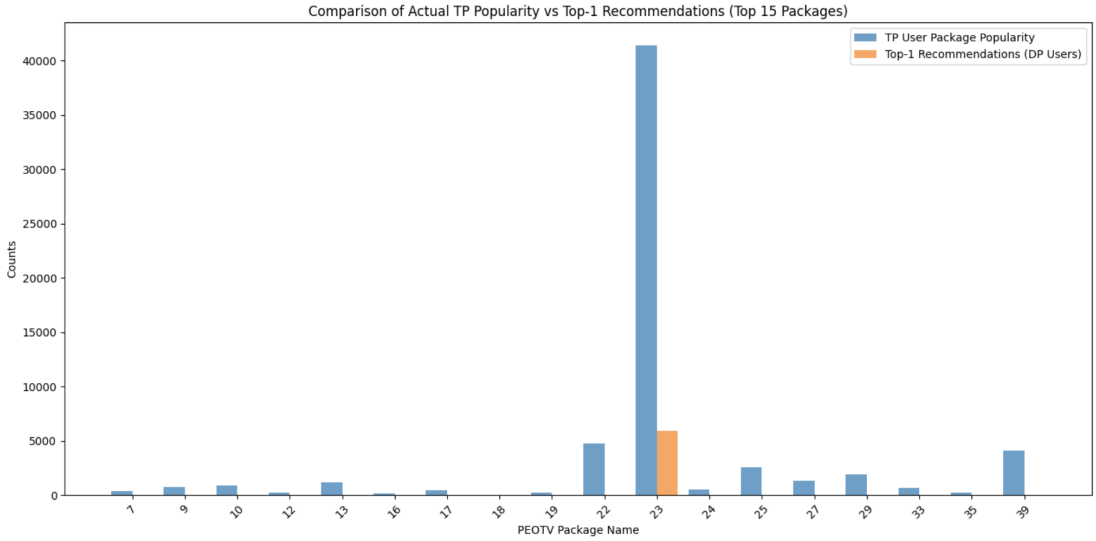

Fig. 4.13: Comparison of Actual TP Popularity vs Top-1 Recommendations (Top 15 Packages)

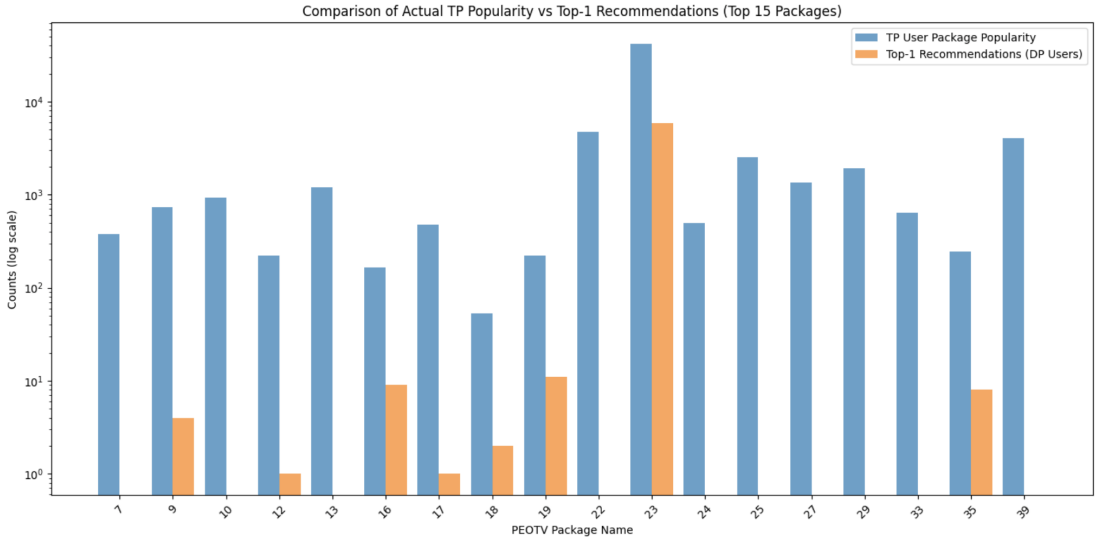

Fig. 4.14: Comparison of Actual TP Popularity vs Top-1 Recommendations (Top 15 Packages) - Log Scale

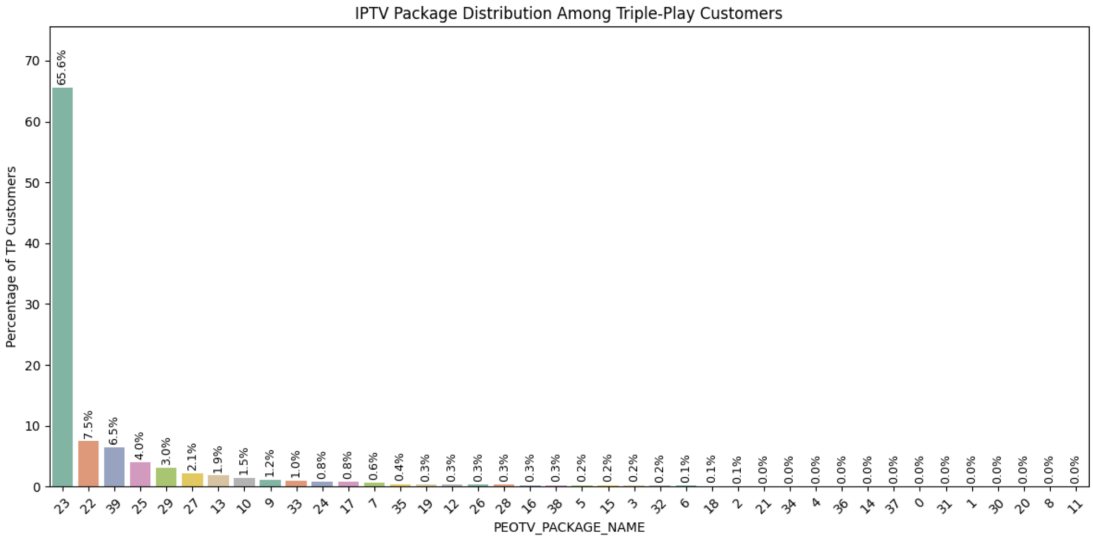

Fig. 4.15: IPTV Package distribution across Triple-Play customers

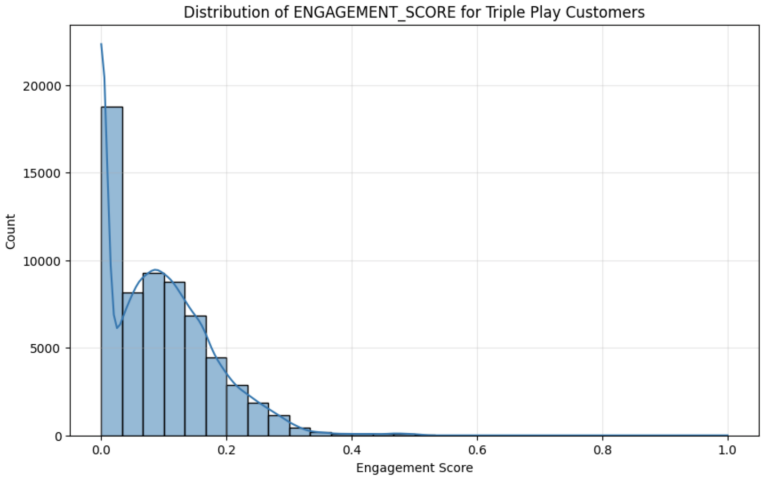

Fig. 4.16: Customer-Package Interaction Matrix

4.4 Verification Against Standard Benchmarks

To verify the effectiveness and generalizability of the proposed customer segmentation and recommendation pipeline, performance was evaluated using widely accepted benchmarking metrics across both components of the methodology.

4.4.1 Clustering Evaluation Benchmarks

The segmentation stage, which employed both K-Means and Gaussian Mixture Models (GMM), was assessed using multiple internal clustering validation indices, namely:

- **Silhouette Score**: Measures the compactness of clusters and the degree of separation between them, balancing intra-cluster cohesion and inter-cluster separation.
- **Davies-Bouldin Index (DBI)**: Quantifies average similarity between each cluster and its most similar counterpart, with lower values indicating better clustering.
- **Calinski-Harabasz Index (CHI)**: Evaluates cluster dispersion by comparing between-cluster variance to within-cluster variance, with higher values denoting more distinct clusters.

These standard metrics were computed for multiple cluster configurations, and the results consistently supported the selection of GMM as the superior model for capturing overlapping behavioral patterns in the customer base.

4.4.2 Recommendation Evaluation Benchmarks

The SVD-based collaborative filtering recommender was evaluated through a comprehensive set of performance metrics:

- **Root Mean Square Error (RMSE)** and **Mean Absolute Error (MAE)**: Computed on holdout test sets with standard train-test splits and 5-fold cross-validation, yielding mean RMSE of approximately 0.0793 and MAE around 0.0607, demonstrating robust prediction accuracy on engagement scores
- Top-N Ranking Metrics: Given the recommendation task focus, ranking-based evaluations were performed using simulated metrics:
  - **Simulated Recall@5**: Approximately 0.234, indicating moderate success in retrieving relevant packages within the top-5 recommendations.

  - **Simulated Normalized Discounted Cumulative Gain (NDCG@5)**: High value around 0.818, reflecting effective ranking and prioritization of relevant packages.

- **Coverage**: Measured as the proportion of unique packages recommended to users relative to the full package catalog, with an observed value of approximately 20%, indicating a focused but limited scope of recommendations.

- **Diversity**: Evaluated using pairwise cosine similarity of item embeddings derived from the model, producing a high diversity score (0.974), suggesting the recommender delivers meaningful personalized variation despite coverage constraints.

These evaluations collectively validate the model's effectiveness under standard recommendation benchmarks. The SVD-based system exhibited strong alignment with dominant customer preferences, supported by its accuracy and ranking fidelity. However, the observed limitations in recommendation breadth underscore the challenges posed by sparse and imbalanced engagement data, suggesting that future enhancements such as hybrid approaches which may further improve system performance.

CHAPTER 5

CONCLUSION AND FUTURE WORK

5.1 Conclusion

This study explored a customer segmentation-driven approach to predicting triple-play service adoption, integrating Gaussian Mixture Models (GMM) for clustering and a colloborative filtering-based recommendation approach using Singular Value Decomposition (SVD) for cross-selling recommendations. The approach specifically targeted double-play customers with the highest propensity to adopt triple-play services, enabling optimized marketing strategies through personalized service recommendations.

5.1.1 Customer Segmentation

- GMM was chosen over K-Means clustering due to its superior performance in capturing underlying customer distributions.

- The optimal number of clusters was determined using the Bayesian Information Criterion (BIC), the Akaike Information Criterion (AIC) and Silhouette scores, identifying 8 primary customer segments.

- In particular, three clusters (Clusters 1, 4, and 6) were identified as high-value Triple-Play segments and were used as behavioral benchmarks to inform IPTV recommendations for Double-Play customers.

5.1.2 Cross-Selling Recommendation Engine

- A collaborative filtering-based recommendation system was developed using Singular Value Decomposition (SVD) to predict IPTV package adoption for Double-Play (DP) customers, leveraging behavioral similarities with existing Triple-Play (TP) subscribers.

- The system utilized historical engagement scores to generate personalized recommendations.

- Despite extreme sparsity in the customer-package interaction matrix (sparsity = 98.74%), the model effectively identified dominant customer preferences. Specifically, the top-1 recommendations predominantly featured a single IPTV package referred to as Package-23 which aligned with its real-world popularity.

- These results demonstrate the model's ability to deliver practical, targeted cross-selling suggestions based on latent preference patterns, though future improvements may focus on increasing catalog coverage and incorporating hybrid recommendation strategies to address long-tail package underrepresentation.

5.1.3 Business Implications

- The findings enable data-driven marketing strategies, helping to increase adoption rates and enhance customer retention.
- The segmentation model can be used for personalized customer engagement, ensuring targeted promotional efforts for high-potential customers.

In general, this research provides a practical framework for customer segmentation and predictive modeling in the telecom industry, offering valuable insights for marketing and business strategy optimization.

5.2 Future Work

While this study achieved significant results, there are several areas for further improvement and exploration:

Enhanced Feature Engineering

- Incorporating additional behavioral metrics such as customer interactions, complaints, and support tickets to refine the accuracy of the model.
- Analyze temporal patterns in service usage to predict seasonality-driven adoption trends.
- Introducing behavioral data such as watch history, session duration, and channel preferences to enrich user-item interactions.

Advanced Modeling Techniques

- Exploring hybrid recommendation models to improve predictive performance.
- Introducing a graph-based approach to recommend packages for new users based on demographic similarities.

Business Implications

- Conduct a practical validation of the proposed cross-selling recommendation framework by deploying it in a real-world environment and collecting customer feedback to assess its effectiveness and user satisfaction.

- Implementing A/B testing frameworks to validate recommendation effectiveness through controlled experiments comparing customer adoption rates between model-suggested packages and existing recommendation strategies, measuring both short-term conversion and long-term retention metrics.
- Investigate the potential for cross-selling Value-Added Services (VAS) such as premium content, gaming bundles or home automation solutions.

5.3 Final Remarks

This research demonstrates the effectiveness of customer segmentation and cross-selling prediction in improving service adoption strategies. Using machine learning-driven insights, telecommunication providers can deliver personalized offers, optimize marketing strategies, and enhance the customer experience. Future advancements in feature engineering, hybrid recommendation approaches, and real-time deployment will further refine this approach, ensuring continued improvements in customer engagement and revenue growth.

REFERENCES

[1] T. A. Alghamdi and N. Javaid, "A survey of preprocessing methods used for analysis of big data originated from smart grids," *IEEE Access*, vol. 10, pp. 29 149–29 171, 2022.

[2] A. Tawakuli and T. Engel, "Towards normalizing the design phase of data preprocessing pipelines for iot data," in *2022 IEEE International Conference on Big Data (Big Data)*, 2022, pp. 4589–4594.

[3] Z. Mundargi, S. Bhatti, A. Chandra, A. Kamble, B. Jiby, and R. Arole, "Prepy - a customize library for data preprocessing in python," in *2023 International Conference for Advancement in Technology (ICONAT)*, 2023, pp. 1–5.

[4] Z. Guan, T. Ji, X. Qian, Y. Ma, and X. Hong, "A survey on big data preprocessing," in *2017 5th Intl Conf on Applied Computing and Information Technology/4th Intl Conf on Computational Science/Intelligence and Applied Informatics/2nd Intl Conf on Big Data, Cloud Computing, Data Science (ACIT-CSII-BCD)*, 2017, pp. 241–247.

[5] J. Abasova, J. Janosik, V. Simoncicova, and P. Tanuska, "Proposal of effective preprocessing techniques of financial data," in *2018 IEEE 22nd International Conference on Intelligent Engineering Systems (INES)*, 2018, pp. 000 293–000 298.

[6] V. Desai and D. H A, "A hybrid approach to data pre-processing methods," in *2020 IEEE International Conference for Innovation in Technology (INOCON)*, 2020, pp. 1–4.

[7] B. Song, "A path to implementing a fresh produce e-commerce customer segmentation method based on clustering algorithms," in *2023 IEEE 7th Information Technology and Mechatronics Engineering Conference (ITOEC)*, vol. 7, 2023, pp. 1047–1050.

[8] S. Ren, Q. Sun, and Y. Shi, "Customer segmentation of bank based on data warehouse and data mining," in *2010 2nd IEEE International Conference on Information Management and Engineering*, 2010, pp. 349–353.

[9] A. Afzal, L. Khan, M. Z. Hussain, M. Zulkifl Hasan, M. Mustafa, A. Khalid, R. Awan, F. Ashraf, Z. A. Khan, and A. Javaid, "Customer segmentation using hierarchical clustering," in *2024 IEEE 9th International Conference for Convergence in Technology (I2CT)*, 2024, pp. 1–6.

[10] S. Lukas, R. M. Suryaputri, and P. Widjaja, "Comparison of clustering results using k-means, gaussian mixture models, based on seven sectors of country electricity and correlation with gross national income," in *2024 2nd International Conference on Technology Innovation and Its Applications (ICTIIA)*, 2024, pp. 1–4.

[11] S. Manimozhi, D. Ruby, and K. Biruntha, "An elegant evaluation: Triangulating clustering methods for customer segmentation," in *2024 Second International Conference on Emerging Trends in Information Technology and Engineering (ICETITE)*, 2024, pp. 1–6.

[12] C. Yuan and H. Yang, "Research on k-value selection method of k-means clustering algorithm," *J*, vol. 2, no. 2, pp. 226–235, June 2019. [Online]. Available: https://doi.org/10.3390/j2020016

[13] E. Nandapala and K. Jayasena, "The practical approach in customers segmentation by using the k-means algorithm," in *2020 IEEE 15th International Conference on Industrial and Information Systems (ICIIS)*, 2020, pp. 344–349.

[14] M. Namvar, M. R. Gholamian, and S. KhakAbi, "A two phase clustering method for intelligent customer segmentation," in *2010 International Conference on Intelligent Systems, Modelling and Simulation*, 2010, pp. 215–219.

[15] A. A. Nugraha and M. Wasesa, "Customer segmentation and preference modeling of indonesian mobile telecommunication industry: A data mining approach," in *2021 6th International Conference on Management in Emerging Markets (ICMEM)*, 2021, pp. 1–6.

[16] H. Mahmood, T. Mehmood, and L. A. Al-Essa, "Optimizing clustering algorithms for anti-microbial evaluation data: A majority score-based evaluation of k-means, gaussian mixture model, and multivariate t-distribution mixtures," *IEEE Access*, vol. 11, pp. 79 793–79 800, 2023.

[17] N. D. Sugiharto, D. Elbert, J. Arnold, I. S. Edbert, and D. Suhartono, "Mall customer clustering using gaussian mixture model, k-means, and birch algorithm," in *2023 6th International Conference on Information and Communications Technology (ICOIACT)*, 2023, pp. 212–217.

[18] P. Falcarin, A. Vetrò, J. Yu, and S. Islam, "A recommender system for telecom users: Experimental evaluation of recommendation algorithms," in *2011 IEEE 10th International Conference on Cybernetic Intelligent Systems (CIS)*, 2011, pp. 81–85.

[19] A. Chakraborty, K. Dey, S. Ghosh, R. Chakraborty, I. Mitra, and P. Nandy, "A novel approach for customer segmentation and product recommendation to boost sales using machine learning," in *2023 10th International Conference on Computing for Sustainable Global Development (INDIACom)*, 2023, pp. 97–103.

[20] Y. Wang, "Movie recommendation system based on svd collaborative filtering," in *CAIBDA 2022; 2nd International Conference on Artificial Intelligence, Big Data and Algorithms*, 2022, pp. 1–7.

[21] Q. Ba, X. Li, and Z. Bai, "Clustering collaborative filtering recommendation system based on svd algorithm," in *2013 IEEE 4th International Conference on Software Engineering and Service Science*, 2013, pp. 963–967.

APPENDIX A

DATASET STRUCTURE

TABLE A.1: Structure of the Dataset used

Column	Type	Description
ACCOUNT_NUM_A	object	Unique account identifier
PROMOTION_INTEG_ID	object	Promotion integration ID
ONE_TIME_CHARGCE_AMT_AVG	float64	Average one-time charge amount
RECURRING_CHARGE_AMT_AVG	float64	Average recurring charge amount
LAST_BILL_CHARGE_AVG	float64	Average last bill charge
LAST_BILL_TOT_AVG	float64	Average last bill total amount
OUTSTANDING_AMT_MEAN	float64	Mean outstanding amount
MSAN_ID	float64	MSAN identifier
MSAN_PORT	object	MSAN physical port
RTOM	object	Regional territory code
LEA_CODE	object	Local exchange area code
CUSTOMER_REF_A	object	Unique customer reference ID
PROMOTION_NAME	object	Marketing promotion name
GENDER	object	Gender (Male, Female, Not Known)
CUSTOMER_PRIORITY	object	Customer priority level
LOYALTY_TIER_NAME	object	Loyalty tier
CUSTOMER_SEGMENT_DESC	object	Marketing segment description
CAT_SP_DP_TP	object	Bundle type (Double Play / Triple Play)
CUSTOMER_TYPE_CAT	object	Customer category

Continued on next page

TABLE A.1 – continued from previous page

Column	Type	Description
CUSTOMER_TYPE	int64	Customer type ID (numeric)
BUSINESS_LINE	object	Business line category
CREDIT_SCORE	float64	Credit score (1–100)
CREDIT_RANK	float64	Credit rank (1–10)
ZIP_CODE	int64	Postal code
CITY	object	Customer city
DISTRICT	object	Customer district
PROVINCE	object	Province or region
PAYMENTAMOUNT_AVG	float64	Average payment amount
BALANCE_FWD_MNY_AVG	float64	Average balance forward money
INVOICE_NET_MNY_AVG	float64	Average invoice net money
INVOICE_TAX_MNY_AVG	float64	Average invoice tax money
BALANCE_OUT_MNY_AVG	float64	Average balance outstanding money
MONTHLY_BILL_TOTAL_AVG	float64	Average monthly bill total
IDD_USAGE_MIN_AVG	float64	Average international direct dial usage (minutes)
IDD_OG_CALLS_AVG	float64	Average international outgoing calls
IINCOMING_USAGE_MIN_AVG	float64	Average incoming usage (minutes)
OUTGOING_USAGE_MIN_AVG	float64	Average outgoing usage (minutes)
VOICE_USAGE_MIN_AVG	float64	Average voice usage (minutes)
VOICE_FREE_USAGE_MIN_AVG	float64	Average free voice usage (minutes)
CALL_COUNT_AVG	float64	Average call count
REMAINING_DATA_VOLUME_AVG	float64	Average remaining data volume

Continued on next page

TABLE A.1 – continued from previous page

Column	Type	Description
USED_DATA_VOLUME_AVG	float64	Average used data volume
REMAINING_EXTRA_GB_VOLUME_AVG	float64	Average remaining extra GB volume
ADD_ON_DATA_GB_USAGE_AVG	float64	Average add-on data GB usage
PEOTV_USAGE_AVG	float64	Average PeoTV usage
MONTHLY_LIVE_TV_USAGE_MIN_AVG	float64	Average monthly live TV usage (minutes)
MONTHLY_TSTV_USAGE_MIN_AVG	float64	Average monthly TSTV usage (minutes)
BASE_BUNDLE_PACKGE	object	Base bundle package
CATEGORY_Broadband	bool	Broadband service category flag
CATEGORY_PeoTV	bool	PeoTV service category flag
CATEGORY_Voice	bool	Voice service category flag
PEOTV_PACKAGE_NAME	object	IPTV package name
SERVICE_USAGE_AVG	float64	Average service usage
BB_USAGE_REMAINING_RATIO	float64	Broadband usage remaining ratio
ENGAGEMENT_SCORE	float64	Customer engagement score