Financial Services Recommendation System

Statistical Modeling for Enhanced Financial Recommendations

Subbhashit Mukherjee
MT2023065
IIIT Bangalore
Bengaluru, India
Subbhashit.Mukherjee@iiitb.ac.in

Aryan Yadav
MT2023013
IIIT Bangalore
Bengaluru, India
Aryan.Yadav@iiitb.ac.in

Gummaraju Sai Hemanth
MT2023010
IIIT Bangalore
Bengaluru, India
email@iiitb.ac.in

Billa Abhignan
MT2023044
IIIT Bangalore
Bengaluru, India
email@iiitb.ac.in

***Abstract*—This paper introduces an innovative approach to enhancing the efficacy of a bank financial service recommender system by integrating advanced statistical techniques, namely Kernel Density Estimation (KDE) and Gaussian Mixture Models (GMM). Initially, synthetic data generation utilizing KDE and GMM enriches the dataset, ensuring a robust representation of user profiles. Subsequently, this enriched dataset is utilized in conjunction with two distinct recommendation methodologies: the first leveraging custom-made transformers to encode user-specific attributes for predictive modeling, and the second incorporating clustering techniques with distance metrics. Through rigorous evaluation and comparative analysis, this paper meticulously examines the effectiveness and performance of these methodologies in recommending tailored financial services to users, thereby contributing to the advancement of personalized banking experiences.**

Keywords—bank financial services, recommender system, Kernel Density Estimation (KDE), Gaussian Mixture Models (GMM), synthetic data generation, personalized banking, predictive modeling, clustering techniques, distance metrics

I. Introduction

In today's banking landscape, the demand for personalized financial services has surged, prompting banks to innovate and meet evolving customer expectations. This paper introduces a pioneering approach aimed at bolstering the effectiveness of bank financial service recommender systems by integrating advanced statistical techniques, namely Kernel Density Estimation (KDE) and Gaussian Mixture Models (GMM).

Central to our methodology is the utilization of KDE and GMM for synthetic data generation, enriching the dataset with comprehensive user profiles. This augmented dataset forms the cornerstone for deploying two distinct recommendation methodologies. The first methodology harnesses custom-made transformers, finely tuned to encode user-specific attributes for predictive modeling. Concurrently, the second methodology integrates clustering techniques empowered with distance metrics to refine recommendation accuracy.

The efficacy and performance of these methodologies are meticulously evaluated through an exhaustive comparative analysis. By closely examining the effectiveness of KDE, GMM, and the recommendation methodologies, this paper delves deep into the intricacies of tailoring financial service recommendations to individual users. Our findings not only illuminate the subtleties of personalized banking experiences but also propel forward the frontier of recommendation systems within the banking sector.

Through this research, we endeavor to furnish invaluable insights that empower financial institutions to optimize their recommendation systems. By doing so, we aim to significantly enhance the standard of personalized banking experiences for customers, ultimately driving greater satisfaction and engagement.

II. Dataset Creation

The dataset utilized in our paper was sourced from Kaggle's Santander Product Recommendation competition, chosen for its relevance to the study's objectives. Upon initial examination, the dataset exhibited several data quality issues, notably skewed distributions in certain target variables and user features. To address these issues, a comprehensive preprocessing approach was adopted. Null values, outliers, and biases in user feature values were identified as primary concerns. Null values were addressed through imputation techniques to maintain the dataset's integrity. Outliers were detected and treated using appropriate methods to minimize their impact on subsequent analysis. Furthermore, biases in user feature values were mitigated through strategic approaches such as binning and data filtering.

The dataset utilized in given the presence of skewed data distributions, particularly in certain target variables and user features, traditional preprocessing methods alone were deemed insufficient to adequately represent the underlying data distribution. To address this limitation and enhance the dataset's diversity and representativeness, synthetic data generation techniques were incorporated. Kernel Density Estimation (KDE) and Gaussian Mixture Models (GMM) were leveraged to generate synthetic data points that closely approximated the distribution of existing data.

A. Gaussian Mixture Models

In this section, we delineate our methodology for employing Gaussian Mixture Models (GMM) with Gaussian multivariate distributions to extract both local and global information from complex datasets, while concurrently determining optimal model complexity using the Akaike

Information Criterion (AIC) and Bayesian Information Criterion (BIC).

We begin by elucidating the rationale behind employing GMM in our analysis, emphasizing its flexibility in capturing intricate data structures while accommodating multiple variables simultaneously. Following this, we detail our approach, beginning with data preprocessing to ensure compatibility with GMM. Subsequently, we fit a GMM to the preprocessed dataset, initializing model parameters and optimizing the fit using appropriate algorithms. Crucially, we employ the AIC and BIC to determine the optimal number of components in our GMM. By generating AIC and BIC plots for varying values of N and analyzing their trends, we identify the model complexity that optimally balances fit and complexity. This rigorous model selection process ensures that our GMM accurately captures the dataset's underlying structure without overfitting [1]. Integrating AIC and BIC analysis into our methodology enhances the reliability and interpretability of our results, providing a robust foundation for extracting insights from multivariate datasets using GMM. Upon fitting the GMM, we extract both local and global information from the dataset. Local information entails analyzing mixture components to identify cluster centroids and covariance matrices, providing insights into individual cluster structures. Simultaneously, we leverage the GMM framework to discern global trends by considering the joint distribution of multiple variables.

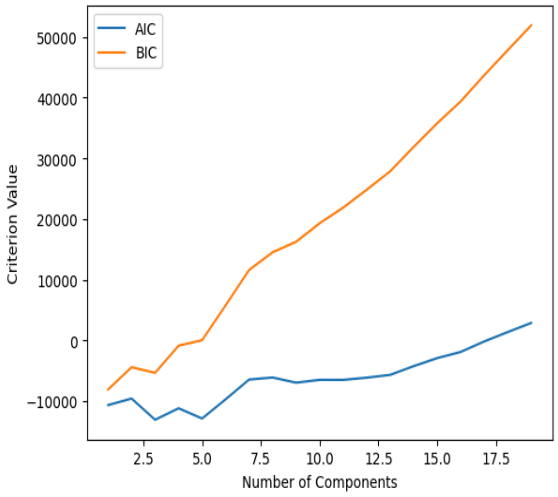

Fig. 1. Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC) Curve

B. *Kernel Density Estimation*

In our paper, we applied Kernel Density Estimation (KDE) to develop synthetic data for a recommendation system targeting the Spanish Santander Bank, using data from a Kaggle competition. Our preference for KDE stemmed from its capability to handle complex aspects of the dataset that Gaussian Mixture Models (GMM) failed to resolve effectively. As a non-parametric approach for estimating the probability density function of a random variable, KDE was chosen for its flexibility and non-reliance on a fixed parametric framework.

The creation of synthetic data through KDE provided a deeper level of analysis and improved the efficiency of the clustering algorithms used in our recommendation system. By closely replicating the statistical attributes of the original data, we managed to overcome issues related to missing values that previously skewed our analysis [2]. This enhanced dataset led to more precise similarity assessments, enabling the subsequent recommendation algorithms to more accurately customize financial product offerings based on individual customer needs.

III. Algorithms Applied

For making bank financial services as recommendations, we have considered two scenarios for making predictions, if the user hasn't purchased a single financial service from the bank, then we will use a transformer to predict products and if some items items were purchased before by the user, then we will apply k means and suitable distance metrics to make predictions.

A. *Transformer*

In this section, we present our novel approach to personalized bank product recommendation using a custom transformer architecture. We leverage a comprehensive set of user features to represent individual users within the transformer model. These features include data_date, gender, seniority, channel_used, province code, province name, activity index, gross income household, segmentation, and age group. On these user features we apply our own custom transformer architecture which is tailored for predicting user preferences based on a comprehensive set of input features. The architecture includes an input layer to receive user features, an embedding layer to convert categorical features into dense representations, and reshaping layers to prepare the input for subsequent processing [3]. The core of the architecture consists of transformer layers, incorporating multi-head attention mechanisms and feed-forward networks to capture complex relationships and higher-level features. Dropout and layer normalization are applied for regularization and stability. Global average pooling aggregates information for final predictions, which are generated by the output layer with a sigmoid activation function.

Layer (type)	Output Shape	Param #	Connected to
input_layer (InputLayer)	(None, 10)	0	-
embedding (Embedding)	(None, 10, 64)	6,400	input_layer[0][0]
reshape (Reshape)	(None, 10, 64)	0	embedding[0][0]
multi_head_attenti… (MultiHeadAttentio…	(None, 10, 64)	33,216	reshape[0][0], reshape[0][0]
dropout_1 (Dropout)	(None, 10, 64)	0	multi_head_atten…
add (Add)	(None, 10, 64)	0	dropout_1[0][0], reshape[0][0]
layer_normalization (LayerNormalizatio…	(None, 10, 64)	128	add[0][0]
dense (Dense)	(None, 10, 32)	2,080	layer_normalizat…
dense_1 (Dense)	(None, 10, 64)	2,112	dense[0][0]
dropout_2 (Dropout)	(None, 10, 64)	0	dense_1[0][0]
add_1 (Add)	(None, 10, 64)	0	dropout_2[0][0], layer_normalizat…
layer_normalizatio… (LayerNormalizatio…	(None, 10, 64)	128	add_1[0][0]
global_average_poo… (GlobalAveragePool…	(None, 64)	0	layer_normalizat…
dense_2 (Dense)	(None, 1)	65	global_average_p…

Total params: 44,129 (172.38 KB)
Trainable params: 44,129 (172.38 KB)
Non-trainable params: 0 (0.00 B)

Fig. 2. Transformer Architecture

Utilizing our custom transformer architecture, we generate probabilities for each user's preference towards our 24 products. These probabilities guide our recommendation system, where we select the top products based on the highest probabilities in descending order.

```
1 prod_current_accounts: 0.742905855178833
2 prod_particular_account: 0.3763580024242401
3 prod_e_account: 0.15566176176071167
4 prod_direct_debit: 0.12461651861667633
5 prod_taxes: 0.11302828043699265
6 prod_payroll_account: 0.1045469418168068
7 prod_credit_card: 0.09911097586154938
8 prod_securities: 0.08380809426307678
9 prod_long_term_deposits: 0.0738743469119072
10 prod_pensions2: 0.06563747674226761
```

Fig. 3. Transformer Top 10 Recommendations

B. KMeans

In this section we discuss the unsupervised learning technique "kmeans" which we performed on our data. KMeans clustering was employed as a pivotal technique in our recommendation system to categorize existing users into distinct groups based on their financial behaviors and preferences [5].

The dataset was preprocessed to address data quality issues and ensure compatibility with the clustering algorithm. Null values, outliers, and biases in user feature values were handled to maintain dataset integrity. Algorithm was applied to the preprocessed data to categorize existing users into distinct clusters. This facilitated efficient grouping of users based on their financial behaviors, enabling targeted marketing efforts and resource allocation.

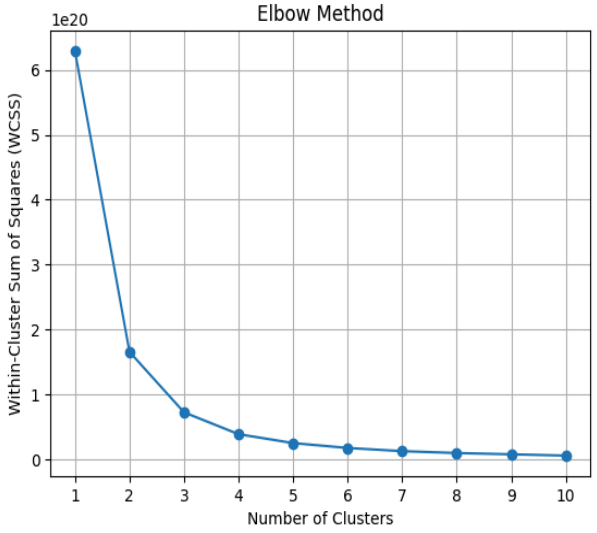

Fig. 4. Optimal clusters by Elbow method

Upon the arrival of a new user with substantial bank service usage, the algorithm assigned them to the appropriate cluster based on their financial behaviors and characteristics.

Leveraging **cosine similarity**, we accurately identified the top 10 customers within their assigned cluster whose financial behaviors closely resembled those of the new user. This metric allowed us to measure the similarity between users' preferences and behaviors, ensuring that recommendations were based on users with the most similar financial profiles. A weighted sum calculation was conducted, integrating product preferences and distance metrics, to assign scores to each product for the new user. Products were then sorted based on these scores, prioritizing those with higher relevance to the user's financial preferences.

```
Column: prod_e_account, Score: 54.99999999999979
Column: prod_current_accounts, Score: 53.99999999999982
Column: prod_long_term_deposits, Score: 50.99999999999983
Column: prod_direct_debit, Score: 39.999999999999865
Column: prod_payroll_account, Score: 30.999999999999915
Column: prod_pensions2, Score: 25.999999999999932
Column: prod_payroll, Score: 22.999999999999943
Column: prod_taxes, Score: 18.999999999999947
Column: prod_credit_card, Score: 16.99999999999996
Column: prod_funds, Score: 14.999999999999956
```

Fig. 5. Kmeans Top 10 Recommendations using cosine similarity metric

Pairwise distance metrics operate by computing the distance between feature vectors representing the financial behaviors and characteristics of users. By employing pairwise distance metrics, we systematically compare the financial preferences of the new user with those of existing users within the same cluster. This comparison allows us to identify users whose preferences closely align with those of the new user, facilitating the generation of personalized recommendations. Once the distances between the new user and other users within their cluster are computed, we assign scores to each product based on its relevance to both the new user and similar users within the cluster.

```
Column: prod_e_account, Score: 0.37293402147947413
Column: prod_current_accounts, Score: 0.37190847035571467
Column: prod_long_term_deposits, Score: 0.29129448278781034
Column: prod_direct_debit, Score: 0.24732954949971683
Column: prod_funds, Score: 0.12622296166614963
Column: prod_taxes, Score: 0.0893060500361404
Column: prod_credit_card, Score: 0.08515076600288012
Column: prod_particular_plus_account, Score: 0.0751242950563124
Column: prod_particular_account, Score: 0.07324047525231864
Column: prod_securities, Score: 0.05371191049503254
Column: prod_mas_particular_account, Score: 0.013668062779178267
```

Fig 6. Kmeans Top 10 Recommendations using pairwise distance metric

IV. Results and Analysis

To gauge the efficacy of the proposed banking product recommender solution we utilize a number of analytical methods these evaluations include, selecting the optimal synthetic data generation algorithms to accurately mimic customer behaviors, implementing strategies that balance exploration and exploitation to optimize recommendation accuracy, exploring association rule mining to show patterns in product relationships and employing the jaccard similarity index to verify the accuracy of our recommendations.

A. Synthetic Data Generation

In our paper, we adopted Kernel Density Estimation (KDE) to synthesize data for our banking recommendation system, utilizing a dataset from a Kaggle competition. We opted for KDE owing to its non-parametric nature, which is particularly effective for handling complex and multi-dimensional data typical in the banking sector. This method stands out from traditional parametric models like Gaussian Mixture Models (GMM) because it does not require the assumption of a predefined distribution. This quality enables a more accurate capture of the statistical traits of the original data. Such flexibility significantly enhances the predictive capability of our recommendation model, enabling it to better accommodate the subtle variations and complexities of financial data.

```
synthetic_data.head()
```

	data_date	employee_index	country_residence	gender	registration_date	new_customer_
0	1.435450e+09	3.0	0.0	1.0	1.437091e+09	0.0
1	1.440720e+09	3.0	0.0	0.0	1.049674e+09	0.0
2	1.453939e+09	3.0	0.0	0.0	1.075680e+09	0.0
3	1.461802e+09	3.0	0.0	0.0	1.399507e+09	0.0
4	1.425082e+09	3.0	0.0	1.0	9.905760e+08	0.0

Fig. 7. Synthetic data generation using KDE

B. Exploration-Exploitation Strategies

In our post-analysis phase, once we've identified the top 10 products along with their respective probabilities using the transformer model, we employ sophisticated exploration-exploitation strategies to delve deeper into user behavior and preferences. These strategies, including Upper Confidence Bound (UCB), Epsilon Greedy, and Thompson Sampling, are instrumental in refining our understanding and optimizing recommendation effectiveness [4].

UCB methodology enables us to explore user product selections by factoring in estimated upper confidence bounds, thereby accommodating prediction uncertainties. By scrutinizing both predicted probabilities and confidence intervals, we discern how users prioritize products with higher confidence levels. This analytical approach not only reveals evolving user preferences but also highlights shifts towards specific products or categories over time.

```
percentage_of_best, percentage_of_worst,percentage_of_second_best

([0.0, 0.994, 0.928, 0.758, 0.494, 0.268, 0.109, 0.083, 0.06],
 [0.0, 0.001, 0.003, 0.01, 0.02, 0.03, 0.038, 0.039, 0.041],
 [0.0, 0.0, 0.003, 0.013, 0.033, 0.052, 0.039, 0.043, 0.043])
```

Fig. 8. Probability change according to UCB for best, second best, and worst product

Epsilon Greedy strategy plays a pivotal role in monitoring user selections and preferences by striking a balance between exploration and exploitation. Through a dynamic allocation of recommendations towards exploring new options while leveraging established preferences, we adapt our strategies in real-time. The analysis of various epsilon values aids in fine-tuning the balance between exploring novel recommendations and exploiting known preferences, thus optimizing user satisfaction.

```
percentage_of_best, percentage_of_worst,percentage_of_second_best

([0.0, 0.994, 0.925, 0.699, 0.497, 0.361, 0.093, 0.074, 0.057],
 [0.0, 0.001, 0.003, 0.012, 0.021, 0.026, 0.039, 0.04, 0.041],
 [0.0, 0.0, 0.003, 0.017, 0.029, 0.041, 0.045, 0.042, 0.044])
```

Fig. 9. Probability change according to Epsilon Greedy for best, second best, and worst product

Thompson Sampling offers a probabilistic approach to exploration-exploitation dynamics, allowing us to balance the exploration of uncertain options with the exploitation of known preferences. By varying alpha and beta parameters, we observe how the algorithm adapts its recommendations to optimize user satisfaction. This analysis sheds light on the trade-offs between exploring new recommendations and exploiting known preferences, guiding the refinement of recommendation strategies

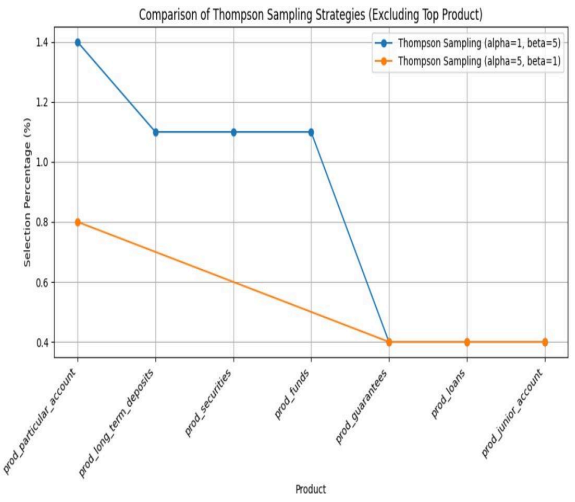

Fig. 10. Plot showing variations in selection percentage by varying alpha and beta in Thompson Sampling. (Prod_particular_account is our second-best product here)

C. *Association Rule Mining*

In our quest to refine our recommendation system, we embraced association rule mining as a pivotal technique to unearth intricate patterns and relationships within our extensive datasets of user product interactions. This method serves as a powerful means to discern significant associations among different products that users engage with over time. Unlike certain complex machine learning models, association rule mining offers transparent and interpretable results, aligning seamlessly with our commitment to clarity and insightfulness [6].

To harness the potential of association rule mining, we developed a bespoke function dubbed "search_association_rules." This function is meticulously crafted to sift through association rules based on user-used products as input, employing predefined thresholds of confidence, lift, and support. By adhering to these stringent criteria, we can identify association rules that hold tangible significance in our recommendation strategy. For instance, upon analysis, if a notable proportion of customers who have purchased products related to taxes and direct debit also demonstrate a propensity to purchase a product associated with pensions, it signals a robust association between these product categories. These insights gleaned from association rule mining empower us to refine our recommendation algorithms and tailor our offerings more effectively to align with user preferences and needs.

Through the strategic implementation of association rule mining, our goal is to uncover valuable insights into user behavior and preferences. By leveraging these insights, we can augment the effectiveness and relevance of our recommendation system, thereby delivering a more personalized and gratifying experience for our users.

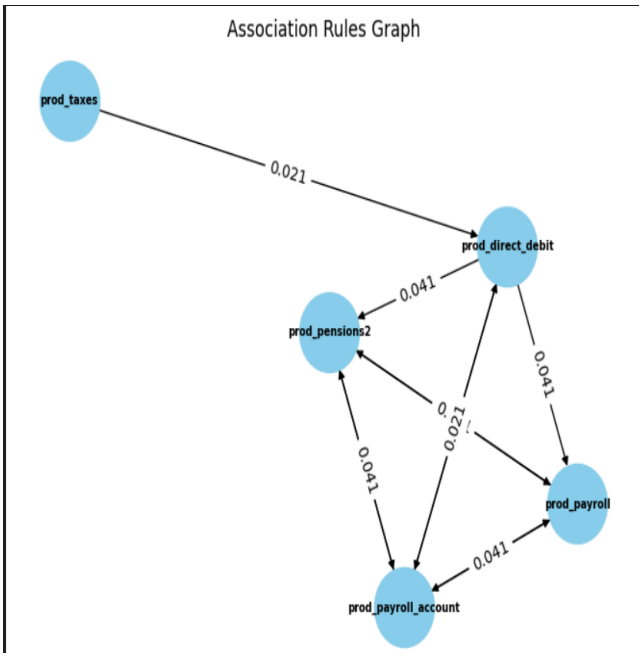

Fig. 11. Association graph for user who already purchased taxes and direct debit

D. *Jaccard Similarity*

We employed the Jaccard Similarity index to assess the performance of our banking product recommendation system. This metric is instrumental in gauging the congruence between the types of products customers actually purchase and those our system predicts. The product categories include a range of banking services such as "Accounts and Deposits" (encompassing savings and payroll accounts), "Investment Products" (including funds and securities), "Loans and Financing" (featuring various loans and mortgages), "Pensions and Retirement", "Payment Services", and "Other Financial Products".

In particular, we computed the Jaccard Similarity for two distinct recommendation methodologies: one utilizing Pairwise Distance and another based on Cosine Similarity. This index is essential for it measures the accuracy with which our model predicts customer preferences, where a higher Jaccard score indicates a more precise recommendation system. This suggests our system's proficiency in recognizing and predicting customer requirements based on their demographic details and past banking activities.

```
set1 = genres
set2 = genres_rec_pd
intersection = len(set1.intersection(set2))
union = len(set1.union(set2))
jaccard_similarity = intersection / union
print("Jaccard Similarity Score:", jaccard_similarity)
```

```
Jaccard Similarity Score: 0.5
```

Fig. 12. Jaccard Similarity Score for sample recommendation

E. *Data visualization*

In our recommendation analysis, we utilized data visualization techniques to gain valuable insights into customer behaviors and preferences. By visualizing key demographic and financial data, we aimed to identify patterns and trends that could inform our recommendation strategy [7].

The division of the age column into groups allowed us to identify distinct patterns and trends within different age cohorts, facilitating more targeted and personalized recommendations. For example, typical age ranges such as (18,25], (25,35], (35,45], and so forth, were chosen to capture distinct life stages and transitions, such as transitioning from education to early career, establishing financial independence, and reaching midlife milestones.

Notably, the (18,25] and (25,35] age groups exhibited a high proportion of customers associated with the "University" segment, indicating a potential correlation between age and educational status.

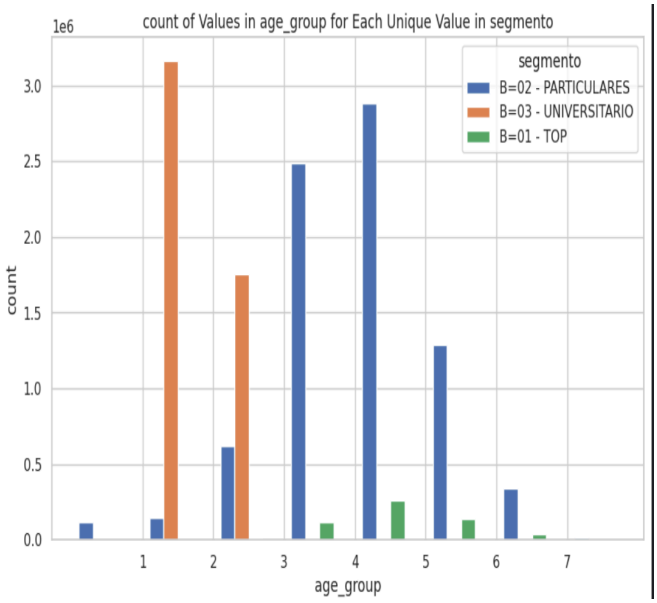

Fig. 13. Plot on "segmentation" column group by age

Additionally, by examining the enrollment patterns based on the month of first account creation, we identified a peak in enrollments in October, particularly within the 18 to 25 age group. This observation led us to speculate potential reasons for this trend, such as the commencement of college or the start of a first job, prompting individuals to open bank accounts.

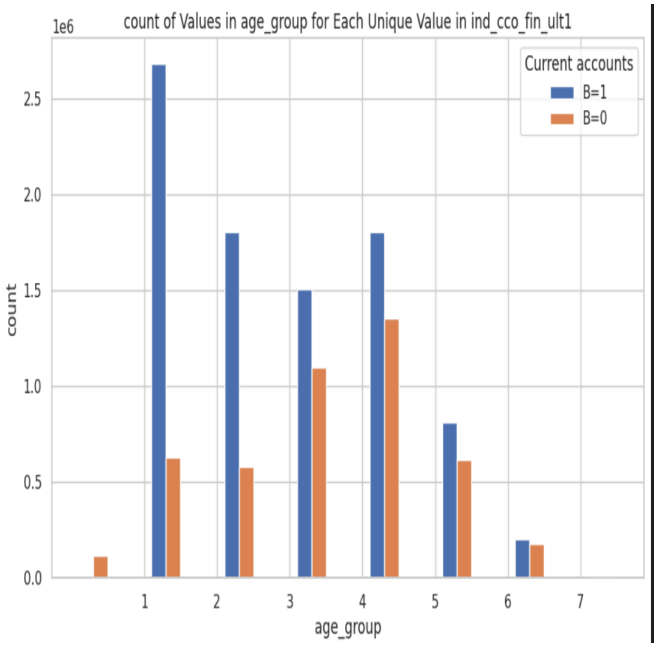

Fig. 14. Plot on "current accounts" columns group by age

Furthermore, our analysis extended to gross household income, which we categorized into exponential bins. By visualizing the distribution of current account openings across income ranges, we observed a concentration of maximum customers within the income range of 30000 to 300000.

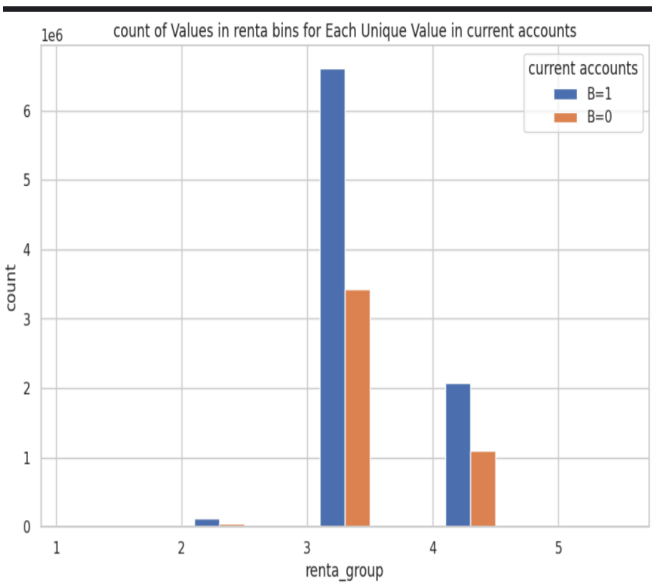

Fig. 15. Plot on "current accounts" column group by gross household income

Conclusion

Our paper introduces a thorough methodology to enhance banking product suggestions by utilizing data processing and machine learning techniques. By incorporating synthetically generated data points through kernel density estimation (KDE) into our existing dataset, we achieved a more balanced distribution which accurately reflects user profiles and target variables. This enhanced dataset reduced the issues arising from data skewness and improved the robustness of our analysis and modeling, leading to more dependable insights. We employed transformer-based models to customize product recommendations based on individual customer demographics, utilizing deep learning to analyze complex data patterns for personalized suggestions. Furthermore, we integrated a model based on the k-means clustering algorithm, enhanced by similarity metrics, to consider customers' previous purchasing behavior. This amalgamation of demographic data and historical purchase patterns was designed to increase the relevance and precision of our recommendations. We evaluated the effectiveness of our recommendation system using various analytical methods. Visual plots were created to illustrate the distribution of recommendations and user engagements, providing a clear graphical depiction of the system's operation. Association rule mining was used to discover correlations in product connections and customer preferences, shedding light on common trends and anomalies. The Jaccard similarity index was also utilized to measure the accuracy of our recommendations by comparing the predicted product categories with those actually bought by customers. These approaches contributed to a comprehensive evaluation of the effectiveness of the recommendation system.

References

[1] Carlo Cavicchi Maurizio, Vichi Maurizio, Vichi Giorgia, Zaccaria Giorgia Zaccaria, "Parsimonious ultrametric Gaussian mixture models".April 2024, DOI: 10.1007/s11222-024-10405-9

[2] John Heine, Erin E. E. Fowler, Anders Berglund, Michael J Schell, Steven A Eschrich, “Techniques to produce and evaluate realistic multivariate synthetic data”, July 2023, DOI: 10.1038/s41598-023-38832-0

[3] Fanfei Meng, Chen-Ao Wang, “Sentiment analysis with adaptive multi-head attention in Transformer ”, March 2024, DOI: 10.54254/2755-2721/50/20241326

[4] Kelsey Hagan, Ivieosa Aimufua, Ann HaynosB, Timothy Walsh, "The explore/exploit trade-off: An ecologically valid and translational framework that can advance mechanistic understanding of eating disorders", February 2024,DOI: 10.1002/eat.24173

[5] Kenneth EzukwokeSamaneh, ZareianSamaneh Zareian , “KMEANS AND KERNEL K MEANS A comparative study of classical and kernel k means for data clustering” December 2019 DOI:10.13140/RG.2.2.29297.43361

[6] Hend Amraoui, Faouzi Mhamdi, “Association Rule Mining for Multifactorial Diseases: Survey and Opportunities” February 2024 DOI:10.1007/978-3-031-51643-6_12

[7] Dada Ishola Durojaye, “8 Analysis and visualization of market segmentation in banking sector using Kmeans machine learning algorithm” . March 2022 ,DOI: 10.33003/fjs-2022-0601-910