Academic Report

Donna Oftadeh
Alberto Outumuro Bueno
Ana Paula Canseco
Miftah Mahmood

Data Science Nuclio Digital School

Abstract

EasyMoney, a fintech startup established by Carol Denver, has transitioned from a period of rapid growth and innovation to one of strategic consolidation. The company, known for its user-friendly online platform and the flagship 'smart piggy bank account,' is recalibrating its focus to maximize value from its existing customer base amid technological challenges and organizational changes.

The inclusion of a Data Scientist, Bob, marks a shift towards a data-centric approach in augmenting revenue streams. This academic report encapsulates a four-month journey of data exploration, customer segmentation using k-modes clustering, and the development of actionable business insights through a recommendation model. Our robust analysis delineated eight distinct customer segments, each characterized by specific behavioral patterns and product preferences.

The project's crux was to pivot EasyMoney's strategy from customer acquisition to monetization and engagement, using a blend of data science methodologies. Through meticulous data cleaning, feature engineering, and iterative modeling, we have paved the way for targeted marketing campaigns and enhanced customer experiences. Our findings suggest a nuanced understanding of the customer lifecycle, necessitating personalized engagement strategies to foster loyalty and growth.

The recommendations put forth in this report are grounded in quantitative evidence and reflect a strategic realignment with EasyMoney's long-term vision, underscoring the power of data science in reshaping business trajectories

.

Keywords: *Customer Segmentation, Data Analysis , K-modes Clustering, Recommendation Model, MLOps Framework, Monetization Strategy, Behavioral Insights, Data-Driven Marketing, Revenue Optimization, User Engagement*

Data Processing

In reviewing our customer database, which contains 5,962,924 customer IDs, we've identified several key points and areas for data cleaning and restructuring to better analyze and utilize our data.

- We confirmed that there are no duplicates across the pk_cid (customer ID) and pk_partition fields, ensuring unique entries for our analysis.
- Most fields, excluding IDs, should be converted to boolean to streamline our analysis process.
- We've identified a significant number of missing values: 133,944 in the segment and 133,033 in the entry_channel fields.
- The em_account_pp column, indicating subscribers to the easyMoney++ account, contains zeros for all entries, suggesting no subscribers. Despite this, the column will be retained for now as it provides insight into product subscription rates.
- Payroll and pension information is missing concurrently, indicating a pattern in missing financial data.
- Additional fields with missing values include gender, salary, and region code.
- Notably, 1,075 entries correspond to deceased individuals. While these cannot be counted as active customers, their data may still offer valuable insights into behavioral patterns.

Data Cleaning

- Convert pk_cid to string and partition fields to datetime format to better reflect their nature and enhance data handling.
- Address missing values, particularly in entry_channel and segment, and ensure entry_date is also in datetime format.
- For product data, cid should be string-formatted and partition dates standardized to datetime. The decision to potentially remove the em_account_pp column will be postponed, acknowledging even zero-subscriber data as relevant for now.
- Boolean conversion is advised for clarity in product-related columns, with special attention to resolving missing payroll and pension plan data, which coincidentally overlaps in rows.
- Socio-demographic data reveals missing values in region codes, gender, and notably high missing salary data. Additionally, the presence of deceased individuals' records prompts a careful approach to leveraging their historical data while acknowledging their status.

Overall, our immediate focus will be on cleaning and structuring this data to better support our analytical and commercial activities, ensuring a more accurate and insightful understanding of our customer base and product engagement.

Analysis

Our dataset comprises 5,962,923 records with a significant number of duplicate entries; only 456,373 unique customer IDs (pk_cid) are identified. This suggests that customer records are updated with each batch update, presenting a unique challenge for analysis. To address this, we plan to:

- Review duplicates meticulously to ensure comprehensive understanding.
- Retain the most recent entry based on the date (pk_date) while removing outdated ones.
- The duplicate IDs across different dates are invaluable for behavioral analysis, offering insights into customer patterns over time. This necessitates a time series analysis, converting dates into datetime format, handling missing values, and creating an aggregated table by date with the date as an index.
- We've identified an error with entry dates, specifically the incorrect date of February 29, 2015, which doesn't exist. To rectify this, we'll adjust these entries to March 1, 2015. Additionally, we aim to transform the entry date into datetime format for more accurate analysis.
- Analysis of a particular customer (ID 1375586) reveals a pattern of product usage and app activity, highlighting the importance of understanding customer engagement over time. This customer's interaction, including periods of inactivity and usage of services like debit cards, provides a microcosm of broader customer behavior trends.
- A significant finding is that 75% of our customers do not have a short-term deposit, with only 0.25% subscribing to this service. A detailed time-based analysis shows fluctuations in subscriber numbers, with a peak in early 2018 followed by a decline and then slight recoveries and drops. This pattern underscores the need for further exploration into the reasons behind these trends, suggesting that product launches may have been rushed or not fully supported by the platform's capacity.

- The impact of deceased customers on our analysis is also a consideration. By converting their status into numerical values, we aim to better understand how this subset of data influences overall trends and insights.

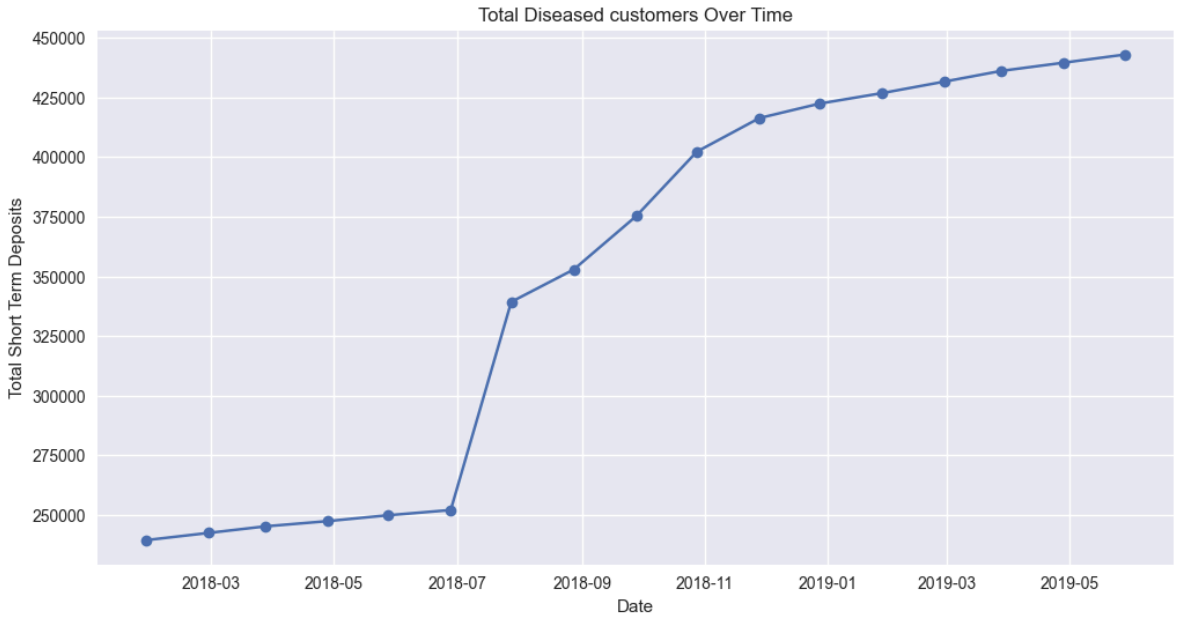
The preliminary analysis indicates a complex interplay of customer behaviors, subscription trends, and data integrity issues. By addressing duplicates, refining our time series approach, and carefully analyzing specific behaviors and trends, we can uncover actionable insights to guide future strategies. This includes reevaluating product launches, platform capacity, and marketing efforts to better meet customer needs and drive engagement.

- Observing the trend in deceased subscribers, we noted an increase from 45 in July 2018 to 86 in June 2019. However, this appeared to be an accumulation over time, rendering it meaningless. To rectify this, we checked whether deceased individuals from one batch appeared in subsequent batches, leading to the decision to remove repetitive deceased customer entries. Subsequent analysis involved sketching charts and examining trends in subscribed and unsubscribed individuals, entry dates, and active users.

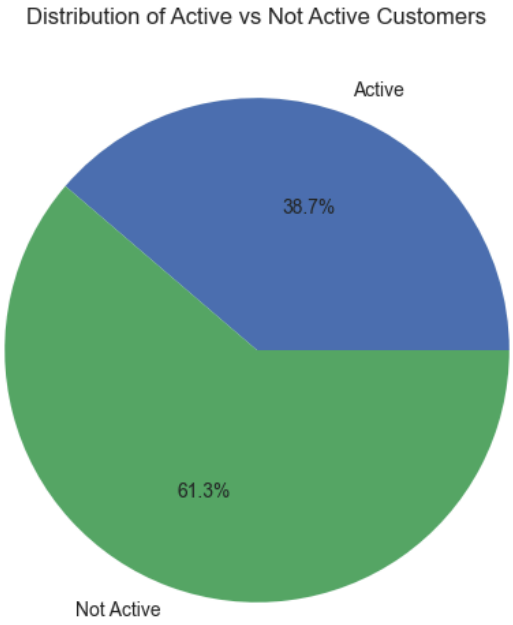
- Following the removal of repetitive deceased customer rows, we conducted another round of analysis to visualize and analyze the data. The revised numbers now appear more logical and reflective of real scenarios.

- Prior to February 2018, the dataset recorded 47 deceased individuals, a cumulative figure from previous years. This number decreased to zero for three consecutive months (February, March, April), followed by fluctuations in subsequent months. Peaks were observed in September 2018 (10 deaths) and February/March 2019 (12 and 10 deaths, respectively).
- We analyzed the distribution of deceased versus alive subscribers, acknowledging the impact of subscriber age and the likelihood of deaths. Our focus shifted towards understanding subscriber dynamics, commercial sections, and product engagement to comprehensively explore the customer journey and funnel.

- Despite the low percentage of deceased subscribers (0.0021%), deemed negligible for now, we retained them for analysis. Total subscriber numbers witnessed a significant increase, particularly in July 2018, underscoring the need for further investigation into active versus inactive subscribers to better understand customer engagement levels and potential churn. This analysis sets the stage for deeper exploration into subscriber behaviors and trends to inform strategic decision-making.

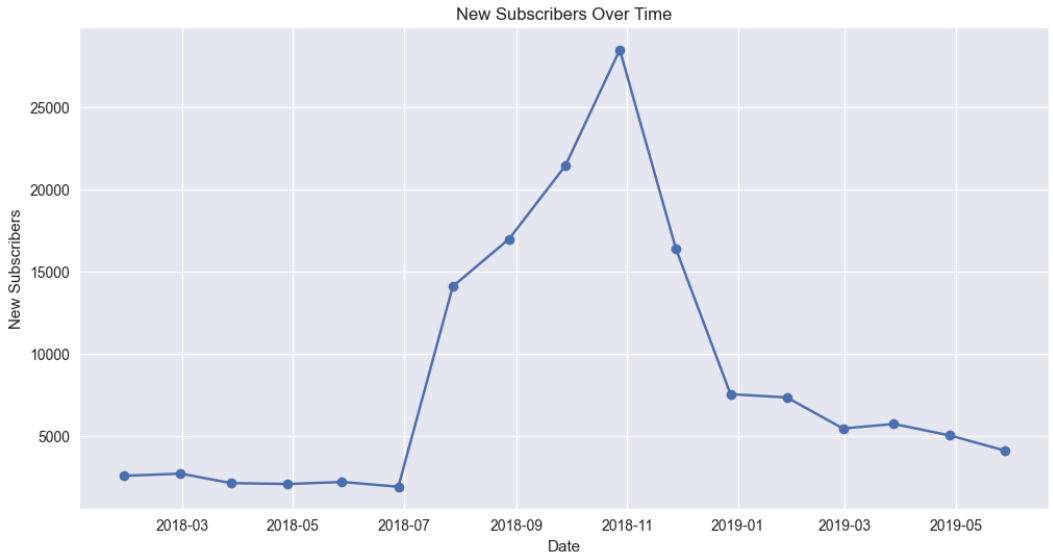


- The analysis of active subscribers reveals that 38.7% of subscribers are currently active, with 61.3% classified as inactive. This disparity underscores the need to target inactive customers for re-engagement efforts to maximize overall customer engagement and satisfaction. Additionally, tracking new subscriptions and unsubscriptions provides valuable insights into acquisition and churn rates, informing strategic decisions aimed at optimizing marketing efforts and enhancing customer retention.

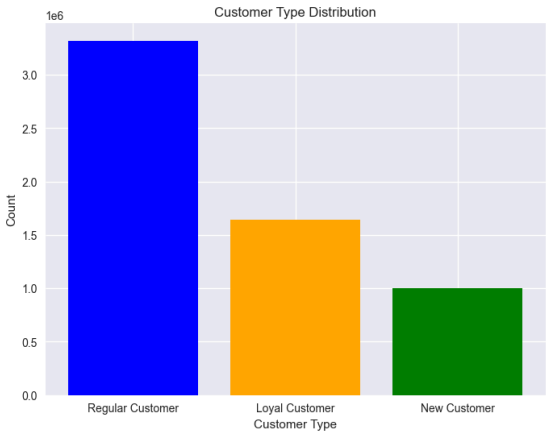


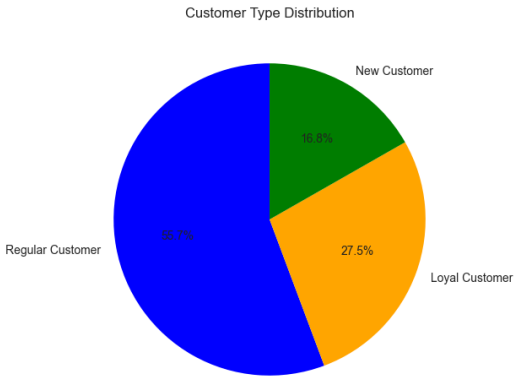
- The analysis of new subscribers reveals dynamic trends over time, with notable fluctuations observed across different months. In January 2018, there were 2590 new subscribers, maintaining a consistent pace until July 2018. However, a significant surge occurred from July to

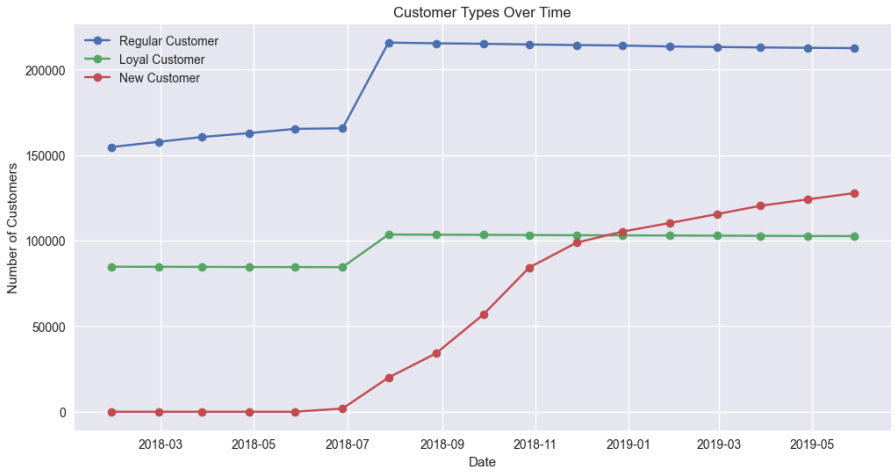
August 2018, with new subscribers skyrocketing from 1925 to 14095. This trend continued until October 2018, peaking at 28462 new subscribers before sharply declining in subsequent months. Possible factors influencing these fluctuations include seasonal variations, shifts in marketing strategies, product launches, competitive dynamics, economic conditions, and external influences such as partner and investor requests or technological advancements. Understanding these trends is vital for adapting strategies and maximizing customer acquisition efforts effectively.



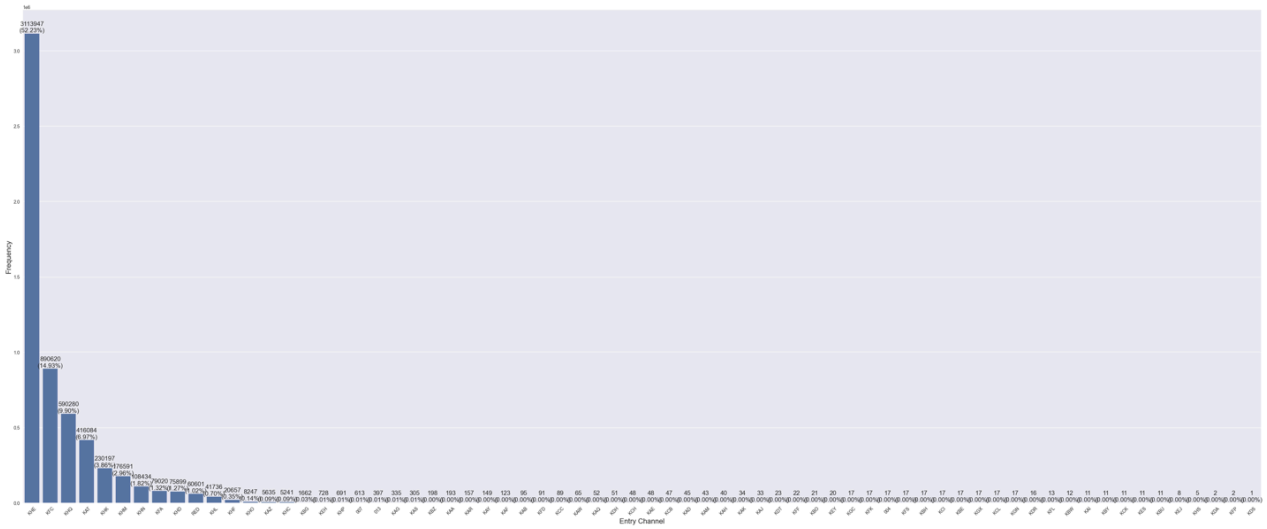
The **feature engineering process** involved categorizing customers based on their tenure, with distinctions made between loyal customers (subscribed for more than 3 years), intermediate customers (subscribed for 1 to 3 years), and new customers (subscribed for less than 1 year). This resulted in a distribution where regular customers accounted for the majority (55.7%), followed by loyal customers (27.5%), and new customers (16.8%). The distribution pattern illustrates a logical progression, with regular customers showing consistent counts over time, while new subscribers exhibit a notable increase starting from July 2018. This surge in new subscribers coincides with an overall rise in the total number of subscribers, particularly among regular and loyal customer segments. The shift towards prioritizing customer retention is evident, with the focus transitioning from acquiring new subscribers to maintaining existing ones. By December 2018, new subscribers surpassed loyal subscribers, highlighting the evolving strategy of Easy Money towards customer retention.

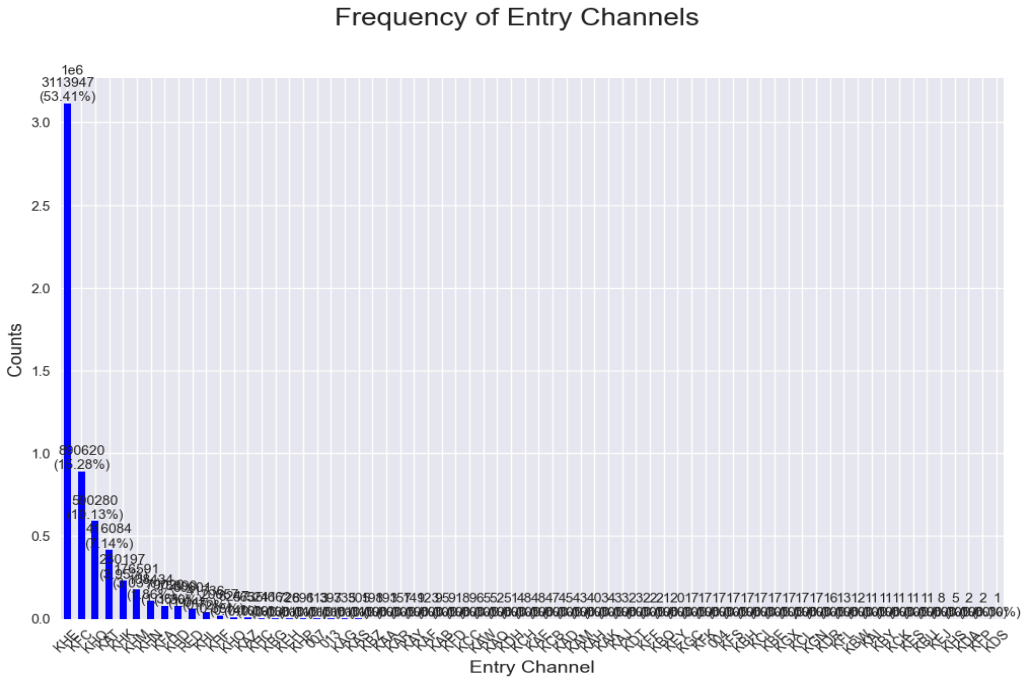


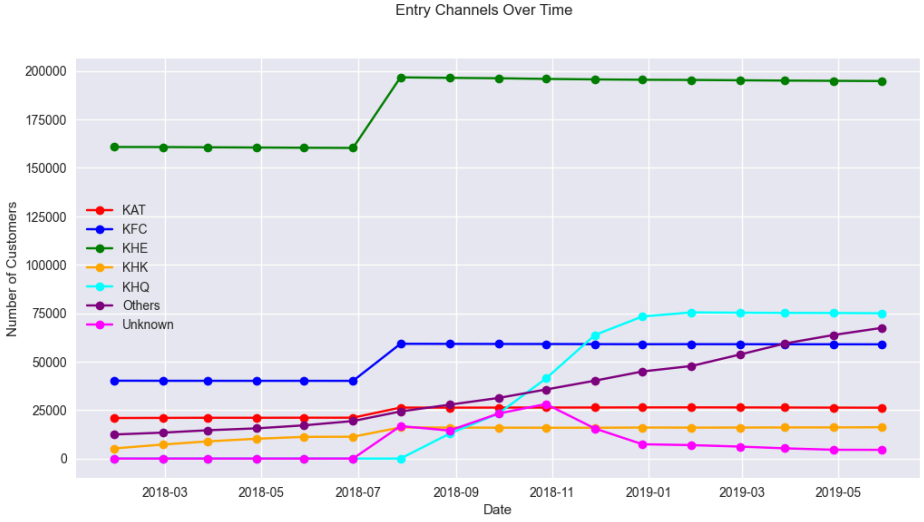




- The analysis of entry channels reveals valuable insights into customer acquisition strategies. The majority of customers were acquired through the KHE channel, comprising 52% of all entries, followed by KFC (15%) and KHQ (10%) channels. Notably, the remaining channels collectively contributed to 10% of the entries, labeled as 'Others.' Encoding and aggregating the channels in a monthly aggregated time series dataframe facilitated further examination. Key observations include significant jumps in subscribers from various channels, particularly KHE, KFC, and KAT, in July, indicating potential shifts in acquisition tactics or external factors. Additionally, the steady increase in subscribers from other channels underscores the importance of monitoring and optimizing entry channels to adapt strategies effectively and capitalize on emerging opportunities. The analysis emphasizes the dynamic nature of customer acquisition and the need for proactive adjustments to ensure sustainable growth and market competitiveness.


Frequency
Entry Channel


Frequency of Entry Channels
1e6
3113947
(53.41%)
890620
(15.28%)
590280
(10.13%)
416084
(7.14%)
Counts
Entry Channel


Entry Channels Over Time
KAT
KFC
KHE
KHK
KHQ
Others
Unknown
Number of Customers
Date

Segmentation (Using K-Modes Clustering)

Step 1: Preparing the Data and Selecting the Number of Clusters

Utilizing the K-Modes algorithm, we aimed to group customers based on behavior patterns, considering the categorical nature of the data.

- Identified potential features for clustering across demographics, financial status, product ownership and usage, and engagement channels.
- Aggregated binary features into meaningful groups to address dimensionality issues.

Step 2: Running K-modes Clustering

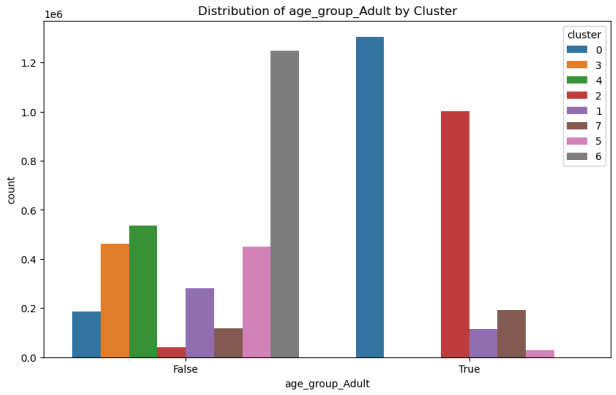
- Employed the K-Modes algorithm with "Cao" initialization, suitable for categorical data.
- Computed K-Modes for 7 and 8 clusters to compare cost and interpretability.
- Evaluated clusters based on cost, interpretability, and usefulness.

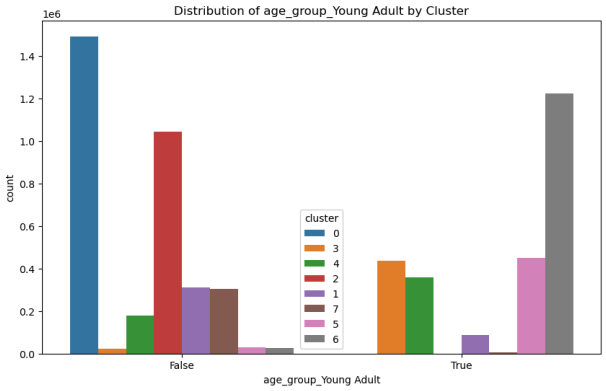
Step 3: Profiled the Clusters

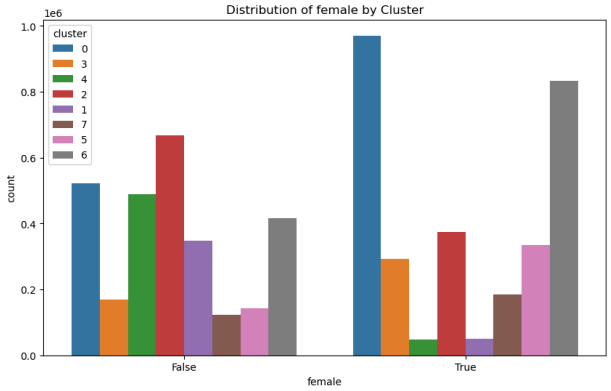
- Evaluated cluster profiles to understand commonalities and distinct characteristics across clusters.
- Examined product ownership, age groups, salary information, and cross-cluster comparisons.
- Identified 8 clusters as providing the best interpretation, segmentation, and cost.
- Additional Considerations and Validation

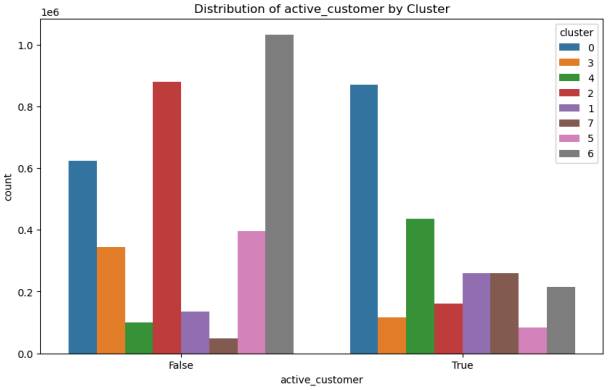
Proposed cross-tabulation of clusters with other categorical variables for further validation of segmentation coherence and alignment with business goals.

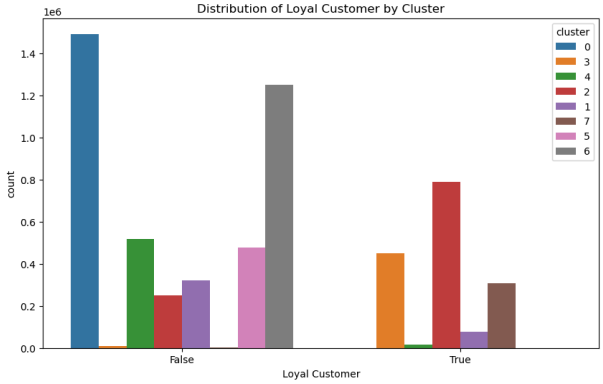
The chosen 8-cluster solution offers a nuanced segmentation that aligns with business objectives, providing actionable insights for targeted marketing strategies and customer engagement efforts. Further analysis and validation methods could enhance the robustness of the clustering results.



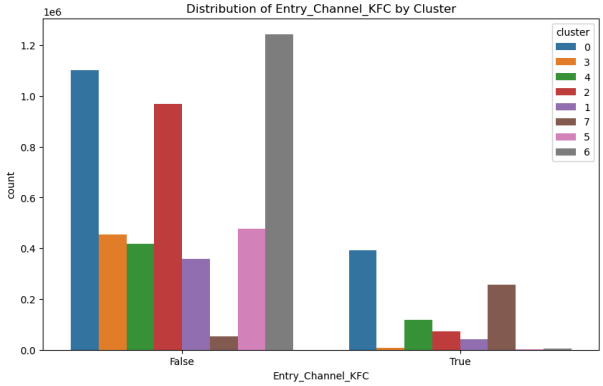


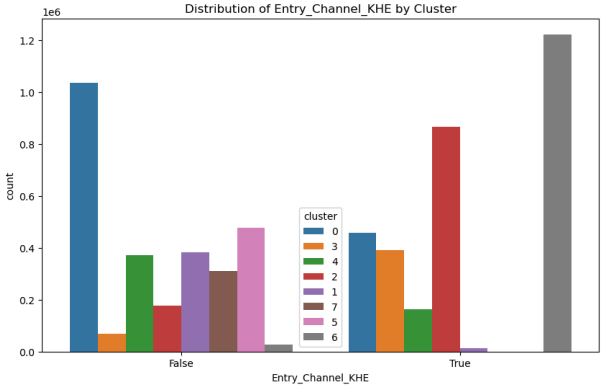

Distribution of female by Cluster
1e6
cluster
0
3
4
2
1
7
5
6
count
False
True
female

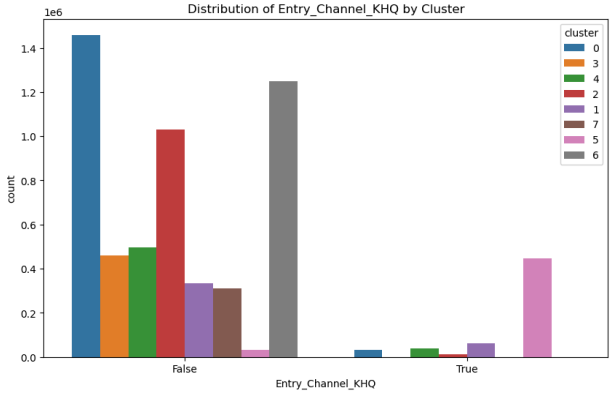

Distribution of active_customer by Cluster
1e6
cluster
0
3
4
2
1
7
5
6
count
False
True
active_customer

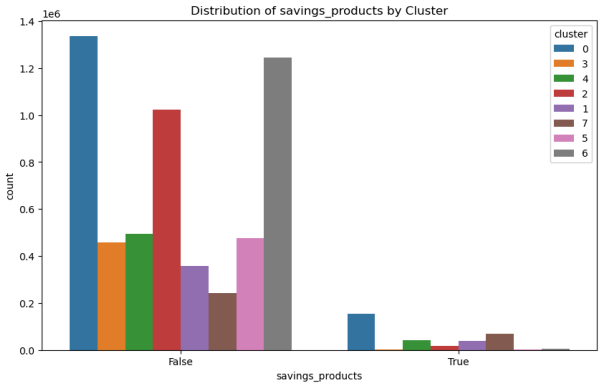

Distribution of Loyal Customer by Cluster
1e6
cluster
0
3
4
2
1
7
5
6
count
False
True
Loyal Customer

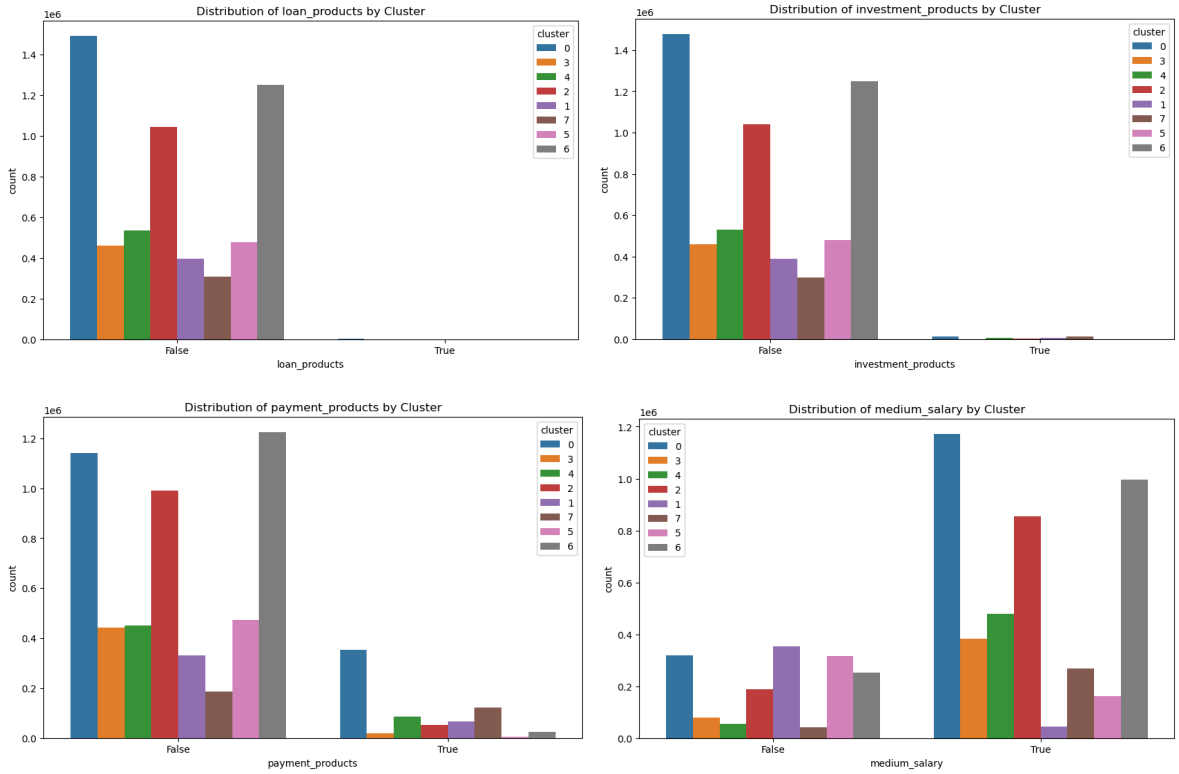

Distribution of Regular Customer by Cluster
1e6
cluster
0
3
4
2
1
7
5
6
count
False
True
Regular Customer


Distribution of Entry_Channel_KFC by Cluster
1e6
cluster
0
3
4
2
1
7
5
6
count
False
True
Entry_Channel_KFC


Distribution of Entry_Channel_KHE by Cluster
1e6
cluster
0
3
4
2
1
7
5
6
count
False
True
Entry_Channel_KHE


Distribution of Entry_Channel_KHQ by Cluster
1e6
cluster
0
3
4
2
1
7
5
6
count
False
True
Entry_Channel_KHQ


Distribution of savings_products by Cluster
1e6
cluster
0
3
4
2
1
7
5
6
count
False
True
savings_products



Interpretation of plots

Entry Channels (KHE, KFC):

Certain entry channels are more associated with specific clusters, suggesting that the way customers are acquired or choose to interact with the company can be indicative of their cluster grouping.

- KFC: associated with cluster 0,7
- KHE: associated with cluster 2,6,0,3
- KHQ: associated with cluster 5

Regular and Loyal Customers:

There is a visible distinction in the distribution of regular and loyal customers across clusters. Some clusters have a higher proportion of loyal customers, while others are predominantly non-regular or non-loyal customers.

- Regular customers: associated with cluster 0,4,6
- Loyal customers: associated with cluster 3,2,7

Active Customers:

Active customer status also varies significantly across clusters, which could reflect the engagement level or recent activity of customers within each cluster.

- Active customers: associated with cluster 0,4

Product Holdings (Savings, Loan, Investment, Payment Products):

The distribution of product holdings such as savings products, loan products, investment products, and payment products suggests that some clusters are more product-focused than others.

- Savings products: associated with cluster 0
- Loan products: not associated with any cluster in particular
- Investment products: not associated with any cluster in particular
- Payment products: associated with cluster 0

Demographics (Age Group, Gender):

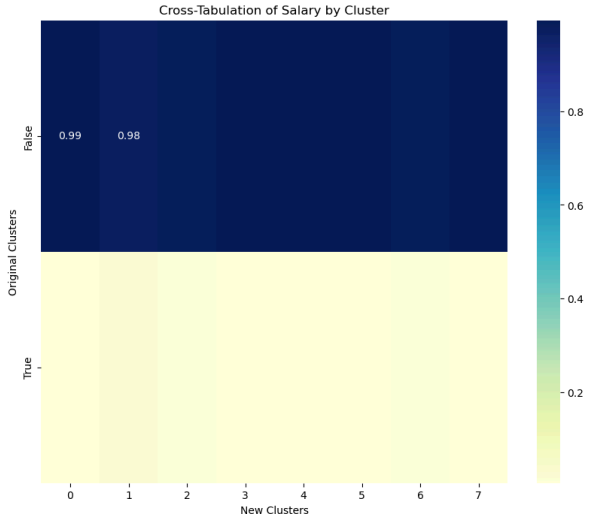
Demographic distributions such as age group (adult, young adult) and gender (female) also show variability across clusters, implying that certain age groups or genders might be more prevalent in specific clusters.

- Adults: associated with cluster 0,2
- Young Adults: associated with cluster 3,4,5,6
- Female: associated with cluster 0,6,3,2,5
- Male: associated with 0,4,2,6
- Salary: associated with 0,2,6

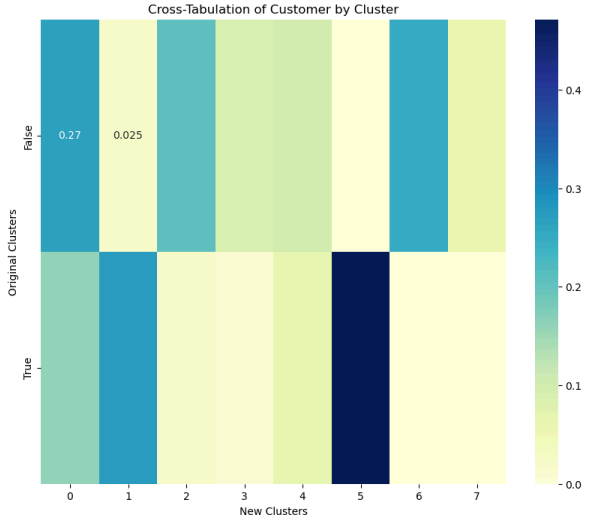
By analyzing these plots, we can conclude that the clusters have distinct characteristics and likely represent different customer segments with unique behaviors and preferences. The visualizations support a segmented approach to customer relationship management, marketing strategies, and product offerings.

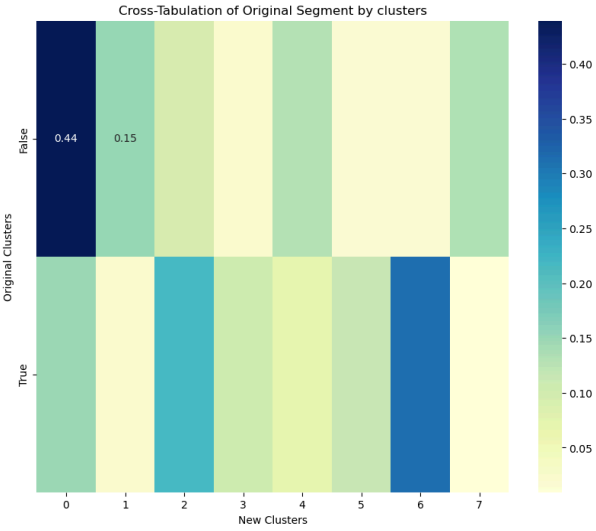
Cross-tabulation

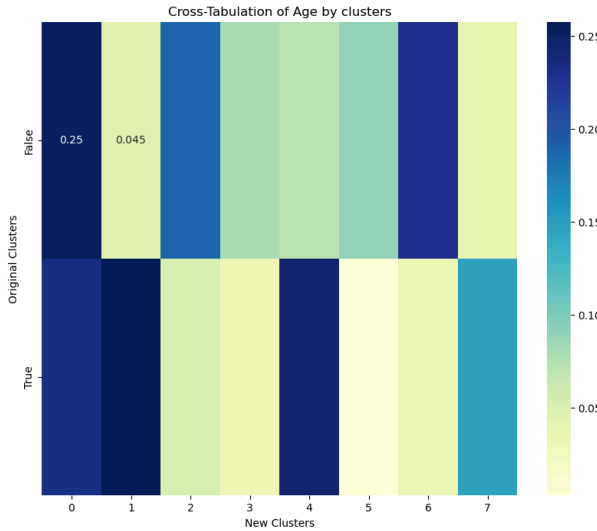
The comparison of clusters with additional demographics or behaviors not utilized in clustering involves selecting a column not used for clustering, such as age or region. By examining the distribution of these variables across different clusters, insights into the cluster's composition can be gained. If certain demographics are predominantly present in specific clusters, it provides valuable information about the cluster's characteristics. Normalizing these cross-tabulations to proportions rather than raw counts is often beneficial for a clearer understanding of the distribution patterns across clusters.

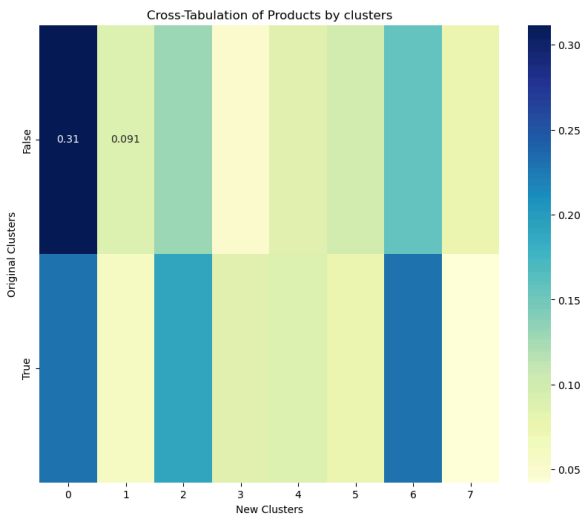

Cross-Tabulation of Salary by Cluster
Original Clusters
False
True
0.99
0.98
New Clusters
0
1
2
3
4
5
6
7
0.8
0.6
0.4
0.2


Cross-Tabulation of Salary by Cluster
Original Clusters
False
True
0.25
0.066
New Clusters
0
1
2
3
4
5
6
7
0.25
0.20
0.15
0.10
0.05


Cross-Tabulation of Customer by Cluster
Original Clusters
False
True
0.27
0.025
New Clusters
0
1
2
3
4
5
6
7
0.4
0.3
0.2
0.1
0.0


Cross-Tabulation of Original Segment by clusters
Original Clusters
False
True
0.44
0.15
New Clusters
0
1
2
3
4
5
6
7
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05


Cross-Tabulation of Age by clusters
Original Clusters
False
True
0.25
0.045
New Clusters
0
1
2
3
4
5
6
7
0.25
0.20
0.15
0.10
0.05


Cross-Tabulation of Products by clusters
Original Clusters
False
True
0.31
0.091
New Clusters
0
1
2
3
4
5
6
7
0.30
0.25
0.20
0.15
0.10
0.05

Interpretation and Decision

Based on the cross-tabulation results, we can have a brief summary highlighting key characteristics for each cluster:

High Salary Cross-Tabulation:

- Across the clusters, high salary earners are a minority, generally below 1%, with Cluster 7 slightly leading at approximately 0.9%. This reaffirms that the high salary group does not represent the majority in any cluster.

Low Salary Cross-Tabulation:

- The data reveals a significant proportion of low salary earners in each cluster, with Cluster 7 having the smallest proportion at around 2.7% and Cluster 0 the largest at about 26.6%. This highlights that the customer base is predominantly composed of individuals with lower salaries.

New Customer Cross-Tabulation:

- Clusters 5 and 3 have the highest proportions of new customers at approximately 47% and 9%, respectively, suggesting recent growth in these segments. In contrast, Clusters 6 and 7 have the lowest, with Cluster 6 having no new customers and Cluster 7 at about 0.15%.

Segment Cross-Tabulation:

- University students have a notably higher representation in Cluster 6 at around 31%, with Cluster 7 having the lowest at about 0.9%. This indicates targeted customer segments, with Cluster 6 potentially being an attractive market for student-related products and services.

Age Group Cross-Tabulation:

- Clusters 1 and 4 have the highest percentages of middle-aged customers, at approximately 25.8% and 24.4% respectively, signifying that these clusters may be more established in their careers and possibly more financially secure.

Products Cross-Tabulation:

- Clusters 6 and 2 have the highest percentages of 'em_account' holders, at approximately 22.9% and 19.1% respectively, indicating a strong engagement with the bank's products. Clusters 5 and 7 have the lowest, with Cluster 5 at around 7.4% and Cluster 7 at 4.2%, suggesting room for increased product uptake and engagement initiatives.

Final Interpretation

Combining all the information provided about customer clusters, we can synthesize a detailed profile for each cluster as follows:

Cluster 0: Emerging Engagers

- Entry Channels: A mix of KFC and KHE suggests varied acquisition strategies.
- Customer Type: A blend of regular customers, active engagement, and a mix of new and established relationships.
- Financial Profile: Dominated by low-salary earners but also associated with savings and payment product holdings.
- Demographics: Broad age group including adults and possibly young adults, but not specifically university students.

Cluster is composed of diversity and inclusivity of various age groups and financial backgrounds. Suggesting their wide-ranging product interests.

Cluster 1: Established Mid-Agers savers

- Entry Channels: Moderate use of KHE indicates traditional engagement.
- Customer Type: Few new customers, indicating longer-term relationships.
- Financial Profile: Low salary earners similar to Cluster 0.
- Demographics: A slightly higher representation of middle-aged customers and a lower proportion of 'em_account' holders, implying stable financial habits.

These customers are likely a key demographic for the bank, with a consistent financial approach. Emphasizes the stability and traditionalism of this cluster.

Cluster 2: Financially Engaged

- Entry Channels: Considerable use of KHE shows established engagement channels.
- Customer Type: Few new customers suggest loyalty and established banking habits.
- Financial Profile: Not the highest but a notable percentage of high salary earners, implying financial stability.
- Demographics: A notable percentage of middle-aged customers, likely financially established.

This cluster's mature and financially stable customer base. Suggests they have a long-standing relationship with the bank and are experienced in financial matters.

Cluster 3: Emerging Customers

- Entry Channels: Mostly associated with KHQ, indicating recent customer acquisition strategies.
- Customer Type: Significant new customers, implying a segment ripe for new relationships and loyalty building.
- Financial Profile: Balance of new engagements with potential for financial growth.
- Demographics: Less focus on university students, suggesting a recent shift away from this demographic.

This cluster comprises newer customers to the bank.

Cluster 4: Balanced Customers

- Entry Channels: Use of KHE similar to Clusters 2 and 3.
- Customer Type: A balance of new and existing customers.
- Financial Profile: Moderate mix of salary levels and 'em_account' engagement.
- Demographics: Likely a diverse age group, possibly reflecting the balanced customer type.

This cluster reflects a balance in customer demographics and financial engagement. There is potential for growth and deeper engagement.

Cluster 5: Growth Seekers

- Entry Channels: Association with KHQ might reflect a digital or innovative acquisition approach.
- Customer Type: Possibly newer segments given the presence of university students.
- Financial Profile: Similar salary distribution to other clusters but with the lowest 'em_account' engagement, suggesting potential for growth.
- Demographics: Presence of university students and young adults indicates this cluster's potential for future development.

This cluster highlights the younger demographic and potential for growth. These customers may be open to trying new financial products and services.

Cluster 6: Aspiring Diversifiers

- Entry Channels: Diverse, not specified which could indicate a variety of acquisition strategies.
- Customer Type: Regular customers with a diversity of engagement levels.
- Financial Profile: Higher representation of low salary earners and a need for financial product education and engagement.
- Demographics: Broader age range with a lean towards middle-aged individuals, not specifically targeting students.

This cluster indicates a customer group aiming to broaden their financial portfolio. Potential desire to improve their financial standing.

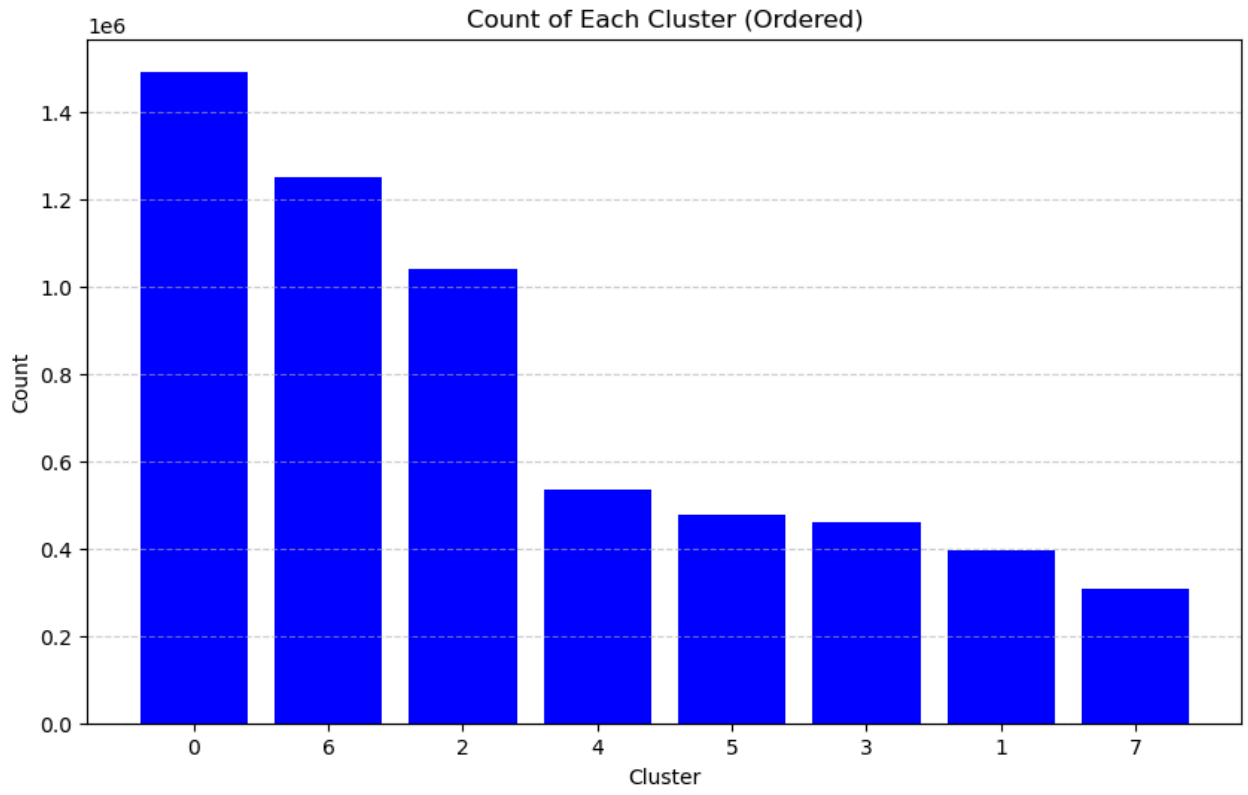
Cluster 7: Premium Innovators

- Entry Channels: Notable use of KFC with some KHE users, indicating a mix of traditional and innovative acquisition approaches.
- Customer Type: Loyal customers indicating strong retention and value in the cluster.
- Financial Profile: The highest percentage of high salary earners, implying a segment with significant financial clout and engagement.
- Demographics: Significant number of university students and middle-aged customers, suggesting a range of product needs and financial sophistication.

This cluster suggests a segment with significant financial resources and a propensity towards adopting new banking technologies or products.

These cluster profiles offer a comprehensive understanding of the customer base, with each cluster showing distinct patterns in terms of demographics, financial status, product preferences, and engagement levels. EasyMoney can use this detailed segmentation to tailor its marketing strategies, product offers, and customer service approaches to meet the unique needs of each customer group.

Monitoring customer behaviour: customers might change their behaviour, product preferences, or interact differently with the services offered, leading them to fit into a different cluster as time progresses. If the clustering model is retrained with new data, the boundaries of clusters may shift, causing customers who were once in one cluster to be reassigned to another. Also, customers go through various life stages, and their financial needs, spending habits, and risk profiles evolve, so it's important
It is essential to monitor these changes as they can provide valuable insights into how customer preferences evolve and help you adapt your strategies.



- Cluster 0 has the highest count, which suggests that the characteristics defining the cluster are the most common among the data points.
- Clusters 6 and 2 follow, with slightly fewer instances than cluster 0, but still among the larger clusters. This indicates that the defining features of these clusters also represent a significant portion of the data points.
- Cluster 4 is in the mid-range in terms of count, indicating a moderate presence of its characteristics within the data.
- Clusters 5, 3, and 1 have increasingly fewer instances, which could imply that they have more specific or less common defining characteristics.
- Cluster 7 has the fewest instances, suggesting that it represents the most unique or specific group among the data points, with characteristics that are less frequently observed than those of the other clusters.

This chart helps us in understanding the prevalence of different segment characteristics within the dataset and could be useful for identifying the most common or unique behaviors, preferences, or other attributes represented in the clusters. It's also useful for resource allocation in marketing campaigns, product development, and customer service enhancements, as it indicates which clusters represent the largest and smallest segments of the population.

Recommendation

In this section, we unveil the intricacies of our recommendation model, designed to be the lynchpin in rejuvenating EasyMoney's market presence and rekindling customer trust. Drawing inspiration from the rich insights gleaned from our meticulous clustering and segmentation efforts, our recommendation model stands poised to infuse every customer interaction with tailored precision, steering them towards products that resonate both with their preferences and the company's bottom line.

Problem: Predict the response rate of the campaign (expected ROI) for 10,000 target customers by considering €10 for each account sold, €40 for savings and investment products (plans, funds, etc.) and €60 for financing products (loans and cards). The recommendation is based on products that interest customers most and also items that make the most ROI.

Approach: Firstly, an in-depth analysis of the dataset is imperative to discern the optimal features for inclusion. By unraveling the nuances of the data and identifying prevailing trends, we pave the way for crafting the most effective recommendation solution. Given our focus on maximizing profitability, our gaze settles upon a select group of 10,000 customers deemed to hold the greatest potential. This elite cadre serves as our campaign's primary targets, meticulously ranked to ensure precision targeting.

Prior to delving into model creation, the dataset undergoes a rigorous purification process, shedding extraneous variables and ensuring its readiness for analysis. With the data cleansed and primed, preprocessing procedures are meticulously executed, laying the groundwork for subsequent modeling endeavors. Thus prepared, we embark upon the intricate journey of modeling, poised to unleash the full potential of our recommendation framework.
In tandem with our endeavor to predict the response rate of our esteemed 10,000 campaign targets, we employ two distinct methodologies, each infused with technical rigor and creative ingenuity.

As the recommendation system is predicated on catering to the preferences of current users while also maximizing ROI for EasyMoney, we employ two distinct recommendation methods: user-item collaborative memory-based and model-based approaches.

1. Model-based Collaborative Filtering

Within the domain of collaborative filtering, our methodology delves deep into the fabric of user interactions, encapsulated within the dataset as binary proxies denoting product ownership (1 for possession, 0 for absence). Diverging from rudimentary heuristics, our approach adopts a model-based paradigm, harnessing the computational prowess of machine learning to discern intricate patterns and nuanced preferences.

At the core of our framework lies the principle of matrix factorization, epitomized through the venerable Singular Value Decomposition (SVD) technique. Empowered by the versatile Surprise library, we harness the formidable capabilities of SVD and evaluate its sophisticated variant, SVD++, to distill the fundamental essence of consumer-product dynamics. Through hyperparameter tuning, we embark on a quest to optimize these models, chiseling away imperfections to reveal their latent potential in exquisite detail.

Armed with our finely-honed models, our gaze shifts towards the ultimate crucible: prognosticating the response rate of our esteemed cohort of 10,000 campaign targets. Equipped with the discerning insights derived from SVD and SVD++, we traverse the intricate labyrinth of customer preferences with unyielding precision. Poised on the precipice of strategy, we stand ready to unveil a campaign narrative that resonates with our audience on a profoundly personal level, igniting sparks of engagement and forging lasting connections in the digital ether.

2. Memory-based collaborative filtering:Collaborative Filtering is based on the analysis of user ratings. In the dataset, the rating is information about the product ownership (1 or 0). In memory based technique recommendations are based on similarity between users. The similarity between users is calculated by the similarity measure function. It uses the cosine distance to create the user-item similarity matrix (based on cosine similarity).

Each model undergoes evaluation and hyperparameter tuning, and the optimal model is chosen based on metrics such as MAE (Mean Absolute Error) and RMSE (Root Mean Squared Error). Subsequently, ROI is calculated using predictions from the model. Additionally, a pessimistic scenario is considered, where all predictions are adjusted by the mean absolute error. This scenario assumes that the model tends to over predict the probability of selecting the product compared to the actual probability. Let's delve into each step in detail:

4.2 Data Preprocessing

Our journey began with a meticulous curation of data, ensuring it gleamed with integrity and fidelity. Rigorous cleansing rituals were performed to purge any imperfections, followed by a symphony of preprocessing techniques orchestrating the transformation of raw data into a harmonious ensemble, ready to feed the hungry maw of our recommendation engine.

Explore the data to understand its characteristics, distributions, and relationships. Identify patterns, trends, and potential insights that could inform the recommendation and prediction models.

A. Creating a user-item interaction matrix representing count

B. Creating a user-item interaction matrix representing ratio

C. Stacking user-item interaction into single column

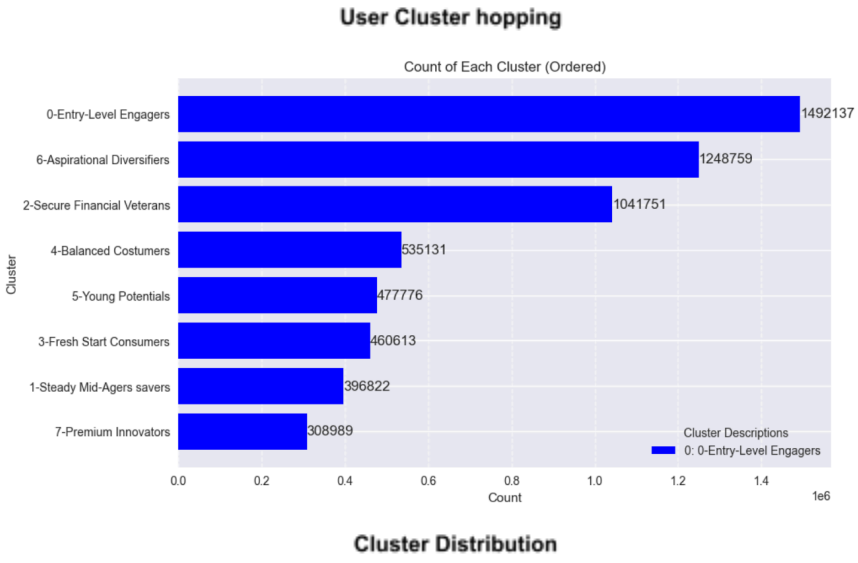
D. Represent Data Correctly

E. Create product popularity matrix with time decay coefficient.

First, let's delve into the clusters and understand how our population is distributed. The data reveals that customers transition between clusters over time, indicating that some customers belong to multiple clusters simultaneously. It is shown below that user 5961556 changed cluster from 2-Financially Engaged to 0-Emerging Engagers. We can see that customers change segments during the time based on their behaviour, maturity and subscription. I am going to keep the last one as the current customers cluster

```
[1005567 rows x 3 columns]
          pk_cid pk_partition  cluster
118      1000306   2018-01-28        4
129      1000306   2018-12-28        0
141      1000381   2018-07-28        3
146      1000381   2018-12-28        2
152      1000383   2018-07-28        3
...          ...          ...      ...
5961495    99283   2018-09-28        3
5961556   994005   2018-08-28        2
5961557   994005   2018-09-28        0
5961921   999706   2018-12-28        1
5961922   999706   2019-01-28        5

[165646 rows x 3 columns]
```



We can observe that clusters 0, 6, and 2 represent the majority of customers, while clusters 7, 1, and 3 represent a smaller portion of the population. The difference in customer distribution between the top three clusters and the bottom three clusters is substantial.

Additionally, there is a significant difference between the top three clusters and the fourth cluster compared to the bottom three clusters.

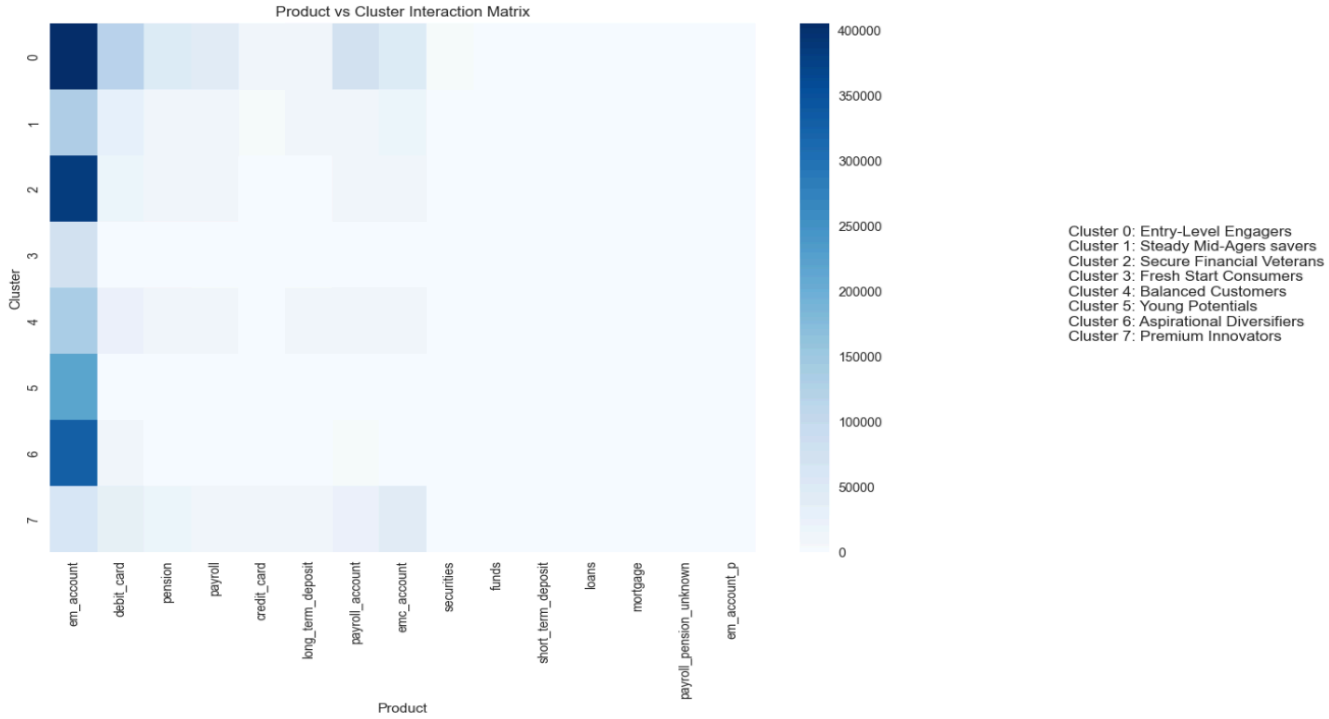
Based on this observation, we can infer that there are distinct segments within our customer base, with clusters 0, 6, and 2 likely representing the most common customer profiles or behaviors. Understanding the characteristics and needs of these dominant clusters can help us tailor our marketing strategies and product offerings to better serve the majority of our customers. Additionally, identifying the smaller clusters (7, 1, and 3) and understanding their unique attributes can help us develop targeted approaches to engage and retain these customer segments, potentially increasing overall customer satisfaction and loyalty.

Let's calculate the product popularity matrix, considering both the popularity for the customer and the ROI (popularity for the business), and their most representative clusters. We will incorporate a decay weight in counting the product subscriptions, where the eldest subscriptions receive a weight of 0.02 and the most recent ones receive a weight of 1. This approach will help us capture the evolving trends in product popularity over time within each cluster. The first metrics is the popularity based on user interaction with the product considering the time decay weight factor.

	Product	Popularity
0	loans	29.434641
1	Total_Short_Term_Deposit	399.833333
2	mortgages	20.565359
3	funds	1287.526144
4	securities	1577.133987
5	credit_card	4521.718954
6	debit_card	38267.513072
7	long_term_deposit	6332.424837
8	payroll	13997.866013
9	pension	14828.362745
10	payroll_pension_unknown	0.539216
11	payroll_account	22785.290850
12	em_account	283907.006536
13	em_account_p	2.000000
14	emc_account	21474.349673

Product Popularity Metrics

We proceeded by conducting an analysis of cluster product interaction matrices.



Cluster Product Interaction Heatmap

The analysis reveals that "em_account" is the most widely used product across all clusters, indicating its importance as a prerequisite for other products. This highlights the necessity of recommending this account as a first step for subscribers who do not have it yet. Conversely, "securities," "funds," "short-term-deposit," "loans," "mortgages," and "em_account_p" are among the least popular products across all clusters. Their low popularity could be attributed to factors such as lack of customer awareness, strong competition from other products, and ineffective marketing strategies. To maximize revenue, EasyMoney should investigate the root causes of their low popularity and consider targeted promotional campaigns or product improvements.

In Cluster 0, "debit card," "payroll_account," and "emc_account" emerge as the most popular products, mirroring trends seen in other clusters. The "Premium Innovators" cluster stands out for its diverse product portfolio, with "debit_card" being particularly popular. However, there is a notable lack of diversity in product usage among "Fresh Start Consumers" (Cluster 3), "Aspirational Diversifiers" (Cluster 6), and "Young Potentials" (Cluster 5), where "em_account" is the sole product in use. Understanding the

needs and behaviors of these clusters is crucial for tailoring products and devising effective marketing strategies.

Overall, EasyMoney should focus on promoting less popular products, considering the significant revenue potential they offer. Additionally, understanding customer needs and preferences in each cluster can guide the development of targeted marketing campaigns and product enhancements, ultimately leading to increased customer engagement and revenue growth. Currently, the primary focus is on maximizing engagement from existing customers. EasyMoney should concentrate its efforts on those customers who are most likely to adopt the recommended services with minimal reliance on content marketing, prioritizing personalized recommendations instead.

Due to resource constraints and memory limitations caused by the sheer volume of data, a careful examination reveals that clusters 3, 6, and 5 lack significant diversity in their datasets for effective recommendation targeting. These clusters should instead be earmarked for a separate campaign aimed at re-engaging inactive customers. With EasyMoney facing pressure from investors and partners to achieve high revenue and response rates swiftly, a strategic approach is imperative.

In light of this, our strategy is to target clusters with a higher likelihood of purchasing recommended products. Given the constraint of 10,000 emails, it is essential to optimize their use by targeting subscribers most inclined to engage in cross-subscriptions. To this end, we propose the removal of the aforementioned clusters from our targeting strategy, as they have not been effectively targeted previously, and lack sufficient historical data on product interactions for accurate prediction.

Furthermore, subscribers categorized as deceased will also be removed from consideration, as their inclusion may introduce noise and potentially mislead interpretations. Despite representing a small percentage of customers, their removal will refine our targeting strategy and enhance the accuracy of our recommendations.

Among the 15 products, data for 7 products are missing, with subscribers rarely having a history of interactions with these products. Among the remaining 8 products, we establish a criterion that subscribers must have activated at least 2 products to be considered potential targets. This criterion ensures that targeted subscribers are more likely to engage further and exhibit a higher response rate to the campaign.

Now, let's proceed with ranking the customer base. We will designate the first 10,000 customers as the test dataset, representing the real target customers. The remaining customers will form the train dataset. It's important to consider that our revenue is approximately €10 for each account sold, €40 for savings and investment products (plans, funds, etc.), and €60 for financing products (loans and cards).. While the manager in EasyMoney initially prioritized a user-item interaction matrix as the bedrock for the recommender, It is also advised to incorporate product profitability. This strategic amalgamation ensures that the selected products not only resonate with users but also contribute to the company's profitability. This integrated approach will effectively guide the selection of the 10,000 customers targeted for the marketing campaign.

To rank the customers, we will take into account their potential Return on Investment (ROI), current interactions and activities, as well as their engagement matrix. This comprehensive approach ensures that our ranking reflects both the revenue potential and the level of engagement of each subscriber.

Then, we introduced Return on Investment (ROI) analysis for the campaign to identify the optimal product recommendations that align with both user preferences and EasyMoney's profitability objectives.

	Product	Popularity	Normalized_Popularity	Weighted_ROI	Normalized_Weighted_ROI	Rank
0	em_account	283907.006536	1.000000	2.839070e+06	1.000000	1.0
1	debit_card	38267.513072	0.134787	2.296051e+06	0.808733	2.0
2	pension	14828.362745	0.052228	5.931345e+05	0.208919	3.0
3	payroll	13997.866013	0.049303	5.599146e+05	0.197218	4.0
4	credit_card	4521.718954	0.015925	2.713031e+05	0.095561	5.0
5	long_term_deposit	6332.424837	0.022303	2.532970e+05	0.089218	6.0
6	payroll_account	22785.290850	0.080254	2.278529e+05	0.080256	7.0
7	emc_account	21474.349673	0.075637	2.147435e+05	0.075639	8.0
8	securities	1577.133987	0.005553	6.308536e+04	0.022220	9.0
9	funds	1287.526144	0.004533	5.150105e+04	0.018140	10.0
10	Total_Short_Term_Deposit	399.833333	0.001406	1.599333e+04	0.005633	11.0
11	loans	29.434641	0.000102	1.766078e+03	0.000622	12.0
12	mortgages	20.565359	0.000071	8.226144e+02	0.000290	13.0
13	em_account_p	2.000000	0.000005	2.000000e+01	0.000007	14.0
14	payroll_pension_unknown	0.539216	0.000000	0.000000e+00	0.000000	15.0

Weighted Product_ROI Popularity and Profitability Product Ranking

Initially, we quantify the engagement level of each user by considering the number of interactions with all products, factoring in their respective time decay. Subsequently, customers are ranked based on this metric, aptly labeled as the "engagement rank" feature.

funds	short_term_deposit	loans	mortgage	payroll_pension_unknown	em_account_p	cluster	engagement_Rank
6.12	0.0	0.00	0.00	0.0	0.0	7	1.0
6.10	0.0	0.00	0.00	0.0	0.0	0	2.0
6.12	0.0	0.00	0.00	0.0	0.0	1	3.0
6.12	0.0	0.00	0.00	0.0	0.0	1	4.0
6.12	0.0	0.00	0.00	0.0	0.0	2	5.0
6.12	0.0	0.00	0.00	0.0	0.0	7	5.0
6.12	0.0	0.00	0.00	0.0	0.0	7	5.0
0.00	0.0	0.00	0.00	0.0	0.0	4	8.0
6.10	0.0	0.00	0.00	0.0	0.0	1	9.0

Customer Engagement Rank

Subsequently, each customer is ranked again based on their potential Return on Investment (ROI) based on the following prices:

```
# Apply ROI values to consider both company and user benefit
roi = {
  'loans': 60,
  'credit_card': 60,
  'debit_card': 60,

  'mortgage': 40,
  'funds': 40,
  'securities': 40,
  'payroll': 40,
  'pension': 40,
  'payroll_pension_unknown': 0,
```

```
  'short_term_deposit': 40,
  'long_term_deposit': 40,

  'payroll_account': 10,
  'em_account': 10,
  'em_account_p': 10,
  'emc_account': 10
}
```

s	short_term_deposit	loans	mortgage	payroll_pension_unknown	em_account_p	cluster	engagement_Rank	potential_roi
2	0.0	0.00	0.00	0.0	0.0	7	1.0	64.571429
0	0.0	0.00	0.00	0.0	0.0	0	2.0	76.285714
2	0.0	0.00	0.00	0.0	0.0	1	3.0	64.571429
2	0.0	0.00	0.00	0.0	0.0	1	4.0	64.571429
2	0.0	0.00	0.00	0.0	0.0	2	5.0	93.142857
2	0.0	0.00	0.00	0.0	0.0	7	5.0	104.857143
2	0.0	0.00	0.00	0.0	0.0	7	5.0	86.285714
0	0.0	0.00	0.00	0.0	0.0	4	8.0	95.714286
0	0.0	0.00	0.00	0.0	0.0	1	9.0	76.285714
2	0.0	0.00	0.00	0.0	0.0	0	10.0	93.142857
0	0.0	0.00	0.00	0.0	0.0	1	11.0	95.714286
0	0.0	0.00	0.00	0.0	0.0	0	12.0	83.142857
0	0.0	0.00	0.00	0.0	0.0	7	13.0	102.571429
0	0.0	0.00	0.00	0.0	0.0	7	14.0	102.571429
0	0.0	0.00	0.00	0.0	0.0	7	15.0	95.714286

Customer potential ROI

The potential_ROI is multiplied by engagement_rate which is assigned based on the engagement ranks:0.3 as the least and 1 the most engaged.

deposit	loans	mortgage	payroll_pension_unknown	em_account_p	cluster	engagement_Rank	potential_roi	engagement_rate
0.0	0.0	0.0	0.0	0.0	7	40653.0	273.714286	0.3
0.0	0.0	0.0	0.0	0.0	7	40653.0	273.714286	0.3
0.0	0.0	0.0	0.0	0.0	0	40653.0	273.714286	0.3
0.0	0.0	0.0	0.0	0.0	2	40653.0	259.428571	0.3
0.0	0.0	0.0	0.0	0.0	4	40653.0	273.714286	0.3
0.0	0.0	0.0	0.0	0.0	7	40653.0	245.714286	0.3
0.0	0.0	0.0	0.0	0.0	1	40653.0	219.714286	0.3
0.0	0.0	0.0	0.0	0.0	7	40653.0	223.142857	0.3
0.0	0.0	0.0	0.0	0.0	7	40653.0	267.142857	0.3
0.0	0.0	0.0	0.0	0.0	4	40653.0	273.714286	0.3

Customer Engagement Rate

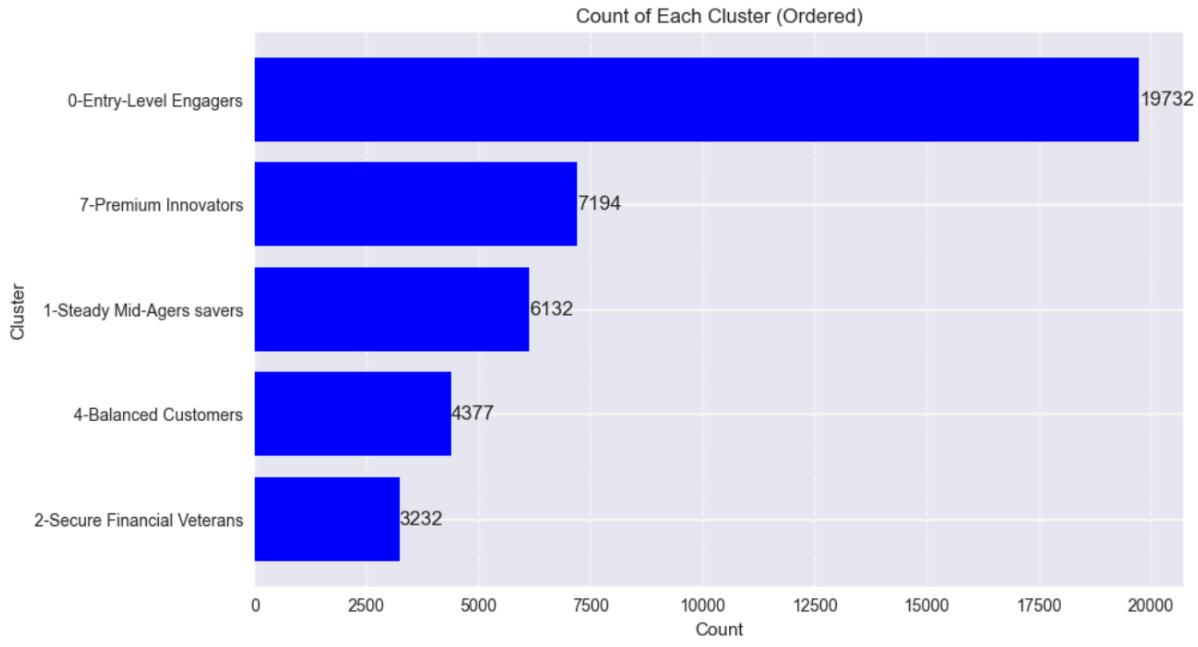
The potential roi is multiplied by engagement rate and the customers ranked based on this metric:

	engagement_ROI	best_customer_to_target_Rank
96479	240.588424	1.0
135263	240.588424	1.0
96630	240.588424	1.0
59970	240.588424	1.0
184425	240.588424	1.0
204631	237.876912	6.0
5532	237.314824	7.0
165429	237.130459	8.0
52380	236.905624	9.0
19288	234.271004	10.0
7120	234.271004	10.0

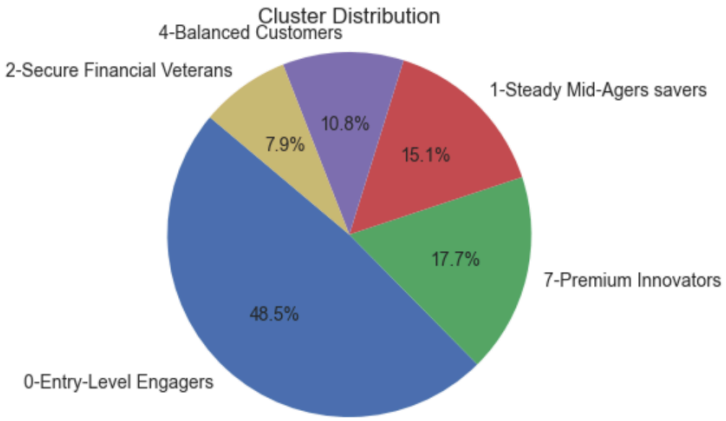
Best Customers to target

Subsequently, the top 50,000 customers are selected: the first 10,000 are designated as the test set, the next 5,000 as the validation set, and the remaining 35,000 as the train set. Due to some customers being ranked equally.

Now, let´s do the final check of cluster distribution and make sure the data is prepared for modeling:



Cluster distribution of customers after reduction



Cluster Distribution Pie Chart

Our analysis reveals that the largest portion of our potential users falls into cluster 0, classified as Entry-level Engagers, closely followed by cluster 7, denoted as Premium Innovators, and cluster 1, termed Steady Mid-Agers (constituting 13.2% of the total). Additionally, there is significant interest observed in cluster 4, recognized as Balanced Customers (10.6%), and cluster 2, identified as Secure Veterans (6.8%).

As a final step, we must ensure the continued status of all our customers as EasyMoney's clientele by confirming their presence in the most recent data batch. There are 33 customers that are unsubscribed from the service (doesn't exist in the last partition) but as their rank is more than 10,000 (the first 10,000 would be our test set and validation set) and the number is ignorable , we keep them.
The dataset will be partitioned into training, testing, and validation sets. Subsequently, the training set will be transformed into a format compatible with the Surprise library, and the model will be trained on this prepared dataset.

```
Train data shape: (25667, 23)
Validation data shape: (5000, 23)
Test data shape: (10000, 23)
```

All three datasets will undergo transformation to conform to the following format before being inputted into our Surprise models.

	pk_cid	service_code	service_selection_ratio	cluster
0	1441539.0	0	0.045918	0
1	1441539.0	1	0.265306	0
2	1441539.0	2	0.229592	0
3	1441539.0	3	0.229592	0
4	1441539.0	6	0.229592	0
...	...	...	...	...
111582	1133822.0	3	0.012987	2
111583	1133822.0	4	0.038961	2
111584	1133822.0	5	0.272727	2
111585	1133822.0	6	0.285714	2
111586	1133822.0	7	0.051948	2

111587 rows × 4 columns

Surprise Friendly datasets with three columns for User_id, service_id and Service Selection Ratio with additional column as cluster

The train, test, and validation datasets should each contain three columns: user_id, service_id, and service_selection ratio. Since we have an additional cluster column, which the model cannot process, the cluster information is concatenated to the service id. Below is the final table format:

	user_id	service_code	service_selection_ratio
0	1441539.0_0	0	0.045918
1	1441539.0_0	1	0.265306
2	1441539.0_0	2	0.229592
3	1441539.0_0	3	0.229592
4	1441539.0_0	6	0.229592

Surprise Friendly datasets with three columns for User_id, service_id and Service Selection Ratio with user_id concat with cluster_id

4.3 Model Selection and Development

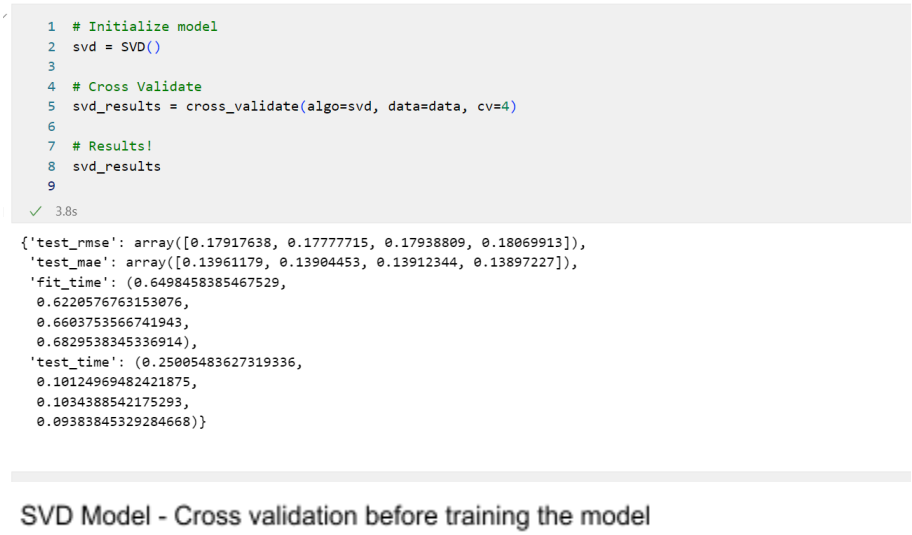
Our data is ready for Surprise modeling. Embracing the duality of simplicity and sophistication, we elected to wield two formidable algorithms: the elegant Singular Value Decomposition (SVD) and the venerable K-Nearest Neighbors (KNN).

The Surprise library is a powerful tool for building and evaluating recommender systems in Python. It provides a convenient interface for implementing collaborative filtering algorithms and evaluating their performance using various metrics.

4.3.1 Singular Value Decomposition (SVD) Model

The SVD model, a stalwart of collaborative filtering, unfurls its latent prowess by unraveling the intricate tapestry of user-item interactions. Through meticulous hyperparameter tuning, we harnessed its potential to distill user preferences into a fine elixir of personalized recommendations.

SVD Initial Cross Validation Model - Performing cross-validation before training the model allows for an initial assessment of the model's performance without fitting it to the entire dataset. This practice helps to identify potential issues or limitations of the chosen algorithm and hyperparameters early in the development process. By splitting the data into multiple folds and evaluating the model's performance on each fold separately, cross-validation provides more robust estimates of performance metrics, such as accuracy and generalization ability. This approach aids in selecting the most suitable algorithm and tuning its parameters before training on the full dataset, ultimately improving the model's predictive performance.



```
# Initialize model
svd = SVD()

# Cross Validate
svd_results = cross_validate(algo=svd, data=data, cv=4)

# Results!
svd_results

✓ 3.8s
```

```
{'test_rmse': array([0.17917638, 0.17777715, 0.17938809, 0.18069913]),
 'test_mae': array([0.13961179, 0.13904453, 0.13912344, 0.13897227]),
 'fit_time': (0.6498458385467529,
  0.6220576763153076,
  0.6603753566741943,
  0.6829538345336914),
 'test_time': (0.25005483627319336,
  0.10124969482421875,
  0.1034388542175293,
  0.09383845329284668)}
```

SVD Model - Cross validation before training the model

It gives a fair evaluation with average of 0.1788 RMSE and 0.1394 MAE. Let's train the model:

```python
# Define the data
# Create an empty DataFrame
# Define the dictionary
Model_Error = {
    'Model': 'SVD_Basic',
    'RMSE': rmse_mean,
    'MAE': mae_mean
}

# Convert the dictionary to a DataFrame
model_error_df = pd.DataFrame([Model_Error])

# Print the DataFrame
model_error_df
```

✓ 0.0s

	Model	RMSE	MAE
0	SVD_Basic	0.178879	0.139475

SVD Model - Cross validation before training the model Results in average

Based on the cross-validation results, the model exhibits fair root mean squared error (RMSE) values, ranging from approximately 0.178 to 0.180, indicating that it performs fair in predicting the response rate.The fit times, which represent the time taken to train the model, are reasonable, with values ranging from approximately 0.582 to 0.649 seconds. However, it's essential to consider the test times as well, which represent the time taken to evaluate the model on the test data, ranging from approximately 0.083 to 0.331 seconds. Overall, the model appears to provide accurate predictions with acceptable computational efficiency.

SVD Trained Model tested on validation set - Let's train the model on train set and test it on validation set. Here is the result:

```
print("RMSE:", rmse)
print("MAE:", mae)
# Define the dictionary for the second row
Model_Error_2 = {
    'Model': 'SVD_trained_val',
    'RMSE': rmse,
    'MAE': mae
}
# Add the second row to the DataFrame using .loc
model_error_df.loc[1] = Model_Error_2
print(model_error_df)
```

[54] ✓ 2.4s

```
RMSE: 0.20202835180741682
MAE: 0.182357803471097
             Model      RMSE       MAE
0        SVD_Basic  0.178879  0.139475
1  SVD_trained_val  0.202028  0.182358
```

SVD-Trained model tested on validation set

These results come from evaluating the model on a separate validation set not used during model training or cross-validation.

From the provided validation results:

Validation RMSE: 0.206

Validation MAE: 0.185

Comparing the cross-validation results with the validation results:

The average CV RMSE (approximately 0.179) is lower than the validation RMSE (0.1981), indicating that, on average, the model performs slightly better on the cross-validation folds compared to the validation set.

Similarly, the average CV MAE (approximately 0.139) is lower than the validation MAE (0.185), suggesting that, on average, the model's predictions are slightly closer to the actual ratings in the cross-validation folds compared to the validation set.

Overall, the model appears to generalize reasonably well based on the cross-validation results, with slightly better performance observed on the cross-validation folds compared to the separate validation set. However, the differences in performance metrics between cross-validation and validation suggest some level of variability or differences in the data distributions between the two sets. Let's hypertune and then get predictions on our test set (target set)

SVD Trained Model - hyperparameter tuning

By implementing GridSearch and defining the parameters that we think are effective on the performance of the model the performance of the model can improve:

```
from surprise.model_selection import GridSearchCV
# Hyperparameter tuning
# Define parameter grid for grid search

param_grid = {'n_epochs': [5, 6,7,10,12,15, 20, 30],
              'lr_all': [.0025, .005, .004,.003,.002, .001, .01]}

# Initialize the GridSearchCV object
gs = GridSearchCV(SVD, param_grid, measures=['MAE'], cv=5)
gs.fit(data)

print(gs.best_score['mae'])
print(gs.best_params['mae'])


✓ 2m 38.0s
```

```
0.13316655042835585
{'n_epochs': 15, 'lr_all': 0.001}
```

SVD Hyperparameter Tuning Using GridSearch

It is evident that by setting the number of epochs to 15 and the learning rate to 0.001, the model achieves the highest performance compared to other parameters specified in the param_grid.

```
34

✓ 3.3s
```

```
RMSE: 0.2068587802635085
MAE: 0.1893344672122607
```

	Model	RMSE	MAE
0	SVD_Basic	0.178879	0.139475
1	SVD_trained_val	0.202028	0.182358
2	SVD_hybertuned_trained_val	0.206859	0.189334

SVD model - Hyperparmeter Tuned

The performance deteriorated with hyperparameter tuning. Therefore, for the time being, the basic cross-validation model emerges as the preferred option.

The deterioration in performance observed after hyperparameter tuning can be attributed to several factors:

Overfitting: Hyperparameter tuning aims to optimize the model's performance on the training data. However, excessive tuning may lead to overfitting, where the model captures noise in the training data rather than generalizing well to unseen data. As a result, the model's performance on the validation or test data may suffer.

Complexity: Increasing the number of hyperparameters to tune increases the complexity of the model. This complexity may lead to difficulties in finding the optimal set of hyperparameters, especially when the dataset is relatively small or noisy.

Limited Data: In some cases, the dataset may be too small or lack diversity to effectively train a complex model with tuned hyperparameters. As a result, the model may struggle to generalize well to unseen data, leading to a decrease in performance.

Randomness: The performance of machine learning algorithms can be sensitive to random variations in the dataset or the optimization process. Hyperparameter tuning involves random search or optimization techniques, which may result in variations in performance across different runs.

Based on these factors, it's crucial to carefully consider whether hyperparameter tuning is appropriate for the specific dataset and problem at hand. In cases where the dataset is limited or the default model performs reasonably well, sticking with the basic cross-validation model may be the most prudent choice to avoid unnecessary complexity and potential overfitting.

SVD ++ - (SVD Plus Plus) Basic Cross Validation

Now let's try implementing SVD which is called SVD++ and its implementation is very easy : replacing SVD() with SVDpp().
SVD++ (SVD Plus Plus) is an extension of the Singular Value Decomposition (SVD) algorithm, which is commonly used in collaborative filtering-based recommendation systems.

Traditional SVD decomposes the user-item interaction matrix into lower-dimensional matrices to capture latent factors representing user preferences and item characteristics. It predicts user-item ratings by computing the dot product of these latent factor matrices.

SVD++ enhances the traditional SVD approach by incorporating implicit feedback in addition to explicit user-item ratings. Implicit feedback refers to indications of user preferences that are inferred from user interactions such as clicks, views, or purchases.

In SVD++, implicit feedback is incorporated by considering the "implicit" interactions of users with items when computing the latent factor matrices. This additional information helps improve the accuracy of recommendations, especially in scenarios where explicit ratings are sparse or unavailable.

Overall, SVD++ is a powerful algorithm for collaborative filtering that leverages both explicit and implicit feedback to provide more accurate and personalized recommendations especially in our case that the rating is the interaction of the customer with services.

```
from surprise import SVDpp
svdpp = SVDpp()
cross_validate(svdpp, data, measures=['RMSE','MAE'], cv=5, verbose=False)
```

58] ✓ 7.5s

```
{'test_rmse': array([0.17972081, 0.18073056, 0.17817956, 0.17880478, 0.18093695]),
 'test_mae': array([0.14014933, 0.14033642, 0.1390862 , 0.13901745, 0.13980165]),
 'fit_time': (0.9344725608825684,
  0.9760725498199463,
  0.9493050575256348,
  1.1836159229278564,
  0.9605457782745361),
 'test_time': (0.4566655158996582,
  0.26642441749572754,
  0.484591007232666,
  0.2529158592224121,
  0.25000834465026855)}
```

SVD++ Model - Basic Cross Validation Performance

9] ✓ 0.0s

	Model	RMSE	MAE
0	SVD_Basic	0.178879	0.139475
1	SVD_trained_val	0.202028	0.182358
2	SVD_hybertuned_trained_val	0.206859	0.189334
3	SVDpp_Basic	0.179064	0.138392

SVD++ Model -Basic Cross Validation Average Performance

Observations indicate that the SVD++ model outperformed other models in terms of Mean Absolute Error during cross-validation, and it also exhibited the second-best performance in terms of Root Mean Squared Error. Now, let's proceed to train the model and evaluate its performance further.

SVD ++ - (SVD Plus Plus) Hyper Parameter Tuned

Define the hyperparameters to be tested and determine the optimal and most impactful parameters.

```
from surprise.model_selection import GridSearchCV
# Hyperparameter tuning
# Define parameter grid for grid search

param_grid = {'n_epochs': [3,5,6,7,10,20,30],
        'lr_all': [.0025, .005, .004,.003,.002, .001, .01]}

# Initialize the GridSearchCV object
gs = GridSearchCV(SVDpp, param_grid, measures=['RMSE','MAE'], cv=5)
gs.fit(data)

print(gs.best_score['mae'])
print(gs.best_params['mae'])
print(gs.best_score['rmse'])
print(gs.best_params['rmse'])
```

✓ 2m 56.5s

```
0.13065424797861122
{'n_epochs': 3, 'lr_all': 0.001}
0.16735402162730256
{'n_epochs': 7, 'lr_all': 0.001}
```

SVD++- Hyperparameter Tuned

Let's train the model with hyper parameters on train set and then test on validation set.

```

# Initialize a new SVD model with the best hyperparameters
svdpp = SVDpp(n_epochs=5, lr_all=0.001)

# Train the new SVDpp model on the entire dataset
trainset = data.build_full_trainset()
svdpp.fit(trainset)
```

✓ 0.4s

```
<surprise.prediction_algorithms.matrix_factorization.SVDpp at 0x1e78aa7e0d0>
```

The result is worse than any other model:

```
RMSE: 0.20930687883257887
MAE: 0.1932281130385172
```

	Model	RMSE	MAE
0	SVD_Basic	0.178879	0.139475
1	SVD_trained_val	0.200400	0.182003
2	SVD_hybertuned_trained_val	0.207447	0.189889
3	SVDpp_Basic	0.179064	0.138392
4	SVDpp_Hypertuned	0.209307	0.193228

SVD++- Hyperparameter Tuned Evaluation

It is evident that the SDVpp is the worst model so farSVD_trained_val is the best model so far. Let's see the results on test set:

```
RMSE: 0.19433912584374907
MAE: 0.1744014186667538
                        Model      RMSE       MAE
0                   SVD_Basic  0.178879  0.139475
1             SVD_trained_val  0.200400  0.182003
2  SVD_hybertuned_trained_val  0.207447  0.189889
3                 SVDpp_Basic  0.179064  0.138392
4            SVDpp_Hypertuned  0.209307  0.193228
5        SVD_1_train_val_test  0.194339  0.174401
```

SVD model trained on trainset and test on testset

The RMSE of 0.193 and MAE of 0.171 indicate the average error in predicting ratings on the test set.

An RMSE of 0.193 suggests that, on average, the predicted ratings differ from the actual ratings by approximately 0.193 units.

Similarly, an MAE of 0.171 means that, on average, the predicted ratings differ from the actual ratings by about 0.171 units.

These values suggest that the SVD_trained_val model's predictions are reasonably accurate, but there is still room for improvement.Now let's get recommendations for the actual 10,000 campaign target customers.

Let's get the recommendation response prediction table for all of the services for a particular user:

	user_id	service_code	pred	Service_Name
0	1114648.0_7	5	0.337177	long_term_deposit
1	1114648.0_7	7	0.306881	emc_account
2	1114648.0_7	6	0.296625	payroll_account
3	1114648.0_7	0	0.281888	em_account
4	1114648.0_7	14	0.240153	em_account_p
...	...	...	...	...
149995	1257213.0_4	10	0.177047	short_term_deposit
149996	1257213.0_4	11	0.149390	loans
149997	1257213.0_4	2	0.134514	pension
149998	1257213.0_4	3	0.132424	payroll
149999	1257213.0_4	13	0.082852	payroll_pension_unknown

Recommendation table example for a particular customer for all of the service

Now, let's calculate the ROI. Here is the approach:

I generate a data frame consisting of three columns: service name, number of users with 0 selected services (denoted as potential_num_user), prediction, and the price of the product.Next, I multiply these values together. Prior to the multiplication by the price, I deduct a pessimistic prediction mean of, say, 0.18 (which is actually 0.168939).

Furthermore, it's crucial to consider the response rate of email campaigns, which typically falls within the range of 1% to 10%, varying based on the marketing strategy and content employed. It's imperative to emphasize that my recommendations are solely derived from the outcomes of the model.

The following table is the dataframe for ROI. It consists of all the columns in the recommendation table and roi per service, whether the subscriber is using the service or not and the final column is the roi per service for each user. The final ROI is calculated as below:

user_id	service_code	pred	Service_Name	service_selection_ratio	roi	service_not_activated	final_roi_per_service_user
1114648.0_7	5	0.337177	long_term_deposit	0.000000	40	1	13.487068
1114648.0_7	7	0.306881	emc_account	0.250000	10	0	0.000000
1114648.0_7	6	0.296625	payroll_account	0.000000	10	1	2.966252
1114648.0_7	0	0.281888	em_account	0.250000	10	0	0.000000
1114648.0_7	14	0.240153	em_account_p	0.000000	10	1	2.401529
...	...	...	...	...	...	...	...
1257213.0_4	9	0.234165	funds	0.000000	40	1	9.366617
1257213.0_4	4	0.226798	credit_card	0.000000	60	1	13.607897
1257213.0_4	8	0.220721	securities	0.000000	40	1	8.828849
1257213.0_4	1	0.217998	debit_card	0.261986	60	0	0.000000
1257213.0_4	12	0.189973	mortgage	0.000000	40	1	7.598931

```
# Calculate ROI for each recommendation

all_recommendations_with_ratio['final_roi_per_service_user'] =
all_recommendations_with_ratio['pred'] *
```

```
all_recommendations_with_ratio['roi'] *
all_recommendations_with_ratio['service_not_activated']


total_roi =
all_recommendations_with_ratio['final_roi_per_service_user'].sum()
```

total_roi = 580,642.9984063023

However, it is safe to consider the mean absolute error and base our scenario on the worst case which is when all the predictions are predicted 0.175 higher. So, first the predictions lower than 0.18 are deleted as the threshold for not interested in using the service. So, the MAE is deducted from each prediction and the ROI is calculated again

```
:# Calculate adjusted ROI for each recommendation

all_recommendations_with_ratio['adjusted_roi_per_service_user'] =
(all_recommendations_with_ratio['pred']-mae) *
all_recommendations_with_ratio['roi'] *
all_recommendations_with_ratio['service_not_activated']
```

In this scenario the ROI is So for our 10,000 campaign for our target customers, the pessimistic roi is 155469 and the normal predicted ROI is 580642 Euros .

total_adjusted_roi = 155,469.7798386232

4.3.2 K-Nearest Neighbors (KNN) Model

On the other flank stands the KNN model, a beacon of simplicity yet a titan in its predictive prowess. Armed with the wisdom of proximity, it forges recommendations based on the collective wisdom of neighboring users, carving a path through the labyrinth of customer preferences with unerring precision.

The model uses cosine similarity function to calculate the similarity between different users and based on that it recommend/predict list of desired products.

```
# Declaring the similarity options.
sim_options = {'name': 'cosine',
               'user_based': True}

# KNN algorithm is used to find similar items
sim_user = KNNBasic(sim_options=sim_options, verbose=True, random_state=11)

# Cross Validate
sim_user_results = cross_validate(algo=sim_user, data=data, cv=4)

# Results!
sim_user_results
```

95] ✓ 46m 1.4s Python

Let’s first get the KNN cross validation performance:

```
{'test_rmse': array([0.17486181, 0.17422347, 0.17431684, 0.17390501]),
 'test_mae': array([0.133904  , 0.13329538, 0.13310264, 0.13275238]),
 'fit_time': (501.76875495910645,
  438.80033826828003,
  461.0615003108978,
  441.63235902786255),
 'test_time': (255.83298420906067,
  190.31322860717773,
  176.99492144584656,
  294.1289436817169)}
```

KNN Model - Basic Cross Validation Evaluation

we get a better error matrix (rmse and mae) for the user similarity than the svd model with :{ 'test_rmse': array([0.17486181, 0.17422347, 0.17431684, 0.17390501]),

'test_mae': array([0.133904 , 0.13329538, 0.13310264, 0.13275238]),

Here is the evaluation on validation set:

```
RMSE: 0.20475907768509885
MAE: 0.19065837163521854
                        Model      RMSE       MAE
0                   SVD_Basic  0.178879  0.139475
1             SVD_trained_val  0.200400  0.182003
2  SVD_hybertuned_trained_val  0.207447  0.189889
3                 SVDpp_Basic  0.179064  0.138392
4            SVDpp_Hypertuned  0.209307  0.193228
5        SVD_1_train_val_test  0.194339  0.174401
6          similarity_knn_val  0.204759  0.190658
```

KNN Model - Trained and tested on validation Evaluation

Now let’s evaluate it on test set:

```
RMSE: 0.2030177341194974
MAE: 0.18817111422183216
                         Model      RMSE       MAE
0                    SVD_Basic  0.178879  0.139475
1              SVD_trained_val  0.200400  0.182003
2  SVD_hybertuned_trained_val  0.207447  0.189889
3                  SVDpp_Basic  0.179064  0.138392
4             SVDpp_Hypertuned  0.209307  0.193228
5          SVD_1_train_val_test  0.194339  0.174401
6            similarity_knn_val  0.204759  0.190658
7           similarity_knn_test  0.203018  0.188171
```

KNN Model - Trained and tested on Test set
Evaluation

Finally, the recommendation function is called to observe its predictions on different services :

	user_id	service_code	pred	Service_Name
0	1114648.0_7	5	0.337177	long_term_deposit
1	1114648.0_7	7	0.306881	emc_account
2	1114648.0_7	6	0.296625	payroll_account
3	1114648.0_7	0	0.281888	em_account
4	1114648.0_7	14	0.240153	em_account_p
...	...	...	...	...
149995	1257213.0_4	10	0.177047	short_term_deposit
149996	1257213.0_4	11	0.149390	loans
149997	1257213.0_4	2	0.134514	pension
149998	1257213.0_4	3	0.132424	payroll
149999	1257213.0_4	13	0.082852	payroll_pension_unknown

150000 rows × 4 columns

KNN Model - Recommendation table

So here is the snapshot of the performance of various models:

```
RMSE: 0.2030177341194974
MAE: 0.18817111422183216
                         Model      RMSE       MAE
0                    SVD_Basic  0.178879  0.139475
1              SVD_trained_val  0.200400  0.182003
2  SVD_hybertuned_trained_val  0.207447  0.189889
3                  SVDpp_Basic  0.179064  0.138392
4             SVDpp_Hypertuned  0.209307  0.193228
5          SVD_1_train_val_test  0.194339  0.174401
6            similarity_knn_val  0.204759  0.190658
7           similarity_knn_test  0.203018  0.188171
```

Model Performance metrics

It's notable that the basic models for SVD and SVD++ performed well even without training, indicating that the data is simple and easily predictable. However, the performance of the SVD++ model was unexpectedly poor. This could be attributed to several factors such as inadequate tuning of hyperparameters, insufficient data preprocessing, or mismatches between the model's complexity and the dataset's characteristics. Further investigation is warranted to pinpoint the specific reasons behind the underperformance of the SVD++ model.

In conclusion, the recommendation task has been executed, with notable success achieved using models such as SVD and SVD++. However, it's crucial to address any unexpected performance issues observed, particularly with the SVD++ model, through further investigation and refinement of the modeling process.

Moving forward, considering the potential benefits of employing advanced techniques like neural networks and gradient boosting models can be explored. These models offer the advantage of capturing complex patterns in the data, potentially leading to improved recommendation accuracy. Additionally, hybrid models, which combine multiple recommendation techniques, can be investigated to leverage the strengths of different algorithms and enhance recommendation quality further.

To enhance recommendations and response rates, EasyMoney can take several actions:

Data Enrichment: Incorporating additional data sources such as customer demographics, browsing history, or transactional data can provide deeper insights into user preferences and behavior, leading to more personalized recommendations.
Continuous Model Evaluation and Improvement: Regularly assessing the performance of recommendation models and iterating on them based on feedback and changing user preferences can ensure that recommendations remain relevant and effective over time.
A/B Testing: Implementing A/B testing methodologies to evaluate the impact of different recommendation strategies and fine-tune them based on real-world user interactions and feedback.
Feedback Loop Integration: Incorporating mechanisms for collecting user feedback on recommendations can help in refining the models and improving their accuracy over time.
Dynamic Content Personalization: Implementing dynamic content personalization techniques to tailor recommendations in real-time based on user interactions, contextual information, and browsing behavior can enhance user engagement and conversion rates.

By adopting these strategies and continually iterating on the recommendation models based on insights gained from data analysis and user feedback, EasyMoney can optimize its recommendation engine to deliver personalized, relevant, and high-converting recommendations, ultimately driving better response rates and customer satisfaction.

Monitoring - Campaign KPIs to monitor

Campaign Name: "EasyMoney's PiggyBank Bonanza: Elevating Customer Experience Through Personalized PiggyAccount Bundles"

Objective: To enhance customer engagement and increase revenue by delivering personalized PiggyAccount product bundles tailored to each customer's financial goals and preferences.

Segmentation Strategy:

- Segment customers based on their purchase history, commercial traits and sociodemographic information.
- Use machine learning algorithms to identify 8 unique customer segments with distinct preferences.
- Created personalized product bundles recommendation for potential top 10000 customers to increase relevance and customer satisfaction.

Predictive Modeling Approach:

- Utilize collaborative filtering and user-based recommendation systems to predict products that are most likely to appeal to each customer.
- Incorporate real-time data and customer feedback to continuously improve the accuracy of the recommendations.
- Implement reinforcement learning techniques to adapt recommendations based on customer interactions and feedback.

Response Rate Prediction:

- Predict the probability of a customer responding positively to the recommendations based on their past interactions and segment characteristics.
- Use historical response data and machine learning models to estimate the response rate for the current campaign.
- Implement sentiment analysis on customer feedback to gauge the effectiveness of the recommendations.

Expected ROI Calculation:

- Estimate the expected revenue from each customer by analyzing the potential purchases of the recommended products.
- Calculate the cost of the campaign, including personalized marketing materials and personnel costs.
- Determine the expected ROI as (Expected Revenue - Campaign Cost) / Campaign Cost.

Key Performance Indicators (KPIs):

Let's walk through the users' journey and their touchpoints with Easy money and see what we can measure:

Step 1. 10,000 Subscribers receive email for the campaign.

1. **Open Rate**: The percentage of our target subscribers who open your email. This indicates how engaging our email contents are.

2. **Click-Through Rate (CTR)**: The percentage of recipients who click on a link in campaign emails. This measures the effectiveness of the call-to-action and content.

3. **Conversion Rate**: The percentage of recipients who actually subscribed to our recommended products, such as making a purchase or signing up for a service, after clicking on a link in the email.

4. **Bounce Rate**: The percentage of emails that were not delivered successfully to recipients' inboxes. High bounce rates can indicate issues with your email list quality or email deliverability.

5. **Revenue per Email**: The average amount of revenue generated per email sent. This helps you understand the direct impact of your email campaigns on revenue.

6. **ROI (Return on Investment)**: The ratio of the net profit generated by the email campaign to the total cost of the campaign. This helps us to understand the overall effectiveness of your email marketing efforts.

7. **Subscriber Engagement**: Metrics such as the number of shares, forwards, or replies to the emails can indicate how engaged our customers are with our content.

These classic KPIs provide a good starting point for measuring the success of the email marketing campaign. However, it's also important to consider additional KPIs that are specific to Easy Money goals and objectives.

- **ROI Efficiency Ratio per cluster:** We can measure the effectiveness of our recommendations by monitoring this kpi. It is the sum of potential return on investment of the subscribers in each cluster vs the actual ROI they are generating after the campaign. Our goal is to maximize that.

- **Cluster ROI ranking:** We can rank target clusters on the revenue they are generating. It helps us better understand our clusters' behavior and how well they are targeted and defined.

- **Product Efficacy Ratio:** The "Product Efficacy Ratio" could be a useful metric to evaluate the effectiveness of each product in generating ROI. This ratio could be defined as the actual ROI generated by a product divided by the potential ROI of that product. We can measure this for this campaign as well as in general.

  Ranking Leaderboards based on various classic KPIs:

  - **Customer Leaderboard:**
  - Ranking: Rank customers based on their purchase frequency, total subscription and total heart rate on the EasyMoney world.
  - Metrics: Include metrics such as total purchases, total revenue generated, average purchase amount, number of products subscribed to.
  - Insights: Identify high-value customers, trends in customer behavior, and opportunities for further targeted marketing campaigns.

  Product Leaderboard:

  - Ranking: Rank products based on their popularity, revenue generated, or other relevant metrics.
  - Metrics: Include metrics such as total sales, total revenue, number of recent subscribers vs expected number of subscribers.
  - Insights: Identify top-selling products, trends in product demand, and opportunities for product optimization or promotion.

Cluster Leaderboard:

- Ranking: Rank clusters based on their ROI, conversion rate, and positive response rate to the campaign
- Metrics: Include metrics such as total revenue generated by the cluster in the campaign, ROI , etc.
- Insights: Identify high-performing clusters, trends in cluster behavior, and opportunities for targeted marketing or cluster-specific campaigns.

Engagement Index: A composite KPI that measures the overall engagement of customers with the personalized recommendations, taking into account factors such as open rates, click-through rates, subscription, frequency of use for each product and average time spent on EasyMoney platforms.

- **Conversion Quality Score:** A metric that evaluates the quality of conversions generated by the campaign, considering factors such as the frequency of use of the product , time spent, money spent.

- **Recommendation Relevance Rate**: The percentage of customers who find the recommended products relevant to their needs and preferences, as indicated by feedback surveys and percentage subscribed.

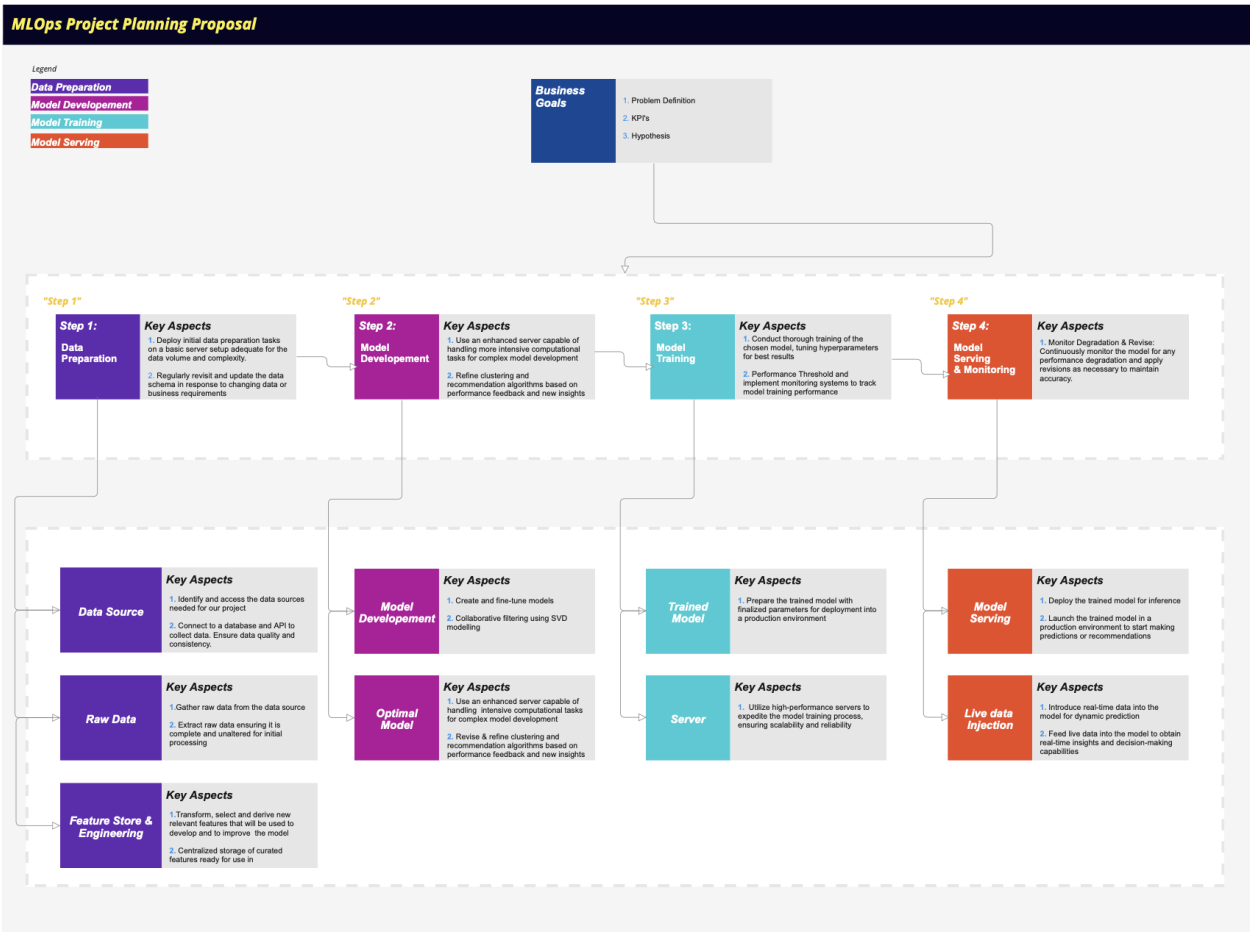
- **Customer Delight Index:** A measure of customer satisfaction and delight resulting from the personalized recommendations, assessed through customer feedback and sentiment analysis.

- **Brand Loyalty Growth**: The increase in brand loyalty among customers who receive personalized recommendations, measured by using the product (total transactions on accounts, amount on loan and credit card and debit card frequency of use and number and amount of transactions, for each product is different) and customer retention rates.

- **Bundle Response Rate**:
    - Definition: The percentage of customers who make a purchase after receiving an offered product bundle.
    - Calculation: Number of customers who purchased after receiving a bundle / Total number of customers who received the bundle * 100.
    - Interpretation: A higher Bundle Response Rate indicates that the offered product bundles are more appealing and effective in generating a response from customers.

MLOps Planning Proposal

The MLOps Planning Proposal at EasyMoney meticulously outlines a comprehensive roadmap designed to leverage methodologies to optimize customer engagement and streamline product offerings. The proposal is a systematic approach that integrates technical development with business objectives, consisting of four pivotal steps: Data Preparation, Model Development, Model Training, and Model Serving & Monitoring, each carrying key aspects essential for the operational execution of the project.



<u>Data Preparation:</u>

This foundational step emphasizes initial data sourcing and preparation tasks tailored to the volume and complexity of data. It focuses on identifying and accessing data sources, ensuring data quality and consistency, and engaging in feature engineering where raw data is transformed and relevant features are defined and stored for model development.

Business Goals:

The preliminary phase, titled "Business Goals," meticulously delineates the core objectives of this venture. It commences with a thorough "Problem Definition," where the intricacies of the challenges faced by EasyMoney are articulated, laying a foundation for the application of machine learning solutions. Following this, "KPIs" are carefully selected and defined, serving as quantifiable benchmarks to gauge the success of the implemented model in aligning with EasyMoney's overarching business aspirations. Additionally, the "Hypothesis" segment anticipates the potential impact of the model on addressing the identified problem and its influence on the predefined KPIs, charting a hypothesis-driven path towards solving EasyMoney's conundrum.

Model Development:

Transitioning to "Model Development," the choice of employing Collaborative Filtering via Singular Value Decomposition (SVD) modeling emerges as a strategic decision underscored by its efficacy within the constraints of available resources and time. Although neural networks represent an ideal paradigm for crafting sophisticated recommendation systems, the selection of SVD modeling is a testament to EasyMoney's pragmatic approach in navigating current limitations, striking a balance between ambition and operational feasibility.

Model Training:

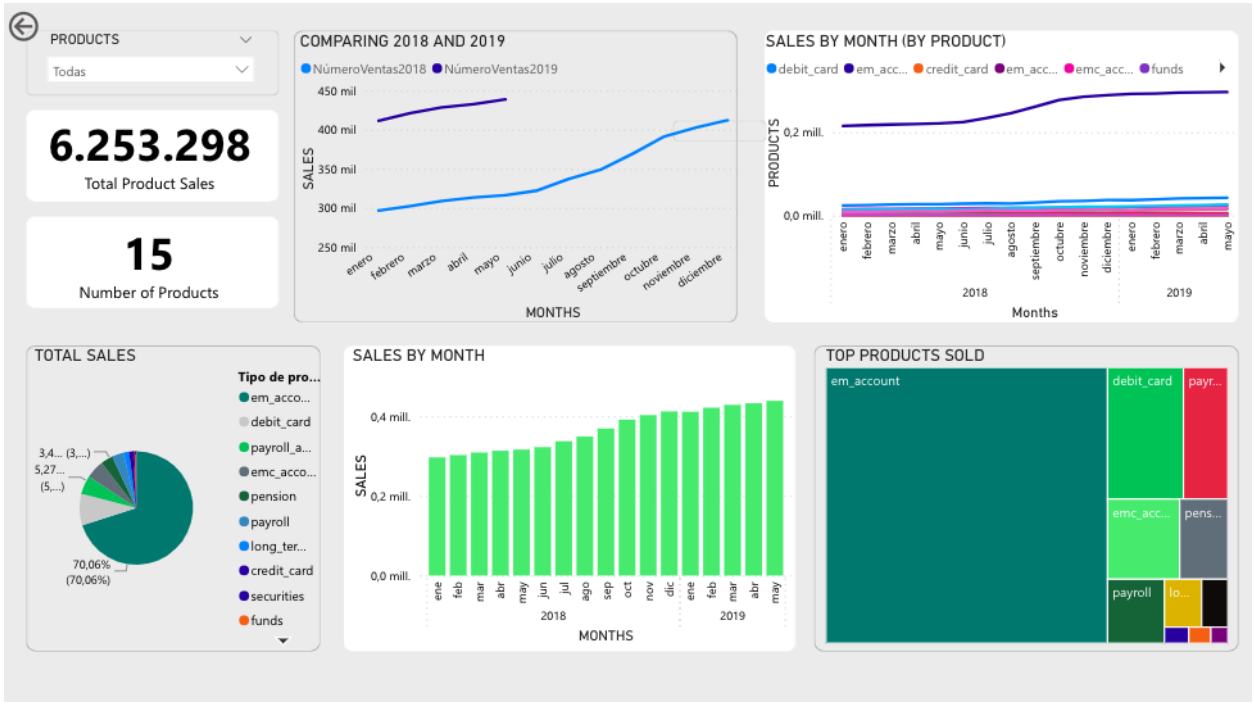
This phase involves the diligent training of the chosen model, adjusting hyperparameters for optimal results, and setting performance thresholds to track the efficacy of model training. High-performance servers are utilized to ensure the scalability and reliability of this process

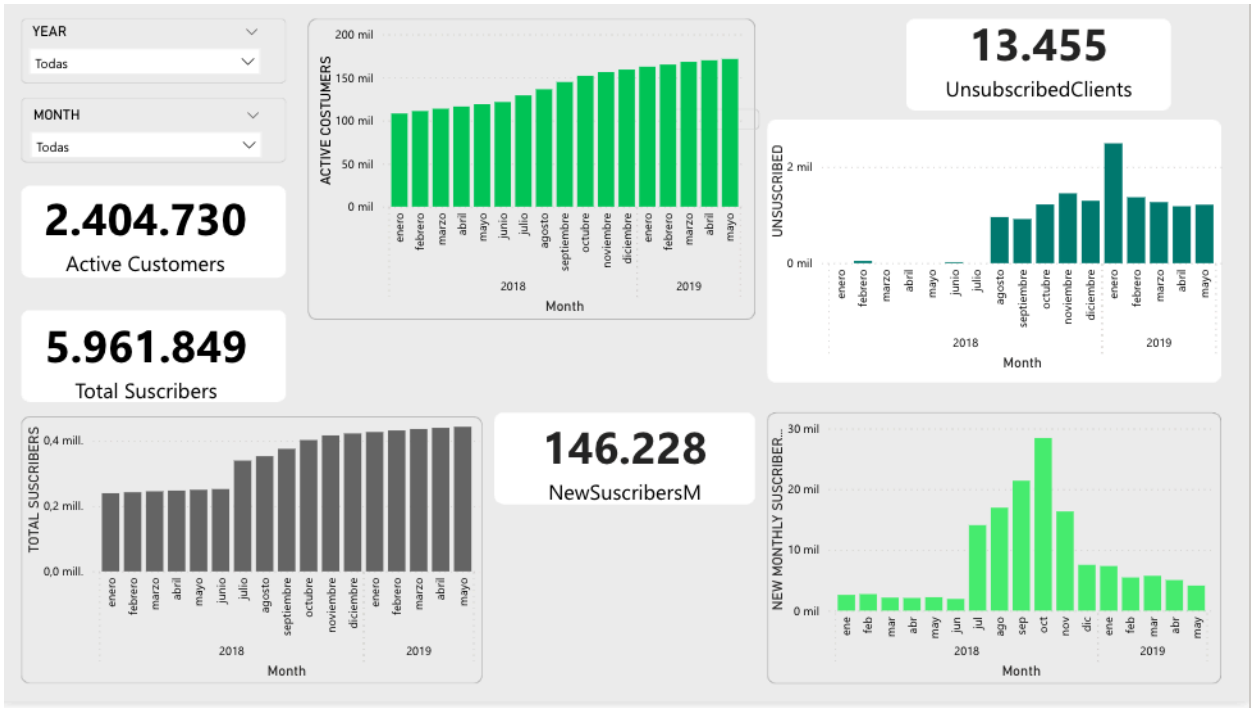
Model Serving:

The "Model Serving" stage is envisioned as a dynamic process that transcends mere deployment. It emphasizes the importance of integrating real-time data and leveraging customer feedback as instrumental components for the continuous refinement of the recommendation system. This iterative process ensures that the model remains responsive and adaptive to evolving customer preferences and market trends.

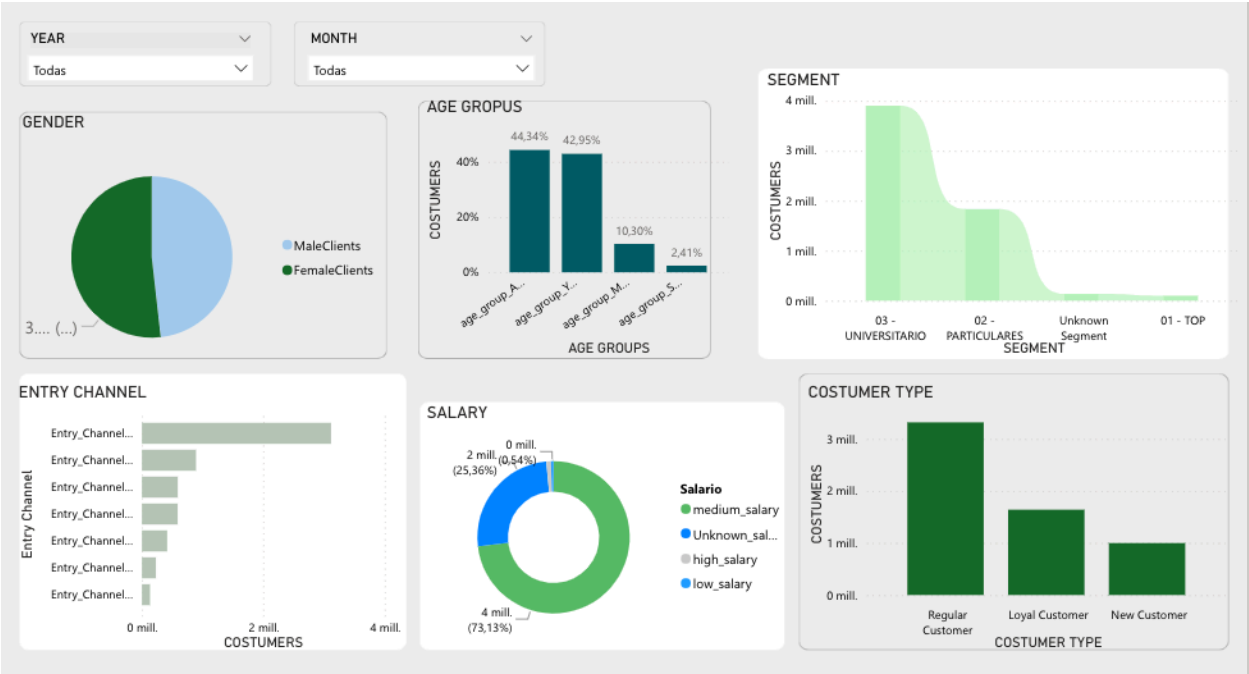
Moreover, the project's lifeblood should be sustained through vigilant "Monitoring" practices. This phase is dedicated to the perpetual assessment of model performance and data pipeline efficiency against the backdrop of the defined KPIs. Utilizing sophisticated monitoring tools and dashboards, EasyMoney commits to an unwavering scrutiny of real-time metrics, poised to initiate timely adjustments or model updates. This ensures the sustained relevance and value delivery of the machine learning model, thereby propelling EasyMoney towards achieving its strategic objectives and fostering a culture of data-driven innovation and customer-centric growth.

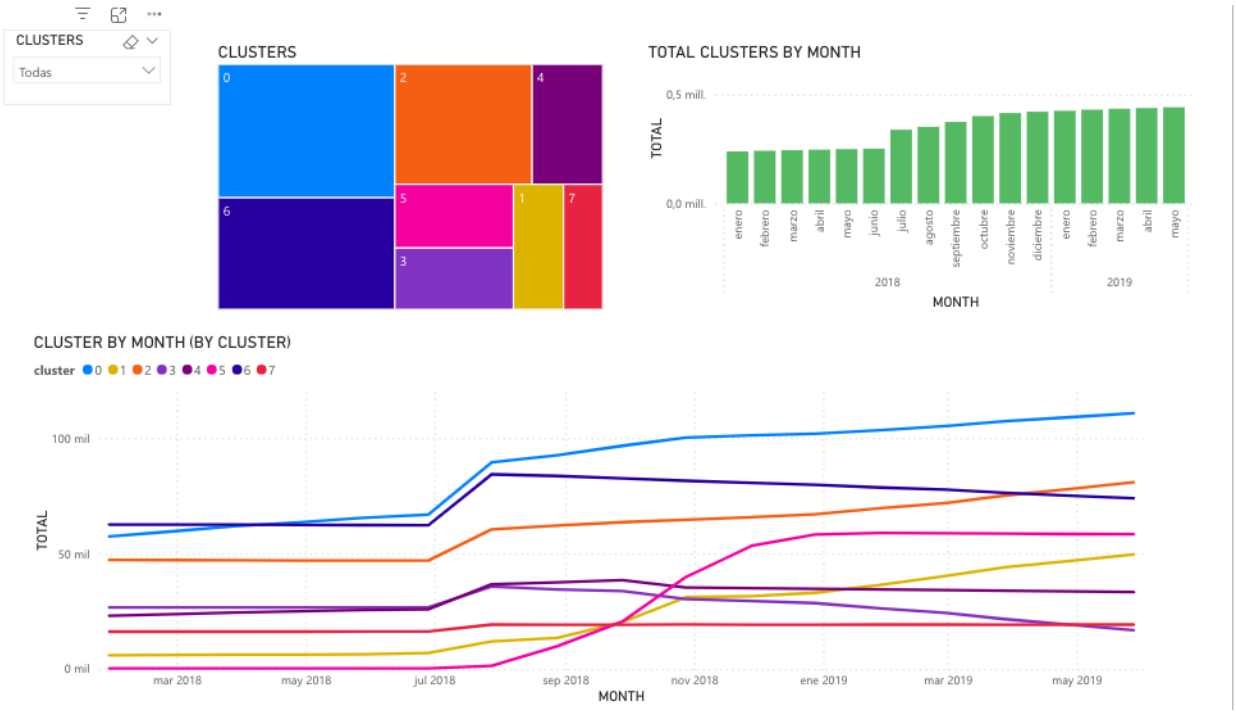
Dashboard


PRODUCTS
Todas
6.253.298
Total Product Sales
15
Number of Products
COMPARING 2018 AND 2019
NúmeroVentas2018
NúmeroVentas2019
SALES
MONTHS
SALES BY MONTH (BY PRODUCT)
debit_card
em_acc...
credit_card
em_acc...
emc_acc...
funds
PRODUCTS
Months
2018
2019
TOTAL SALES
Tipo de pro...
em_acco...
debit_card
payroll_a...
emc_acco...
pension
payroll
long_ter...
credit_card
securities
funds
70,06%
(70,06%)
SALES BY MONTH
SALES
MONTHS
TOP PRODUCTS SOLD
em_account
debit_card
payroll


YEAR
Todas
MONTH
Todas
2.404.730
Active Customers
5.961.849
Total Suscribers
ACTIVE COSTUMERS
Month
13.455
UnsubscribedClients
UNSUSCRIBED
Month
TOTAL SUSCRIBERS
Month
146.228
NewSuscribersM
NEW MONTHLY SUSCRIBER...
Month

easyMoney
Make your dreams come true





Conclusions

Through a rigorous process of data cleaning and explorative analysis, the team processed a vast dataset containing 5,962,924 customer records, rectifying critical data gaps, such as 133,944 missing 'segment' values and 133,033 missing 'entry_channel' values. This foundational work ensured a dataset with 456,373 unique customer IDs, from which the team was able to identify and rectify inconsistencies.

The application of K-modes clustering algorithm segmented customers into eight discernible groups, with Clusters 0, 6, and 2 representing the most prevalent customer profiles. Conversely, Cluster 7 emerged as the most specialized group with distinctive characteristics. This segmentation revealed that a mere 0.25% of customers subscribed to short-term deposits and underscored the importance of product preferences, where 'em_account' proved predominant, and others, like 'securities' and 'loans', exhibited lower popularity.

The study noted a pivotal surge in subscriber numbers in July 2018 and identified that 38.7% of subscribers are active, with new subscriber rates peaking in October 2018 at 28,462 before declining. Cross-tabulation analysis presented a nuanced view of the customer base, where clusters demonstrated varied salary distributions and new customer growth, especially within Clusters 5 and 3.

Crucially, the report advanced the development of an intricate recommendation model aimed at predicting campaign response rates and optimizing ROI. It proposed a sophisticated approach using machine learning algorithms such as SVD and collaborative filtering to recommend products that harmonize user interest with EasyMoney's financial objectives. The ROI analysis delineated a potential profit strategy for 10,000 targeted customers, with earnings projected from accounts and various product lines based on their adoption rates.

The report also underscores the essential role of continuous customer behavior monitoring within the recommendation system to cater to evolving customer preferences and life stages. Additionally, it outlines an MLOps planning proposal to align technical development with EasyMoney's business objectives. This includes hyperparameter tuning and a robust monitoring system for the real-time evaluation of machine learning models to ensure they remain pertinent and optimized.

In essence, the report crystallizes EasyMoney's data-centric methodology as a conduit to foster targeted customer engagement, refine product recommendations, and optimize marketing endeavors. By integrating detailed customer segmentation with actionable predictive analytics and an adaptable MLOps infrastructure, EasyMoney is positioned to enhance customer retention and achieve sustained business growth. This strategic approach is not only aimed at reconciling customer trust but also at reinforcing EasyMoney’s presence in the market with a view towards long-term profitability and customer satisfaction.