U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

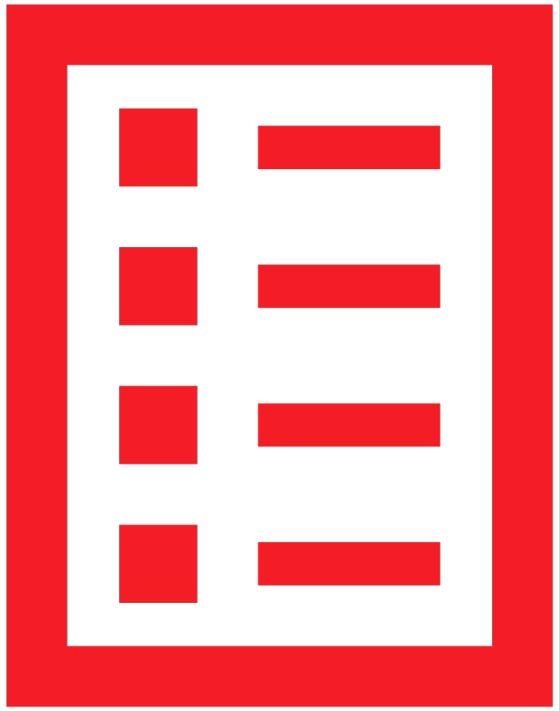
Customized Challenges – Sonae MC

LCED

Master in Data Science & Engineering, 2022/23

Group composed by:
Henrique Ribeiro
Ian Karkles
Luís Henriques
Miguel Veloso
Paulo Portela
Vitor Pereira

Outline

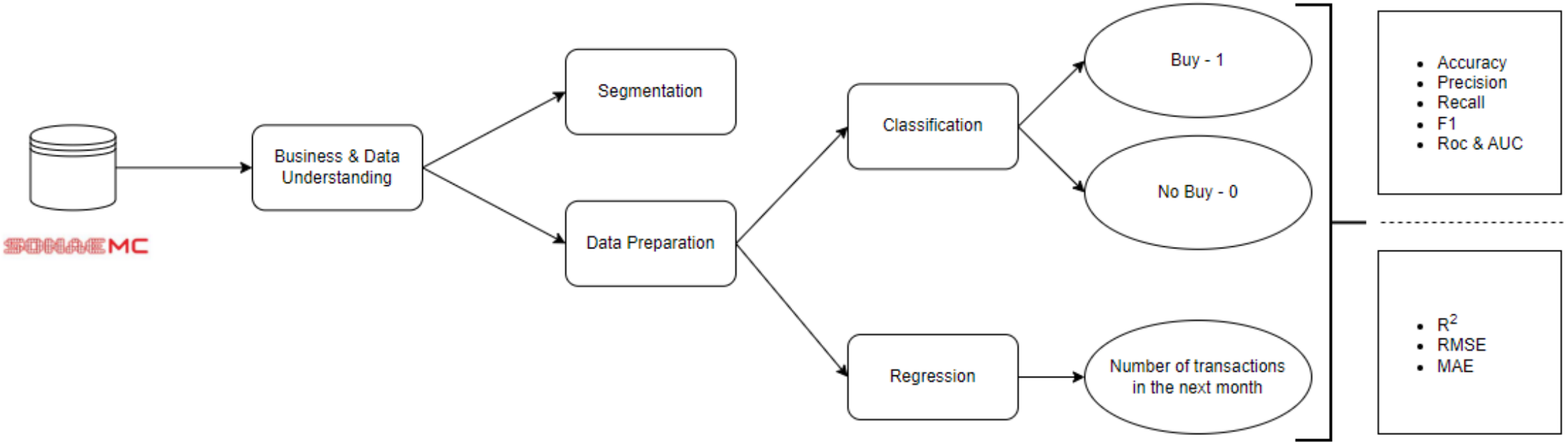


- Business Understanding
- Data Understanding
- Data Preparation
- Modeling
- Evaluation
- Deployment
- Conclusions, limitations and future work
- Annexes

Problem Description

- **Main Goal** → Creation of a customized challenge product based on the 20 most relevant offers, per customer;
- **Secondary Goal** → Predict the number of transactions, per customer, for the upcoming month.

CRISP-DM METHODOLOGY



1. Business Understanding

1.1 Background Summary

- **Company**: SONAE MC
- **Project proponent:** Management Entity of the Continente Card (Advanced Analytics & Insight | Campaign Intelligence)
- **Steering committee FEUP**: Ana Aguiar, Carlos Soares, Vera Miguéis
- **General project area:** Gamification – Customized Challenges Products

SONAE MC **leads the food retail** sector in Portugal

More than **35 years of experience**

Multinational that manages a **diversified portfolio of businesses** in retail, financial services, technology, real estate, and telecommunications

Multi-format business

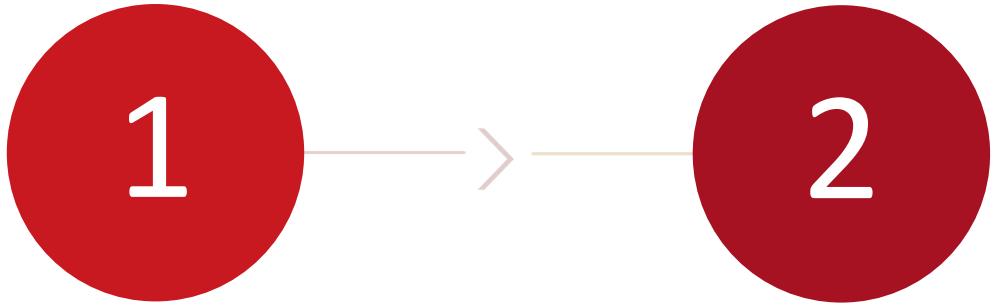
Mission: create **long-term economic and social value**, taking the benefits of progress and innovation to an ever-increasing number of people

In the year **2022**, a **profit of 342 million** euros was recorded, representing an increase of 27.7% over the previous year

1.2 Business Objectives

It is important to consider some additional business requirements, defined by SONAE MC:

- "**Gamification** has been one of the main investments of loyalty programs - mechanics boost sales by focusing on creating an experience that retains experience that retains the customer's attention";
- **Personalized challenges** are one of Sonae MC gamification mechanics;
- Making **recommendations** at the subcategory or brand level;
- **Identify offers that are relevant** to each customer, based on their purchase history in the last 12 months;
- Don't bias the recommendations based on the location of the retail stores and don't consider seasonality;
- If selecting a set of customer segments and subcategories, justify factually.



Identify 20 most relevant offers

- Creation of a customized challenge product based on the **20 most relevant offers** per customer, in a logic of reward (offers where the customer has already made purchases in the last 12 months).

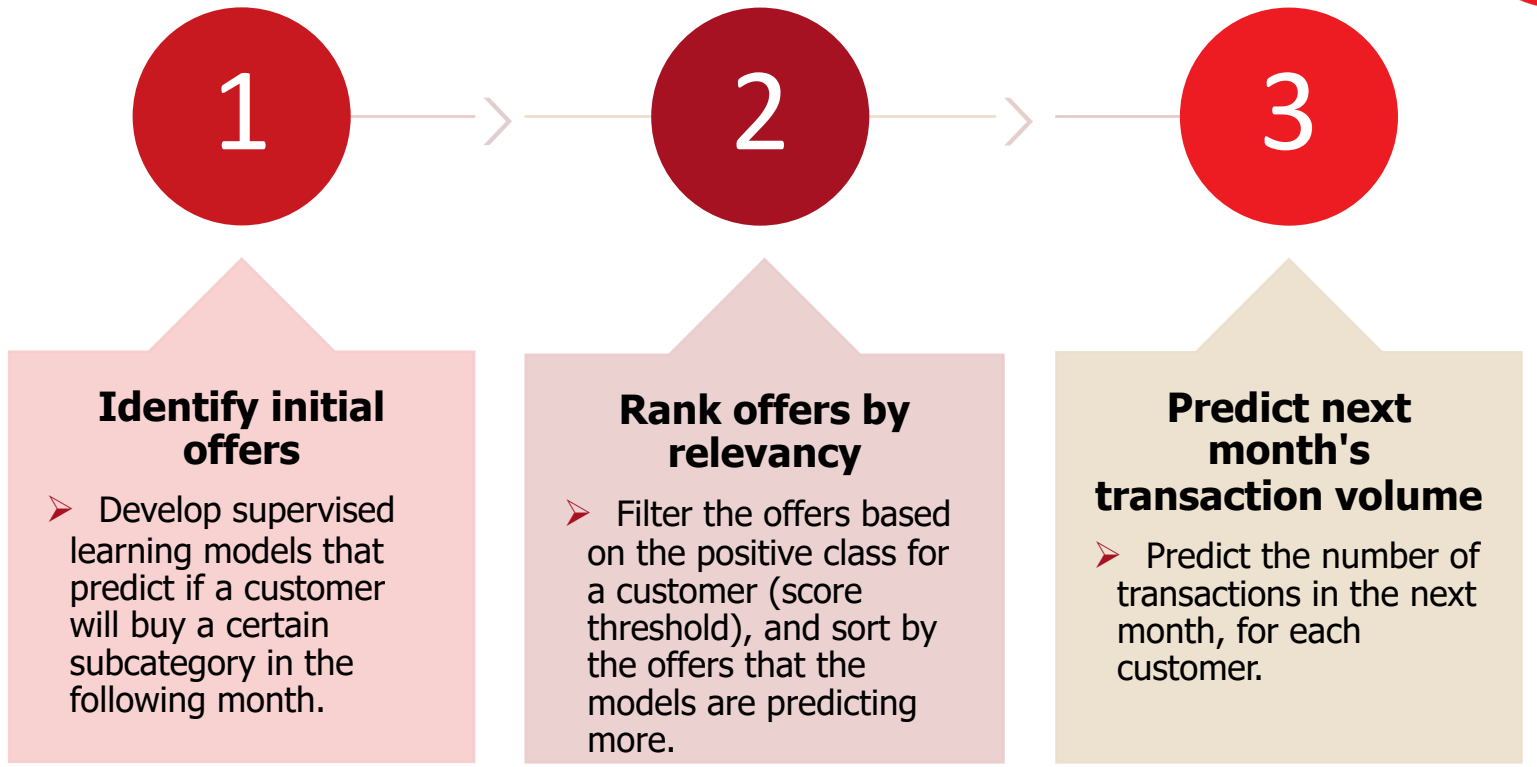
Predict next month's number of visits

- Predict the overall **number of transactions** made at a store/supermarket by a particular customer, for the next month.

1.3 Data Mining Goals

Since SONAE MC did not provide a success criteria, the team took the initiative to develop their own, a baseline model.

- For the main objective of this project, which involves a classification problem, the **baseline model** will recommend the subcategories that a customer purchased in the previous month;
- For the regression problem, the **baseline model** will be the number of transactions made by a customer in the previous month.

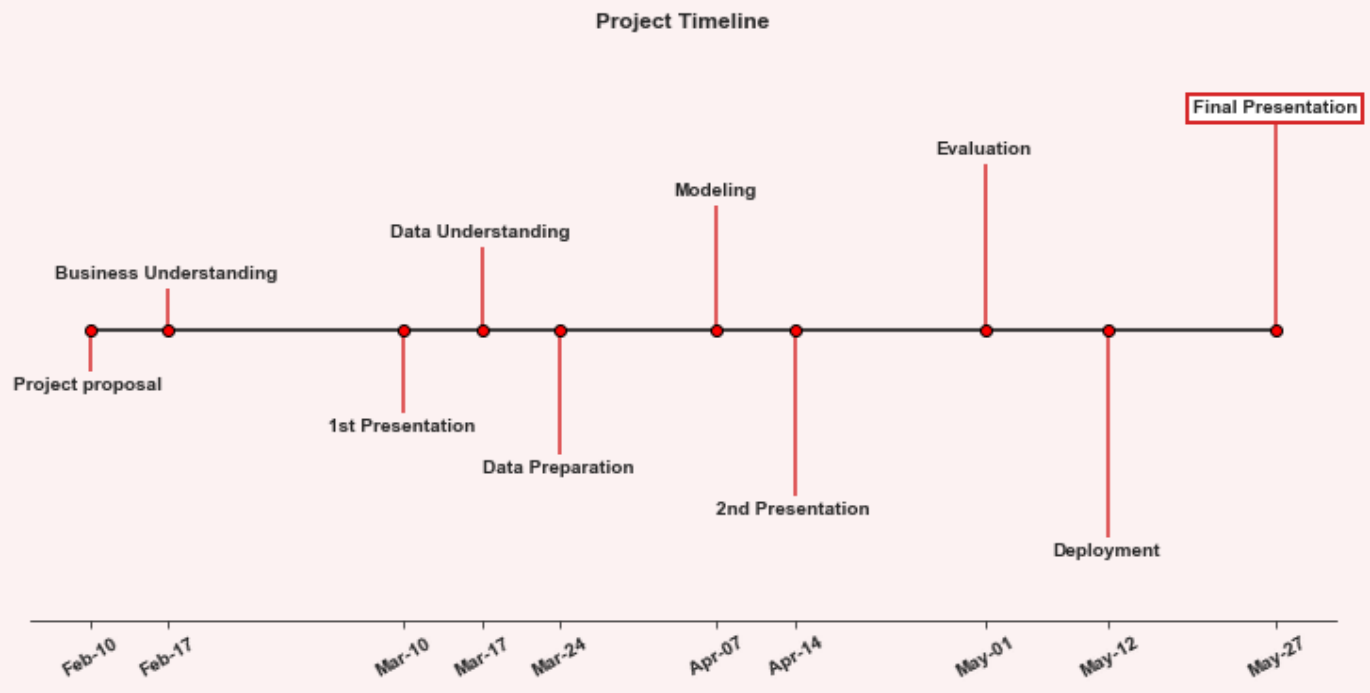


Success Criteria

Achieve significant outcomes compared to a baseline model, significance level of 5%	Recommendations that aren't trivial, but rather very tailored and with a low PERC_TOTAL	R^2 >= 0.15 (defined by the regression's baseline model)

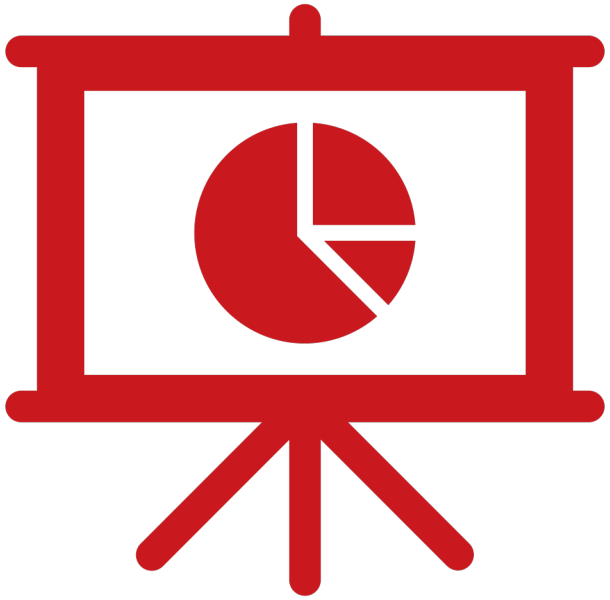
Note: For goal 2, we decided to not focus on filtering positive predictions based on the last 12 months, but rather on filtering positive predictions to recommend 20 tailored subcategories that a customer may have never bought.

1.4 Produce Project Plan



In order to process and develop all the project, several tools were used. The schedule was also adjusted according to the timespan.

- Data collection (MySQL/SQL)
- Data Management (Google Big Query, SQL, Microsoft Excel, Parquet)
- Data Understanding, Preparation, Modeling and Evaluation (Python, SQL, and R)
- Deployment and recommendations (Python/Joblib)
- Version control (Git / GitHub)
- Visualizations (Power BI, Looker, MS Excel, Python, and R)
- Communication (Zoom, Google Meet, and WhatsApp)



2. Data Understanding

2.1 Collect & Describe Data

In the Collect and Describe Data step, we describe the acquired data, including the format, quantity, fields, and any other characteristics identified. We evaluate whether the collected data meets the relevant requirements.

		Features	Observations	Missing Values	Duplicated
Raw Data	Customer	10	93748	35809	0
	Location	5	373	0	0
	Product	18	166207	0	0
	Transaction	14	≈ 66.6 M	10708	0
	Total	**47**	**≈ 66.8 M**	**46517**	**0**

Categorical Variable

SUBCAT_DSC_EXT	Value
Unique	960
Most frequent Subcat	100204 - massas
Frequency of most frequent Subcat	967839
Least frequent Subcat	350502 - medicamentos
Frequency of least frequent Subcat	1

Numerical Features

	FAMILY MEMBERS	QTY	NET_SLS_ AMT	GROSS_SLS _AMT	PROD_DSCNT _ISSUED_AM	TRANS_DSCNT _RAT_AMT	DIRECT_DISCNT _AMT	CONVERSION FACTOR
count	55463664	6,7E+07	66568608	66568608	66568608	66568608	66568608	66568608
mean	3,17	1,29	2,41	2,77	0,01	0,13	0,53	3,33
std	4,79	1,51	4,24	4,8	0,18	0,45	2,13	16,19
min	0	-949,05	-8049,02	-8531,96	-68,95	-130	0	-1
25	2	1	0,97	1,09	0	0	0	0,25
50	3	1	1,62	1,89	0	0	0	0,69
75	4	1	2,82	3,18	0	0,14	0,28	1
max	93	2111,11	8057,08	8540,5	487,5	367,22	999	1800

- The **3rd quartile** number of people **per family** is **4,** which means that 75% of households consist of **4 people**, an extreme of **93 family members** was also recorded;
- **Negative values** in several numerical attributes mean **devolutions**;
- With most of the values in the QTY column being 1, one transaction in specific had a value greater then 2111 units for **chicken steaks**, costing 8540,5 euros;
- With **960 unique subcategories**, the dataset has a wide range of distinct subcategories, from "massas" to "medicamentos".

Note: A features dictionary is presented in the annexes

2.1.1 Collect & Describe Data – Statistical Methods

Here is a list of all **statistical methods** employed during the data understanding phase:

- **Mean**: Calculates the arithmetic average of a set of values, representing a central value of the data;
- **1st , 2nd (Median), and 3rd quartiles**: Determines the value that represents 25%, 50%, and 75% of the numerical data;
- **Standard Deviation**: Measures the dispersion of values around the mean, providing a measure of data variability;
- **Minimum**: Identifies the minimum value, or the smallest observed value, in a dataset;
- **Maximum**: Identifies the maximum value, or the largest observed value, in a dataset;
- **Pearson Correlation***: Assesses the linear relationship between two continuous variables, providing a measure of the strength and direction of that relationship;
- **Variance Inflation Factor (VIF)**: Measures multicollinearity among independent variables in a regression analysis, indicating the influence of each variable on the others. Basically, the VIF of a feature corresponds to a formula that uses the R-squared of the regression performed with that feature as a dependent variable and all others as predictive.
- **Chi-square test of independence***: Tests the association between two categorical variables by comparing observed and expected frequencies in a contingency table;
- **Analysis of Variance (ANOVA)*:** The Mixed Factorial ANOVA, in this case, enables the investigation of how different factors simultaneously, and independently, influence the outcome of a dependent variable, or in this case of the predictive measures for the classification problem, given the success criteria defined;
- **Principal Component Analysis (PCA)**: Reduces the dimensionality of a dataset by identifying linear combinations of the original variables that capture most of the variance, allowing for a more compact representation of the data.

Note 1: The application of these three methods (*) will be discussed in detail in future slides.
Note 2: Although we have specific slides for data understanding, this step was implemented and shown throughout the entire project and presentation, to help the understanding of the reader, and so that should be kept in mind.

2.2 EDA (Exploratory data analysis)

Exploratory Data Analysis (EDA) is a fundamental step in understanding and extracting valuable insights from datasets. To facilitate the understanding and detailed approach implemented to this EDA process, it's possible to divide it into three distinct phases, each with its specific objectives and applied techniques.

1. **First phase -** focused on data quality, covering the datasets Customer, Location, Product and Transaction:
   - Pandas profiling applied to each dataset;
   - Analysis of missing values, null values, outliers, noise and redundancy;
   - Exploration of correlations between variables and univariate statistics;
   - Additional analysis suggested by data quality dimensions.

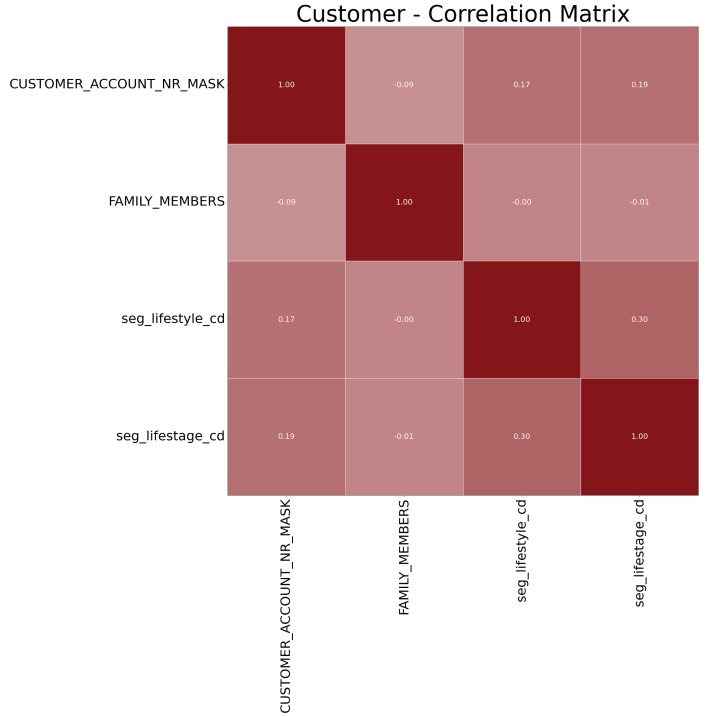
2. **Second phase –** focus on in-depth targeting of some additional requirements defined by SONAE MC:
   - Age segmentation: Focus on the age **range of 25 to 35 years**, as it is the least developed segment among individuals with a source of income and financial stability, as well as being an age group with good digital literacy;
   - Subcategory filtering: Application of **Pareto's principle** (also known as the 80/20 rule) to filter and prioritize the most significant departments, business units, categories and subcategories in the dataset.

3. **Third phase -** creation of segmented visualizations based on the applied filters:
   - Focus on **transactions table** due to extensivety and complexity;
   - Analysis of **transactions data distribution** by month/quarter/year;
   - Impact of **discounts** on customer transactions.

2.2.1 EDA - First Phase

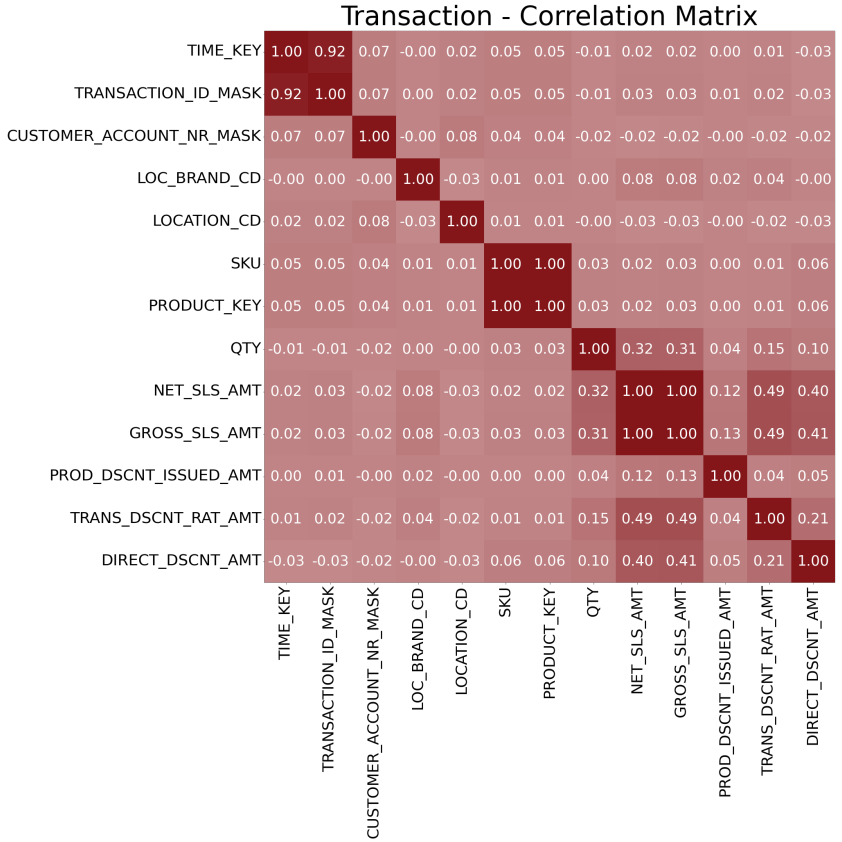
During the project's initial phase, an extensive analysis was carried out on each table - Customers, Products, Stores, and Transactions - utilizing the Pandas Profiling library, as described and detailed below:

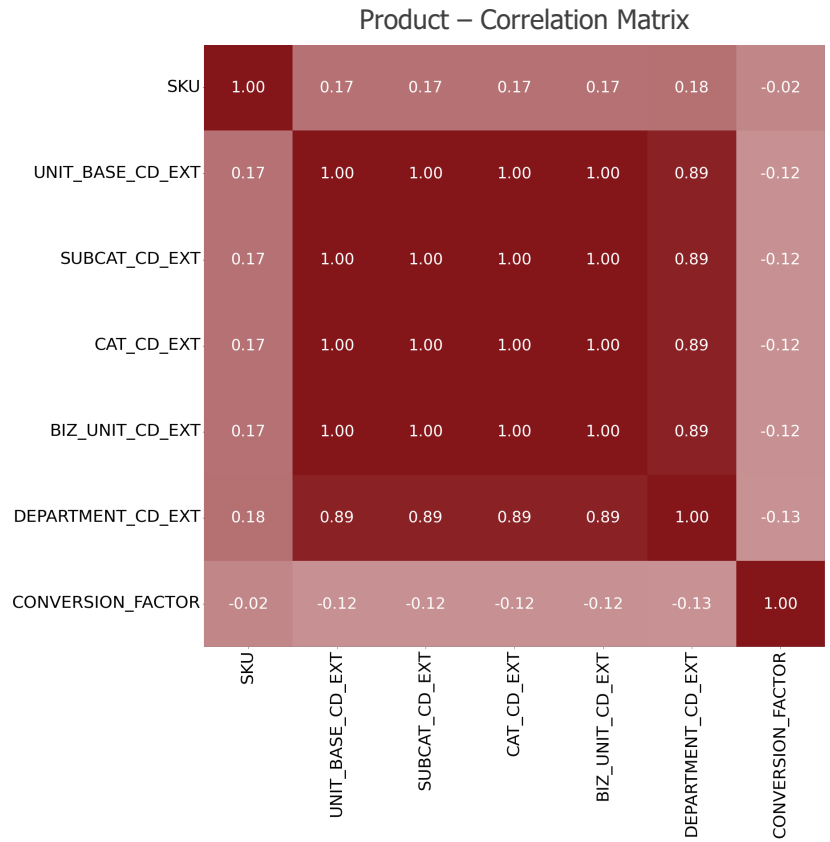
- In the first phase of the work, the **analysis of each table**, namely **Customers, Products, Stores, and Transactions**, was conducted using the Pandas Profiling Library. This involves the gathering of information on correlations, null patterns and other attributes specific to each individual table in the original dataset;
- The **Pandas Profiling Library** is a powerful open-source Python library **that automates the exploratory data analysis (EDA)** process. It provides comprehensive insights and statistical summaries of a given dataset. By invoking the Pandas Profiling function on a DataFrame, a detailed HTML report is generated, which **includes various statistical measures, data types, missing values, correlations, and distribution plots** for each column in the dataset;
- Considering that the customer table contained a significant number of null values initially, the focus was directed towards **filtering and selecting only the relevant values, as well as MV approaches** to be utilized in the subsequent stages of the project.
- Now, **although pandas profiling enables a quick EDA very easily, there're some problems that it has**, one of which being in the correlation matrices. This problematic can be seen in the heatmap on the right, but this will be discussed in the next slide.

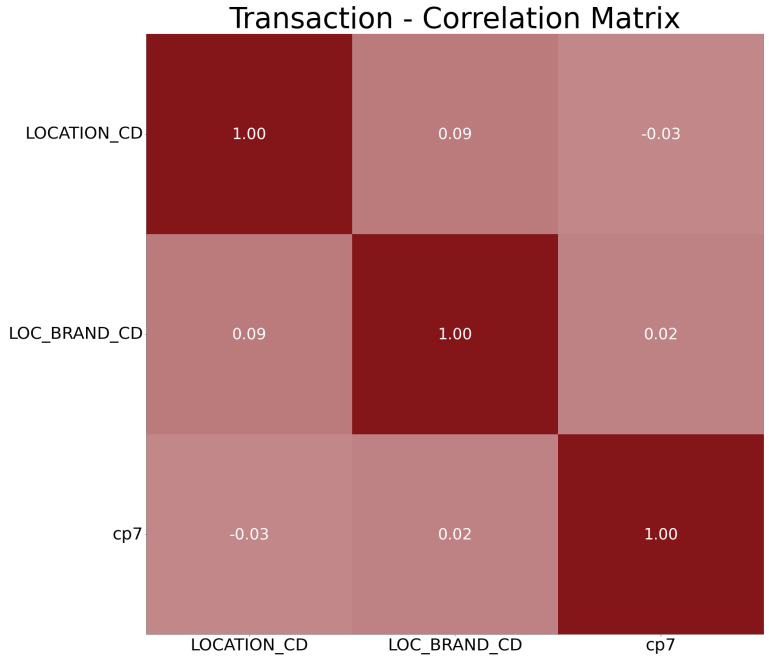


2.2.1 EDA - First Phase

- Due to the transactional nature of the raw data, useful correlations between the fields presented in each column are absent. At this point, it is apparent that binding and modeling the data is necessary to acquire usable metrics for analysis.
- It's important to notice that since **Pandas Profilling doesn't discriminate data types**, **the correlation matrix considers numerical and categorical features**, filtered later by us. In other words, pandas profiling considers variables with numbers as numerical features, however, using categorical variables encoded in numbers to compute correlations is one of the worse mistakes one can make. For example, in the product matrix only the last feature is a numerical variable.
- Using this library not only sensitized us to look more in depth at each feature and determine their data types considering their meanings, but also that this library shouldn't be used for accurate and "by-the-book" EDA. However, **simply as a first glance approach**, we can say that **this library is usefull**!





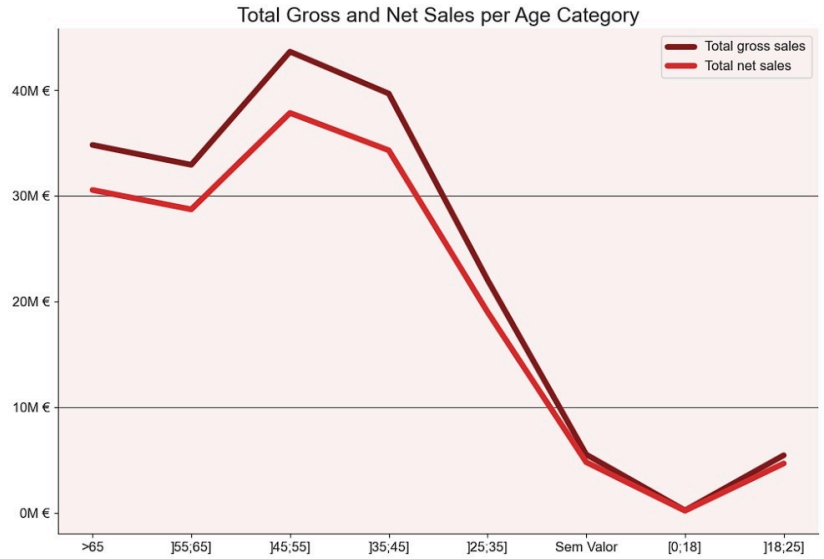


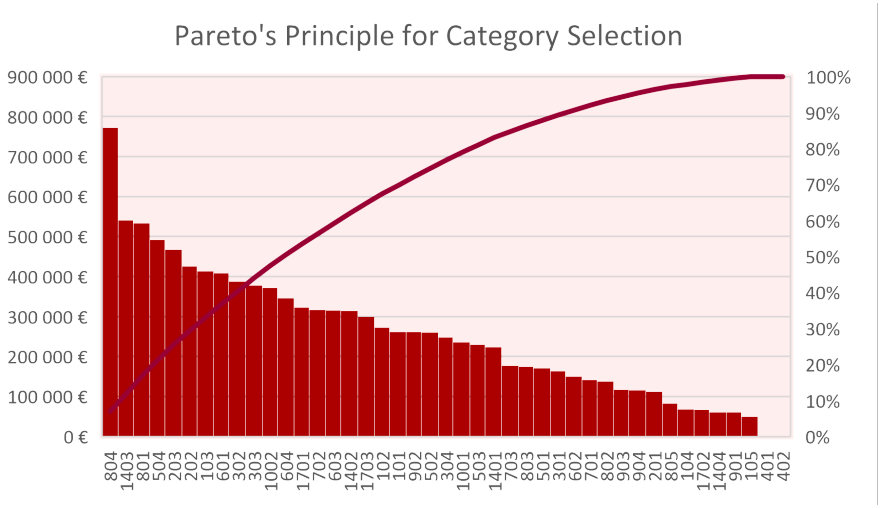
2.2.2 EDA - Second Phase

The main focus of the second phase was to find very specific insights about how to tackle one of SONAE MC's additional business requirements, which was creating a recommendation system for a very specific use case, customer segment and/or set of subcategories.

- The first step done was to identify a specific segment of customers to work with, by creating several line plots to assess gross and net sales across all the segments of several segment features, from lifestage to age segments. Although for all other segment features any insight stood out, when we looked at the **age segment (SEG_AGE),** it was found a possibly **great opportunity of targeting the "perfect" age segment.** On one hand, older segments don't really have digital literacy to use the Continente app, and those segments possibly are more saturated than others given the sales amount being way higher. On the other hand, younger segments don't have the financial stability required to benefit from a recommendation system. Until 25 years usually people don't have that stability, and from 35 years onwards the segments appear to be more saturated. So, the choice was from **25 to 35 years**, with is exactly the segment that is in the middle of the sales gap shown in the line plot on the right;

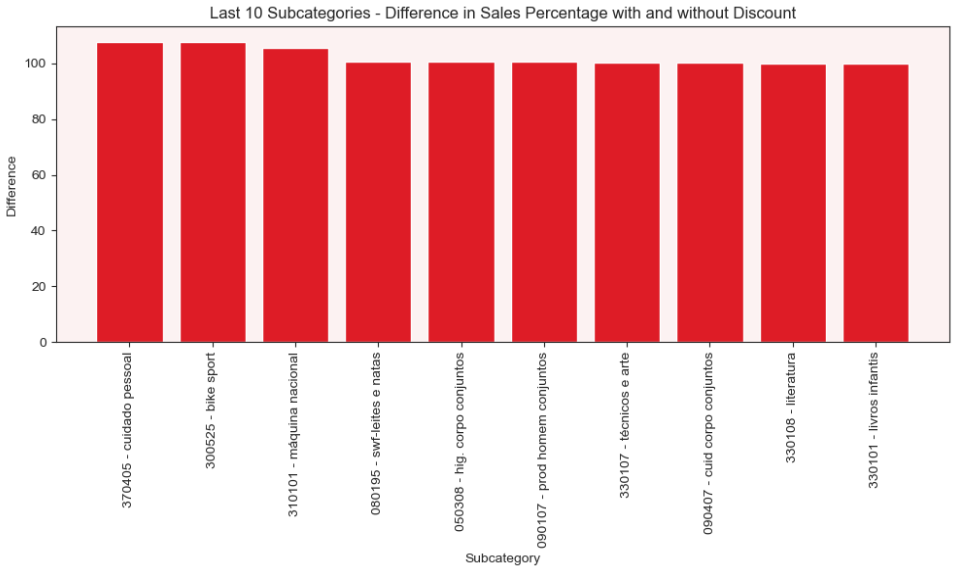
- When it comes to subcategories, a commonly effective method for determining the most significant products or subcategories is **Pareto's Law**, which suggests that 80% of the outcomes typically arise from 20% of the available sources. Thus, we computed a Pareto diagram to transition from a department-level analysis to a more detailed examination of subcategories. In this instance, the diagram revealed that the "Alimentar" category had the tallest bar, and then we were able to select the appropriate categories as the curve reached 80%. Then, we just looked at the x-axis and saw what were the categories we were going to use, at that 80% level.

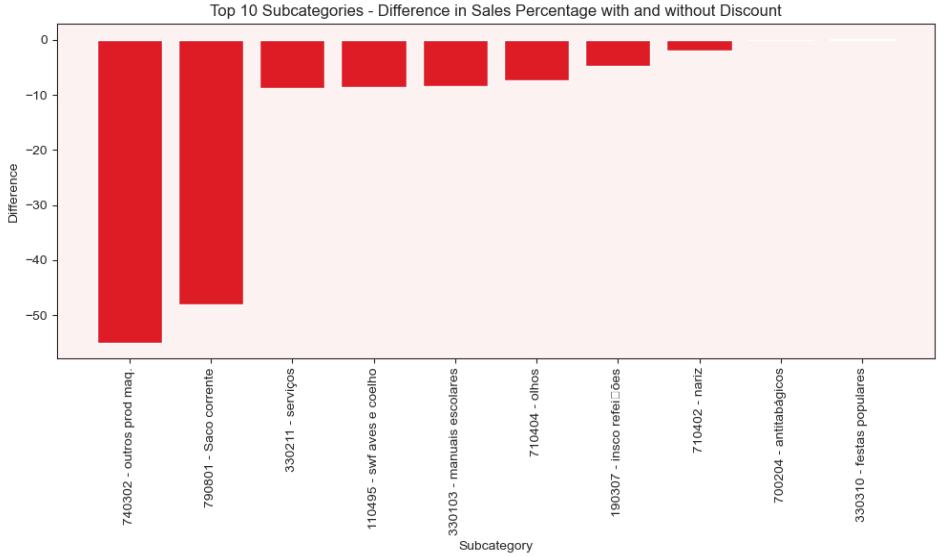




2.2.3 EDA - Third Phase

- An emphasis was placed on analyzing the **transaction table** in order to gain insights into its distribution, identify patterns, and understand the underlying data;
- In light of the outlined objectives, a decision was made to conduct a **study on transactions** involving discounted and non-discounted products. This study specifically focuses on three types of discounts: **product discount, direct discount and transaction discount**;
- The analysis encompassed an examination of whether there were products exclusively sold with a discount or the opposite scenario. Moreover, the analysis considered both the **quantity of products sold and the net revenue** as crucial factors;
- During the analysis, filters based on **subcategory, time space and age** category were applied. Additionally, **three new columns** were generated to display the percentages of items sold with a discount, without a discount and the difference between these two columns in terms of both net revenue and quantity;
- Ultimately, it was observed that there exist products and subcategories that are **exclusively sold or predominantly sold with a discount**. Conversely, there are also products that are exclusively sold without any discount. Exploring the sales potential within these groups could prove beneficial, although additional data pertaining to the promotion periods of these items would be necessary for further analysis.
- Analyzing the graphs, we observe that the subcategories 'Cuidado Pessoal', 'Bike Sport' and 'Máquina Nacional' show a **significant sales trend** when discounts are applied. However, it is important to point out that some subcategories have an opposite behavior. Specifically, the subcategories 'Outros Prod. Maq.' and 'Saco Corrente' present a **negative difference between the percentage of sales with and without discount.**



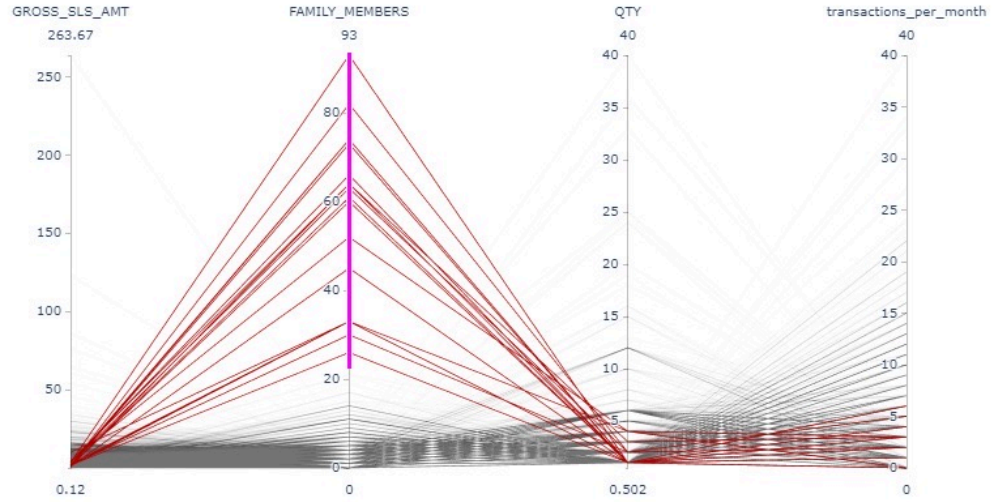


2.3 Verify Data Quality

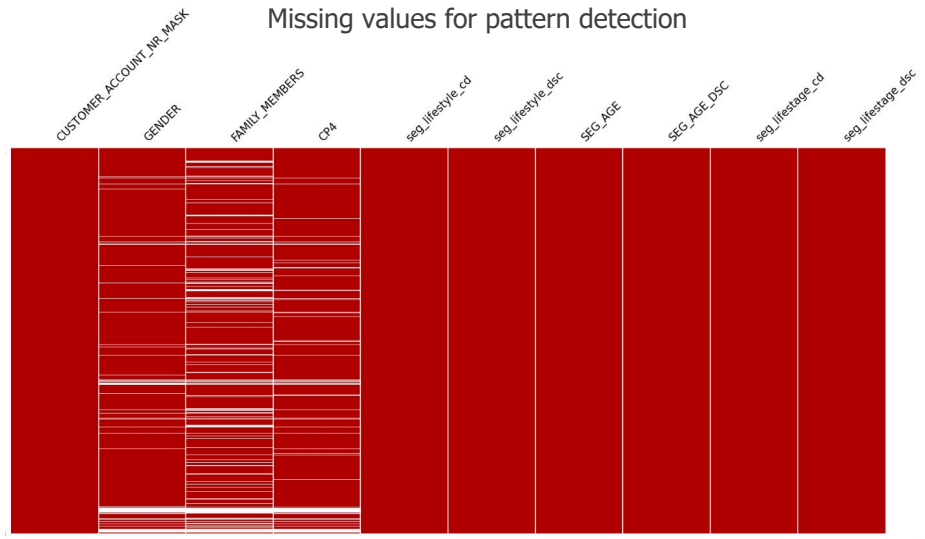
The main data quality key-points identified in the different datasets are presented below.

Accuracy	➢ The data is an accurate representation of the real world, even though there are negative values. ➢ These negative values primarily correspond to product returns or refunds, indicated by the variables QTY (quantity) and GROSS_SLS_AMT (gross sales amount).
Completeness	➢ There are missing values in the dataset, specifically within the Customer dataset, giving more emphasis to the **FAMILY_MEMBERS** variable
Consistency	➢ The **FAMILY_MEMBERS** column contains **outliers** that exhibit low values for both QTY and GROSS_SLS_AMT, which is unexpected since these values would typically be higher for households with many people.
Timeliness	➢ The data spans from January 2021 to December 2022.
Interpretability	➢ The data was straightforward and easy to interpret, aided by the metadata provided by SONAE MC.
Believability	➢ Data was retrieved from the SONAE MC's infrastructure.

Data inconsistency in the FAMILY_MEMBERS variable



Missing values for pattern detection



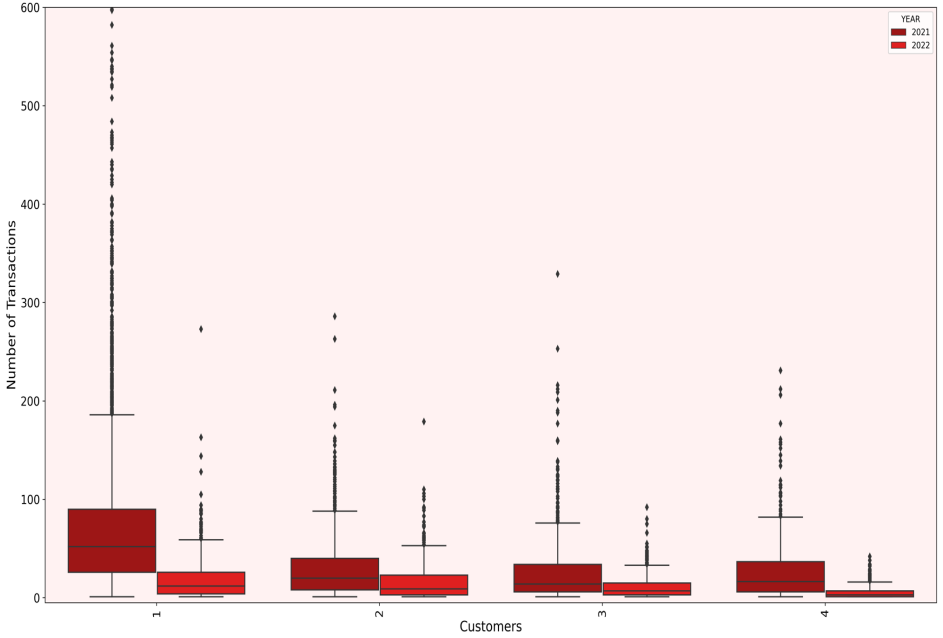
3. Data Preparation

3.1 Select and Clean Data

The following list of points corresponds to the data cleaning and selection process performed:

1. The **Location dataset has been removed** given SONAE MC's additional business requirements identified in the business understanding step;
2. We **dropped 27 columns** considered not relevant for the predictive question (after creating the final dataset, with filter methods);
3. For **outliers**, the approach used was DBSCAN (detects noisy data and creates non-convex clusters), in which 2 hyperparameters were defined, and a 3D scatterplot was created for a visual perception of the outliers. Additionally, to detect outliers, boxplots were used and those values were replaced by the mean value of the corresponding column;
4. As for **missing values**, k-NN was used for Gender and Family Members (the latter being discretized to categorical);
5. For **redundancy**, in terms of **filter methods**, a Pearson correlation matrix was created for the numerical variables, and chi-squared test of independence was performed with bootstrap sampling, at a 5% significance level, to find associations between categorical variables. For the **wrapper methods,** backward and forward selection were tested/used with different stopping criteria. **VIF** measure was computed **for regression**, to assess multicollinearity.

Quarterly transaction volumes across 2021 and 2022



3.1.1 Select and Clean Data - Outliers

The **DBSCAN** technique was employed for **outlier detection**, and a comprehensive description of this methodology will be provided below.

- DBSCAN is **computationally expensive** but has the capability to detect outliers, labeled as -1. Outliers are data points that do not fit into any cluster, and therefore considered noise or anomalies. DBSCAN's ability to identify outliers is valuable in applications like fraud detection or anomaly detection in network traffic.

- As for **DBSCAN's hyperparameters**, **epsilon** gives the maximum distance between instances for them to be considered part of the same density-based cluster. On the other hand, the **minimum number of instances** specifies the minimum number of points that must fall within the epsilon distance to form a cluster. Adjusting these parameters affects the formation of clusters and the inclusion of noise points. The **distance metric** used was the euclidean distance. These hyperparameters were tested (except the distance metric) and the number of detected outliers recorded, and as expected, **by reducing epsilon and increasing the minimum number of instances, the outliers recorded increased**, like shown in the table on the right.

- To ensure that the clustering process is **not biased by differences in feature scales**, the features used (numerical), created in the construct data phase, were rescaled before applying DBSCAN. **Feature normalization** is the process of scaling the features so their values lie within the same scale. This approach prevents features with larger scales from dominating the distance calculations, ensuring that all features have the potential to contribute equally to the clustering process;

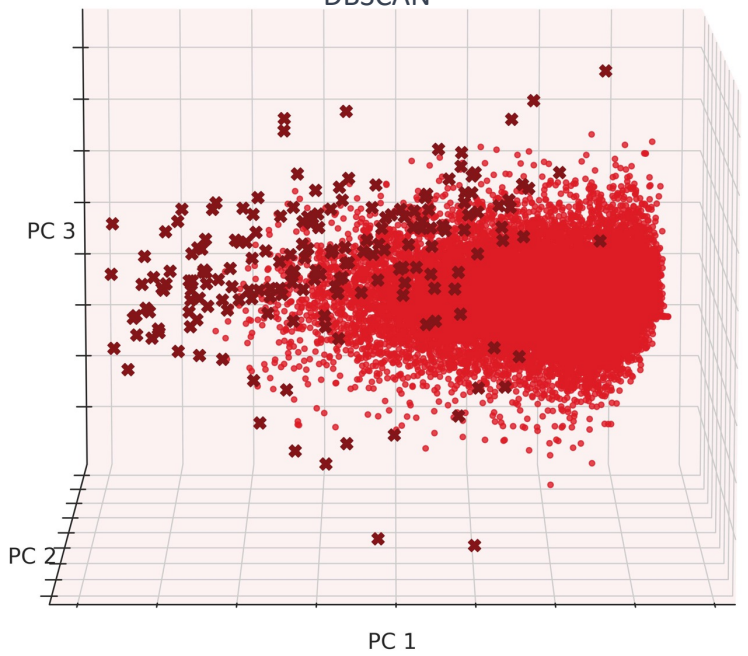
- To handle the large dataset we had, efficiently, **batch processing** was used. The data is divided into smaller subsets or batches, typically containing 50000 instances each. Batch processing reduces memory usage and computational load, making it feasible to apply DBSCAN to large datasets. The results obtained from each batch can be combined or analyzed separately based on specific application requirements.

- On the other hand, **boxplots were also used to detect outliers**. In this approach, with non-normalized data and for each column individually, the values were filtered if their absolute value was larger than 1.5 times the inter-quartile range (IQR), and then those values were replaced by the average of that column. This was the approach chosen given **DBSCAN's computational requirements** and the size of the final dataset. We tested applying DBSCAN with a **batch with 100000 instances and 56 columns** and consumed **more than 70GB of RAM in 5 secs.**

Results with a batch size of 50000 instances

Epsilon	Min_instances	Outliers detected
1.0	500	0
0.4	1000	9465

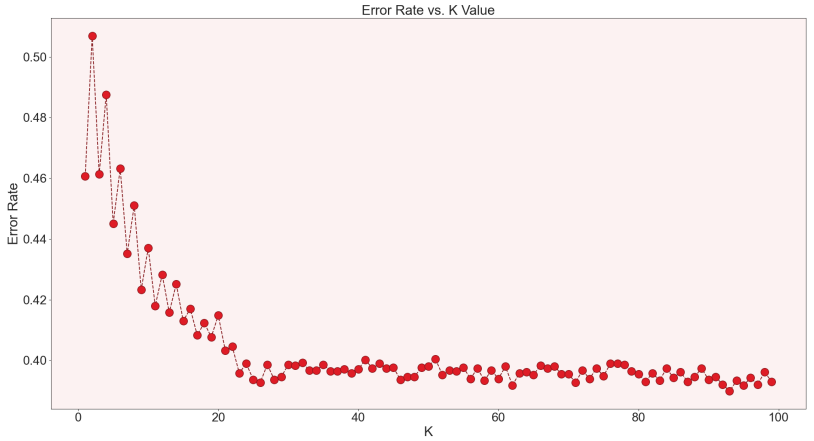
Outlier Detection Results: 3D Scatterplot with DBSCAN

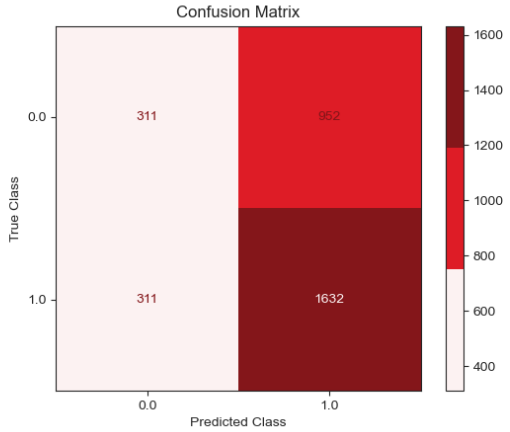


3.1.2 Select and Clean Data - Missing values

Gender

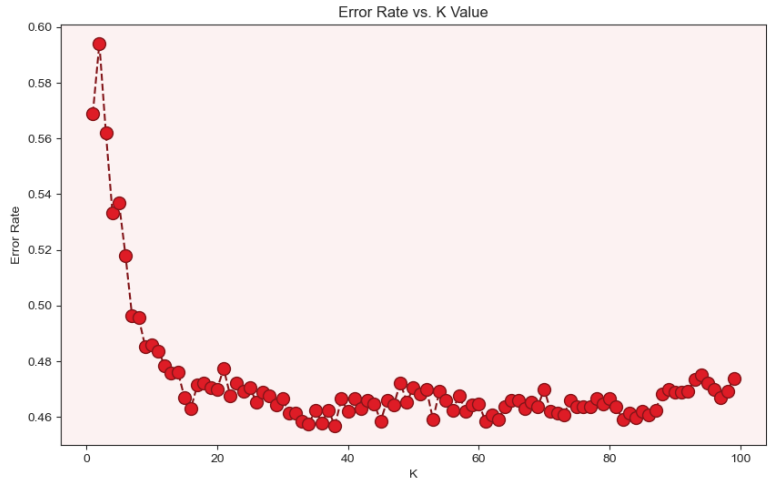
- Multiple models, including SVM, k-NN, and Random Forest were evaluated to make predictions. Among them, **k-NN** yielded the most favorable outcomes by utilizing a hyperparameter (number of neighbors, k) value of 130 neighbors;
- To determine the optimal value for k, an error rate vs k value analysis was employed to assess the best k value, in a validation set. The results in the test set are shown in the heatmaps.

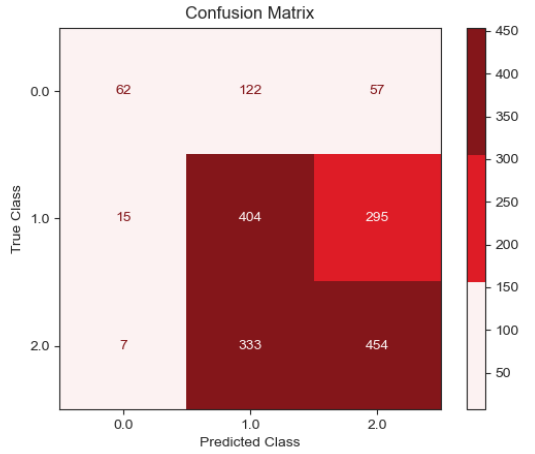




Family Members

- Values **above 9** were deemed inconsistent or incorrect and replaced with NaN, based on outlier detection with boxplots. A regression approach was attempted but did not yield satisfactory results. The problem was, then, reframed as a classification task, so the values were discretized. The value of k used was 38;
- After testing several discretization methods in various models such as k-NN, Random Forest, and Logistic Regression, it was determined that manual discretization achieved the most favorable outcomes when combined with the **k-NN method**.





Note: To apply k-NN, and the other models, and appropriate set up was implemented, from data rescaling, one-hot encoding to training-validation-testing split, to the customer's table, with an additional numerical feature, number of transactions.

3.1.3 Select and Clean Data - Redundancy

In this analysis, three filter methods were employed: the Pearson correlation matrix, the chi-square test of independence, and the VIF measure for multicollinearity. As for wrapper methods, forward and backward selection were tested.

Pearson Correlation Matrix

- The Pearson correlation matrix determines the strength and direction of linear relationships between numerical variables. A threshold of 0.8 identifies highly correlated features, aiding in the detection of correlations and guiding feature selection.

Chi-Square Test of Independence with Bootstrap Sampling

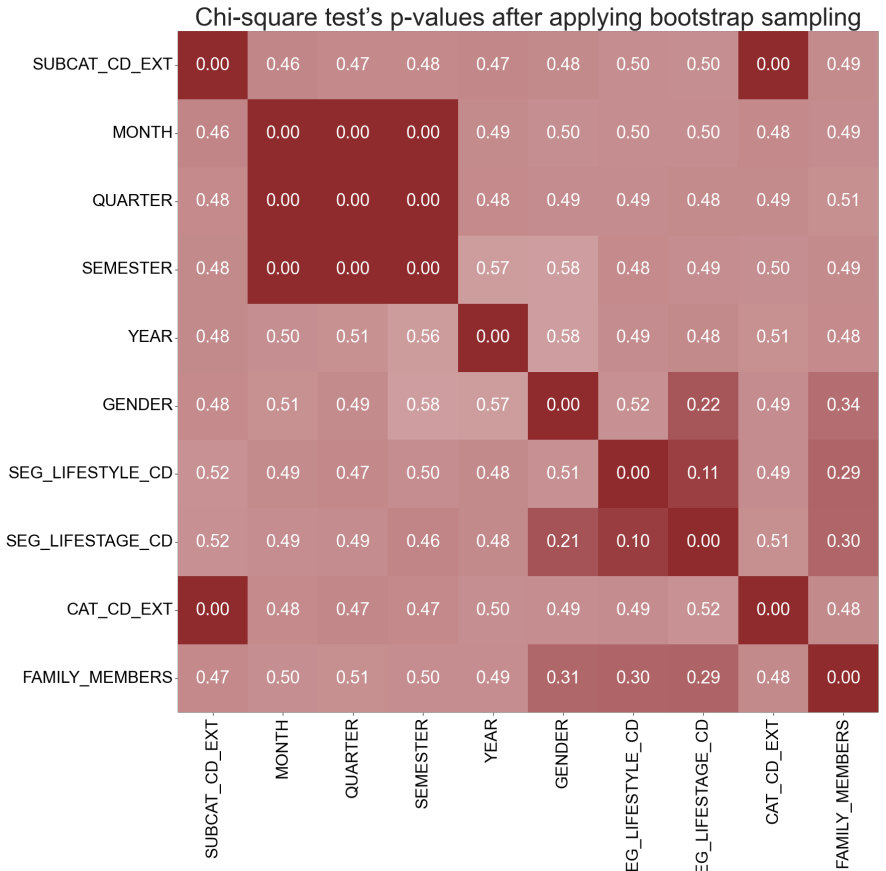
- This test evaluates associations between categorical variables by comparing expected and observed frequency distributions. Bootstrap sampling provides more accurate statistical significance estimation. Some associations lack significance at the 5% level. For bootstrap sampling, 1000 samples with 1000 instances (with replacement) was the structure employed, where for each combination of categorical features the final p-value resulted from the average of all p-values from all the samples of a given combination.

VIF Measure to detect Multicollinearity

- The VIF measure detects multicollinearity by assessing associations between individual features and the combined effect of others. A threshold of 10 indicates high multicollinearity. It helps identify problematic features affecting regression coefficients and predictions. This threshold is an empiric value commonly used.

Wrapper Methods

- Forward and backward selection were the wrapper methods tested. Depending on the classification technique the stopping criteria was different, as well as the wrapper method itself. The stopping criteria implemented was if no improvement of, for example, 0.5%, was recorded in two consecutive iterations of the wrapper method.



3.2 Construct, Integrate and Format Data

The following list of points corresponds to construct, integrate and format of the data:

Construct Data

- **Data discretization** – entropy, equal-width, and manually to FAMILY_MEMBERS.
- **2 datasets were created** - one with **features in days**, for example, last 30/90/180/360 days and another one **by months/quarter/semester/year**;
- After this segregation, the **features** were **divided into 4 categories**:
  - Explicit features, Customer features, Subcategory features and Customer-subcategory features;
- **Creation of records**, based on all possible combinations of customers, subcategories, month and year, given our target granularity (27 M);
- For each client, all combinations prior to the first transaction were filtered (22 M);
- In the end, due to the complexity and size of the data, only the top 1000 customers with the most transactions in the year 2021 were considered (3.4 M).

Integrate Data

- With transactional data, the **Customer table** (PK: Customer_Id) and the **Product table** (PK; SKU) were combined;
- **Aggregated values** were calculated at the characteristics engineering stage - for example, the total quantity purchased by a customer in the last 90 days.

Format Data

- For **categorical variables** the One Hot Encoding (dummy variables) technique was applied and for **numeric variables** the Min-Max (or normalization) and Z-Score (or standardization) techniques were applied.

Explicit Features

- SEG_LIFESTYLE_CD/1/2/3/4/5/6
- GENDER F/M
- FAMILY_MEMBERS_(0, 0)/(1,2)/(3,8)
- FULLDATE, MONTH, QUARTER, SEMESTER, YEAR

Customer Features

- CUST_NUM_TRANSACTIONS_MONTH/QUARTER/SEMESTER/YEAR
- CUST_TOTAL_QTY_BOUGHT_MONTH/QUARTER/SEMESTER/YEAR
- CUST_NUM_UNIQUENT_SUBCAT_MONTH/QUARTER/SEMESTER/YEAR
- CUST_AVG_DAYS_SINCE_PRIOR_TRANSACTION_MONTH/QUARTER/SEMESTER/YEAR
- CUST_AVG_BASKET_SIZE_MONTH/QUARTER/SEMESTER/YEAR

Subcategory Features

- SUBCAT_NUM_TRANSACTIONS_MONTH/QUARTER/SEMESTER/YEAR
- SUBCAT_TOTAL_QTY_BOUGHT_MONTH/QUARTER/SEMESTER/YEAR
- SUBCAT_NUM_UNIQUE_CUST_MONTH/QUARTER/SEMESTER/YEAR

Customer-Subcategory Features

- CUSTSUBCAT_NUM_TRANSACTIONS_MONTH/QUARTER/SEMESTER/YEAR
- CUSTSUBCAT_TOTAL_QTY_BOUGHT_MONTH/QUARTER/SEMESTER/YEAR
- CUSTSUBCAT_AVG_DAYS_SINCE_PRIOR_TRANSACTION_MONTH/QUARTER/SEMESTER/YEAR

3.2.1 Construct Data – Feature Engineering

The following steps were considered/applied during the Feature Engineering phase:

- The **transactional data** was inapropriate for generating recommendations because the granularity of a transaction is different from the target defined. To address this, an aggregated dataset was created, encompassing unique combinations of customer ID, subcategory ID, month, and year. These **recommendations** were intended for individual customers, targeting specific subcategories and specific months. After conducting the Exploratory Data Analysis (EDA), a subset of 7 965 customers and 141 subcategories were selected based on customer scores higher than one. The customer's score is the average number of transactions per month, after it's first transaction recorded in the data;

- The resulting dataset contained a total of 26 953 560 rows. To refine it further, all combinations of month/year rows prior to each customer's first transaction were removed, resulting in the elimination of 22 197 207 rows. The dataset was then filtered to include only the top 1,000 customers with the highest number of transactions in 2021, leaving **3 365 388 rows**;

- The **selected features** encompassed explicit factors such as time, location, and segment, as well as customer-specific attributes like the number of transactions and the number of unique subcategories purchased. Additionally, subcategory-specific features were considered, such as the number of transactions and the number of customers who bought a particular subcategory. Then, customer-subcategory features specify the behaviour of a specific customer with a specific subcategory. So, in a given month, customer feature's values will be the same for all subcategories (for a customer), subcategory feature's values will be the same for all customers (for a subcategory), and customer-subcategory features will always differ because even for one customer, it's preferences change depending on the subcategory, like quantity bought.

- Furthermore, features were computed for two groups of timestamps -month, quarter, semester, and year - and rolling windows that spanned different durations, such as the last 30 days, 90 days, 180 days, and 360 days, although we ended up using as final dataset the one without days given weaker predictive results and a huge bug in some features;

- **Data leakage** issues were encountered during the process, leading to increased complexity in the feature engineering.

3.2.1 Construct Data – Feature Engineering

Within the realm of Feature Engineering, a **clustering** methodology has been devised, incorporating three distinct approaches utilizing existing customer data:

1. K-means: groups similar objects into sets called clusters. The algorithm assigns each data point to the nearest cluster based on their **Euclidean** distance. The objective of **K-means** is to minimize the variance within each cluster, making them as homogeneous as possible. The elbow curve was created to find the best number of clusters (centroids) to use.

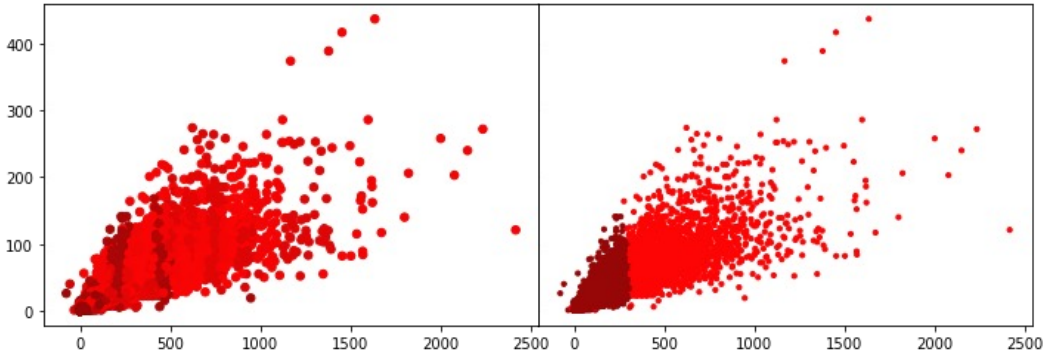
2. Birch Cluster: is a **hierarchical clustering** algorithm that constructs a tree-like structure to represent the data. It divides the data into subclusters and uses a clustering feature called "**branching factor**" to control the size and compactness of the clusters. Birch clustering is efficient for large datasets and provides a compact representation of the data hierarchy;

3. Affinity Propagation: identifies clusters by exchanging messages between data points to find the most representative exemplars. It computes similarity between data points and iteratively updates responsibilities and availabilities. This approach is particularly useful when the number of clusters is unknown, as it automatically determines the optimal number of clusters based on the data itself.

Clustering results

Clustering technique	Silhouette score	Run time (seconds)
K-means	0.1493	0.2
BIRCH Clustering	0.1740	7.1
Affinity Propagation	0.4219	519

Affinity Propagation vs K-means using net revenue and "total_transactions" as plot axes for clustering



Looking at the silhouette scores, we can see that **with affinity propagation we get clusters less overlapped and more compacted**, comparing with K-means and BIRCH Clusters. Notice that the silhouette score goes from -1 to 1, and the higher this score is the more condensed and distinct the clusters are. So, the clusters chosen were the ones given by the affinity propagation technique.

3.2.2 Construct Data – Data Discretization and Balancing

Data Discretization

- To predict the missing values for "**Gender**" and "**Family Members**," a **k-NN** technique of discretization was employed.
- Treating "Gender" as a classification problem yielded satisfactory results, however, addressing "Family Members" as a regression problem did not produce the desired outcomes. As a result, "**Family Members**" was discretized into a classification problem using an entropy-based algorithm and other methods. This first approach resulted in the creation of three bins:
  - Bin 0: [0,2] Bin 1: [3,3] Bin 2: [4,8]
- As a **second approach**, equal-width binning was employed, resulting in the creation of the following bins:
  - Bin 0: [0,2] Bin 1: [3,5] Bin 2: [6,8]
- Given the results of the k-NN with these two discretization methods, a **manual discretization** was applied, considering the behavior of families with more or fewer children. Three bins were created with the following distribution:
  - Bin 0: [0,0] Bin 1: [1, 2] Bin 2: [3,8]
- For the DM phase, the **manual discretization was chosen**, and in fact the models found this discretization useful as the wrapper method deployed found the categories of this feature relevant for the predictions.

k-NN model with Entropy based results

	Precision	Recall	F1-Score	Support
Results	0.36	0.36	0.35	2623

k-NN model with Equal-Width results

	Precision	Recall	F1-Score	Support
Results	0.63	0.48	0.47	4896

k-NN model with Manual Discretization results

	Precision	Recall	F1-Score	Support
Results	0.64	0.62	0.63	6994

Data Balancing

- To address the classification problem, the **dataset needed to be balanced**. To achieve this, three approaches were implemented: random undersampling, random oversampling and SMOTE.
- These three approaches were tested with a sample dataset to choose the one to be used in the final dataset. Now, because the classification results were approximately the same, and given that with both oversampling techniques we would be training with +700k instances per iteration (explained in modeling), as well as choosing hyperparameters and features, **random undersampling was chosen**.

3.2.3 Data Integration

The following points outline the considerations and steps taken during the data integration phase:

Data Integration

- **Multiple tables (customer, product, transaction)** were combined during the data integration process in a MySQL database;
- **Inner joins** were used to merge the tables into a single integrated table, excluding the location table;
- The **primary keys** (CUSTOMER_ACCOUNT_NR_MASK and SKU) were utilized to establish relationships between the tables;
- The **transaction table**, containing the integrated data, did not have a relevant primary key for the merging process, but it had foreign keys for the other two columns.
- The integration of the tables enabled the **creation of aggregated features**, providing insights at a higher level of abstraction;
- The aggregated features encompassed **explicit**, **customer-related, subcategory-related, and customer-subcategory-related features**, allowing analysis and exploration of data from different dimensions.

Product	Customer	Transaction
• SKU (CAT) • PRODUCT_DSC (CAT) • UNIT_BASE_CD_EXT (CAT) • UNIT_BASE_DSC_EXT (CAT) • SUBCAT_CD_EXT (CAT) • SUBCAT_DSC_EXT (CAT) • CAT_CD_EXT (CAT) • CAT_DSC_EXT (CAT) • BIZ_UNIT_CD_EXT (CAT) • BIZ_UNIT_DSC_EXT (CAT) • DEPARTMNT_CD_EXT (CAT) • DEPARTMENT_DSC_EXT (CAT)	•CUSTOMER_ACCOUNT_NR_MASK (CAT) •GENDER (CAT) •FAMILY MEMBERS (NUM) •CP4 (CAT) •SEG_LIFESTYLE_CD (CAT) •SEG_LIFESTYLE_DSC (CAT) •SEG_AGE (CAT) •SEG_AGE_DSC (CAT) •SEG_LIFESTAGE_CD (CAT) •SEG_LIFASTAGE_DSC (CAT)	•TIMEKEY (CAT) •TRANSACTION_ID_MASK (CAT) •CUSTOMER_ACCOUNT_NR_MASK (CAT) •LOC_BRAND_CD (CAT) •LOCATION_CD (CAT) •POS_TP_CD (CAT) •SKU (CAT) •PRODUCT_KEY (CAT) •QTY (NUM) •NET_SLS_AMT (NUM) •GROSS_SLS_AMT (NUM) •PRODUCT_DSCNT_ISSUE (NUM) •TRANS-DSCNT_RATIO (NUM) •DIRECT_DSCNT_AMT (NUM)

3.2.4 Data Formatting

The following points outline the considerations and steps taken during the data formatting phase:

Data Formatting

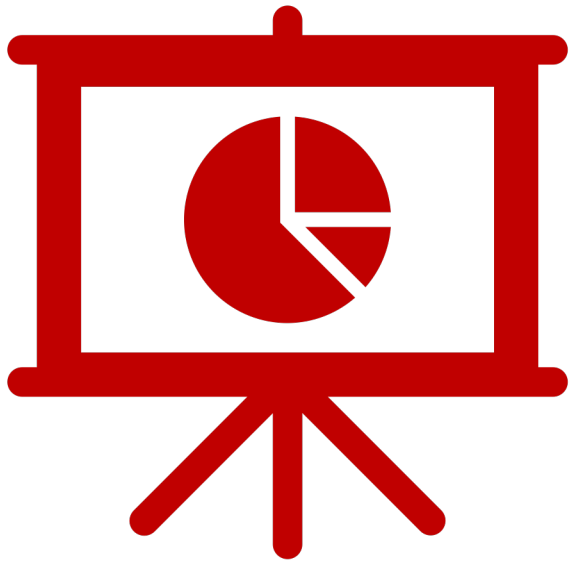
- The **formatting of the data** involved processing both numerical and categorical features;
- **Numerical features** were rescaled using two methods: **z-score (standardization) and min-max (normalization)**;
  - The **z-score rescaling** method standardized the numerical features by subtracting the mean and dividing by the standard deviation;
  - The **min-max rescaling** method normalized the numerical features to a range between 0 and 1, ensuring consistency with the binary values computed for categorical variables;
- Categorical features were transformed into **dummy variables**. This means that each category within the categorical features was assigned its own specific column;
- By applying the **min-max method** to the numerical variables, all features, both numerical and categorical, were normalized to the same numerical interval, 0 to 1;
- The **numerical features** considered for rescaling included the aggregated variables derived, mentioned on the Data Construct phase;
- The categorical features considered for **dummy variable** computation were the explicit features previously mentioned, and some others.

Normalization using Min-Max scaling

SUBCAT_TOTAL_QTY_BOUGHT_QUARTER	SUBCAT_TOTAL_QTY_BOUGHT_QUARTER (min-max)
4304	0.15
8444	0.30
13130	0.47
4200	0.15
9015	0.32

One-hot encoding

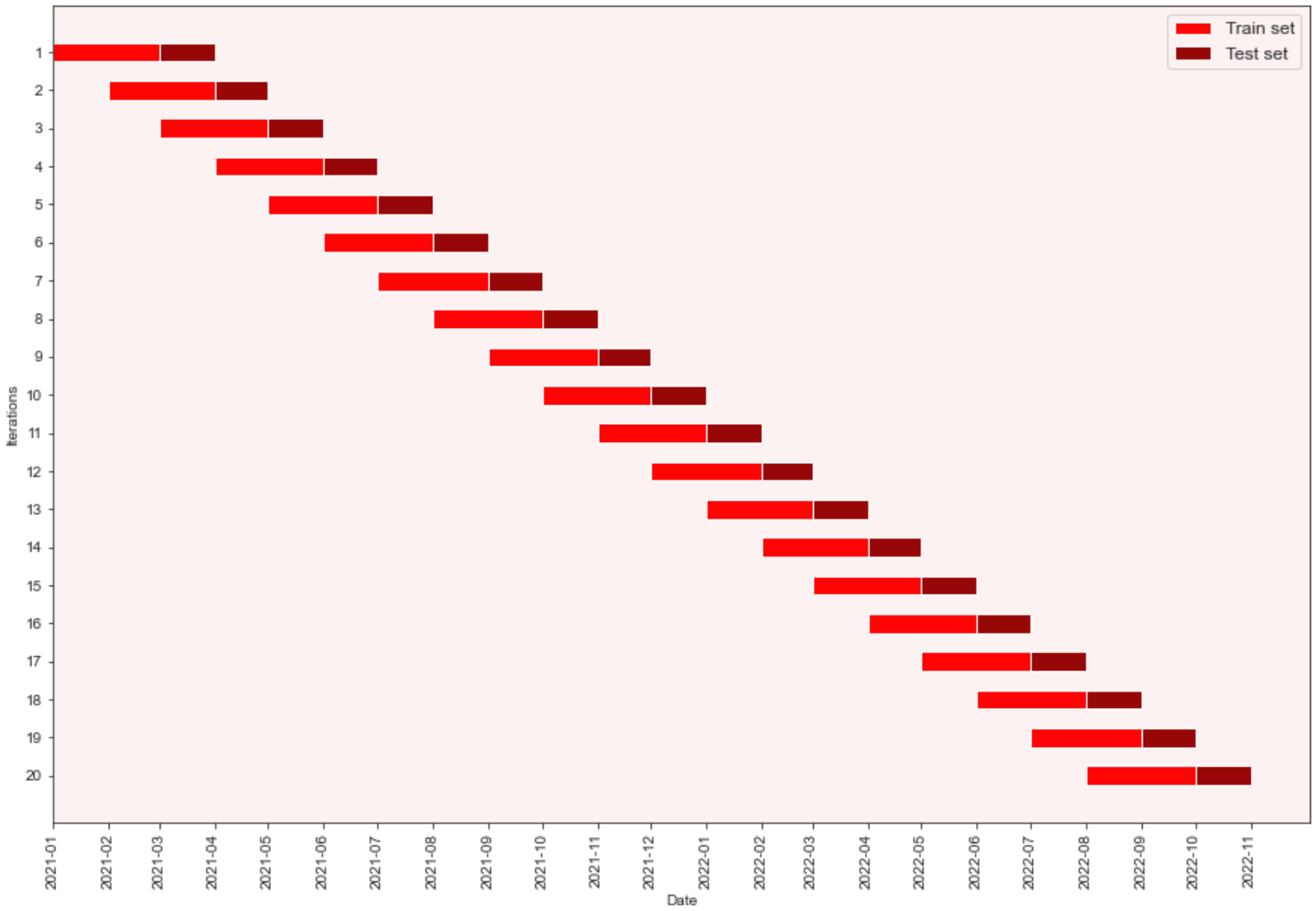
CAT_CD_EXT	CAT_CD_EXT_804	CAT_CD_EXT_1403	CAT_CD_EXT_702
804	1	0	0
1403	0	1	0
1403	0	1	0
702	0	0	1
804	1	0	0



4. Modeling

4.1 TECHNIQUES & TEST DESIGN

- **Time-series Cross Validation – Range (01-2021, 11-2022)**



75% Train

25% Test

For each iteration:

- **Three months** are used for <u>training</u> and the **next month** is used for <u>testing</u>. At the end of the iteration the time window advances to the next month;
- The **hyperparameter tunning** and the **selected features** change each iteration.
- **5-Fold Cross Validation** -> <u>5/10 possible random combinations of hyperparameters</u>.
- Validation set retrieved from part of the training set.

4.2 BUILD & ASSESS MODELS - Classification

Naïve Bayes - Based on Bayes' theorem, Naive Bayes is a probabilistic classification method. It assumes that, given the class labels, the characteristics in a dataset are conditionally independent of one another, therefore being naïve. Naïve Bayes determines the likelihood that a data point belongs to each class and places it in the class where that likelihood is highest.

Artificial Neural Networks - The structure and operation of biological neural networks serve as the basis for computing models known as artificial neural networks (ANNs). Artificial neurons coupled to one another and arranged into layers make up ANN's. Each neuron receives input from previous neurons, and then sends the results to the neurons of the next layer. For many different tasks, including classification, regression, and pattern recognition, ANNs are extensively employed.

Gradient Boosting - Gradient Boosting is an ensemble learning technique that builds an additive model by sequentially training weak learners (typically decision trees) in a stage-wise manner. In each stage, the algorithm fits the new weak learner to the residuals (errors) of the previous stage. Gradient Boosting iteratively minimizes a loss function by adjusting the parameters of the weak learners.

Logistic Regression - A logistic function is used to represent the connection between the output of a linear regression and the binary output, for a given threshold. In order to categorize the data, logistic regression calculates the probability of the target class and applies a decision threshold.

Random Forest - Random Forest uses several decision trees to provide predictions. The final forecast is made by averaging or voting the predictions of several trees, each of which is trained on a different random subset of the training data. Random Forests are renowned for their capacity to manage interactions between features and high-dimensional datasets, in a non-sequential manner.

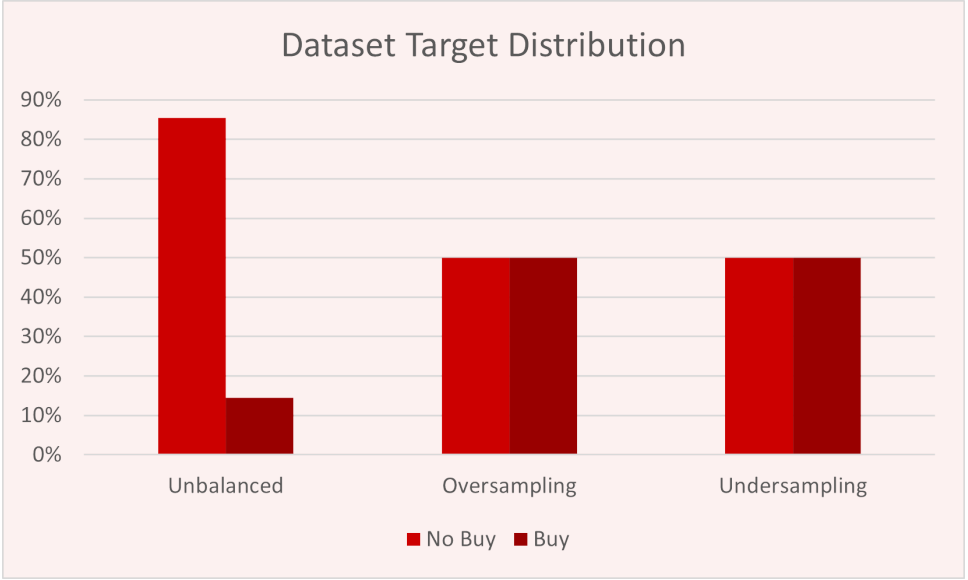
4.2 BUILD & ASSESS MODELS - Classification

Below will be presented key aspects of hyperparameter tuning and handling imbalanced data for target prediction:

- **Hyperparameter tunning**
  - We applied a systematic process with **5-fold cross validation**, in which for each hyperparameter validation the predictive measures were averaged. Afterwards, the best combination is chosen, based on the **f1-score.** Given the high number of possible combinations of hyperparameters, the combinations were defined randomly with a maximum limit of 5/10 possible combinations. We used the f1-score because while using the recall the precision of the models decreased a lot, and the opposite while using the precision;
  - This was done for all models except for **Logistic Regression** and **Naive Bayes**, since these do not have hyperparameters.

- **Target** - If a customer bought a subcategory in the following month
  - Buy –> 1
  - No buy –> 0

- **Imbalanced Data**
  - Undersampling: Reduces instances in the majority class to address imbalanced data;
  - Oversampling: Increases instances in the minority class to balance class distribution;
  - SMOTE: Generates synthetic instances for the minority class to mitigate overfitting and handle complex patterns.



- **Evaluation Metrics** -> F1-Score, Recall, Precision, Accuracy, AUC – ROC and Time.

- **Feature selection:** The exact same approach as in hyperparameter tuning was deployed, with forward selection. Notice that, when entering the training pipeline, we first apply forward selections, and then we do hyperparameter tuning with the selected features.

4.2 BUILD & ASSESS MODELS – Classification

Naïve Bayes

Feature Selection -> FAMILY_MEMBERS, GENDER, Category and CUST_AVG_DAYS

Forward Selection Stopping Criteria -> - 0.5 %

Logistic Regression

Feature Selection -> FAMILY_MEMBERS, GENDER and Category

Forward Selection Stopping Criteria -> - 0.1 %

Artificial Neural Networks

Hyperparameter Tunning -> adaptive learning rate, 1 hidden layer with 256/512 nodes and 20% validation fraction

Feature Selection -> FAMILY_MEMBERS, Category, Year and CUST_SUBCAT_NUM_TRANSACTIONS_SEMESTER

Forward Selection Stopping Criteria -> - 0.5 %

Random Forest

Hyperparameter Tunning -> gini criteria and 500 estimators

No wrapped method was applied

Gradient Boosting

Hyperparameter Tunning -> 500 estimators and 20% of validation fraction

Feature Selection -> FAMILY_MEMBERS, CUSTSUBCAT_AVG_DAYS_SINCE_PRIOR_TRANSACTIONS and GENDER

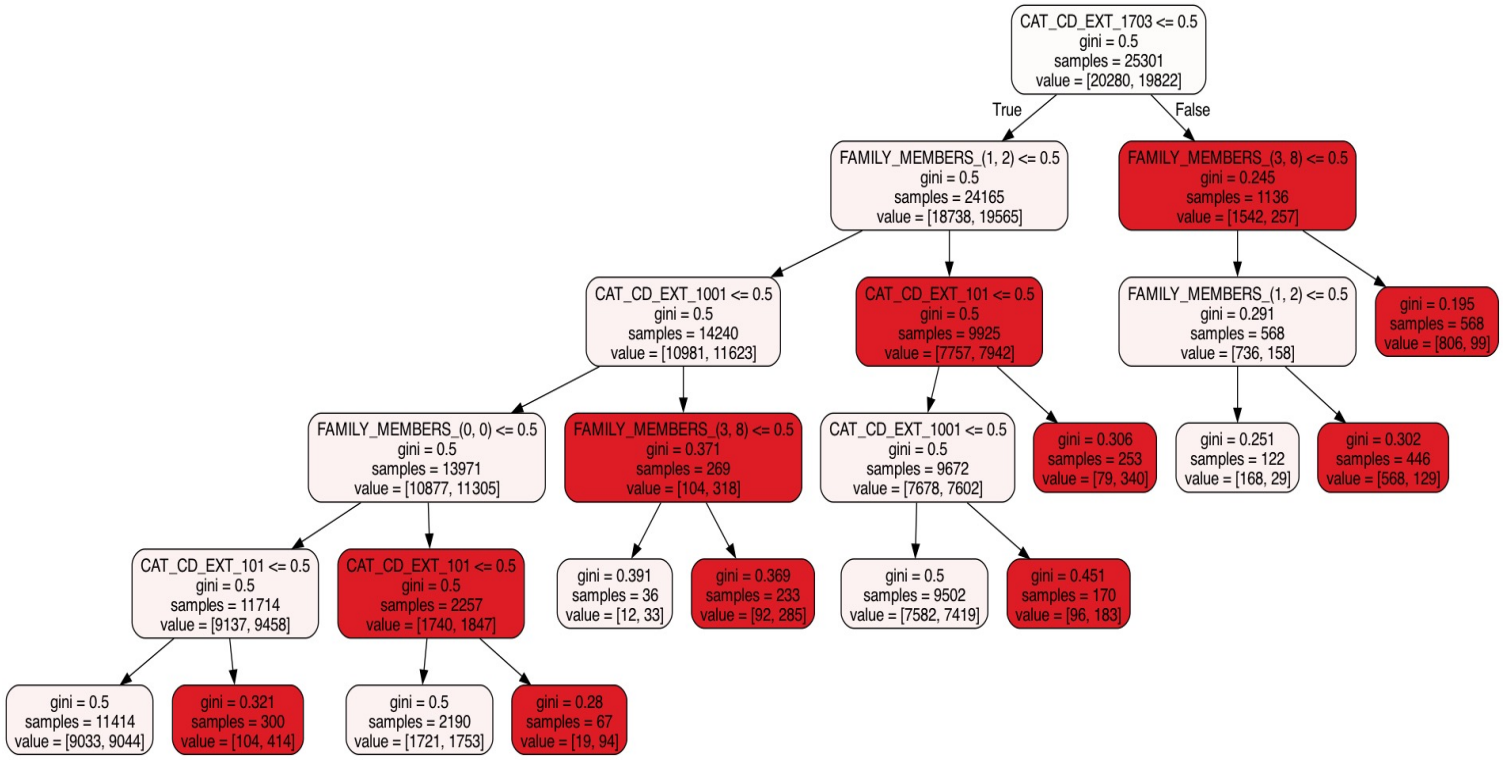
Forward Selection Stopping Criteria -> - 0.5 %

- **A Negative stopping criteria** for forward selection was applied for the algorithm to find "hidden" combinations of features that result in better predictive power, and therefore not being stuck in a possible "local optimum";
- **Random Forest** does not have forward/backward selection because the algorithm already does feature selection;
- The **score threshold** set for all models was **50%**, which satisfied the predefined success criteria.
- Here we are already **showing the features and hyperparameters** that the **pipeline selected the most**, for each ML technique. So, in fact, we used more hyperparameter values, as well as hyperparameters themselves.

4.2.1 White-box model – Decision Tree from RF

In this analysis, we selected a random decision tree, from a total of 500 trees, from an iteration of November 2022. The idea here is to interpret a decision tree the random forest created. This is possible given the fact that a decision tree is a white-box model.

- For this case, the Random Forest algorithm selected features, for example 'FAMILY_MEMBERS_(0, 0)', 'CAT_CD_EXT_101', 'CAT_CD_EXT_1703', 'YEAR_2021', and 'YEAR_2022';
- The decision tree constructed by the Random Forest algorithm started with the feature 'CAT_CD_EXT_1703' as the root node. If the value for that feature and for a given instance is lower or equal to 0.5, the tree follows the **true path**, and if it is greater than 0.5, the **false path** is taken in the subsequent branch. Notice that the false paths are in a vivid red for perception reasons (as well as many other details in the visualizations created);
- For example, starting from the root node, if an instance belongs to the category 1703 then the false path is followed (because 1 <= 0.5 is false), and if for that instance the customer has from 3 to 8 family members, then the probability of guessing wrong the classification target for that instance is 19.5% (gini index), and there're 568 instances that met the characteristics mentioned above in this leaf node.
- In fact, we can see that even when using several features to segment the instances, there're still some leaf nodes with a probability of guessing wrong the target of 50%, the worse possible value for this criteria.



4.3 BUILD & ASSESS MODELS - Regression

Linear Regression -> To model the relationship between a dependent variable and one or more independent variables, a statistical method known as linear regression is used. It assumes that the variables are related linearly and looks for the line that minimizes the sum of squared residuals;

Ridge Regression -> Is a linear regression technique that modifies the OLS objective function by adding a penalty component. The squared coefficient sum plus a regularization parameter make up this penalty term. By reducing the effects of multicollinearity and bringing the coefficient estimates closer to zero, ridge regression can enhance model performance;

Lasso Regression - > Is a statistical method that combines variable selection and regularization by adding a penalty term to the objective function. With L1 regularization, it encourages sparsity by shrinking some coefficients to zero, automatically selecting important variables. It is valuable for high-dimensional datasets, aiding in the identification of relevant variables and enhancing model interpretability.

Elastic Net Regression -> Is a statistical technique that combines the benefits of both Ridge Regression and Lasso Regression. It introduces a hybrid penalty term that includes both the absolute value of the coefficients (L1 regularization) and the squared value of the coefficients (L2 regularization). This allows for variable selection and addresses multicollinearity, resulting in a more flexible and robust regression model;

XGBoost -> Decision-tree based machine learning algorithm that uses the boosting method to improve predictive performance. It combines multiple decision trees to create a more powerful model by sequentially adjusting for the residual errors of previous trees. XGBoost is known for its accuracy and ability to handle different types of data and offers advanced regularization features to avoid overfitting;

Gradient Boosting Regression -> Is a powerful machine learning algorithm that combines weak prediction models, like decision trees, to create an accurate predictive model. It minimizes errors by sequentially optimizing the model using gradient descent, resulting in a robust and effective model capable of capturing complex relationships in the data.

Random Forest Regression -> Another ensemble method that mixes various regression trees to create predictions is the random forest, but now for regression. A random subset of the training data is used to construct each tree in the random forest, and the predictions from all the trees are averaged to provide the final prediction. The variable importance measurements provided by random forest regression are resilient against overfitting, and it can handle nonlinear interactions between variables;

4.3 BUILD & ASSESS MODELS - Regression

Below will be presented key aspects of filtering, hyperparameter tuning and feature selection for target prediction:

- **Hyperparameter tunning**
  - We applied a systematic process with **5-fold cross-validation**, in which the predictive measures were averaged for each hyperparameter validation. Afterwards, the best combination is chosen, based on the **R^2**. Given the high number of possible combinations of hyperparameters, the combinations were defined randomly with a maximum limit of **more than 100 possible combinations.** Because here we have around 24k instances (1000 customers*24 months), we were able to test way more combinations of hyperparameters, comparing with the classification problem;
  - This was done for all models except for **Linear Regression**, since it does not have hyperparameters.

- **Target** - Predict the **next purchase** for a given customer, creating even more personalized promotions for the customers.

- **Feature Selection** - For wrapper methods, several techniques were tested:
  - Select K-Best**:** Selects the K best features based on a relevance metric, such as chi-squared or F1-score;
  - SequentialFeatureSelector (MeanSquaredError): Algorithm that performs sequential feature selection based on minimizing the Mean Squared Error;
  - SequentialFeatureSelector (R-squared Error): Variant of SequentialFeatureSelector that uses R-squared Error as the criterion for the wrapper methods. This was the one implemented in the final ML pipeline, with forward selection;
  - Permutation Feature Importance: Evaluates the importance of each feature by randomly shuffling its values and measuring the resulting impact on model performance metrics.

- **Evaluation Metrics** -> R^2, RMSE, MAE and Time.

4.3 BUILD & ASSESS MODELS - Regression

Linear Regression	Ridge & Lasso Regression	Elastic Regression	XGBoost	Random Forest Regression	Gradient Boosting Regression
Feature Selection-> FAMILY_MEMBERS, CUST_NUM_TRANSACTIONS_MONTH and CUST_SUBCAT_TRANSACTIONS_MONTH	**Hyperparameter Tunning ->** Range (0.001, 1, 0.001) –> 0.996	**Hyperparameter Tunning ->** Range (0.001, 1, 0.001) –> 0.996	**Hyperparameter Tunning ->** L. Rate (0.01), Max depth (3), Max estimators (500)	**Hyperparameter Tunning ->** Samples Split & Leef Range (1,30,1) -> (22, 18)	**Hyperparameter Tunning ->** Range(0.001, 1, 0.001) -> 0.671
	Feature Selection-> GENDER_M, CUSTSUBCAT_NUM_TRANSACTIONS_YEAR and CUST_AVG_BASKET_SIZE_MONTH	**Feature Selection->** CUST_NUM_TRANSACTIONS_MONTH, CUST_AVG_BASKET_SIZE_MONTH and CUSTSUBCAT_NUM_TRANSACTIONS_YEAR	**Feature Selection->** SEG_LIFESTAGE_2, GENDER_F and CUST_NUM_TRANSACTIONS_MONTH	**Feature Selection->** FAMILY_MEMBERS_(0, 0), FAMILY_MEMBERS_(1, 2) and CUST_AVG_BASKET_SIZE_MONTH	**Feature Selection->** CUST_NUM_TRANSACTIONS_MONTH, SEG_LIFESTAGE_CD_1 and FAMILY_MEMBERS_(3, 8)

- Although Random Forest inherently incorporates feature selection, and the absence of a wrapper method for the classification problem, **forward selection** was tested for the regression problem;
- The dataset for this problem (regression) was smaller, with around 24,000 instances, allowing for a practical examination of numerous hyperparameter combinations. For instance, more than 300 combinations were tested for the random forest.
- Here we are already **showing the features and hyperparameters** that the **pipeline selected the most**, for each ML technique. So, in fact, we used more hyperparameter values, as well as hyperparameters themselves.

4.3.1 White-box model – Linear Regression

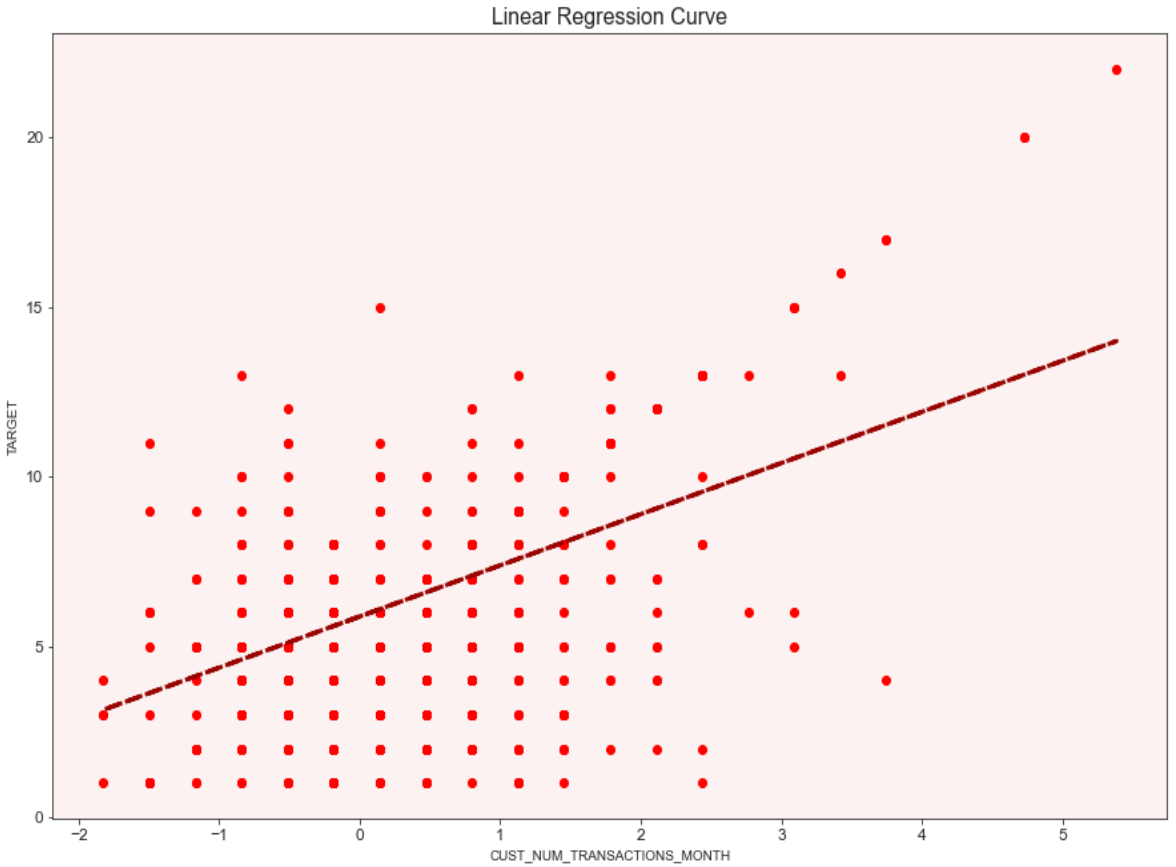
CUST_NUM_TRANSACTONS_NEXT_MONTH (TARGET) $= 5.888 + 1.505 * \boldsymbol{CUST\_NUM\_TRANSACTIONS\_MONTH} + (-0.599) * CUST\_AVG\_DAYS\_SINCE\_PRIOR\_TRANSACTION\_YEAR + 0.020 * CUSTSUBCAT\_NUM\_TRANSACTIONS\_QUARTER + 0.012 * CUSTSUBCAT\_NUM\_TRANSACTIONS\_SEMESTER + 0.012 * CUSTSUBCAT\_AVG\_DAYS\_SINCE\_PRIOR\_TRANSACTION\_QUARTER + 0.008 * CUSTSUBCAT\_AVG\_DAYS\_SINCE\_PRIOR\_TRANSACTION\_SEMESTER + 1.000 * SEG\_LIFESTAGE\_CD\_1 + (-0.429) * SEG\_LIFESTAGE\_CD\_2 + (-0.723) * SEG\_LIFESTAGE\_CD\_3 + (-0.442) * SEG\_LIFESTAGE\_CD\_4 + (-0.321) * SEG\_LIFESTAGE\_CD\_5 + 0.915 * \boldsymbol{SEG\_LIFESTAGE\_CD\_6} + 0.054 * \boldsymbol{FAMILY\_MEMBERS}\_(3,8)$

- The coefficients in a regression model indicate the direction of the effect on the dependent variable and can be either positive or negative. A positive coefficient suggests a **positive association**, implying that an increase in the relevant independent variable corresponds to an increase in the target. Conversely, a **negative coefficient** indicates that an increase in the independent variable is associated with a decrease in the target;

- By examining the coefficients, we can determine the relative importance of each variable. Certain variables, such as **CUST_NUM_TRANSACTIONS_MONTH,** or even **SEG_LIFESTAGE_CD_6**, have a more relative impact compared to all the other features;

- Among all the family segments, the variable **FAMILY_MEMBERS_(3,8)** is assumed to exert the greatest influence on the model because it was the only age segment selected.

- Now, we should be very careful with these coefficient because we can't say that the target would be 7.393 if the number of transactions in the current month is 1, because this model used normalized data.



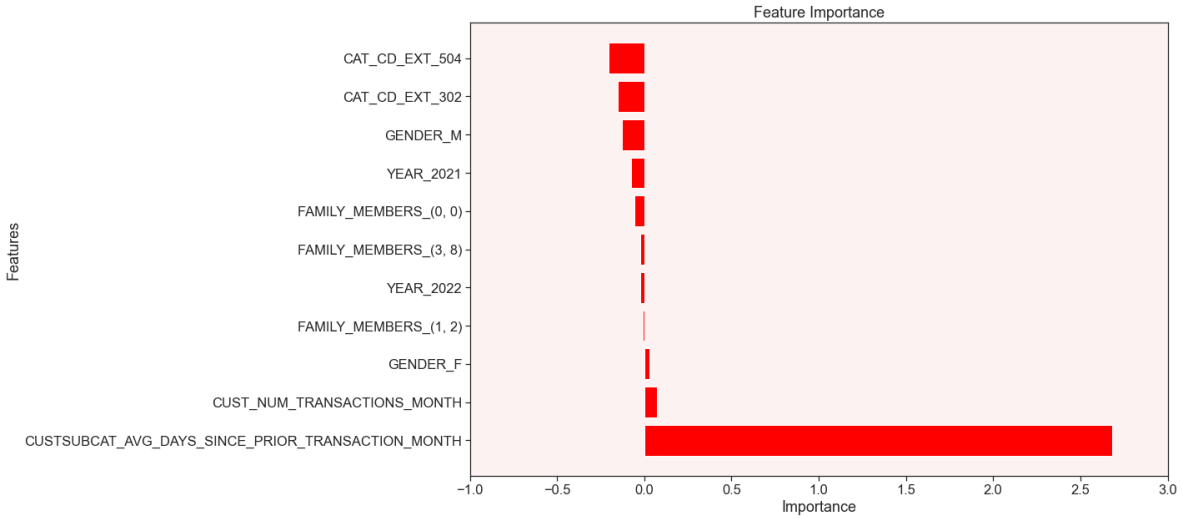
Note: Although it is a multiple linear regression, for visualization purposes we used **only the slope of the feature with greater importance**.

4.4 Feature Importance – Classification & Regression

To determine feature importance across all iterations of the time-series cross validation, the top 11 features with the highest mode were selected, and then they were ordered by the highest average of feature importance.
It should be noted that these feature importance values aren't associated with a specific iteration, meaning that these values don't show if in a given month one specific feature becomes on of the most important ones.

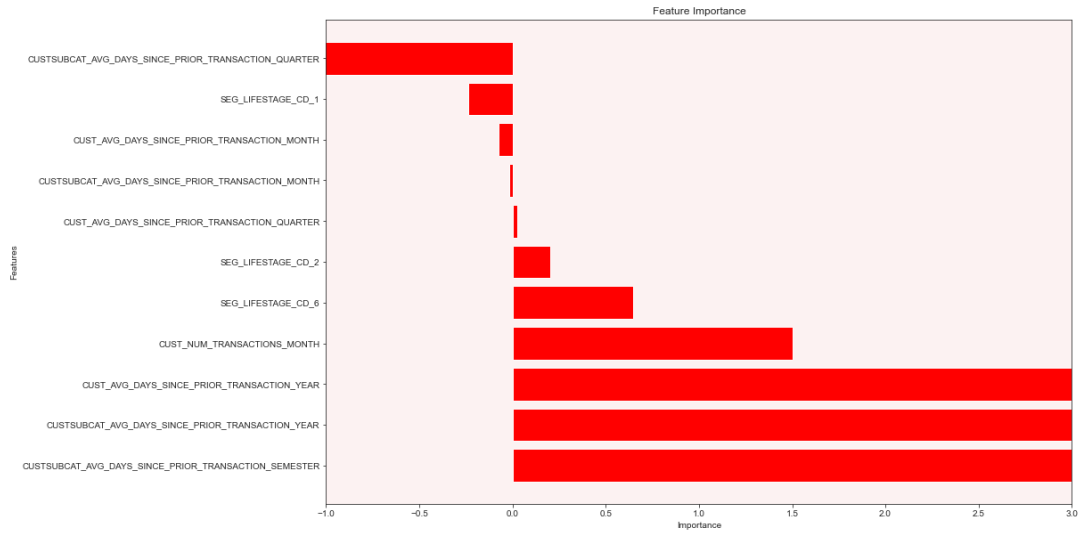
Classification

In the feature importance analysis for the **Classification** problem, the **Random Forest Classification** model was employed. Notably, there is a significant disparity observed in the importance of the features, particularly in the last feature illustrated in the plot below, which possesses a notable feature importance value of 2.5. This outcome emphasizes that the **last feature** holds the **highest degree of significance** among all the features considered. So, we can see that the predicts are very influenced by that last feature.



Regression

In the Regression problem, the feature importance analysis was conducted using the **Elastic Net** Regression model. Notably, when compared to the classification problem, the importance of the features is more pronounced, indicating a higher degree of significance. Furthermore, it is noteworthy that among all the features, the **last three features** stand out as the **most important ones**, displaying a greater impact on the regression model's predictions.



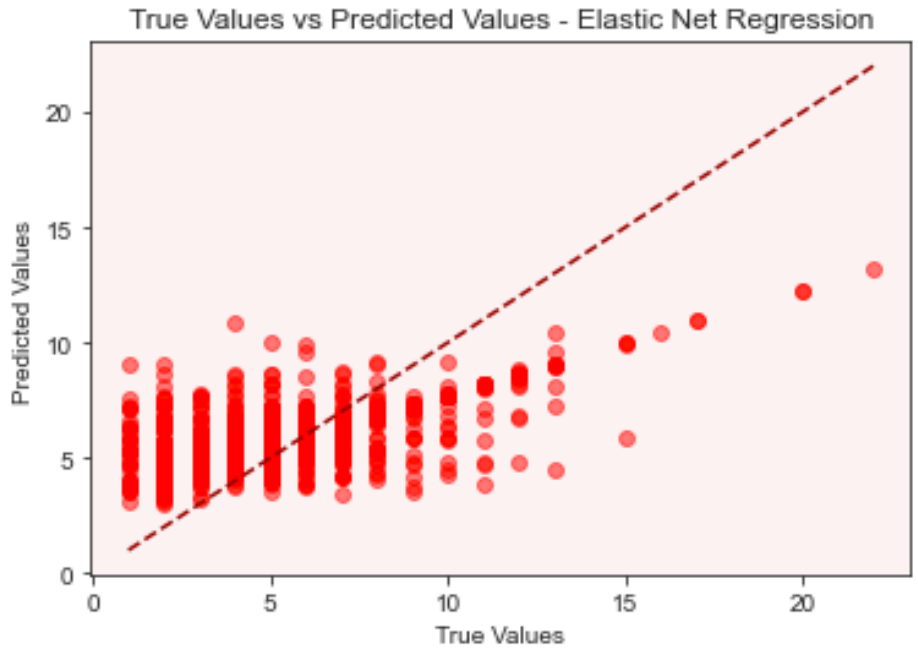
5. Evaluation

5.1 Evaluate Results - Regression

Model	R^2 Mean	MSE Mean	RMSE Mean	MAE Mean	MAPE Mean	Time *
Linear Regression	0,25	6,27	2,50	1,76	0,47	1,2
Ridge Regression	0,25	6,24	2,49	1,75	0,46	1,8
Lasso Regression	0,26	6,19	2,48	1,76	0,47	1,54
Elastic Net Regression	0,26	6,18	2,48	1,76	0,47	1,51
Gradient Boosting Regression	0,24	6,37	2,52	1,76	0,47	18,35
XGBoost Regression	0,24	6,38	2,52	1,76	0,47	12,18
Random Forest Regression	0,25	6,26	2,5	1,76	0,47	67,24

* Time in minutes, for all iterations

- All the tested methods **succeeded** in meeting the specified success criteria (R2 >= 0.15);
- Even though the results have been achieved, t**here is room for improvement**, indicating the need for a dedicated feature engineering process tailored to regression in order to attain the desired target outcomes;
- We can see that, overall, the models are able to predict 25% of the target's variability, by looking at the R squared. Also, on average, the predicted values are 47% away from the real values (MAPE).
- In our opinion, the **lack of sufficient data points or the presence of outliers** might have hindered the effectiveness of the tested methods and feature selection, thereby impacting the ability to achieve better results.



5.1.1 Evaluate Hyper Parameters - Regression

To enhance our understanding of the regression model's behavior, an approach was employed where **one hyperparameter was fixed at a time**, allowing us to observe the resulting impact on the predictive power of the models:

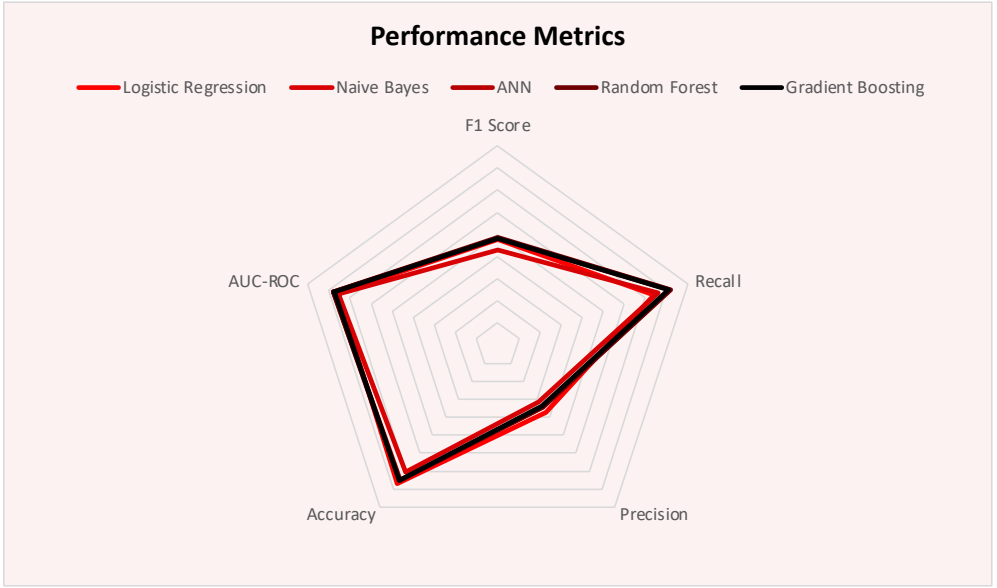
- In both **Lasso** and **Ridge** regression, the selected hyperparameters remained constant throughout all the iterations. Their impact on the results consistently remained the same, although in some iterations, the outcomes were better than in others;
- In **Random Forest**, the optimal values for the two trained hyperparameters, **min_samples_leaf** and **min_samples_split**, are 18 and 22, respectively. The selection of these values in iterations consistently led to a substantial improvement in the results;
- In **Gradient Boosting**, among all the trained hyperparameters, **alpha**=0.671 consistently led to better results when it was selected in the iterations;
- It should be noted that the utilization of a **sliding window** technique resulted in a wide range of selected hyperparameters, thereby contributing to an increased variability of the data.

5.2 Evaluate Results - Classification

Model	F1 Score	Recall	Precision	Accuracy	AUC-ROC	Time *
Logistic Regression	0,4773	0,7387	0,3713	0,7725	0,7591	12
Naive Bayes	0,4292	0,7597	0,3119	0,7034	0,7525	10
ANN	0,4831	0,8144	0,345	0,7504	0,7768	524
Random Forest	0,4857	0,8179	0,346	0,7529	0,7798	220
Gradient Boosting	0,4807	0,8122	0,3397	0,7512	0,7766	310

*Average time, in minutes

- The **ROC-AUC** results enable us to understand that the models are much better that guess work. This measure considers the model's true-positive and false-positive rate's trade-offs for several thresholds, giving a general measurement of the performance. With a ROC curve we can see those rate's evolution, and possibly find the threshold that best fits the trade-off we're willing to accept**;**
- By looking at those ROC curves and by several tests, we found that 0.5 enables the models to have the best balance between precision and recall. From the start we defined that we're willing to have more false positives because we want to recommend new offers to the customers, but still getting right the vast majority of true positives;
- With all of that in mind, and given the marginal differences, we decided to **select the Random Forest** as our "best model";
- On the other hand, in the next slides we'll go in detail into the predictive measures, as well as into the recommendations, and therefore to test the concepts employed, the team decided to filter this research into a single model, and compare it with the success criteria, the baseline model.



5.2.1 Evaluate Hyper Parameters - Classification

To enhance our understanding of the classification model's behavior, an approach was employed where **one hyperparameter was fixed at a time**, allowing us to observe the resulting impact on the predictive power of the models:

- One of the significant hyperparameters in machine learning methods is the **number of neurons in the artificial neural network**. In the main training, the choice of neurons ranged from 256 to 512, with a preference for 512. However, reducing the number to 32 neurons resulted in slightly worse outcomes. This is expected as fewer neurons imply fewer parameters to optimize, leading to reduced flexibility. The average recall measure decreased to 78.6% compared to the results, while other metrics remained relatively unchanged;

- The **number of estimators** in Random Forest and Gradient Boosting had a significant impact on the predictive power of classification models. These models utilize decision trees for predictions, with Random Forest operating in parallel and Gradient Boosting in a sequential manner. When the number of decision trees is low, both the voting mechanism in Random Forest and the prediction improvement in Gradient Boosting are severely affected, as observed in the experiments. In the case of Gradient Boosting, the recall achieved was 75.2%, with an f1-score of 47.2% and precision of 37.04%. For Random Forest with 50 estimators, the recall was 76.7%, with an f1-score of 47.9% and precision of 36.9%. Interestingly, the precision in these experiments surpassed the precision observed in the results;

These conclusions highlight some of the patterns observed during the time series cross-validation iterations. However, there were additional patterns that emerged throughout the iterations:

- The **choice of the number of neurons**, specifically between 256 and 512, **had a notable impact on predictions**. Opting for 256 neurons consistently resulted in a lower f1-score, typically around 3%-4%, compared to iterations with 512 neurons;
- In a few instances where a batch size of 5000 was combined with 256 neurons, the f1-score decreased by approximately 2% when compared to using a batch size of 1000;
- During hyperparameter tuning, the **adaptive learning** rate was favored in most cases. As a result, the decision was made to **fix the learning rate parameter to adaptive instead of using alternatives like constant**.

5.3 Evaluate Metrics – Analysis of Variance

The analysis presented below is a comprehensive evaluation of model performance using the analysis of variance (ANOVA), delving into the assessment of metrics considering the success criteria of statistically significant improvements:

- After selecting the most promising ML model, the **Random Forest**, it was time to **compare it with the baseline model**, and here, as well as for the recommendations, we'll specify for **November 2022** not only because it's the most recent month to which we can evaluate and compare predictive measures and recommendation, as well as the fact that the customer's behaviour is very different between years and across quarters (presented here), but also because it's a proof of concept.

- Now, a general way to compare these two models would be simply by seeing the predictive measures for each one, and then saying if the ML model chosen to be compared had better predictive measures or not. However, from out perspective, this is too simple given the complexity of the problem, and in fact this approach says nothing about the predictions for a specific segment. On one hand, the differences between the two models may be due to chance, meaning that the **differences between the models may not be significant**. On the other hand, to **assess the model's performance for individual segments**, from lifestyle to gender, a very complex, robust, an complete statistical test would be required, and for that, the choice was an **Analysis of Variance, ANOVA**.

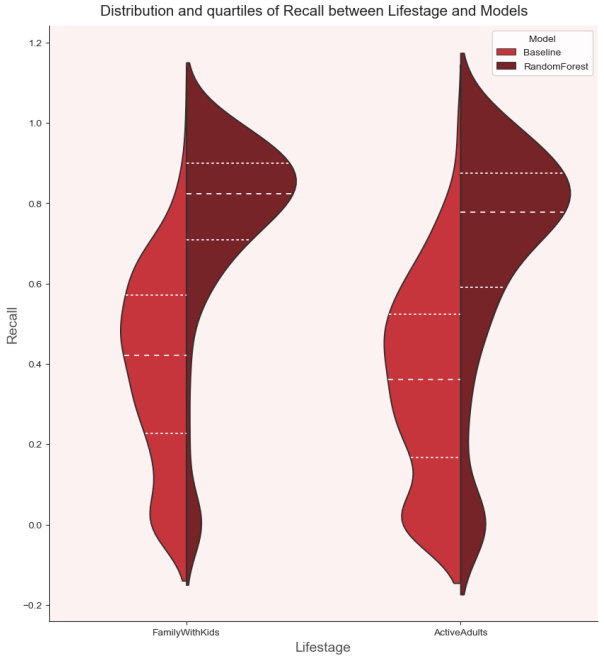
- To make this happen, we calculated the **predictive measures individually for all 1000 customers (subjects).** Then, because the ANOVA requires each cell of the experiment to be balanced, we removed segments and features that didn't have more or less the same number of subjects per segment. Because of that, we ended up dropping the gender, lifestyle, and removed segments from lifestage and family members that had relatively few subjects (2 segments for lifestage and family members were kept).

- There're several types of ANOVA's, however the appropriate format for this experiment is a **Mixed Factorial ANOVA**; factorial because there're **three factors** (Lifestage, Family members, and Model – Random Forest and Baseline), and mixed because we have **between-subjects factors** (independent samples) – Lifestage and Family members –, and **within-subjects factors** (repeated measures) – Model. Notice that one customer only belongs to one segment of lifestage and family members, however he/she is tested on both models. Then, to enter in more detail on the segments and models, simpler ANOVA's were developed.

5.3.1 Evaluate Metrics – Analysis of Variance

	Main ANOVA results (results aggregated from each ANOVA made to each predictive measure)											
	Recall			Precision			F1-score			ROC-AUC		
Effect	F statistic	p-value	significant	F statistic	p-value	significant	F statistic	p-value	significant	F statistic	p-value	significant
Family members	0,8279	0,3632		5,0188	0,0254	*	2,5470	0,1109		1,8565	0,1734	
Lifestage	9,6365	0,0020	*	13,2267	0,0003	*	14,2817	0,0002	*	2,4963	0,1145	
Model	2133,1140	6,80E-226	*	313,0037	3,44E-59	*	136,6059	3,32E-29	*	745,3539	7,16E-116	*
Family members : Lifestage	4,0941	0,0434	*	4,53E-04	0,9830		0,3925	0,5312		4,1137	0,0429	*
Family members : Model	3,6415	0,0567		1,5783	0,2094		2,9350	0,0871		2,7707	0,0964	
Lifestage : Model	4,0764	0,0438	* (1)	0,0270	0,8695		2,5507	0,1106		0,0364	0,8488	
Family members : Lifestage : Model	1,4259	0,2328		1,0791	0,2992		2,2715	0,1322		0,1092	0,7412	

- Looking at these results, we can see that, for all these predictive measures, the differences between the baseline and the random forest are **not due to luck** (predictive measures for each in the annex).
- A very interesting result comes from the recall. The **interaction between Lifestage and Model (1) is statistically significant**, meaning that the models have a different behaviour depending on the lifestage segments. By fixing the Model's levels, more **targeted ANOVAs** showed that there're **statistical differences between the lifestage segments for each model** (p-value (baseline) = 0.0341 and p-value (RF) = 5.87e-4). Nonetheless, the **random forest has statistically way higher recall values**, across all segments and customers, than the baseline model. All these results are supported by the violin plots on the right, although the interaction doesn't seem that significant given that the distributions and quartiles, for each model, don't seem too different between lifestage segments;
- The **precision** is the only predictive measure where the **random forest has statistically lower values**, compared with the baseline model;
- Although the significance level considered was 5%, we can see **p-values lower than a significance level of 10%**, mainly on the **interaction between family members and the models;**
- From these results, we conclude that the **random forest is statistically better than the baseline model**, besides precision, and that **segments affect their predictive power**.



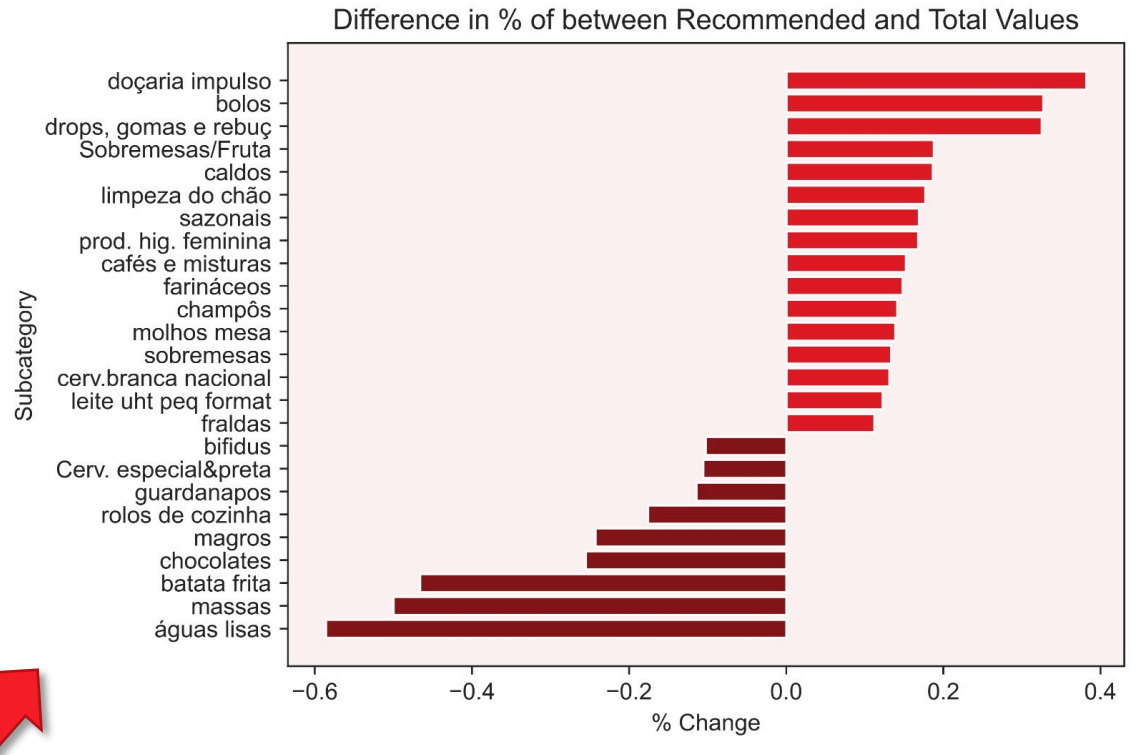
5.4 Evaluate Results - Recommendations

- Since the best model for classification was Random Forest, Model Persistence was used to save the model in a "joblib" file, being saved only the last iteration of the time-series cross validation, with the features and hyperparameters selected for that iteration;

- Then, the "joblib" file was used to run for November 2022 to get the top 20 recommendations per customer considering the proportion of subcategory appearances in the transactional data and the proportion of recommendations for each subcategory;

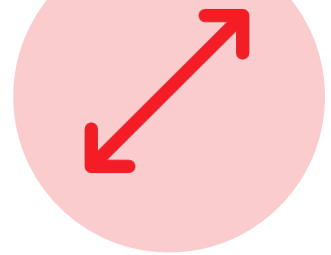
- Top 20 recommendations were defined based on a filter applied, that defines the highest percentual difference between the proportions mentioned above.

- Looking at an example of recommendations for a customer, we can see that trivial recommendations, such as "massas" and "águas lisas" decreased in the recommendation, while more tailored subcategories, such as "fraldas", "bolos", and "caldos" are recommended the most. We'll see that some of these subcategories really have a low percentage of appearances in the transactional data.



5.4.1 Evaluate Results - Recommendations

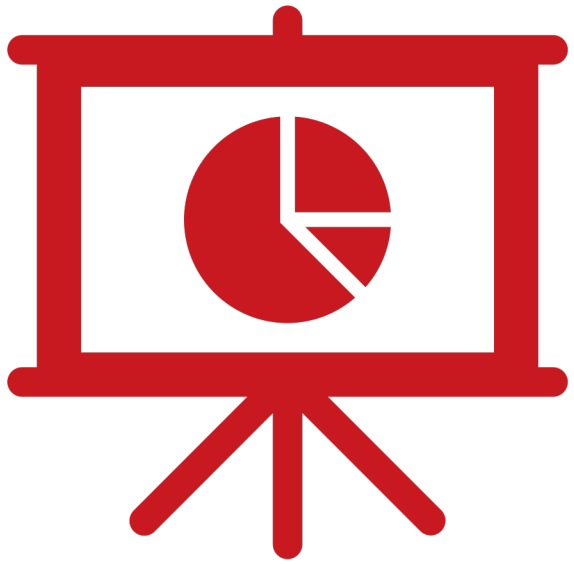
The choice of the recommendation system tactic was done focusing on the **personalization** for the client whilst keeping in mind the **profitability** of the recommendation to the seller. By choosing the 20 subcategories with the highest percentual difference between recommendations and actual transactions it allows for a considerable profit for SONAE MC.



However, these recommendations are still **highly personalized** to the customer (as we will be seeing in an example of the deployment) due to the nature of the models.

Initially, the tactic was sorting the recommendations by the **score** given by the model (between 0 and 1), however this approach would give highly generic recommendations that wouldn't be interesting neither for the client nor the seller. For example, a gallon of water would have a very high score but wouldn't bring any profit for MCSONAE, because the client would probably still buy it without a discount.

One approach considered was to **combine the model's score with the percentage difference** mentioned in the previous slides, to explore its potential. This could allow for an even personalized offers for the client and profitable business for the seller.



6. Deployment

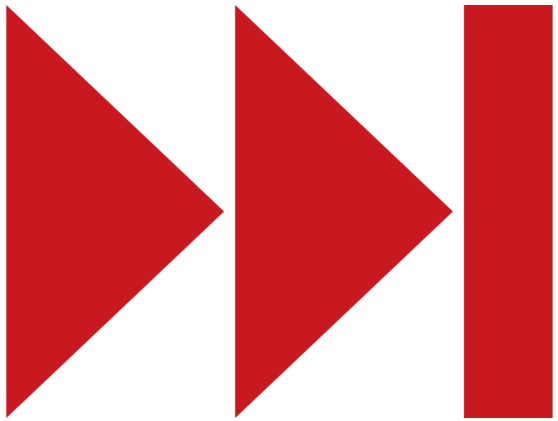
6. Deployment

Recommendations for December of 2022

SUBCAT_DSC_EXT	PERC_RECS_TOTAL	PERC_TOTAL	DIFFERENCE_PERC_TOTAL
020204 - doçaria impulso	1.73	1.34	0.38
020305 - bolos	1.29	0.96	0.33
020203 - drops, gomas e rebuç	1.11	0.78	0.33
080405 - infantis	0.79	0.48	0.32
100101 - óleos	1.08	0.80	0.27
020301 - bol. tradicionais	1.30	1.03	0.27
010207 - sal	0.99	0.72	0.27
050402 - toalhitas bebé&crian	1.28	1.02	0.25
060307 - acessórios limpeza	1.11	0.87	0.24
010302 - conservas carne	1.37	1.14	0.23
080401 - sólidos tradicionais	0.83	0.60	0.22
100202 - a□úcar	1.17	0.95	0.22
030204 - Ref sem gás	0.49	0.28	0.21
020303 - bolachas salgadas	0.82	0.61	0.20
030202 - ice tea	1.20	1.00	0.20
140204 - Sobremesas/Fruta	0.83	0.64	0.19
010203 - caldos	0.67	0.49	0.19
060301 - limpeza do chão	1.21	1.04	0.18
020205 - sazonais	0.53	0.36	0.17
050202 - prod. hig. feminina	1.30	1.13	0.17

- **Customer ID:** 13031606
- **Age segment:** 25-35 years
- **Lifestage segment:** Family with kids
- Two different approaches were tested;
- We recommended **20 subcategories**, based on the difference between the proportions:

$$PERC_{RECS} - PERC_{TOTAL}$$



7. Conclusions, limitations and future work

7.Conclusions, limitations and future work

- **Conclusions**
  - In general, it was very hard to achieve good results given our current knowledge and limited experience with projects of this complexity
  - Not only that, but the entire pipeline, from the relational database until the deployment became a tough challenge
  - Regarding regression, it is not possible to obtain conclusions given the models' performance
  - Overall, we are very satisfied with proposed problem and respective outcome and results

- **Limitations**
  - Computational resources
  - Experience

- **Future Work**
  - As future work, it would be interesting to predict the top 5 subcategories with highest potential of increased spending

8. Annexes

Features Dictionary

Feature	Group	Type	Description
SEG_LIFESTYLE_CD	Explicit	Categorical	Segment used to distinguish the types of customers
GENDER	Explicit	Categorical	Gender of each client
FAMILY_MEMBERS	Explicit	Categorical	Total number of members in each family
TARGET	Explicit	Binary	If a consumer bought a certain subcategory - 1; O - Otherwise
FULLDATE	Explicit	Categorical	Date on which the transaction was made
CUS_NUM_TRANSACTIONS_Month	Customer	Numerical	Total number of transactions made by the consumer in the respective month
CUS_NUM_TRANSACTIONS_QUARTER	Customer	Numerical	Total number of transactions made by the consumer in the respective quarter
CUS_NUM_TRANSACTIONS_SEMESTER	Customer	Numerical	Total number of transactions made by the consumer in the respective semester
CUS_NUM_TRANSACTIONS_YEAR	Customer	Numerical	Total number of transactions made by the consumer in the respective year
CUST_TOTAL_QTY_BOUGHT_MONTH	Customer	Numerical	Total quantity purchased by the consumer in the respective month
CUST_TOTAL_QTY_BOUGHT_QUARTER	Customer	Numerical	Total quantity purchased by the consumer in the respective quarter
CUST_TOTAL_QTY_BOUGHT_SEMESTER	Customer	Numerical	Total quantity purchased by the consumer in the respective semester
CUST_TOTAL_QTY_BOUGHT_YEAR	Customer	Numerical	Total quantity purchased by the consumer in the respective year
CUST_NUM_UNIQUE_SUBCAT_MONTH	Customer	Numerical	Total number of unique subcategories purchased in the respective month
CUST_NUM_UNIQUE_SUBCAT_QUARTER	Customer	Numerical	Total number of unique subcategories purchased in the respective quarter
CUST_NUM_UNIQUE_SUBCAT_SEMESTER	Customer	Numerical	Total number of unique subcategories purchased in the respective month
CUST_NUM_UNIQUE_SUBCAT_YEAR	Customer	Numerical	Total number of unique subcategories purchased in the respective month
CUST_AVG_DAYS_SINCE_PRIOR_TRANSACTION_MONTH	Customer	Numerical	Average number of days since the last transaction made by the consumer in a respective month
CUST_AVG_DAYS_SINCE_PRIOR_TRANSACTION_QUARTER	Customer	Numerical	Average number of days since the last transaction made by the consumer in a respective quarter
CUST_AVG_DAYS_SINCE_PRIOR_TRANSACTION_SEMESTER	Customer	Numerical	Average number of days since the last transaction made by the consumer in a respective semester
CUST_AVG_DAYS_SINCE_PRIOR_TRANSACTION_YEAR	Customer	Numerical	Average number of days since the last transaction made by the consumer in a respective year
CUST_AVG_BASKET_SIZE_MONTH	Customer	Numerical	Average size of the last shopping cart made by the consumer in a respective month
CUST_AVG_BASKET_SIZE_QUARTER	Customer	Numerical	Average size of the last shopping cart made by the consumer in a respective quarter
CUST_AVG_BASKET_SIZE_SEMESTER	Customer	Numerical	Average size of the last shopping cart made by the consumer in a respective semester
CUST_AVG_BASKET_SIZE_YEAR	Customer	Numerical	Average size of the last shopping cart made by the consumer in a respective year
SUBCAT_NUM_TRANSACTION_MONTH	Subcategory	Numerical	Total number of times a subcategory was purchased in a given month
SUBCAT_NUM_TRANSACTION_QUARTER	Subcategory	Numerical	Total number of times a subcategory was purchased in a given quarter
SUBCAT_NUM_TRANSACTION_SEMESTER	Subcategory	Numerical	Total number of times a subcategory was purchased in a given semester
SUBCAT_NUM_TRANSACTION_YEAR	Subcategory	Numerical	Total number of times a subcategory was purchased in a given year
SUBCAT_TOTAL_QTY_BOUGHT_MONTH	Subcategory	Numerical	Total quantity that a subcategory was purchased in a given month
SUBCAT_TOTAL_QTY_BOUGHT_QUARTER	Subcategory	Numerical	Total quantity that a subcategory was purchased in a given quarter
SUBCAT_TOTAL_QTY_BOUGHT_SEMESTER	Subcategory	Numerical	Total quantity that a subcategory was purchased in a given semester
SUBCAT_TOTAL_QTY_BOUGHT_YEAR	Subcategory	Numerical	Total quantity that a subcategory was purchased in a given year
CUSTSUBCAT_NUM_TRANSACTIONS_MONTH	Customer-subcategory	Numerical	Total number of transitions where a category appears as purchased by a consumer in a given month
CUSTSUBCAT_NUM_TRANSACTIONS_QUARTER	Customer-subcategory	Numerical	Total number of transitions where a category appears as purchased by a consumer in a given month
CUSTSUBCAT_NUM_TRANSACTIONS_SEMESTER	Customer-subcategory	Numerical	Total number of transitions where a category appears as purchased by a consumer in a given month
CUSTSUBCAT_NUM_TRANSACTIONS_YEAR	Customer-subcategory	Numerical	Total number of transitions where a category appears as purchased by a consumer in a given month
CUSTSUBCAT_TOTAL_QTY_BOUGHT_MONTH	Customer-subcategory	Numerical	Total number of times a subcategory was purchased by a consumer in a given month
CUSTSUBCAT_TOTAL_QTY_BOUGHT_QUARTER	Customer-subcategory	Numerical	Total number of times a subcategory was purchased by a consumer in a given quarter
CUSTSUBCAT_TOTAL_QTY_BOUGHT_SEMESTER	Customer-subcategory	Numerical	Total number of times a subcategory was purchased by a consumer in a given semester
CUSTSUBCAT_TOTAL_QTY_BOUGHT_YEAR	Customer-subcategory	Numerical	Total number of times a subcategory was purchased by a consumer in a given year
CUSTSUBCAT_AVG_DAYS_SINCE_PRIOR_TRANSACTION_MONTH	Customer-subcategory	Numerical	Average number of days since a consumer last bought a category in a given month
CUSTSUBCAT_AVG_DAYS_SINCE_PRIOR_TRANSACTION_QUARTER	Customer-subcategory	Numerical	Average number of days since a consumer last bought a category in a given quarter
CUSTSUBCAT_AVG_DAYS_SINCE_PRIOR_TRANSACTION_SEMESTER	Customer-subcategory	Numerical	Average number of days since a consumer last bought a category in a given semester
CUSTSUBCAT_AVG_DAYS_SINCE_PRIOR_TRANSACTION_YEAR	Customer-subcategory	Numerical	Average number of days since a consumer last bought a category in a given year

Baseline vs Random Forest

Predictive Measures for November 2022

Models	Precision	Recall	F1-Score	ROC-AUC
Baseline	0.4599	0.4290	0.4439	0.6750
Random Forest	0.3362	0.7985	0.4732	0.7756