University of Southern Denmark, Kolding
Faculty of Business and Social Sciences
Business School

Delivery date: 02.01.2026
Number of keystrokes: 55.573 (including 800 keystrokes per)

Cleaning & preparing retail data

Data Science for Business Development

Authors of the report:
Ahsanul Haque Khan
Dimitrios Liaros
Daniel Alexandru Radulescu
Ludwig Wegner

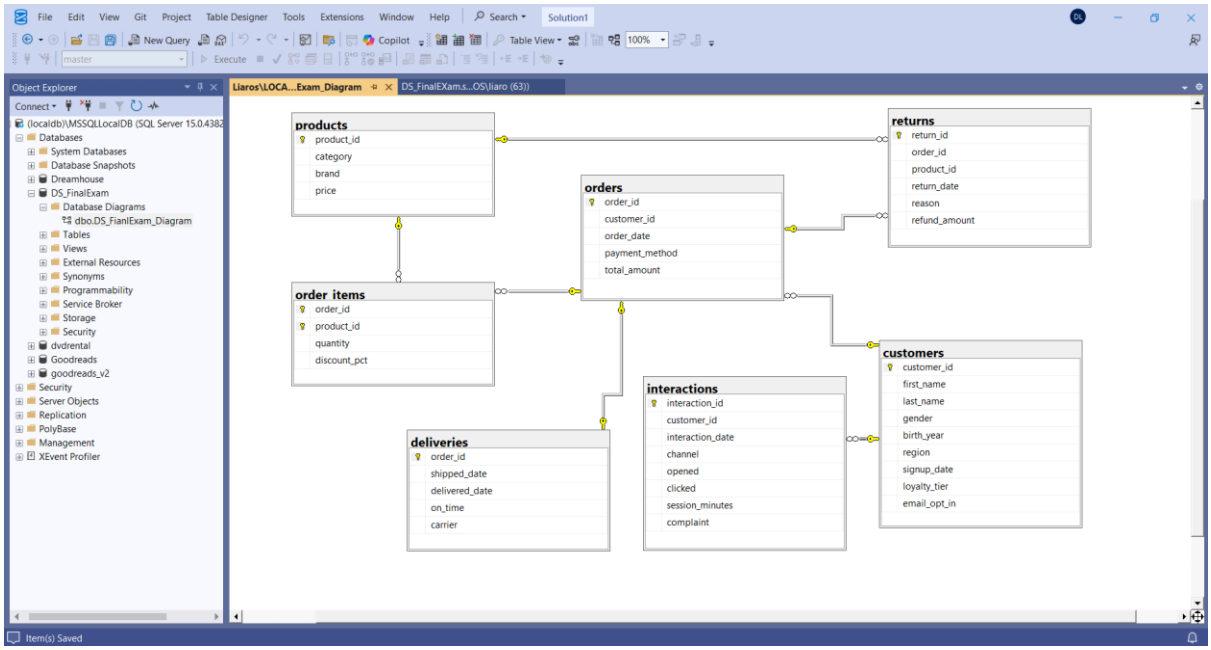
Lecturer:
Jonas Husum Dalstrup

Table of contents

1. SQL and Database Analysis

We created the database keeping in mind the guidelines that we were given and the story the data needed to tell. First, we established the core tables, including customers, products, and orders. These are the variables we thought are most relevant. Afterwards, we assigned a separate primary key to each table to ensure that information can be identified clearly and distinctly. The next step involved the identification of the foreign keys to establish a distinct relationship between the tables. This avoids confusion while retrieving the necessary and useful information, like the customer who ordered different goods in one order or different orders altogether. In addition, we carefully placed data types and constraints, making sure that identifiers and date fields could not be null, which helped maintain data quality from the beginning.

The table we found most complex was the interactions table. After a lot of discussion, we decided to keep it simple and only convert the session minutes to integers. The reason for this is that the dataset we had to insert into this table was not that substantial, and the guidelines in the description file were straightforward. After establishing a basic structure, we went ahead and developed a diagram that showed how each table was related. Finally, we loaded the data provided to us in order to create a functional database.



2. Data Preparation

In the beginning, we handled the dataset the same way a real business would. Prior to our attempt to clean the dataset, it was important to understand what the situation was. The dataset, even though messy, resembles how an actual mid-sized business operates daily. Customers often ignore their e-mails, do not bother to leave a satisfaction score, and a small group of people have an actual interest in those things. Our goal was to make sure that it still told an insightful story about the company's customers and operations.

We started by taking a general look at the data, i.e. what kind of variables were involved, how complete they were, and if they had missing values. At first sight, it was obvious that most of the data gaps appeared in fields where customer behavior is not mandatory, for example, e-mail engagement and satisfaction scores. That observation was the cornerstone of the cleaning process, since we had to think about why data was missing and not how to replace it. Based on our given case, we focused on the most important variables from the company's perspective (related to e.g. marketing engagement, returns, future value, customer satisfaction, and spending behavior). We thought that these variables can provide valuable information about retention, targeting, and operational efficiency. Thus, we did not proceed with aggressive cleaning that would potentially lead to serious alterations of the dataset.

For spending-related metrics, missing values were treated carefully. We observed that most of the customers spend relatively little compared to the few that spend a lot. So, replacing the missing values with the average spending amount helped keep the dataset realistic and avoid extreme values from dominating it.

Regarding marketing engagement, we proceed with a different mindset. It made more sense to assume that values missing from e-mail click rates are equal to disengagement, rather than average interest. Realistically, these missing values are translated as no interaction, but replacing them with zero keeps the data aligned with the marketing logic and makes it easier to identify engaged versus disengaged customers. Satisfaction scores were handled with extreme care. Zero values imply very strong dissatisfaction, which cannot be accurate. Simultaneously, dropping customers without a satisfaction score would lead the dataset to be biased towards the ones that gave feedback. Thus, replacing zero values with a typical one allowed those customers to remain in the dataset, while keeping the satisfaction scale meaningful.

After the completion of the cleaning process, we investigated the structure and the consistency of the dataset. Regional and demographic data were standardized so they behaved as true categories. This does not mess with the information, though it will prevent potential issues when comparing performance across regions or customer groups. Additionally, we checked the dataset for duplicate values and abnormal/extreme spending values. The reason behind this action is that duplicates can inflate results, while extreme values can look suspicious. These kinds of values can exist in an e-commerce context, but as

a safety net, we decide to treat those values as insightful signals.

Before the completion of the entire cleaning process, we verified that all the dimensions were complete, that no rows were unintentionally deleted, that the dataset still makes sense and then saved it as a new .csv file. Only after we did these actions was it decided that we could move on with further analysis.

There were other options regarding the cleaning procedure. One of them would have been to drop all the rows with missing data, but that would have decreased the size of the dataset and lead to less efficient data. Another approach could be to use more advanced imputation code, making it more difficult to interpret the results. The chosen way of approach is the golden link between simplicity and interpretability.

In the end, this whole process was not about making the dataset flawless. It was about making it solid, transparent, and trustworthy. The finalized dataset reflects real customer behavior, without the noise affecting data-driven decisions.

3. Correlation Matrix analysis

3.1 Modifications and Their Importance

- **Numeric filter**: In this, *select_dtypes(include='number')* has been used to select only numeric data and avoid any noise from data that has meaning and importance, such as gender and region. This modification makes PCA mathematically more accurate**.**
- **Grid format**: The heatmap was re-adjusted to be a better-shaped cube. Also, since there are a lot of values in decimals, we chose a *(square=True)* heatmap with a light gray border, specifying the *(linecolor)*. This was done considering the heatmap has a 30x30 structure, so as not to miss any rows in the variable display**.**
- **Preparation for Standardization**: To have the full summary table (Mean, Std, Median) together with the heatmap, we check for the existence of variables in different scales. This can be observed in the heatmap, where variables with the highest correlations also have the highest variances.

3.2 Key Findings and Behavioral Data Points

The heatmap generated by the correlation matrix revealed several patterns in the data set that define the customer base. From there, we are able to identify the blocks of high and low correlation. By doing this, we can predict a pattern in customer behavior, which reduces data

complexity. Meaning we can do that without taking into account every column in the data set.

3.2.1 Financial Growth (Strong Positive Correlations)

From the heatmap, we get to observe that there is a near-perfect correlation between total_spent, total_orders, and future_3m_spend (correlation ranging from 0.80 to 0.94). **Insight:** This confirms that we can follow the past data. Customers who have spent heavily in the past are the ones who are most likely to drive revenue in the future. Combining these into a single Value Index in the PCA.

3.2.2 Past orders vs. Last order (Strong Negative Correlations)

From the heatmap, we get to see one of the most striking findings, the negative relationship between orders_last_3m and recency_days (-0.79).
Insight: This acts as a pulse check for the whole business. While the order frequency increases, the time since the last order drops distinctly. If a customer who is going against this trend (high recency despite high past orders) should be a target candidate for marketing campaigns to bring them back or make them active.

Friction & Satisfaction:

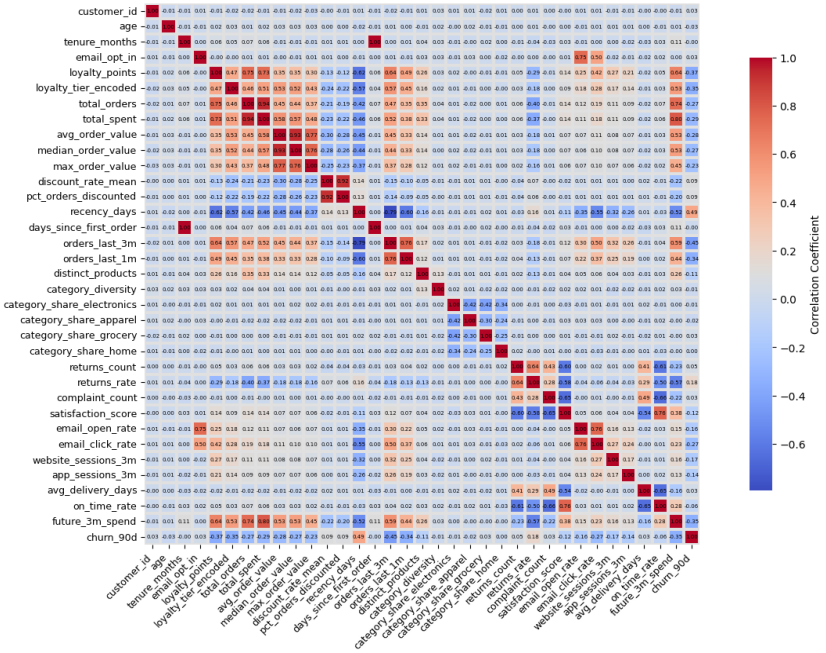
The heatmap showed a significant negative correlation between complaint_count and satisfaction_score (-0.65).
Insight: An interesting finding we get to see from the heatmap between complaint_count and recency_days. complaint_count had almost zero correlation with recency_days (0.01). Pointing out the fact that customers don't necessarily stop shopping when they are unhappy or unsatisfied. This gives the company a window of opportunity to resolve its shortcomings, finding the issue before the customer actually churns.

Digital Engagement vs. Loyalty:

There is a moderate positive correlation between email_open_rate and loyalty_points (0.44)
Insight: We know that digital engagement is a leading indicator of loyalty. Customers who show loyalty to the brand's "top-of-funnel" content (emails) are more likely to participate in the "bottom-of-funnel" rewards program. The main goals of these clients are to join rewards programs and accrue points. We also see from the data that if a customer stops opening emails, there is a high probability that their loyalty participation will decline soon.

```
PS D:\Semester_1\Data science for business dev\final_project> & "D:/Semester_1/Data science for business dev/final_project/.venv/Scripts/Activate.ps1"
(.venv) PS D:\Semester_1\Data science for business dev\final_project> & "D:/Semester_1/Data science for business dev/final_project/.venv/Scripts/python.exe" "d:/Semester_1/Data science for business dev/final_project/correlation.py"
--- Data Summary Table (All Numeric Columns) ---
                              mean          std     min       max    median  missing
age                      39.553500    11.551668  18.000    80.000    40.000        0
app_sessions_3m           1.554333     1.669784   0.000     9.000     1.000        0
avg_delivery_days         3.471375     1.114593   1.000     7.680     3.460        0
avg_order_value          60.141325    24.564777   5.000   146.470    60.210        0
category_diversity        1.366167     0.621680   1.000     4.000     1.000        0
category_share_apparel    0.250707     0.108715   0.001     0.833     0.252        0
category_share_electronics 0.293037    0.124200   0.000     0.818     0.294        0
category_share_grocery    0.253322     0.108761   0.000     0.836     0.250        0
category_share_home       0.202933     0.095443   0.000     0.747     0.198        0
churn_90d                 0.546500     0.497875   0.000     1.000     1.000        0
complaint_count           1.308500     1.105832   0.000     5.000     1.000        0
customer_id            3000.500000  1732.195139   1.000  6000.000  3000.500        0
days_since_first_order  721.189833   349.493011  30.000  2139.000   717.000        0
discount_rate_mean        0.394081     0.115719   0.110     0.600     0.384        0
distinct_products         5.481000     3.095088   0.000    21.000     5.000        0
email_click_rate          0.101866     0.115298   0.000     0.526     0.064        0
email_open_rate           0.324616     0.247184   0.000     0.971     0.362        0
email_opt_in              0.757000     0.428931   0.000     1.000     1.000        0
future_3m_spend          27.669002    43.128878   0.000   371.290     4.690        0
gender                         NaN          NaN     NaN       NaN       NaN        0
loyalty_points          130.252500   163.731689   0.000  1577.000    75.000        0
loyalty_tier_encoded      0.650000     0.726352   0.000     2.000     0.500        0
max_order_value         100.397340    31.974415  10.000   203.450   100.275        0
median_order_value       60.160268    26.246396   5.000   152.070    60.220        0
on_time_rate              0.767456     0.133472   0.322     1.000     0.768        0
orders_last_1m            0.600333     0.927583   0.000     6.000     0.000        0
orders_last_3m            1.509000     1.742917   0.000     9.000     1.000        0
pct_orders_discounted     0.444011     0.126718   0.025     0.781     0.439        0
recency_days            120.961500    48.135121   0.000   301.000   121.000        0
region                         NaN          NaN     NaN       NaN       NaN        0
returns_count             1.622667     1.194930   0.000     6.000     2.000        0
returns_rate              0.673502     0.409930   0.000     1.000     1.000        0
satisfaction_score        6.690768     1.774486   1.000    10.000     6.700        0
tenure_months            24.040333    11.651867   1.000    71.000    24.000        0
total_orders              1.842667     2.350426   0.000    26.000     1.000        0
total_spent             138.400128   234.072000   0.000  3034.260    59.520        0
website_sessions_3m       1.641833     1.748922   0.000    11.000     1.000        0
```

Cleaned Dataset correlation Heatmap)



3.3 PCA (Principal Component Analysis)

3.3.1 Code Modifications

Targeted vs. Global PCA: From the correlation of the data we analyzed based on the variance, we ran two separate PCA executions. The first one is the Targeted Index (4 variables), which is used to create a specific Customer Power score. The second one is the Global Dimensionality Reduction, which encompasses all 30 columns in the dataset. The idea behind this modification is to simplify the full dataset.

Standardization: The dataset contains a handful of different units (e.g., total_spent is in euros vs. recency_days is in time). We used preprocessing.scale() to ensure that columns with high values do not unfairly dominate the analysis.

3.3.2 How We Performed the PCA

1. Selection: We identified the numeric columns first and dropped the customer_id column as it was creating noise
2. Standardization: Then we transformed every variable to have a mean of 0 and a standard deviation of 1.
3. Decomposition: We also calculate the Eigenvalues through the PCA algorithm to determine the variance explained by each new expand
4. Loading Generation: We then reordered the components into the loadings table to see how much weight each original variable (e.g., Age or Spend) contributes to each Principal Component

3.3.3 Key Findings and Interpretations

The Value Index (The 4-Column Combo): The targeted PCA was highly correlated.

- Retention: PC1 and PC2 together accounted for 90.12% of the information from the four columns we selected.
- Interpretation: PC1 (71% variance) functions as the growth engine. Since total_spent and future_3m_spend have a high positive correlation, a high score highlights the most valuable financial assets.

3.3.4 Global Behavioral Patterns

- **80% threshold**: At PC12, we attain the 80% threshold. This mostly means that we can accurately describe a customer's whole profile (30 columns) using only 12, while retaining significant accuracy
- **PC1 (21.5%)**: If we observe the loading for PC1 in the 30-column table, we see high weights for total_spent (0.318), loyalty_points (0.305), and orders_last_3m (0.318). These components denote active loyalty.
- **PC2 (8.7%)**: We can see that two columns, email_open_rate (0.441) and email_click_rate (0.441), are the main drivers of PC2. This determines a certain subset of digital participants who are the most approachable through marketing campaigns, but it may take more time to convert them into active customers.
- **PC3 (8.5%)**: We also observe that compared to returns_count (-0.484) and complaint_count (-0.415), satisfaction_core (.502) dominates the PC3. The stress in the customer experience is adequately depicted.

```
(.venv) PS D:\Semester_1\Data science for business dev\final_project> & "D:/Semester_1/Data science for business dev/final_project/.venv/Scripts/python.exe" "d:/Semester_1/Data science for busi
ness dev/final_project/PCA.py"
--- PCA (n=2) Summary: Comprehensive Value Index ---
   Eigenvalue  Proportion of Variance  Cumulative proportion
0    2.833971                0.708375               0.708375
1    0.788631                0.197125               0.905499

--- PCA (n=30) Summary (Standardized) ---
      Eigenvalue  Proportion of Variance  Cumulative proportion
PC1       6.4434                  0.2147                 0.2147
PC2       2.6094                  0.0870                 0.3017
PC3       2.5745                  0.0858                 0.3875
PC4       2.0265                  0.0675                 0.4550
PC5       1.7399                  0.0580                 0.5130
PC6       1.4768                  0.0492                 0.5623
PC7       1.4092                  0.0470                 0.6092
PC8       1.3794                  0.0460                 0.6552
PC9       1.3031                  0.0434                 0.6986
PC10      1.2243                  0.0408                 0.7394
PC11      1.0182                  0.0339                 0.7734
PC12      0.9862                  0.0329                 0.8062
PC13      0.8623                  0.0287                 0.8350
PC14      0.7960                  0.0265                 0.8615
PC15      0.7646                  0.0255                 0.8870
PC16      0.7173                  0.0239                 0.9109
PC17      0.5139                  0.0171                 0.9280
PC18      0.4368                  0.0146                 0.9426
PC19      0.2884                  0.0096                 0.9522
PC20      0.2858                  0.0095                 0.9617
PC21      0.2341                  0.0078                 0.9695
PC22      0.2245                  0.0075                 0.9770
PC23      0.1808                  0.0060                 0.9830
PC24      0.1649                  0.0055                 0.9885
PC25      0.1500                  0.0050                 0.9935
PC26      0.0799                  0.0027                 0.9962
PC27      0.0666                  0.0022                 0.9984
PC28      0.0481                  0.0016                 1.0000
PC29      0.0003                  0.0000                 1.0000
PC30      0.0000                  0.0000                 1.0000
```

4. Defining Prediction Goals

In this part of the project, based on the insights gathered from the Correlation Matrix and PCA, we have decided to focus on revenue forecasting for the regression analysis. From here, we transition to predictive modeling.

Prediction goal: Forecasting/predicting future spend

- **Target variable (continuous):** future_3m_spend
- **Definition:** From this variable, we will be able to predict or forecast an expenditure or a monetary value that a customer will incur within the next 3 months
- **Business Justification:** Regardless of the size of a company, predicted revenue is one of the most important factors of business intelligence. By predicting future spend, a company can determine its financial targets, categorize customers based on the spend, what sort of optimization it would need for the logistics, and what changes to make for operations. With this, a company can also justify marketing expenditure. The way we tried to make the model is that it will not simply look into how much a customer has spent in a time frame; instead, this will allow the business to be forward-looking and categorize high-potential growth segments.

```
(.venv) PS D:\Semester_1\Data science for business dev\final_project> & "D:/Semester_1/Data science for business dev/final_project/.venv/Scripts/python.exe" "d:/Semester_1/Data science for busi
ness dev/final_project/PCA.py"

--- PCA (n=30) Components (Loadings) ---
                              PC1     PC2     PC3     PC4     PC5     PC6     PC7     PC8     PC9    PC10  ...    PC21    PC22    PC23    PC24    PC25    PC26    PC27    PC28    PC29    PC30
age                        0.0111 -0.0051 -0.0097 -0.0110  0.0194 -0.0177  0.0067 -0.0497 -0.0688 -0.0912  ... -0.0087 -0.0005 -0.0000 -0.0031  0.0121 -0.0022  0.0015  0.0007  0.0000 -0.0000
tenure_months              0.0157 -0.0379  0.0889  0.6723 -0.1335  0.0214 -0.0734 -0.1137 -0.0153 -0.0041  ... -0.0016 -0.0060 -0.0018 -0.0003  0.0055 -0.0009 -0.0020 -0.0012  0.7071  0.0013
email_opt_in               0.0357  0.3443  0.0842 -0.0439 -0.4462  0.0385  0.3471 -0.0138  0.0182  0.0153  ...  0.0472 -0.3870  0.1360  0.2349 -0.4184 -0.0110  0.0011 -0.0090 -0.0003  0.0001
loyalty_points             0.3089  0.0684  0.0355  0.0770  0.1739  0.0079  0.0200  0.2649  0.0013  0.0438  ...  0.2709  0.0248  0.7408  0.0444 -0.0100  0.0038  0.0224 -0.0436 -0.0010 -0.0000
loyalty_tier_encoded       0.2811 -0.0212 -0.0526  0.0128  0.0096 -0.0169 -0.0724 -0.1107  0.0071 -0.0218  ...  0.0440  0.0811  0.0337 -0.0103 -0.0148  0.0077  0.0000 -0.0138 -0.0003  0.0000
total_orders               0.2983 -0.0985  0.0103  0.0944  0.1302  0.0102  0.2301  0.3282 -0.0005  0.0589  ... -0.0444 -0.0345 -0.3041 -0.0716 -0.0231 -0.0060 -0.0217  0.7102 -0.0000  0.0001
total_spent                0.3126 -0.1221 -0.0192  0.0750  0.1111  0.0086  0.2397  0.2425 -0.0064  0.0399  ...  0.0784 -0.0128 -0.3948 -0.0580  0.0009 -0.0051  0.0026 -0.6877  0.0004 -0.0001
avg_order_value            0.2871 -0.2010 -0.1326 -0.0598 -0.0706 -0.0088  0.1636 -0.3516 -0.0203 -0.0445  ...  0.0337  0.1297  0.0823  0.0130  0.0283 -0.0476  0.7239  0.0684 -0.0007 -0.0000
median_order_value         0.2842 -0.2024 -0.1307 -0.0626 -0.0579 -0.0086  0.1646 -0.3523 -0.0170 -0.0484  ...  0.0336  0.1574  0.1312 -0.0000  0.0062  0.0608 -0.6833  0.0438  0.0002  0.0001
max_order_value            0.2491 -0.1846 -0.1207 -0.0618 -0.0768 -0.0039  0.1757 -0.3625 -0.0255 -0.0526  ... -0.0313 -0.1292 -0.0659 -0.0056 -0.0084  0.0037 -0.0386 -0.0008  0.0002 -0.0000
discount_rate_mean        -0.1325  0.2443  0.1570  0.0804  0.4698 -0.0015  0.3054 -0.2615 -0.0414 -0.0132  ...  0.0017 -0.0174 -0.0154  0.0075 -0.0210  0.7069  0.0588  0.0041 -0.0000 -0.0001
pct_orders_discounted     -0.1257  0.2398  0.1562  0.0786  0.4753 -0.0004  0.3091 -0.2699 -0.0429 -0.0156  ... -0.0100 -0.0078  0.0091  0.0021  0.0145 -0.7021 -0.0490 -0.0006 -0.0002  0.0001
recency_days              -0.3058 -0.1651 -0.0025  0.0213 -0.0901  0.0149  0.2081  0.0751  0.0061  0.0033  ...  0.2194  0.5130  0.0860  0.3518  0.3027  0.0111  0.0007  0.0228 -0.0001  0.0000
days_since_first_order     0.0156 -0.0377  0.0888  0.6722 -0.1336  0.0216 -0.0737 -0.1138 -0.0150 -0.0041  ... -0.0027 -0.0061 -0.0030 -0.0006  0.0062 -0.0006 -0.0030 -0.0017 -0.7071 -0.0012
orders_last_3m             0.3188  0.1479  0.0014 -0.0082  0.1298 -0.0152 -0.2194 -0.0305 -0.0042 -0.0115  ...  0.0086 -0.2404 -0.1444  0.7063  0.3718  0.0041 -0.0200  0.0230  0.0002  0.0001
orders_last_1m             0.2576  0.1340 -0.0045  0.0029  0.1471 -0.0234 -0.2443 -0.0403 -0.0100 -0.0177  ...  0.0398  0.1914  0.0528 -0.3313 -0.1563  0.0005  0.0104  0.0009  0.0001  0.0000
distinct_products          0.1233 -0.0366  0.0146  0.0864  0.0914 -0.0016  0.2733  0.3634 -0.0299 -0.0009  ...  0.0201  0.0360  0.0314 -0.0039  0.0031  0.0000  0.0019 -0.0139 -0.0000  0.0000
category_diversity         0.0143 -0.0061  0.0133  0.0615 -0.0118  0.0654  0.1503  0.1659 -0.0895 -0.0701  ... -0.0108  0.0030  0.0058 -0.0015  0.0038  0.0078 -0.0020  0.0044 -0.0003 -0.0001
category_share_electronics 0.0078 -0.0102 -0.0227 -0.0238  0.0396  0.8123 -0.0431 -0.0107  0.0097 -0.1059  ...  0.0086 -0.0080 -0.0022 -0.0076  0.0046  0.0010  0.0009  0.0010 -0.0009  0.5658
category_share_apparel    -0.0008  0.0396  0.0306 -0.0127 -0.0517 -0.3789 -0.0317  0.0600 -0.6955 -0.3196  ... -0.0067  0.0066 -0.0016  0.0012 -0.0004  0.0002 -0.0009  0.0021 -0.0010  0.4952
category_share_grocery    -0.0049 -0.0063  0.0113  0.0402  0.0103 -0.3852  0.0752  0.0277  0.6994 -0.3202  ... -0.0109 -0.0036  0.0052 -0.0018  0.0032 -0.0004  0.0023 -0.0012 -0.0008  0.4955
category_share_home       -0.0036 -0.0247 -0.0182 -0.0004 -0.0044 -0.1866  0.0065 -0.0860 -0.0174  0.8667  ...  0.0087  0.0071 -0.0014  0.0106 -0.0093 -0.0007 -0.0026 -0.0028 -0.0009  0.4348
returns_count             -0.0012  0.1478 -0.4907  0.1165  0.0653 -0.0147  0.0464  0.0948 -0.0202  0.0127  ... -0.4078  0.0704  0.1572  0.0549  0.0237  0.0151  0.0047 -0.0640  0.0001 -0.0000
returns_rate              -0.1393  0.1853 -0.4167  0.0529 -0.0117 -0.0229 -0.0842 -0.0848 -0.0115 -0.0390  ...  0.6267 -0.0930 -0.1973 -0.0767 -0.0110 -0.0129 -0.0094  0.0909 -0.0006  0.0001
complaint_count           -0.0190  0.1271 -0.4209  0.0827  0.0638 -0.0088  0.0904  0.0643  0.0034  0.0153  ...  0.3507 -0.0540 -0.0446 -0.0498  0.0023  0.0011 -0.0047  0.0086  0.0004  0.0001
satisfaction_score         0.0786 -0.1480  0.5085 -0.0907 -0.0407 -0.0098 -0.0564 -0.0290 -0.0005  0.0034  ...  0.4193 -0.0380 -0.0428 -0.0517  0.0096  0.0070 -0.0046  0.0088 -0.0003  0.0001
email_open_rate            0.1332  0.4468  0.1035 -0.0496 -0.3308  0.0261  0.1703 -0.0220  0.0204  0.0321  ... -0.0478 -0.0114  0.0174 -0.3914  0.6858  0.0180 -0.0086  0.0013  0.0005  0.0000
email_click_rate           0.1873  0.4450  0.0940 -0.0256 -0.1714  0.0290 -0.0119 -0.0112  0.0252  0.0087  ... -0.0064  0.6364 -0.2322  0.1843 -0.3108 -0.0123 -0.0006  0.0103  0.0000 -0.0001
website_sessions_3m        0.1158  0.1745  0.0339  0.0018  0.1497 -0.0478 -0.3091 -0.0212  0.0670 -0.0186  ... -0.0070 -0.0138 -0.0265 -0.0096 -0.0122  0.0051  0.0017  0.0017  0.0002  0.0000
app_sessions_3m            0.0957  0.1521  0.0319 -0.0298  0.1165 -0.0000 -0.2772 -0.0198  0.0338  0.0243  ... -0.0076 -0.0252 -0.0005 -0.0122 -0.0038  0.0014  0.0002  0.0016  0.0002  0.0000

[30 rows x 30 columns]
(.venv) PS D:\Semester_1\Data science for business dev\final_project>
```

4.1 Predictors

We have selected four distinct predictors for the model to observe the customer journey from a different perspective.

- total_spent (Total Expenditure): From the PCA we observed that total expenditure is one of the main factors for future behavior (PC1). A customer's budget and long-term value can be identified through this.
- tenure_months (Longevity): From this we get the measures for brand trust. When it comes to stable and predictable revenue streams, long-term customers are more valuable than new acquisitions.
- satisfaction_score (Customer satisfaction): From this predictor, we tried to make the bridge between customer satisfaction and the company's revenue. It also allows us to quantify the ROI of customer happiness. Which leads to how much a satisfied customer generates extra revenue.
- recency_days (Purchase drive): With this predictor, we can measure the customer's relationship with the company. This factor, time since the last purchase, is often reflected as a strong indicator of when a customer will return soon.
- discount_rate_mean (Price sensitivity): This helps the model to figure out which customers are more price sensitive whose purchase pattern depends more on the discount. So in the future they might have a higher volume, but the profit margin won't be high.
- loyalty_points (Stored value): Customers with higher loyalty points might as well be an indication of the right incentive to spend.
- email_open_rate (Digital Engagement): A customer needs to have a digital footprint before they actually become loyal to a certain brand. From this, we can link marketing performance directly to the business.
- avg_delivery_days (Logistics support): This predictor acts as the operation friction. It can check if delays in the delivery are acting as an obstacle to future spending.

5. Predictive Modeling

5.1 Regression Analysis

We implemented a multiple linear regression to predict future 3 m spend with the alignment of the eight predictors chosen in the regression analysis. All of the predictors were

standardized so that the comparison across the variables stays valid (e.g., comparing days to euros).

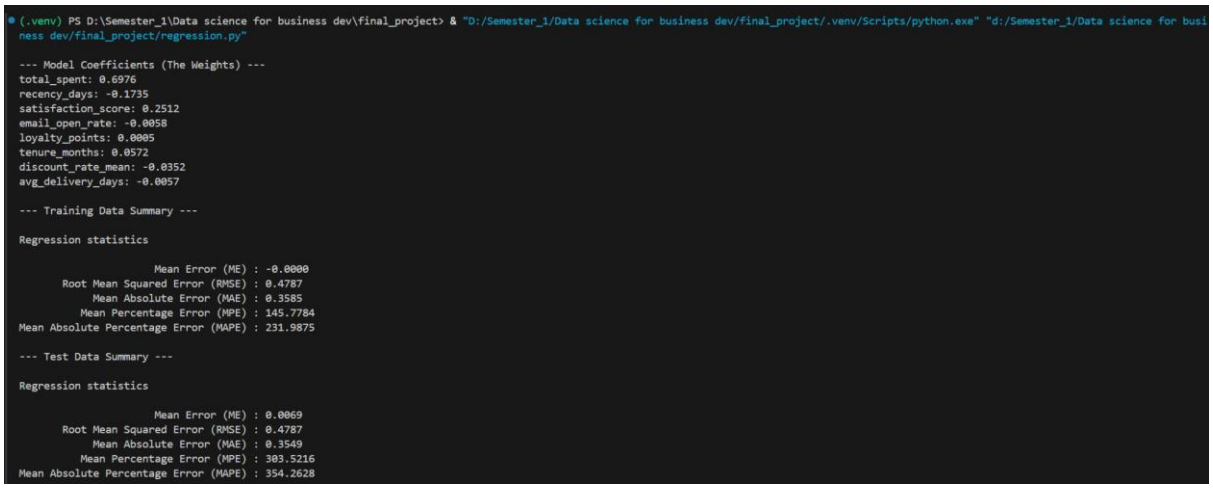
Mirror RMSE justification:

We can see that the model demonstrates stability. The root mean squared error (RMSE) is almost identical in both the training data and test data. In the python code, it has been rounded to the 4th decimal. When you did a deeper check, the numbers are slightly different at 5th place.

Training RMSE: 0.478688...

Test RMSE: 0.478737...

This mirror number and consistency prove that the model is not overfitting. Instead, it applies equally well to the new unseen data.



5.1.1 Ranking Prediction

For the business, the standardized coefficients act as a significant list. We are sharing the top 3 to show what those coefficient numbers actually mean in the real world.

1. **total_spent:**
   Coefficient: 0.6976
   1 std of Past Spend is approx. $234.07. 1 std of Future Spend is around $43.13
   0.6976 X 43.13 = $30.09

We can say that for every extra $234.07 a customer has spent in the past, the model predicts that the customer will spend an additional $30.09 within the next 3 months. From here, we can say that this is the strongest predictor among all. And best for predicting future 3-month spending

2. **satisfaction_score:**
   Coefficient: 0.2512
   1 std of satisfaction_score is approx. 1.77 points.
   0.2512 X 43.13 = $10.83
   We observed that for every 1.77point increase in satisfaction (e.g., from 7/10 to 8.77/10), the model predicts an extra $10.83 in spend. Customer satisfaction is a direct revenue driver. Even if a small increase in customer sentiment can yield to a significant financial return.
3. **recency_days:**
   Coefficient: -0.1735$. 1 std of recency_days is approx. 48.14 days.
   -0.1735 X 43.13 = -$7.48
   If a customer passes 48 days without a purchase, the model predicts that the future spend shall drop by $7.48. The longer the days without purchase increase, the harder it gets them to bring them back.
4. **tenure_months:**
   Coefficient: 0.0572. 1 std of Tenure is approx. 11.65 months.
   0.0572 X 43.13 = $2.47
   If a customer stays with a brand approximately for a year (11.65 months), the customer's predicted quarterly spend increases by $2.47. Although it is a smaller change and less powerful than immediate satisfaction.
5. **discount_rate_mean:**
   Coefficient: -0.0352. 1 std of Discount Rate is approx. 0.12 (12%).
   -0.0352 X 43.13 = -$1.52
   If a customer's average discount increases by 12%, their predicted total spend actually decreases by $1.52

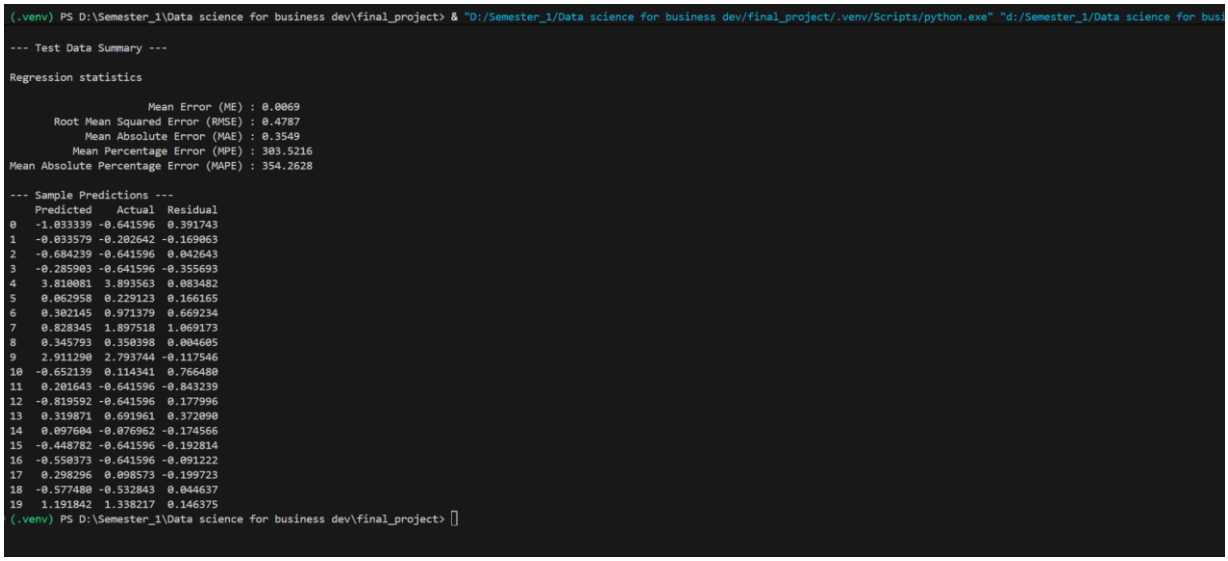
The rest of the variables are almost very small on the final dollar amount. We still have taken them as the findings were interesting, but financially not that impactful.

5.1.2 Analysis of Sample Predictions

- **Row 9: (**Predicted: 2.91 | Actual: 2.79 | Residual: -0.11)
  Analysis: The model recognized a higher spend. The model flagged this as a top-tier customer and was slightly optimistic. The prediction recommends that, although their metrics have been elite, their expenditure might be slightly less than suggested - it may be subject to personal budget cycles.
- **Row 7:** (Predicted: 0.82 | Actual: 1.89 | Residual: 1.07)
  Analysis: The model predicted a customer who can be tagged as a high spender, but

unfortunately, the customer outperformed (Actual 1.89). The model predicted the customer to spend a moderate amount using the predictors, but in reality, the customer didn't just spend more, but went on a shopping spree.

- **Row 11** (Predicted: 0.20 | Actual: -0.64 | Residual: -0.84) Analysis: The model predicted a slightly above-average spender, but the customer spent zero (standardized to -0.64). As the model went through the predictors, the number came out good. But even with those good numbers, the customer suddenly stopped purchasing, which can be explained in the next part.



5.1.3 Optional: Regression on PCA Components vs. Original Components

The idea of this analysis is to compare the standard Linear Regression model (using all 30 original features) against a Principal Component Regression (PCR) model. From this we tried to evaluate whether reducing the dimensionality of the data through PCA can maintain predictive accuracy while improving model stability.

Performance Comparison:

- Regression on Original Features: R^2 = 0.8375
- Regression on 30 PCA Components: R^2 = 0.8380
- Regression on 10 PCA Components: R^2 = 0.7825

We can see that there is a small increase in R^2 and this happened because of PCA's ability to handle multicollinearity. There are columns like total_spend and total_orders that are highly correlated. But PCA transforms them into independent component which creates a more stable situation and reduces noise

Dimensionality Reduction:

We mostly prioritize n=10 rather than n=30 for few reasons

- We take 10 components first, which captures the majority of meaningful data for the customer behaviour. The rest doesn't have that much of an impact and leads to noise which might lead to an overfitting model
- While gaining 93% of original models performance while using 66% lesser variables indicates a much efficient model.
- One of the efficient part of using 10 components is easier to map all the predictors rather than going for 30 components which make things a bit more complex

The approach we can have if we follow PCA shows that we can reduce the complexity of the dataset. In terms of losing a very minimal predictive power. The deduction to 10 components hence brings better accuracy and long term stability.

5.2 Classification

The objective of the analysis is to develop a predictive regression model and classification system that could identify customers who are likely to stop doing business with the company in the next 90 days.

The goals are:

1. To build accurate classification models to predict customer churn (stay or leave)
2. Identify the key features that make the customers churn
3. Compare model performance and interpretability
4. Provide business insights on the models findings.

5.2.1 Classification – churn90

We chose the *"churn_90d"* binary variable as it is fundamental for the prosperity of the business: the customers either stay (1) or they leave(0). This represents a crucial business challenge with direct impact in the financial part of the company. Also, the binary nature of the variable provides clear information about the retention of the customers: we either successfully maintain and intervene to keep a customer (1) or we fail to do this (0) and we have to reallocate resources to prevent this or face the consequences. Churn's classification offers insights across all customers' perspectives, as it reveals how purchase behavior, service interactions, product satisfaction and engagement all together

determine the retention outcomes.
The analysis was implemented in Python executed in Visual Studio Code and using the following libraries: pandas (for data manipulation, importing csv, creating DataFrames), numpy, scikit-learn (for machine learning algorithms and evaluation metrics such as: RandomForestClassifier, LogisticRegression, train_test_split, confusion_matrix), matplotlib (for visualization), dmba (which provides functions like classificationSummary( ) for model evaluation).

The database contains 6000 rows with 37 features, from which we selected 20 that we thought are the most relevant for business. We divided them in:

1. **Customer activity features:**
   '*recency_days*': representing the number of days since the last order. It is really important, customers with high recency are the ones who are most likely to leave
   '*orders_last_3m*': counts the orders placed in the last 3 months. Recent purchases represent ongoing interest in buying from the company
   '*total_orders*':all the orders placed. It is a customer value indicator and it shows their engagement
   '*tenure_months*': the length of customer relationships with the company in months.
   '*avg_order_value*': the average value of each transaction. It also shows the customer value
   '*distinct_products*': number of unique products. Some customers might want to have a variety of products. Usually the ones who try more products are often more engaged.
   '*category_diversity*': product categories to buy from. Customers who buy from different categories are less likely to churn
2. **Satisfaction features:**
   '*satisfaction_score*': reported satisfaction on a scale from 0 to 10. It's a direct measure of customer's happiness
   '*complaint_count*': the number of complaints registered. It's a direct measure of customer's dissapoinment
   '*loyalty_point*': the number of accumulated number point, measuring engagement
   '*returns_rate*': proportion of orders returned. It shows products and services quality and satisfaction or disappointment
   '*pct_orders_discounted*': customer's necessity of promotions and discounts. It shows how much customers rely on promotions when buying something
3. **Engagement features:**
   '*email_open_rate*': proportion of emails opened, most likely related to marketing
   '*website_session_3m*': number of website visits in the last 3 months, showing digital engagement
   '*app_session_3m*': number of mobile app sessions in the last 3 months, showing mobile engagement

'*avg_delivery_days*': average days for delivery showing service speed and satisfaction or disappointment
'*on_time_rate*': percentage of orders delivered before or on the promised day. It shows service reliability.

4. **Demographic:**
'*region*': geographical location. Each region varies in market dynamics
'*age*': customer age. It shows generational preferences
'*gender*': gender identification. It displays gender preferences

The features we decided to remove are:

- 'customer_id' - irrelevant
- 'total_spent' - highly correlated to 'total_orders' and 'avg_order_value',
- 'median_order_value' - highly correlated to 'avg_order_value',
- 'max_order_value' - not important for typical customer behavior,
- 'days_since_first_order' - has the same information as 'tenure_months',
- 'loyalty_tier_encoded - encoded version of 'loyalty_points', same information
- 'category_share_electronics', 'category_share_apparel', 'category_share_grocery', 'category_share_home' – we only kept 'category_diversity' which has them all
- 'returns_count' - we kept 'returns_rate'
- 'email_click_rate' - highly correlated with 'email_open_rate'
- 'orders_last_1m' - same with 'orders_last_3m'
- 'future_3m_spend' - this is what churn affects
- 'discount_rate_mean' - we already have 'pct_orders_discounted', we don't need the discount rate

Before doing the two classifications, some preprocessing steps were made. This included the categorical variable encoding for "region" and "gender" features which were converted to gender using the pd.get_dummies().
For the model evaluation, the database was split into training and testing sets using a ratio of 80% to 20%: 4800 customers (80%) used for model training and 1200 customers (20%) for model evaluation. Two classification models were implemented in this project, each representing different approaches to classification problems: logistic regression for interpretability and random forest for performance.

5.2.2 Model 1: Logistic Regression

Logistic regression is a statistical and machine learning technique used for classification problems, especially in the binary problems ('churn_90') by modelling the probability of an event happening. It functions like a calculator that estimates how likely a customer will leave

or stay. Instead of saying “yes” or “no”, it provides a probability percentage (like 70% chance of the customer to stay).
It is important to the business because every feature gets a clear coefficient (or “importance score”). It can display data like: “Each complaint increases the risk of churn by xx %” or “Each day without a purchase increases the churn risk by xx %”. It also translates directly to the business actions and insights. For example, if a customer has a 0-30% chance of staying, the business has to take action. If it has an 80-100% chance of staying, the business can approach a "maintenance mode” and focus on other customers who are more likely to leave.

5.2.2 Model 2: Random Forest

Random Forest is a machine learning method for classification that builds a selected amount of decision trees during the training. For classification, it operates using the principle of collecting “intelligence” through the assemblance of decision trees. Rather than relying on a single predictive model, it uses mutiple independent trained classifiers to get a more accurate prediction.
It is important to the business because it combines multiple perspectives to make more reliable predictions. Each of the selected trees analyzes the customer data in a slightly different way, for example some focus more on recent purchase behaviour, other look more into the satisfaction score or the return patterns.
Another important part of the random forest is its ability to provide feature importance rankings that show exactly which features matter the most for customer retention. This helps the business prioritize their retention efforts by focusing on the most influential factors and modify and adapt to the market.
In addition, Random Forest results apply directly into business actions. When the majority of trees agree a customer will leave, the business has to intervene fast. When most trees predict churn, retention campaigns should start. When the trees are evenly split the human insight should intervene to take a decision and approach and when the majority of trees agree a customer will stay, the focus should be on the features that make the customer leave.

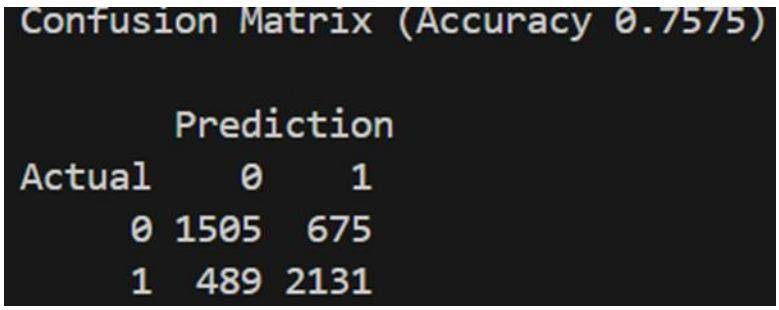
Both models were trained and evaluated using the same methodology to ensure fair comparison and both were created on the 4800 customer training set. For Random Forest the feature importance was selected for interpretation and coefficients for Logistic Regression. In the evaluation process, models generated predictions based on the 1200 customers test set. The performance metrics that were calculated are: accuracy, confusion matrix, precision and recall. The training and test performance were compared to see how well the model performs on the training data and how it performs on the test data, measuring the gap between them.

5.3 Interpretability vs Performance Analysis

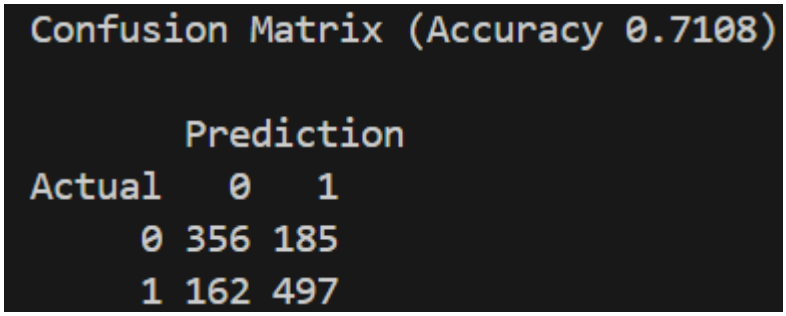
5.3.1 Random Forest Results

The training set of Random Forest accuracy was 75,75%, meaning that 3 636 predictions were correct out of 4800. This indicates that the model correctly classified approximately 3 out of 4 customers. The test set performance had an accuracy of 71,08%, meaning that 853 predictions out of 1200 were correct. The gap between them is 4.67%. The performance drop indicates moderate overfitting, as the model learned some patterns specific to the training data that don't generalize perfectly to the new customers. It is not a huge gap, but it suggests that the model has memorized some noise in addition to learning patterns.

Confusion Matrix – Train Set



Confusion Matrix – Test Set



1. *True Negative (356):* customers who stayed and the model predicted they would stay. This represents correct predictions.

2. *False Positive (185)* : customers who stayed, but the model predicted incorrectly they would churn. This represents wasted retention efforts; the company spends resources on customers who were not at risk.
3. *False Negative (162*): customers who churned but the model didn't identify them. These are missed opportunities; the business lost customers who could have been potentially kept. 555 out of 555 actual customers who churn were missed.
4. *True Positive (497):* Customers who churned and the model got them right. This represents successful identification of at risk customers.

Performance metrics calculated:

- *Precision (* Positive Predicted Value): 497/(497+185) = 72,9% When the model predicts a churn, it's correct 72,9% of the time.
- *Recall* (True Positive Rate): 497 / (497+162) = 75,4% Out of 100 customers who churn, we successfully identify 75.

Feature importance analysis

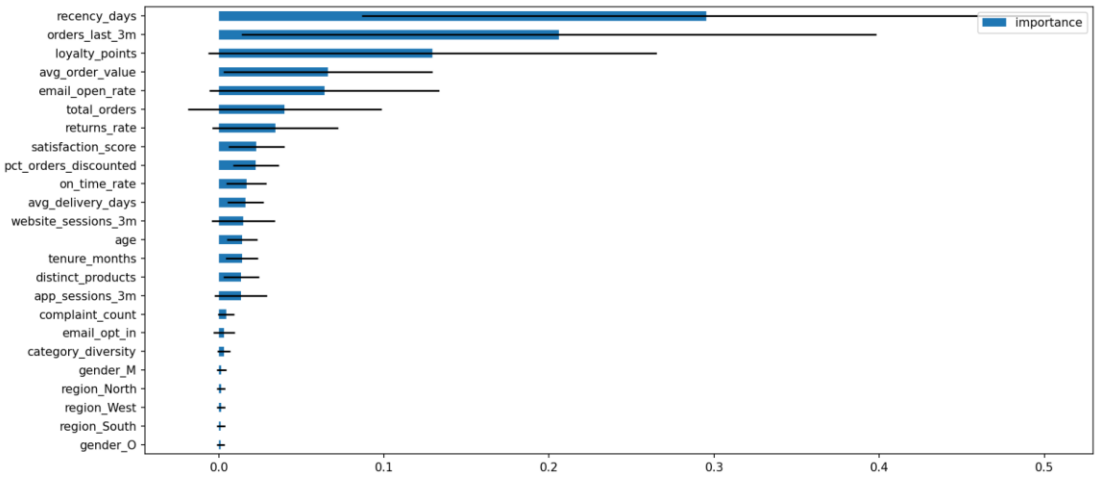
One of Random Forest's key strengths is providing feature importance scores which show each variable contribution to the predictions from highest to weakest.

The most important features result are as it follows:

1. recency_days (Importance 0.2951, Std: 0.2084) . It dominates the model with 29.51% of total predictive power. It is by far the most important churn indicator. High standard deviation indicates variability across trees, but consistent importance.
2. orders_last_3m (Importance: 0.2061, Std: 0.1922) : Second strongest predictor with 20,61% importance meaning recent purchasing frequency is a strong indicator. Together with recency_days, these two features account for 50,12% of the model predictions.
3. loyalty_point (Importance: 0.1294, Std: 0.1358) : Third place with 12,94% importance. High standard deviation means its importance across different branches.
4. avg_order_value and email_open_rate with 6,4% respectively 6,3% importance.

The next ones in order of importance are: total_orders, returns_rate, satisfaction_score, pct_orders_discounted, on_time_rate, avg_delivery_days, website_sessions_3m, age, tenure_months, distinct_products, app_sessions_3m, category_diversity, email_opt_in.

Complaint_count surprisingly only has 0,4% importance. Also demographic contribution is minimal, combined only 0,7%.



The model reveals a clear hierarchy of churn drivers, with behavioral activities dominating over satisfaction scores, demographics and tenure. The top 3 features account for 63,06% of all predictive power.

5.3.2 Logistic Regression Results

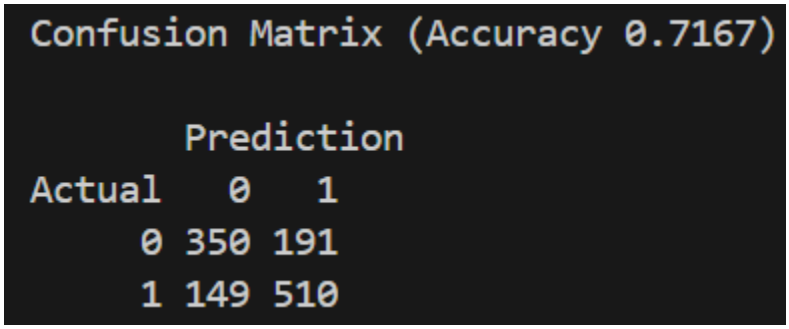
The training set of Logistic Regression accuracy was 72,25%, which means 3 464 predictions were right out of 4800, lower by 3.45% compared to the Random Forest results. The test set had an accuracy of 71,67%, which means 860 predictions were right out of 1200. The gap between them is only 0,5% which demonstrates excellent generalization. The model performs almost identical on new customers as it did during the training.

Confusion Matrix – Train Set

Confusion Matrix (Accuracy 0.7225)

Actual	Prediction 0	1
0	1418	762
1	570	2050

Confusion Matrix – Test Set



Compared to Random Forest:

1. *True Negatives (350)*: 6 fewer than RF. Slightly more staying customers misclassified
2. *False Positives (191)*: again, 6 fewer than RF
3. *False Negatives (149)*: 13 fewer than RF. It predicted more churners than RF
4. *True Positive: (510)*: again, 13 fewer than RF – identifies more at risk customers who actually churn.

Performance metrics calculated:

- *Precision*: 510/(510+192) = 72,64%
- *Recall:* 511/(511*148) = 77,54%

The coefficients with highest positive coefficients:

1. pct_orders_discounted (0.928): Customers who buy mostly on discounts are very likely to leave
2. returns_rate (0.575): customers with high returns rates are unhappy
3. email_opt_in (0,408): even though some customers opted for marketing email, if they don't open them or they are not engaged they are most likely not loyal
4. on_time_rate (0,192): a surprising discovery. It might mean that customers who receive on time deliveries is not enough to keep them loyal

The coefficients with highest negative coefficients:

1. discount_rate_mean (-0.785) : customers who get good discounts stay loyal
2. email_open_rate(-0.513) : customers who open their email are less likely to churn
3. orders_last_3m(-0,114): customers who were active recently will most likely keep going to be active
4. gender_M(-0,046): means that male customers are more loyal

The highest coefficients contribute much more to the predictions while many coefficients contribute minimally. This is the same as Random Forest feature importance, which validates consistency across models.

5.4 Evaluation

The project demonstrates a fundamental principle in machine learning: the most complex model is not always the best for business solutions. The choice between Random Forest and Logistic Regression the interpretability and performance trade off.

While Random Forest shows higher training accuracy, Logistic Regression shows higher accuracy in the test set which means it will identify more customesr who churn. In addition, Logistic Regression's advantage is its interpretability. Each feature gets a clear impact score or coefficient that shows positive score (which makes customers more likely to leave) or negative score (which makes customers more likely to stay). Random Forest has low interpretability because with 500 trees up to 6 levels of depth like in our model, it contains thousands of decisions that a human cannot comprehend. Also, it cannot explain why a specific customer received a churn score beyond saying that "the trees voted this way". However, the interpretability that will get from Random Forest is its feature importance ranking.

In Logistic Regression's case, it's simplicity and stability makes it easier to understand. Every feature has a precise numerical impact that can be explained, it can show which feature drove a customer churn which makes individual predictions explainable and it's easy to communicate: positive coefficents means churn, negative means customesr will stay. Furthermore, coefficients translates directly into business insights (example: each day of inactivity increase the churn by x% or for every unit increase in X%, churn probability changes by Y%) .

Evaluation	Random Forest	Logistic Regression	Winner
Train set accuracy	75,75%	72,25%	Random Forest
Test set accuracy	71,08%	71,67%	Logistic Regression
Stability	-4,67%	0,58%	Logistic Regression
Precision	72,9%	72,64%	Random Forest
Recall	75,4%	77,54%	Logistic Regression
Interpretability	Lower	Higher	Logistic Regression

In our case and database, Logistic Regression would be recommended as a churn prediction model. The marginal training accuracy of Random forest is outweighed by Logistic Regression's higher test accuracy, recall, stability, very close precision rate and complete interpretability. In business context, a model that can perform reliably on new customers and can explain the decision easier is overall better than the one who can memorize the training set better and is slightly more precise.

6. Power BI Visualization

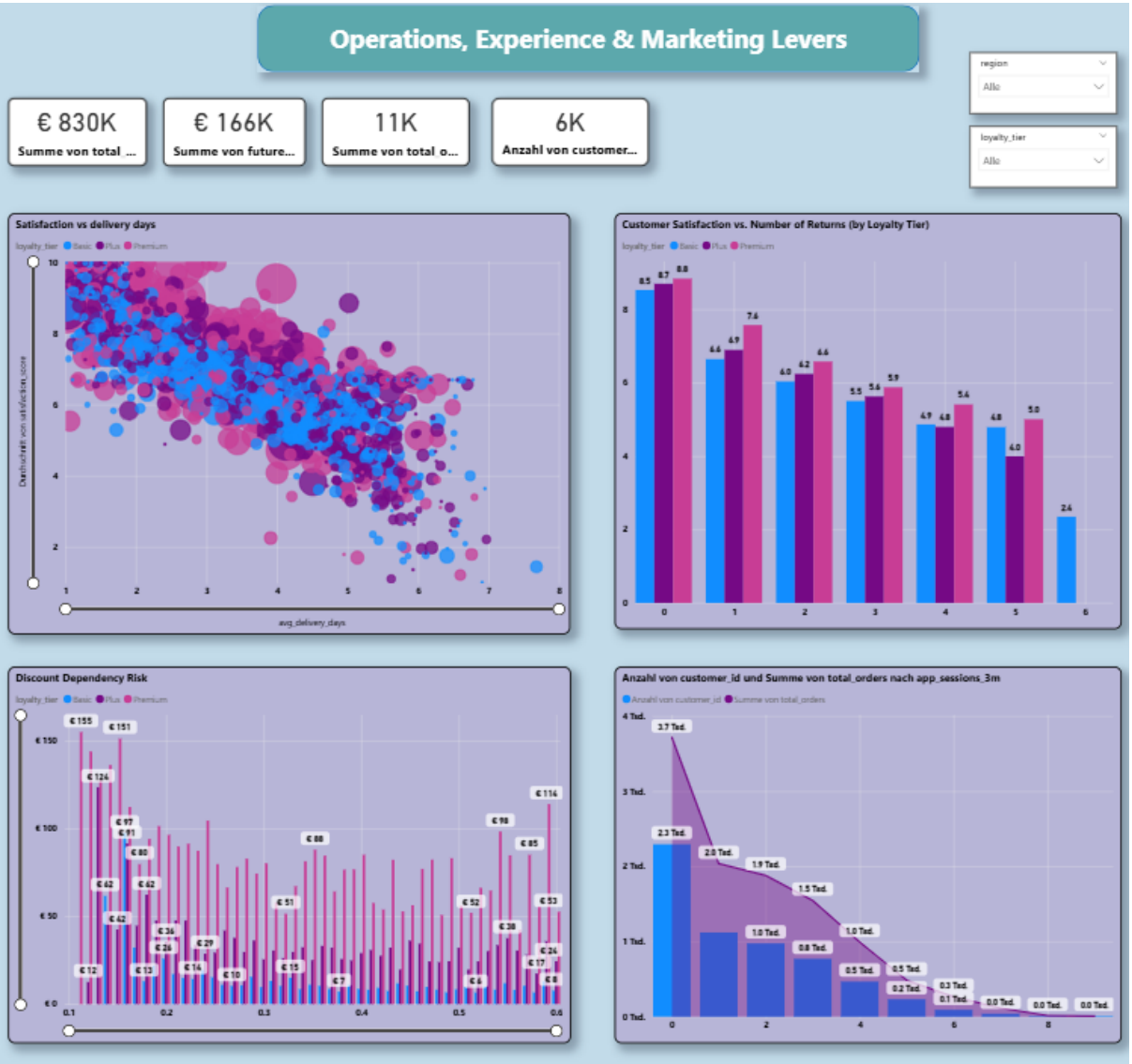
6.1 Customer Value & Retention

This dashboard looks at customer value and retention based on several relevant KPIs. Specifically, it provides an overview of past revenue data vs. future spending forecasts, churn rate and total spent of the different loyalty tiers, and where most customers are from. Filter options also enable the viewer to apply criteria related to e.g. geographical distribution and customer loyalty.

The scatter chart relating total spent to future spent reinforces the correlation between past and projected customer value. A customer who spent a lot in the past is therefore more likely

to spend more in the future, just like customers in higher loyalty tiers are associated with higher spending. The chart looking at churn rate by loyalty level also illustrates that customers at higher tiers churn at a considerably lower rate, supporting the importance of loyalty schemes to the business. Additional bar graphs show the data on total expenditure and number of customers by regions and by level of loyalty, visualizing that Premium customers from the North spend the most, while Basic Customers from the West spend the least. These tables help in strategic planning and prioritization of certain customer segments based on who is spending the most today and most likely to spend the most in the future.

6.2 Operations, Experience & Marketing

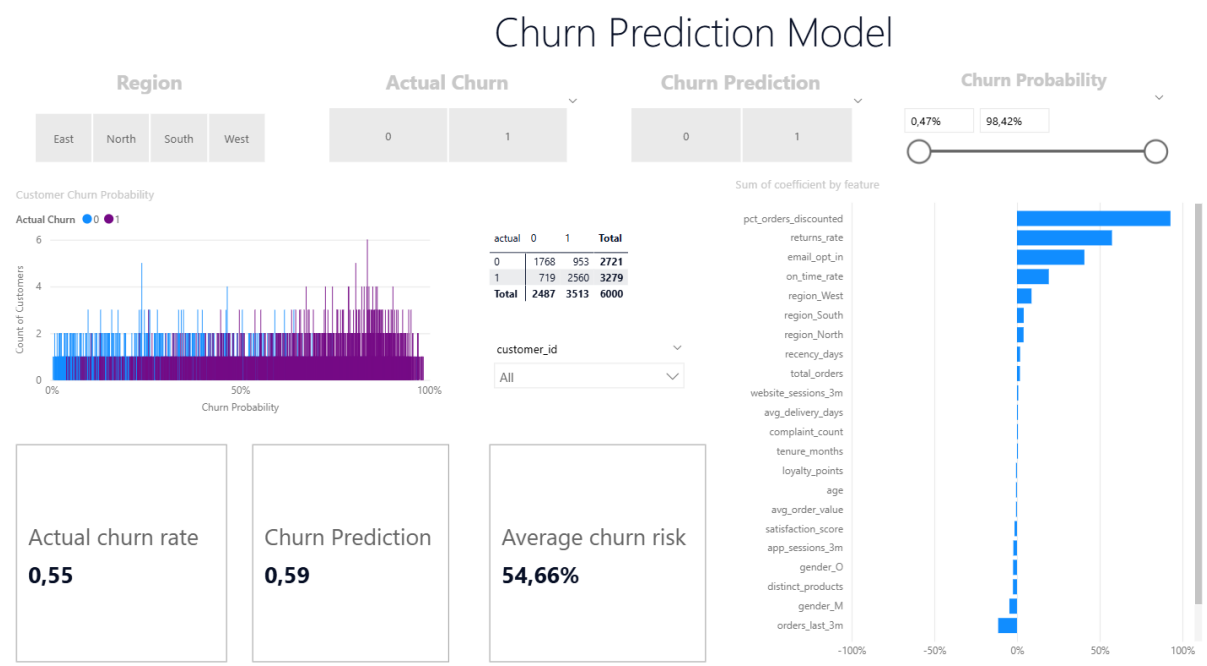


This dashboard examines how different factors like the amount of delivery days, product returns, or discounts influence customer satisfaction throughout different loyalty tiers. It

points to nonfinancial elements impacting customer perception, potentially driving whether customers will stay loyal with a brand over time.

Looking at the tables, we can observe that delivery days and product returns are negatively correlated. Therefore, the longer a delivery takes or the more products are returned, the less satisfied the customer. This pattern is consistent across all loyalty tiers.
The discount dependency risk analysis reveals that there is an inverted relationship between discounts and future spending. While discounts can promote short-term sales, especially high-tier customers have the tendency to rely too strongly on discounts. From a business perspective, the risk of this is consumers being conditioned to buy only when there is a discount.
The last chart shows that many customers only visit the app a few times, yet those sessions contribute to most of the orders. Unfortunately, more sessions in the app do not necessarily convert to more orders, which creates a problem of usability or app value. All in all, more sessions in the app does not necessarily mean more orders being placed.

6.3 Churn Prediction



This table offers an interactive and actionable visual representation of churn metrics, and can therefore be helpful to assist operational decision making. It integrates actual churn results, churn class, and churn probabilities, so that the user will be able to assess the model results both at a macro- and individual customer level. This is facilitated through filters such as region, actual churn, predicted churn, and probability threshold.

The confusion matrix focuses on differentiating churners from non-churners, and assists in making classification errors immediately visible. In addition, a probability distribution of users churning helps understanding how confidently the model splits users into the ones who stay and the ones who churn. Key performance metrics relate to actual churn, predicted churn, and average churn risk, providing another customer retention overview.

The feature contribution bar chart makes the results more easy to interpret. It ranks variables based on their predicted relevance for churn. By doing so, businesses are better able to explain why a consumer is at risk of churning. Overall, this table presents a statistical model of churn, now made clear, filterable, and ready for a business setting.

7. Conclusion

The key insights in this project was obtained through the correlation analysis, dimensionality reduction, and prediction modeling. Through the correlation matrix, we were able to discern the behavior and financial patterns within the customer data set, enabling us to look past the superficial level of customer information to see how they are truly engaging with the business. High positive correlations among total_spent, total_orders, and future_3m_spend confirmed the observation that past spending behavior is the key predictor of future earnings. On the other hand, the negative relationship of recency_days to past order behavior reinforced the importance of pulse checks for customer engagement and early warning systems against customer churn.

The correlation analysis also suggested the point that customer discontent does not always lead to customer churn. The data on the number of complaints demonstrated an insignificant link with the recency score, which means that the consumers will continue to purchase the merchandise despite their discontent. The link between the engagement score and the points earned also demonstrated that digital engagement acts as the leading determinant of customer loyalty.

Based on the above observations, we proceeded to perform Principal Component Analysis on the dataset. The targeted PCA revealed that most variables were sufficient enough to cover most aspects of financial value on the client side, while the global PCA revealed that 80% of the total information from 30 variables could be covered by 12 principal components, which confirmed that client behavior could be captured without any loss of accuracy. The principal components explained various behavioral patterns on the client side, dividing financial value, online engagement, and client behaviors.

These findings were directly applied to the predictive modeling component. Within the regression analysis, the training- and test RMSE were practically the same, which indicated strong generalization. Spend directly correlated to revenue, and other factors like

satisfaction and recency were shown to directly affect future revenue. From this we could observe that customer experience is not just subjective, but a relevant and measurable factor for financial performance.

For the classification, we observed that the importance of recent activity and engagement drives the result for churn significantly, even more than demographics and long-run features. Though Random Forest has high training accuracy, Logistic Regression offers good opportunities for cause-effect analysis, making it more practical for business use.

8. Appendix

8.1 Queries

8.1.1 Ludwig

Query 1: Revenue per payment method

Identifies which payment methods generate the highest total revenue. Useful for optimizing payment options, negotiating fees, and prioritizing the most profitable payment channels.

```
SELECT payment_method, SUM(total_amount) AS total_revenue
FROM orders
GROUP BY payment_method;
```

Query 2: Orders per month

Shows how order volume changes over time on a monthly basis. Useful for identifying seasonality, demand trends, and planning inventory and staffing.

```
SELECT
  YEAR(order_date) AS year,
  MONTH(order_date) AS month,
  COUNT(*) AS order_count
FROM orders
```

```
GROUP BY YEAR(order_date), MONTH(order_date)

ORDER BY year, month;
```

Query 3: Number of customers per region

Displays in which regions the customers are located. This could support regional marketing efforts and help in showing where the customer base is the strongest.

```
SELECT region, COUNT(*) AS customer_count

FROM customers

GROUP BY region;
```

Query 4: Product return rate

Calculates the return rate of every product by completed orders. Supports the identification of problem items, enhancement of quality control, and a reduction in reverse logistics costs.

```
SELECT

   p.product_id,

   p.brand,

   COUNT(DISTINCT r.return_id) * 1.0 / COUNT(DISTINCT oi.order_id) AS return_rate

FROM products p

JOIN order_items oi ON p.product_id = oi.product_id

LEFT JOIN returns r

   ON oi.order_id = r.order_id

  AND oi.product_id = r.product_id

GROUP BY p.product_id, p.brand;
```

Query 5: On-time delivery rate

Measures the percentage of deliveries that take place on time. It serves as a guide for evaluating the performance of the logistics and how it impacts customer satisfaction.

```
SELECT
  SUM(CASE WHEN on_time = 1 THEN 1 ELSE 0 END) * 1.0 / COUNT(*) AS on_time_rate
FROM deliveries;
```

Query 6: Average delivery time in days

Calculates the time from an order being placed to that same order being delivered. This helps evaluate shipping efficiency and find delays in fulfillment.

```
SELECT
  d.order_id,
  DATEDIFF(DAY, o.order_date, d.delivered_date) AS delivery_days
FROM deliveries d
JOIN orders o ON d.order_id = o.order_id
WHERE d.delivered_date IS NOT NULL;
```

Query 7: Average spend by loyalty tier

To see how average order value (AOV) differs between loyalty tiers. It can help in assessing the efficiency of loyalty programs or tailoring incentives to different customer groups.

```
SELECT
  c.loyalty_tier,
  AVG(o.total_amount) AS avg_spend
FROM customers c
JOIN orders o ON c.customer_id = o.customer_id
GROUP BY c.loyalty_tier
ORDER BY avg_spend DESC;
```

Query 8: Click-through rates per channel

Allows to analyze customer engagement by computing the click-through rate for each of the channels. This might help in comparing the effectiveness between different marketing and communication channels.

```
SELECT
   channel,
   SUM(CASE WHEN clicked = 1 THEN 1 ELSE 0 END) * 1.0 /
   NULLIF(SUM(CASE WHEN opened = 1 THEN 1 ELSE 0 END), 0) AS ctr
FROM interactions
GROUP BY channel;
```

8.1.2 Dimitrios

Query 1: Top customers by lifetime value
Identifies the customers who drive the most revenue. Useful for VIP programs, retention offers, and prioritizing high-value sections instead of throwing marketing budget out.

```
SELECT TOP 20 c.customer_id, c.first_name, c.last_name, SUM(o.total_amount) AS lifetime_value
FROM customers c JOIN orders o ON c.customer_id = o.customer_id
GROUP BY c.customer_id, c.first_name, c.last_name
ORDER BY lifetime_value DESC;
```

Query 2: Delivery speed by carrier
See which carriers are fast and which ones are slow. This information helps the company improve shipping, cut better deals with carriers, and make customers happier.

```
SELECT carrier, AVG(DATEDIFF(DAY, shipped_date, delivered_date)) AS avg_delivery_days
FROM deliveries
WHERE shipped_date IS NOT NULL AND delivered_date IS NOT NULL
GROUP BY carrier;
```

Query 3: Average order value by region
This shows how much people spend on average in different areas. Useful for tailored pricing, promotions, product mix, and marketing strategies per region, instead of using a general approach.

```
SELECT c.region, AVG(o.total_amount) AS avg_order_value

FROM orders o

JOIN customers c ON o.customer_id = c.customer_id

GROUP BY c.region;
```

Query 4: Orders, average order value, and lifetime value per customer
This query shows which customers spend a lot, and which ones don't, or how often they buy. It also provides insights into who might stop buying and how to make things more personal for customers.

```
SELECT customer_id, COUNT(*) AS total_orders,
 AVG(total_amount) AS avg_order_value,
 SUM(total_amount) AS lifetime_value

FROM orders

GROUP BY customer_id;
```

Query 5: Most popular products
If the company knows what products are most popular, it makes it easier to decide what to stock, how to group its products, and what to suggest as extra buys. It also shows which products must focus on advertising.

```
SELECT TOP 10 p.product_id, p.category, COUNT(oi.order_id) AS total_orders

FROM order_items oi

JOIN products p ON oi.product_id = p.product_id

GROUP BY p.product_id, p.category

ORDER BY total_orders DESC;
```

Query 6: Monthly revenue trend.
It helps the company see its development, its seasonality, and detect any anomalies. This information helps predict sales or check how ads are working out. It also points out problems before they get big.

```
SELECT FORMAT(order_date, 'MM-yyyy') AS month,
        COUNT(*) AS num_orders,
        SUM(total_amount) AS total_revenue
FROM orders
GROUP BY FORMAT(order_date, 'MM-yyyy')
ORDER BY month;
```

Query 7: Most used channel per customer.
This tells us which way customers like to interact. It's important to understand how we engage with them, making sure our messages are personal and avoiding bothering them with irrelevant content.

```
SELECT i.customer_id, i.channel, COUNT(*) AS interaction_count
FROM interactions i
GROUP BY customer_id, channel
HAVING COUNT(*) = (SELECT MAX(channel_count)
FROM ( SELECT COUNT(*) AS channel_count
FROM interactions
WHERE customer_id = i.customer_id
GROUP BY channel) sub)
ORDER BY customer_id;
```

Query 8: Active vs inactive customers.
This helps the company identify those who remain engaged and those who have lost interest. We can tell which customers are disengaging, offer benefits to our loyal customers, and improve the management of our entire customer base.

```
SELECT c.customer_id,

CASE
    WHEN MAX(o.order_date) >= DATEADD(MONTH, -3, GETDATE()) THEN 'Active'
    WHEN MAX(o.order_date) < DATEADD(MONTH, -6, GETDATE()) THEN 'Inactive'
    ELSE 'Occasional'
 END AS customer_status

FROM customers c

LEFT JOIN orders o ON c.customer_id = o.customer_id

GROUP BY c.customer_id;
```

8.1.3 Alex

1. Best 10 selling products by company

It shows the most important products sold by the company in terms of revenue. Knowing this, we can prepare restocking options and be prepared to always have them available to sell as they produce the most revenue. Also it's important for marketing campaigns and companies can develop new products based on the ones that are selling the best.

```
SELECT TOP 10

p.product_id,

p.category,

p.brand,

SUM(oi.quantity) AS total_sold

FROM order_items oi

JOIN products p ON oi.product_id = p.product_id

GROUP BY p.product_id, p.category, p.brand

ORDER BY total_sold DESC;
```

2. Most common return reason

It shows why each product was returned and what quality problems it had. We can adress these problems to each department involved and we can prevent this from happening again in the future and fix the current problems.

```
SELECT reason,

COUNT(*) AS return_count

FROM returns

GROUP BY reason

ORDER BY return_count DESC;
```

3. Complaints per region

It shows the geographical areas with the most issues. If a region has much more issues than the others, most likely the customers isn't the problem but there might be a breakdown in the local operations.

```
SELECT c.region,

COUNT(i.interaction_id) AS complaint_count

FROM interactions i

JOIN customers c ON i.customer_id = c.customer_id

WHERE i.complaint = 1

GROUP BY c.region

ORDER BY complaint_count DESC;
```

4. Top brands by revenue

This one shows the most profitable brands so we can strengthen partnerships with them, negotiate better terms with suppliers and for marketing campaigns. Also, it's different from the first query because even if a brand sells a lot of products, their price and profit margin might be different. For example, a brand that sells 100 products for 1000 euros each will produce more money than a brand that sells 1000 products for 1 euro each.

```
SELECT TOP 10 p.brand,

SUM(o.total_amount) AS total_revenue

FROM orders o

JOIN order_items oi ON o.order_id = oi.order_id

JOIN products p ON oi.product_id = p.product_id

GROUP BY p.brand

ORDER BY total_revenue DESC;
```

5. Product category performance per region

This can show the different preferences across regions. This is useful for marketing campaigns and to expand the product catalogue with similar products that are selling well in a region compared to others, creating better expansion opportunities

```
SELECT c.region,p.category,


COUNT(DISTINCT o.order_id) AS order_count,


SUM(o.total_amount) AS regional_category_revenue


FROM orders o

JOIN customers c ON o.customer_id = c.customer_id

JOIN order_items oi ON o.order_id = oi.order_id

JOIN products p ON oi.product_id = p.product_id
```

```
GROUP BY c.region, p.category

ORDER BY c.region, regional_category_revenue DESC;
```

6. Customer by spending

This one shows the customers who are bringing the most revenue to the company. This could be used for different loyalty programs, marketing campaigns (see their profile and know how to adapt to that type of customer). Also we can create special discounts so they can spend more money and come back to us.

```
SELECT TOP 10

   c.customer_id,

   c.first_name,

   c.last_name,

   c.loyalty_tier,

COUNT(DISTINCT o.order_id) AS total_orders,

SUM(o.total_amount) AS total_spent,

MAX(o.order_date) AS last_order_date

FROM customers c

JOIN orders o ON c.customer_id = o.customer_id

GROUP BY c.customer_id, c.first_name, c.last_name, c.loyalty_tier

ORDER BY total_spent DESC;
```

7. Months with the highest number of orders and monthly revenue

This one analyzes the peak and dry season and it helps the business prepare the staff and inventory for when the busy season starts.

```
SELECT

   YEAR(order_date) AS year,
```

```
    MONTH(order_date) AS month,
    SUM(total_amount) AS monthly_revenue,
    COUNT(*) AS number_of_orders
FROM orders
GROUP BY YEAR(order_date), MONTH(order_date)
ORDER BY number_of_orders DESC;
```

8. How much do customers spend on average per order by region?

We can identify the regions that are “the richest”, where we can usually sell more premium products and we can tailor the price strategies regionally.

```
SELECT
    c.region,
    AVG(o.total_amount) AS average_order_value,
    COUNT(o.order_id) AS number_of_orders
FROM customers c
JOIN orders o ON c.customer_id = o.customer_id
GROUP BY c.region
ORDER BY average_order_value DESC;
```

8.1.4 Ahsan

```
– 1. Find products priced above $100, sorted by price –
-- Business Value: Identifies high-ticket items that drive the most revenue --
SELECT *
FROM products
```

```
WHERE price > 100

ORDER BY price DESC


-- 2.Identify customers who have opted into email marketing

-- Provides a clean list for the marketing team to send newsletters and promotions

SELECT first_name, last_name, email_opt_in

FROM customers

WHERE email_opt_in = 1


-- 4.Number of orders placed in each region

-- Helps in logistics planning and deciding where to build new warehouses

SELECT c.region, COUNT(o.order_id) AS order_count

FROM customers c

JOIN orders o ON c.customer_id = o.customer_id

GROUP BY c.region


-- 5.Brand-Level Return Severity Report

-- This  provides a direct look at Revenue Leakage and Product Quality

SELECT p.brand, p.product_id, COUNT(r.return_id) AS total_items_returned,
SUM(r.refund_amount) AS total_refund_amount

FROM products p

JOIN returns r ON p.product_id = r.product_id

GROUP BY p.brand, p.product_id

ORDER BY total_refund_amount DESC
```

```
-- 6.Late Delivery

-- This query identifies any tier of customers who experienced late deliveries

SELECT c.customer_id, c.first_name, c.last_name, c.loyalty_tier, o.order_id, d.carrier,
d.on_time

FROM customers c

JOIN orders o ON c.customer_id = o.customer_id

JOIN deliveries d ON o.order_id = d.order_id

WHERE d.on_time = 0

ORDER BY loyalty_tier DESC


-- 7. Regional Product Category Popularity

-- This tells the business what to stock in different warehouses.

SELECT c.region, p.category, SUM(oi.quantity) AS units_sold

FROM customers c

JOIN orders o ON c.customer_id = o.customer_id

JOIN order_items oi ON o.order_id = oi.order_id

JOIN products p ON oi.product_id = p.product_id

GROUP BY c.region, p.category

ORDER BY c.region, units_sold DESC


-- 8.Category wise Revenue

-- This calculates Total Revenue per Category

SELECT p.category, SUM(oi.quantity * p.price) AS gross_revenue

FROM products p

JOIN order_items oi ON p.product_id = oi.product_id
```

```
JOIN orders o ON oi.order_id = o.order_id

GROUP BY p.category

ORDER BY gross_revenue DESC
```