TÉCNICO LISBOA

Data Science Engineering
Project Report

Team nr: 3	**Student 1 :** Francisco Lourenço	**IST nr:** 13018
	Student 2 : João Valente Martins	**IST nr:** 13020
	Student 3 : Eugenia Cozma	**IST nr:** 13016

This document presents a template for the Data Science Project report. It specifies the mandatory format and suggests the structure to follow. All text with grey background shall be replaced with the analysis made over the datasets.

1 DATA PROFILING

Dataset 1 Link: https://www.kaggle.com/datasets/mkechinov/ecommerce-behavior-data-from-multi-category-store

Dataset 2 Link: https://drive.google.com/file/d/12H5kAUEgcMeELN4VeS56UZPnTRgKXO97/view (official description)

Dataset 3 Link: https://www.kaggle.com/datasets/mkechinov/direct-messaging/data?select=messages-demo.csv

Filtered Dataset 1 with only add-to-cart users. Filtered North American Users for Dataset 2. Filtered Dataset 3 to only opened campaigns. All datasets are about online stores' user behavior.

Feature Generation

In all datasets additional datetime variables were generated from date or timestamp variables. In dataset 1 and 2 there was split of symbolic hierarchical variables (category code X, page path level X).

Dataset	Variable	Formula
Dataset 1	category code	df cart['category code lvl 1'] = df cart['category code'].str.split(".").str[0] df cart['category code lvl 2'] = df cart['category code'].str.split(".").str[1]
Dataset 2	page location	split columns = data['page location'].str.split('/', n=4, expand=True) data['domain'] = split columns[0] # url domain data['page path level 1'] = split columns[1].replace('', pd.NA) data['page path level 2'] = split columns[2].replace('', pd.NA) data['page path level 3'] = split columns[3].replace('', pd.NA)

Dataset 1, 2, 3	event time, event timestamp, event date, sent at	df cart['event time'] = pd.to datetime(df cart['event time'], utc=True)
		df cart['week of month'] = df cart['event time'].apply(lambda x: (x.day - 1) // 7 + 1)
		df cart['is weekend'] = df cart['event time'].dt.weekday.apply(lambda x: "weekend" if x >= 5 else "weekday")
		
		df cart['day of week'] = df cart['event time'].dt.day name()# Monday=0, Sunday=6
		df cart['day'] = df cart['event time'].dt.day
		df cart['hour'] = df cart['event time'].dt.hour
		df cart['min'] = df cart['event time'].dt.minute
		def get time of day(hour):
		if 5 <= hour < 12:
		return 'morning'
		elif 12 <= hour < 18:
		return 'afternoon'
		elif 18 <= hour < 22:
		return 'evening'
		else:
		return 'night'
		
		df cart['time of day']= df cart['hour'].apply(get time of day)

Data Dimensionality

In all 3 datasets, we have way more records than variables, all types of variables except date (because they were transformed to numeric), and overall, more discrete variables than numeric ones. Dataset 1 has more numeric variables, dataset 2 more symbolic and dataset 3 more binary. This analysis already includes the generated features. Datasets 2 and 3 have around 50% of missing values in some variables which will have to be dealt with in a later step. The 3 datasets were sampled from original datasets with millions of records, to reduce slow data processing.

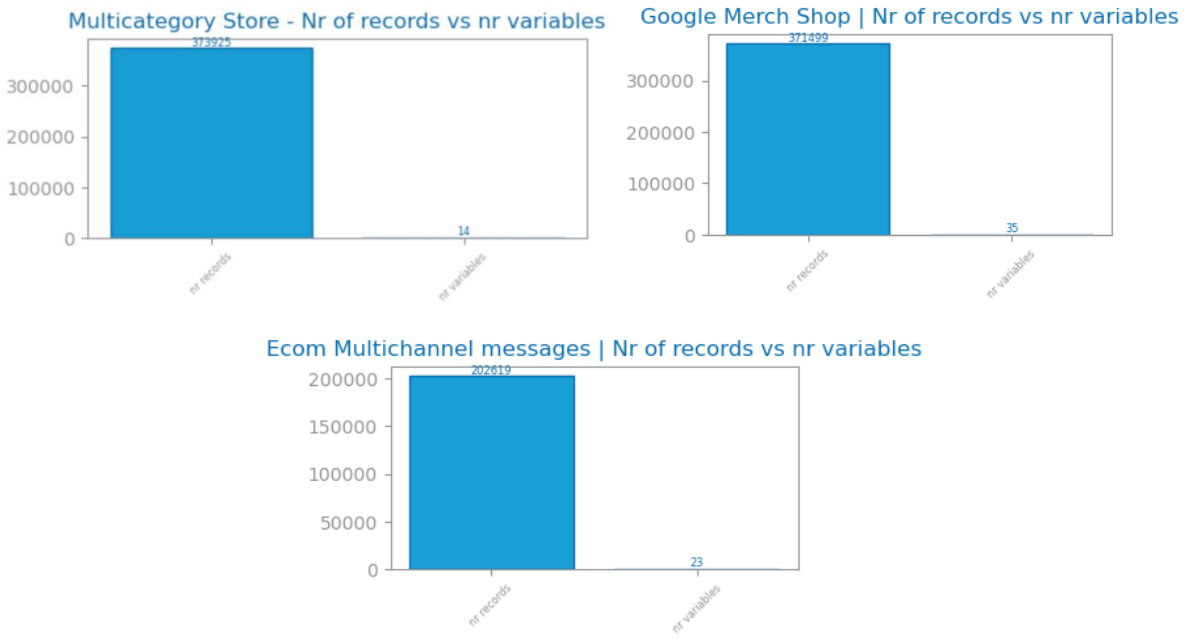


Figure 1 Nr Records x Nr variables for dataset 1 (left), dataset 2 (middle), and dataset 3 (right)

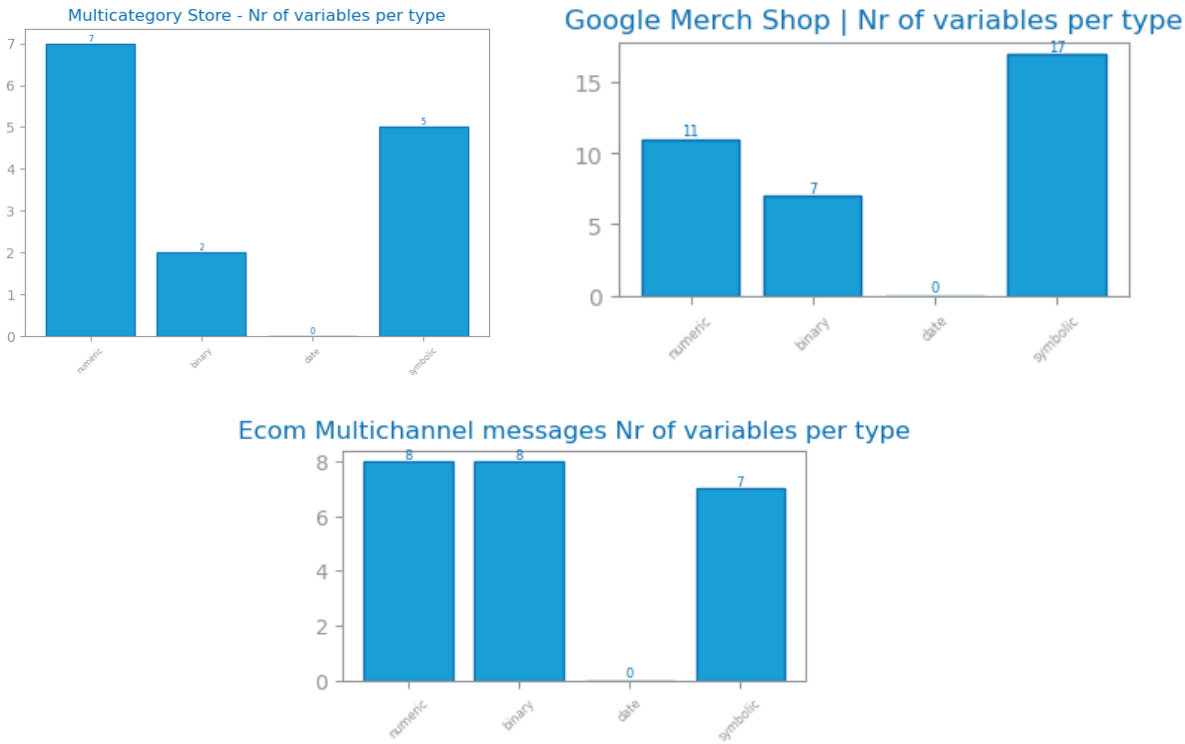


Figure 2 Nr variables per type for dataset 1 (left), dataset 2 (middle), and dataset 3 (right)

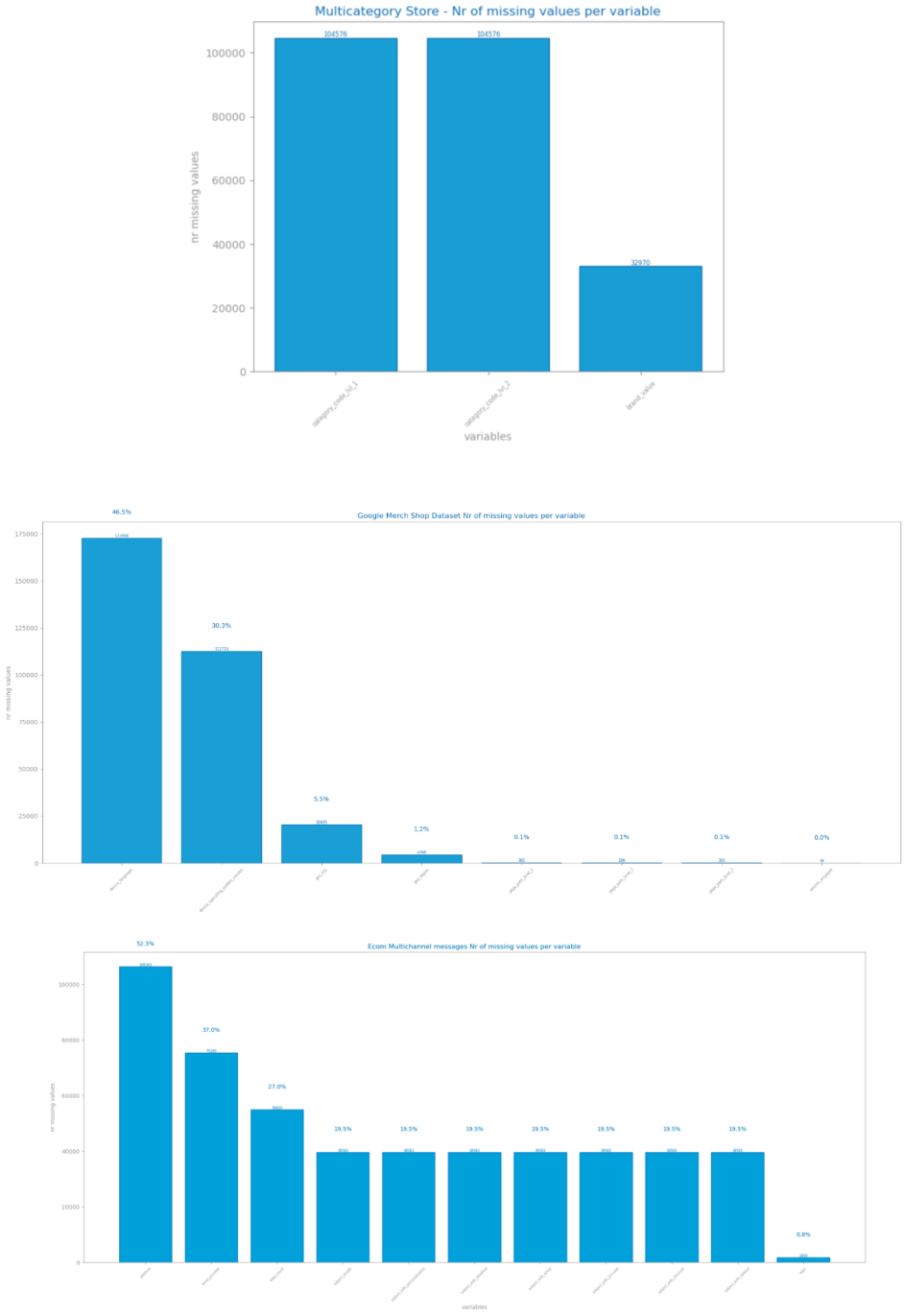


Figure 3 Nr missing values for dataset 1 (left), dataset 2 (middle), and dataset 3 (right)

Data Distribution

All datasets are unbalanced ranging from 13% to 25% minority class. In all datasets, most of its symbolic variables were biased towards higher frequency of a few values which could affect models' performances. The 3 datasets had different ranges for date generated variables. There are also different kinds of numeric variables with different ranges across the datasets like brand value in dataset 1, engagement time msec in dataset 2 and total count in dataset 3. Some of these are the variables with the most outliers.

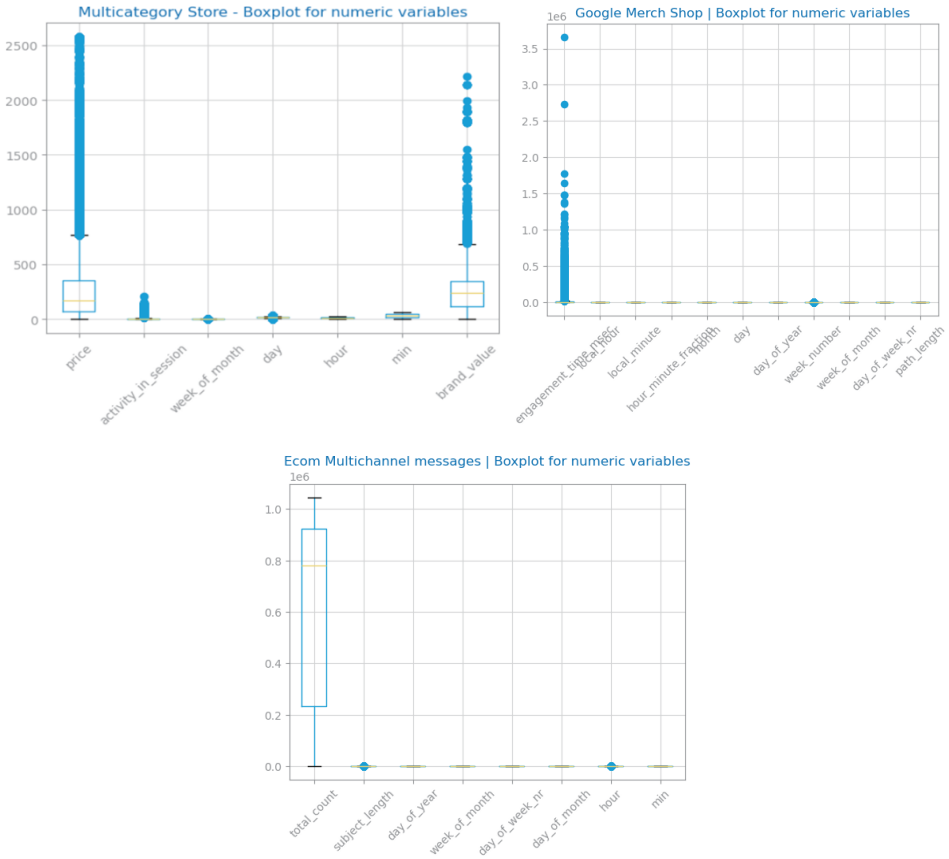


Figure 4 Global boxplots dataset 1 (left), dataset 2 (middle), and dataset 3 (right)

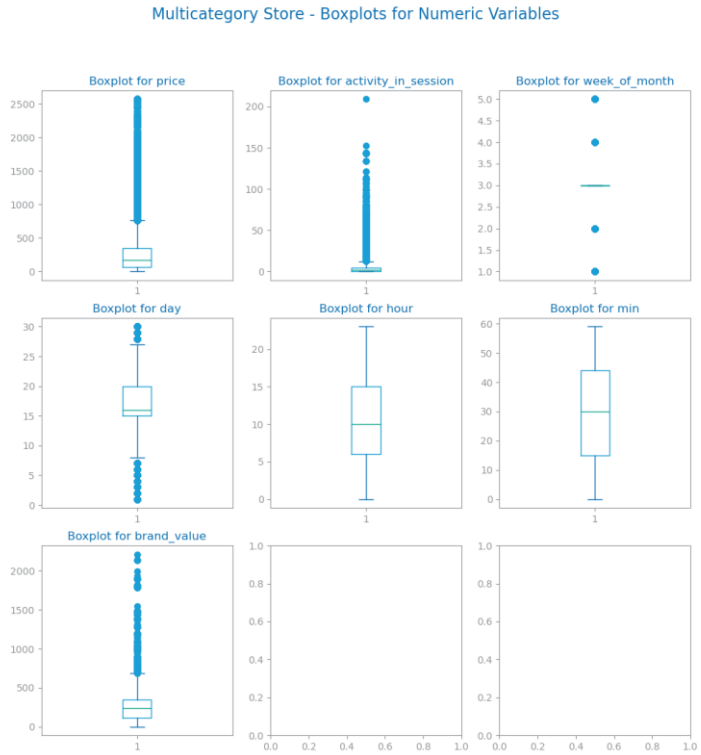


Figure 5 Single variable boxplots for dataset 1

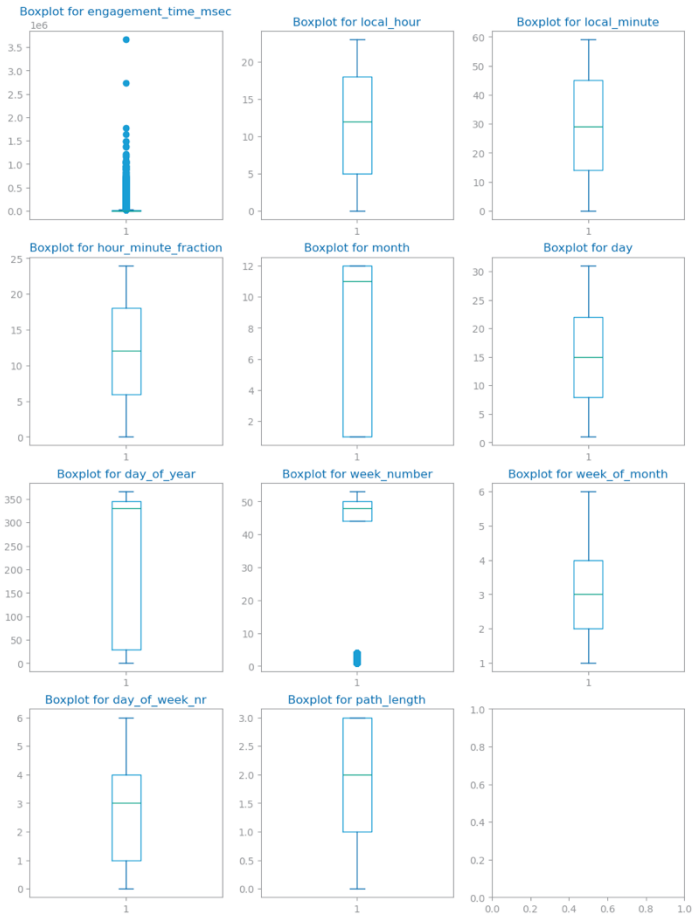


Figure 6 Single variable boxplots s for dataset 2

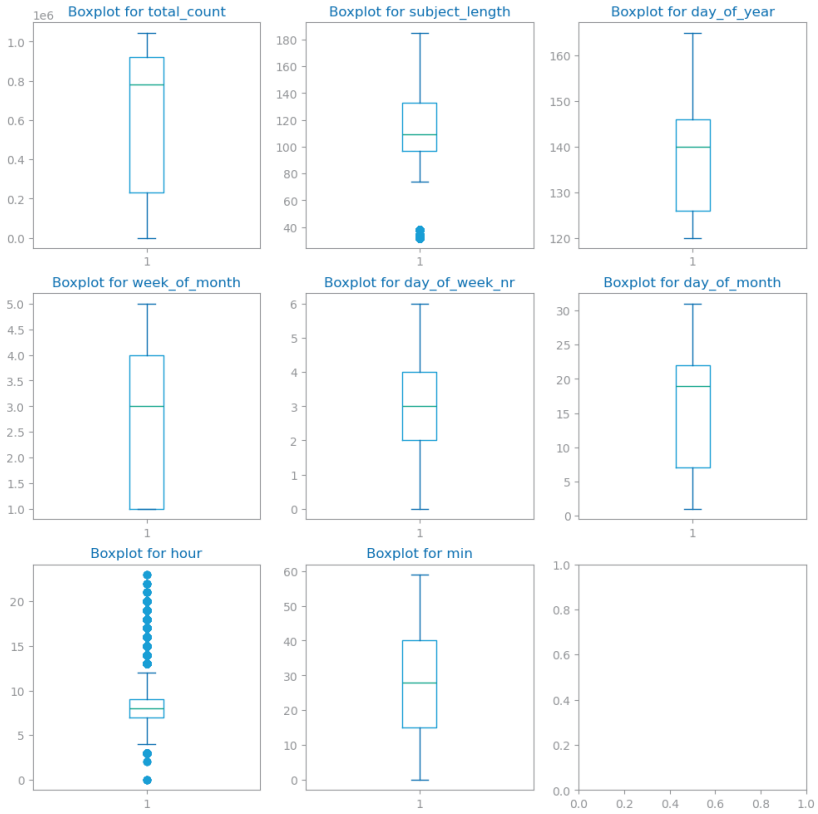


Figure 7 Single variable boxplots for dataset 3

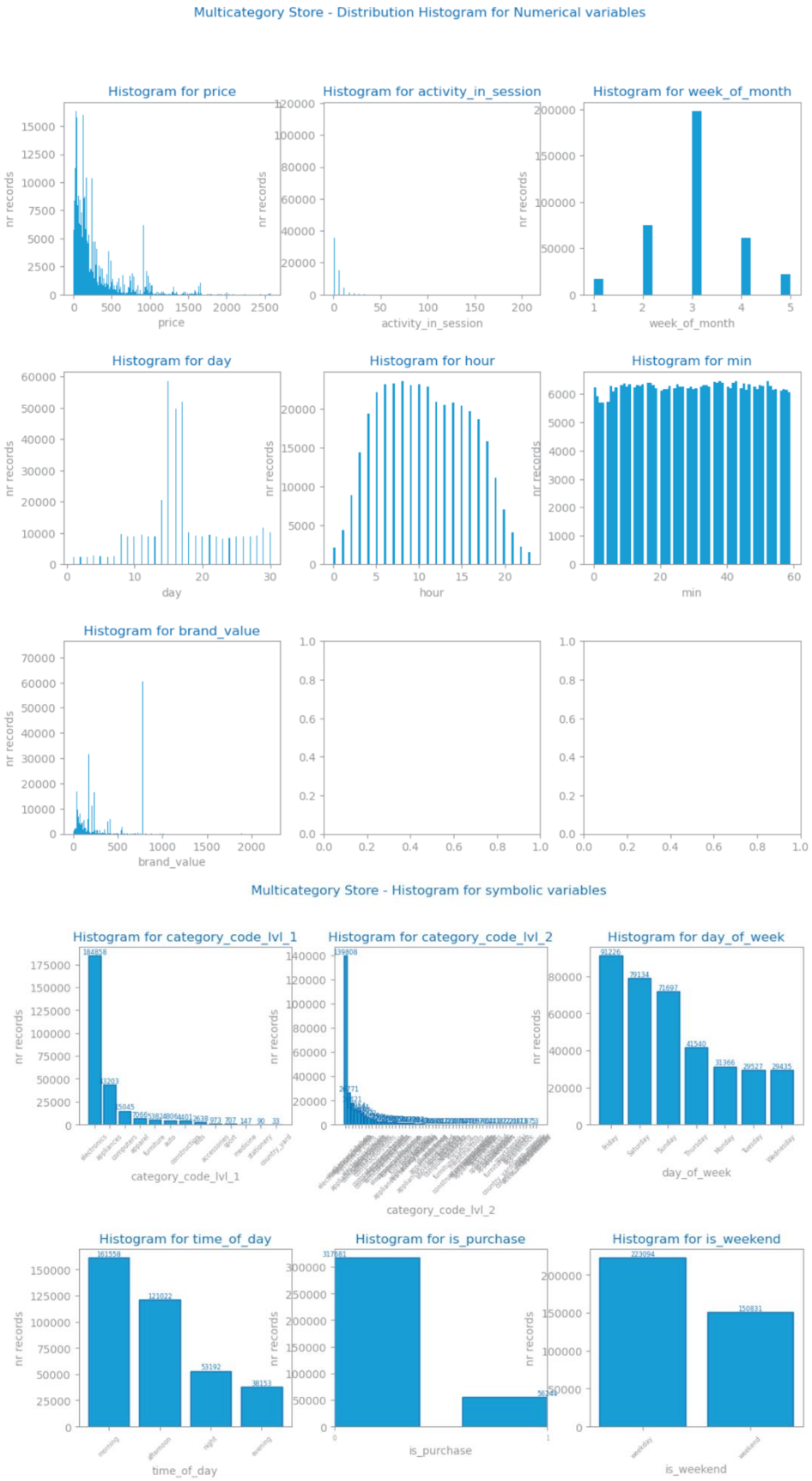
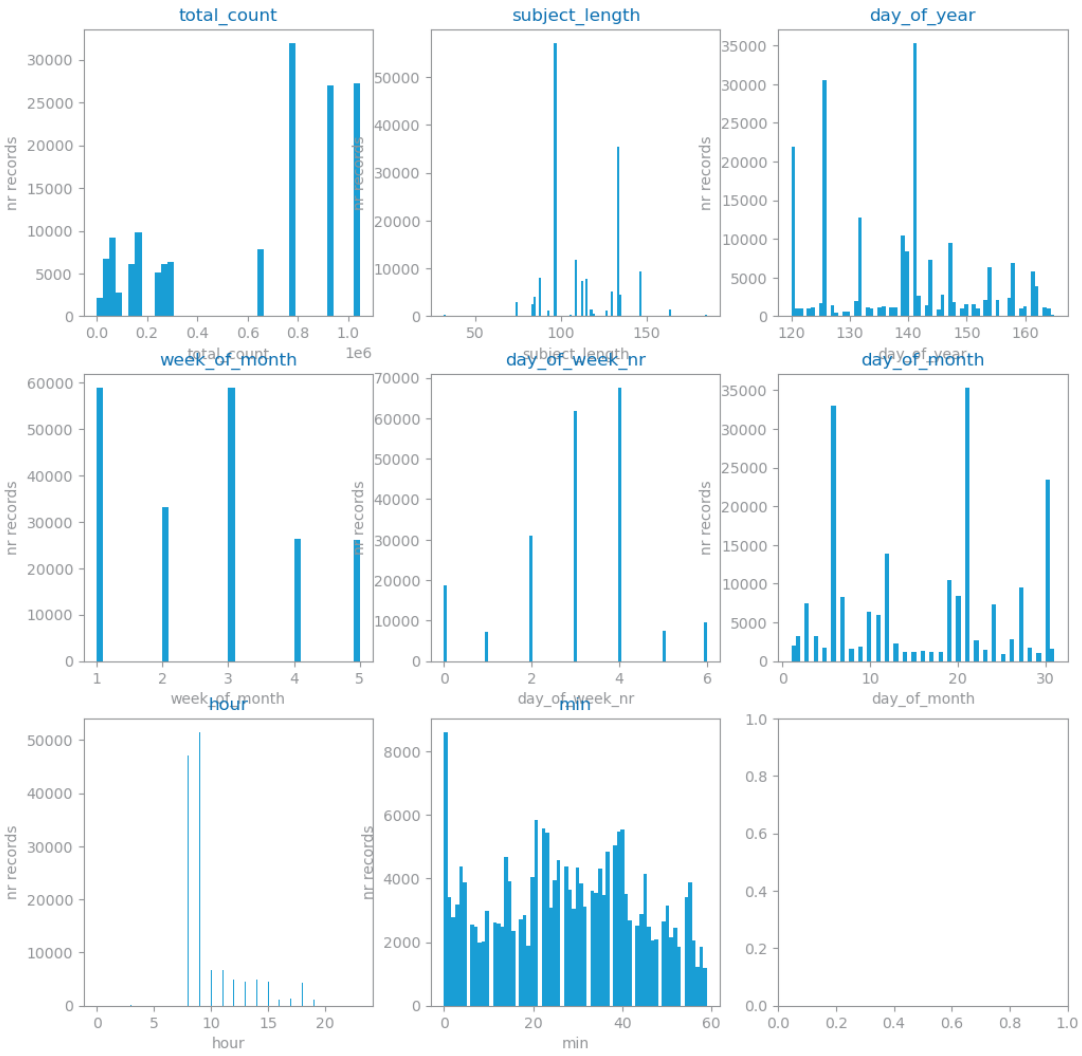


Figure 8 Histograms for dataset 1 (with distributions is enough)

Figure 9 Histograms for dataset 2 (with distributions is enough)


Ecom Multichannel messages | Histogram for numeric variables
total_count
subject_length
day_of_year
week_of_month
day_of_week_nr
day_of_month
hour
min
nr records
1e6

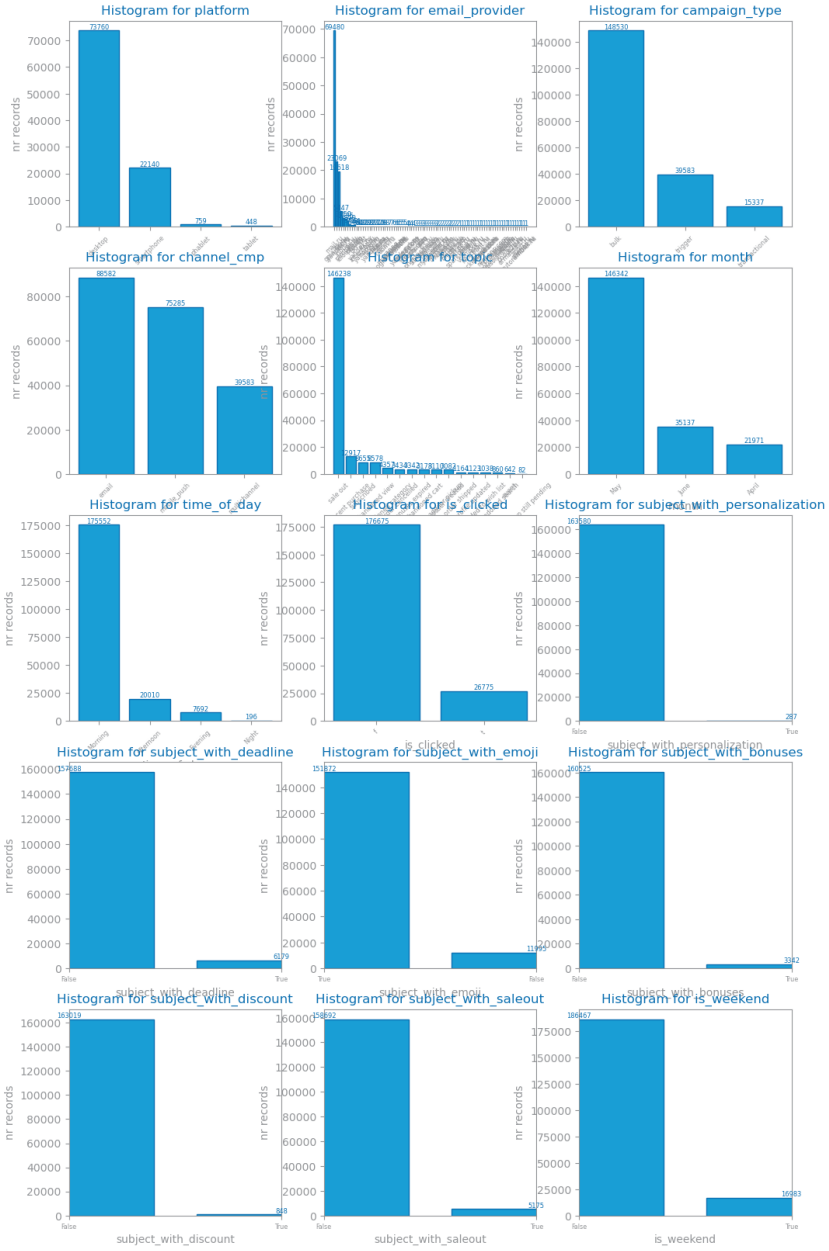


Figure 10 Histograms for dataset 3 (with distributions is enough)

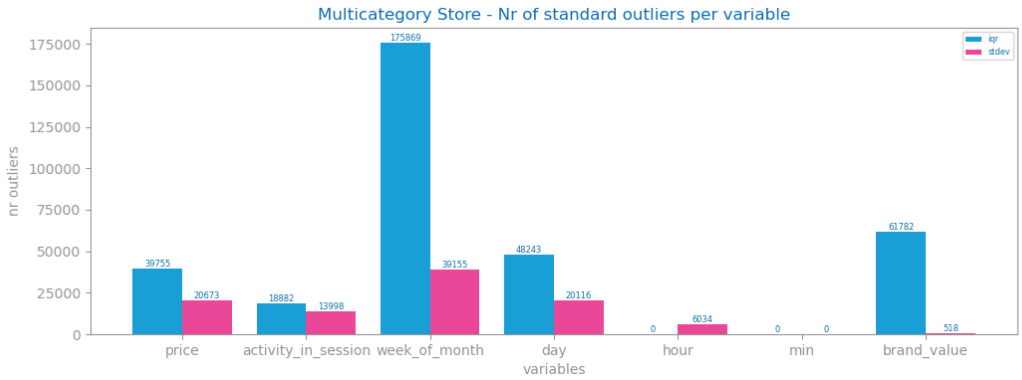


Figure 11 Outliers study dataset 1

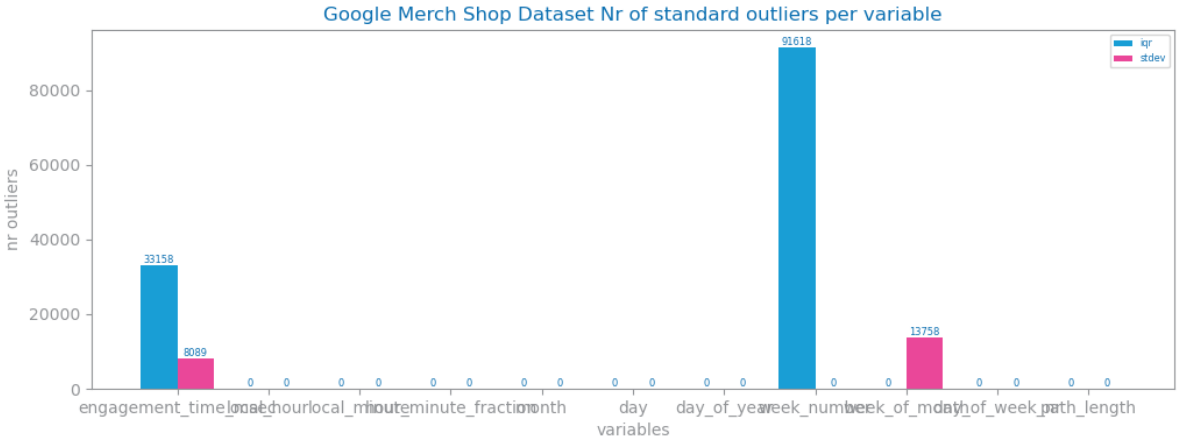


Figure 12 Outliers study for dataset 2

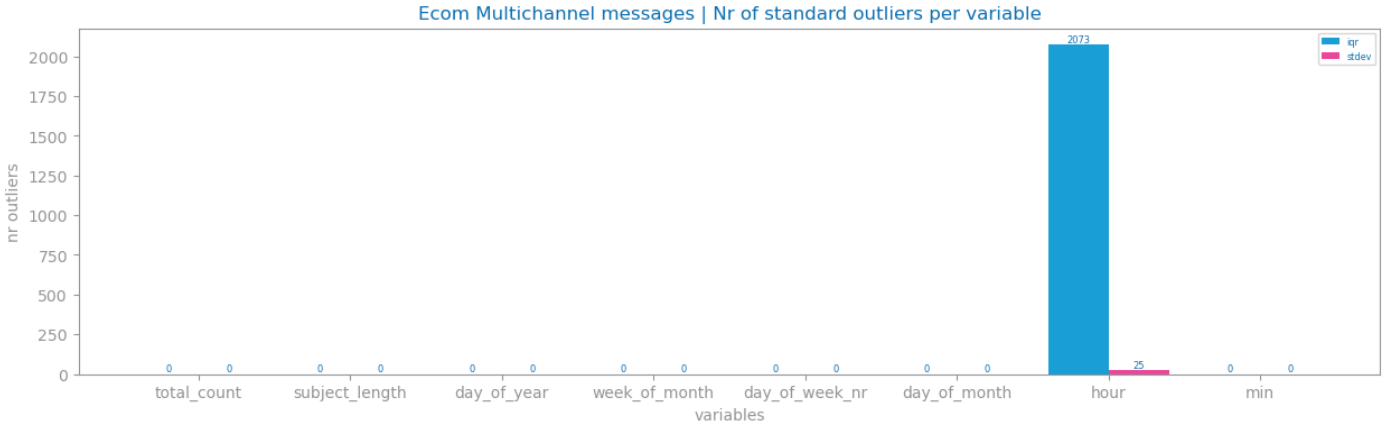


Figure 13 Outliers study for dataset 3

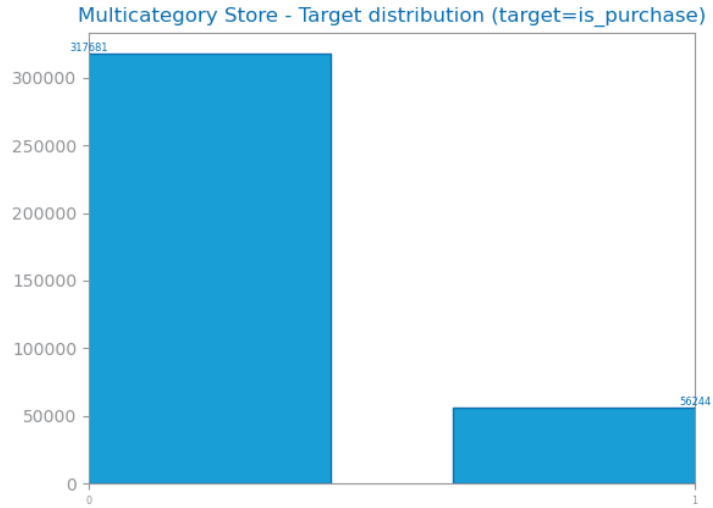


Figure 14 Class distribution for dataset 1

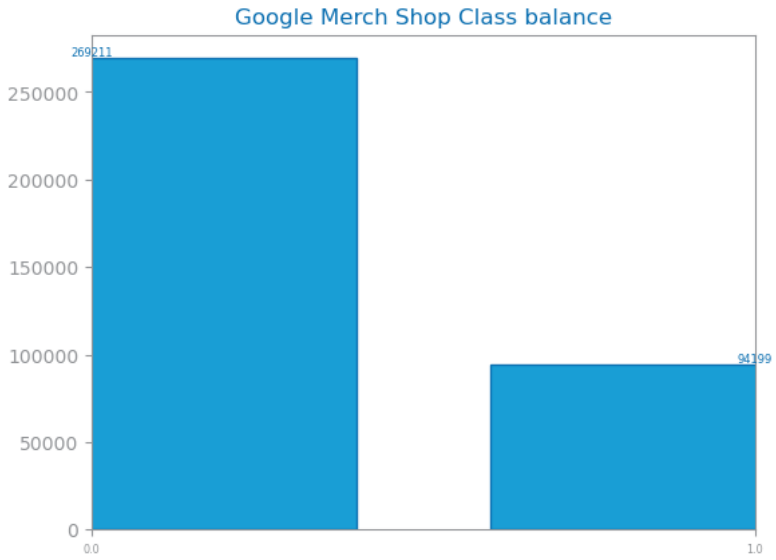


Figure 15 Class distribution for dataset 2

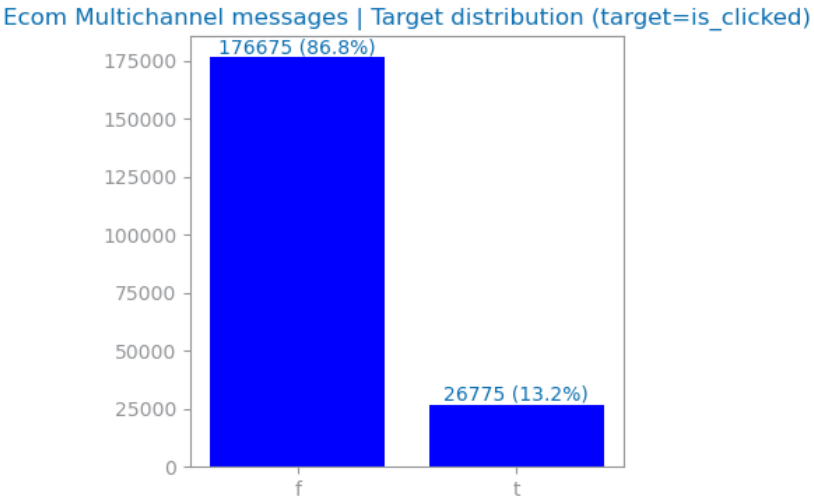


Figure 16 Class distribution for dataset 3

Data Granularity

All datasets have high datetime granularity ranging from year to minute detail which allows for time-specific trend analysis (eg. “Do users buy more frequently at certain times of day?”). Dataset 1 has hierarchical category code lvl X granularity that provides more detailed product buying trends. Dataset 2 has hierarchical granularity on page path level X which allows detailed user journey analysis and spatial data on user’s location (from country to city).

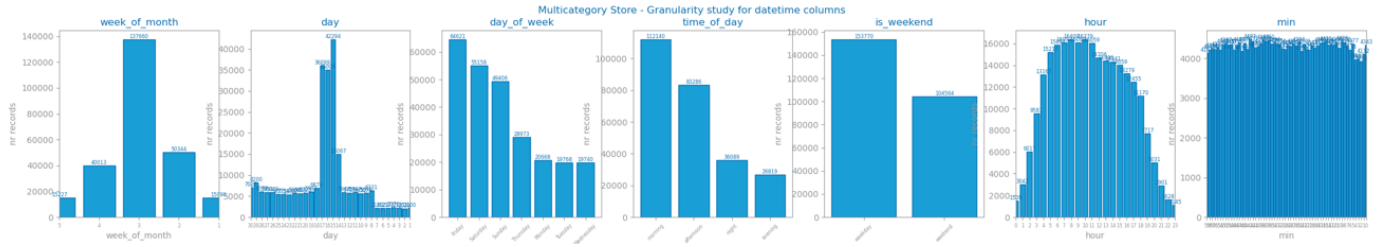
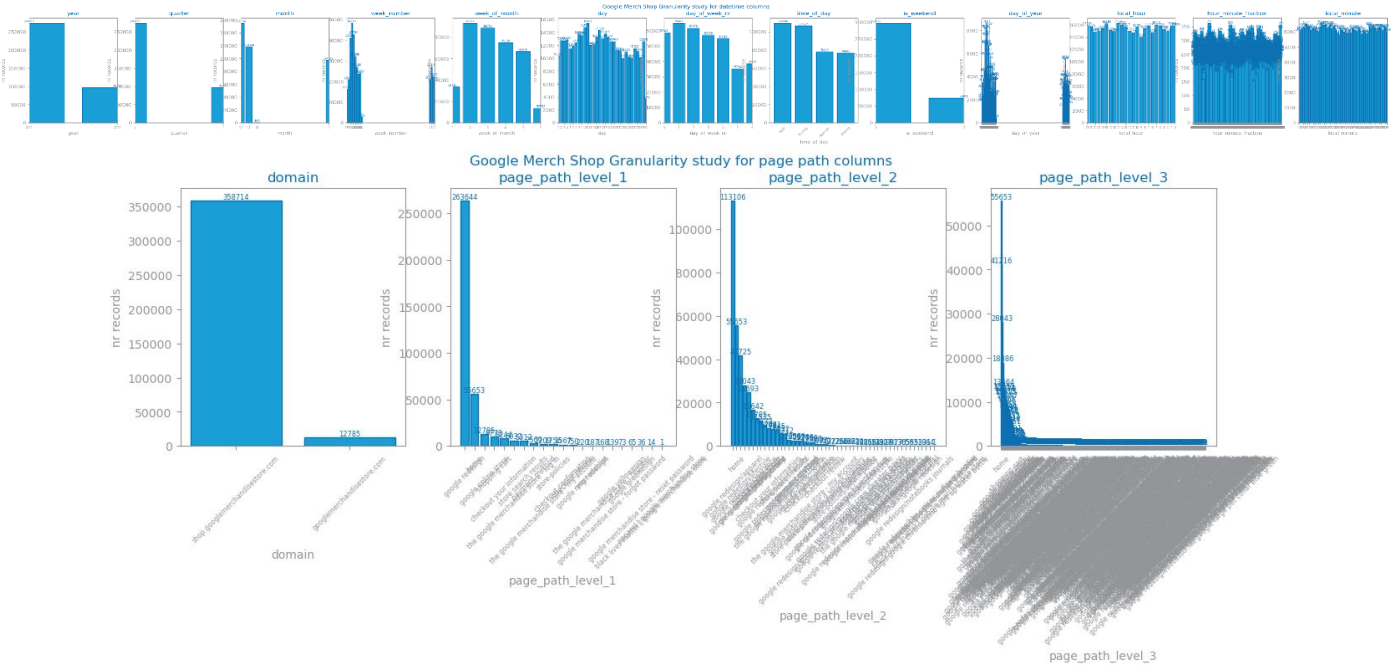


Figure 17 Granularity analysis for dataset 1



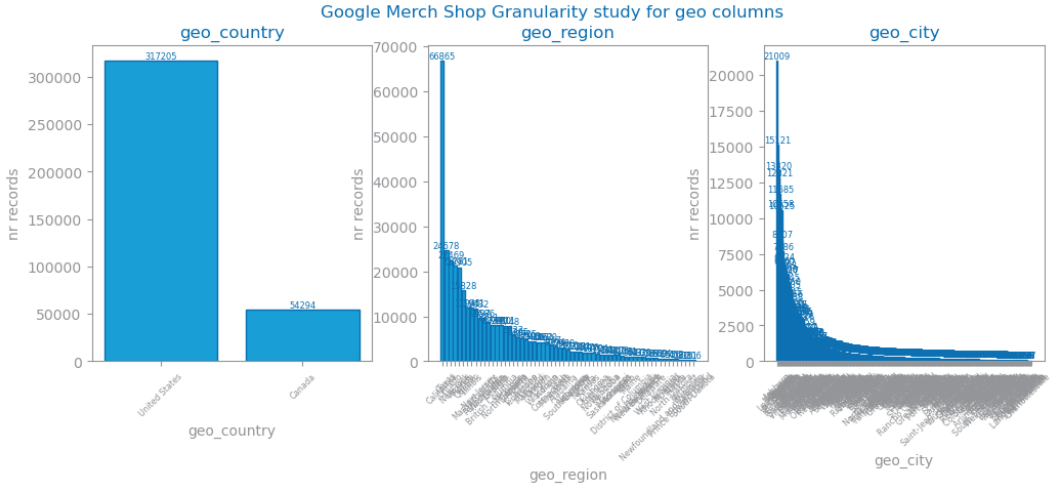


Figure 18 Granularity analysis for dataset 2

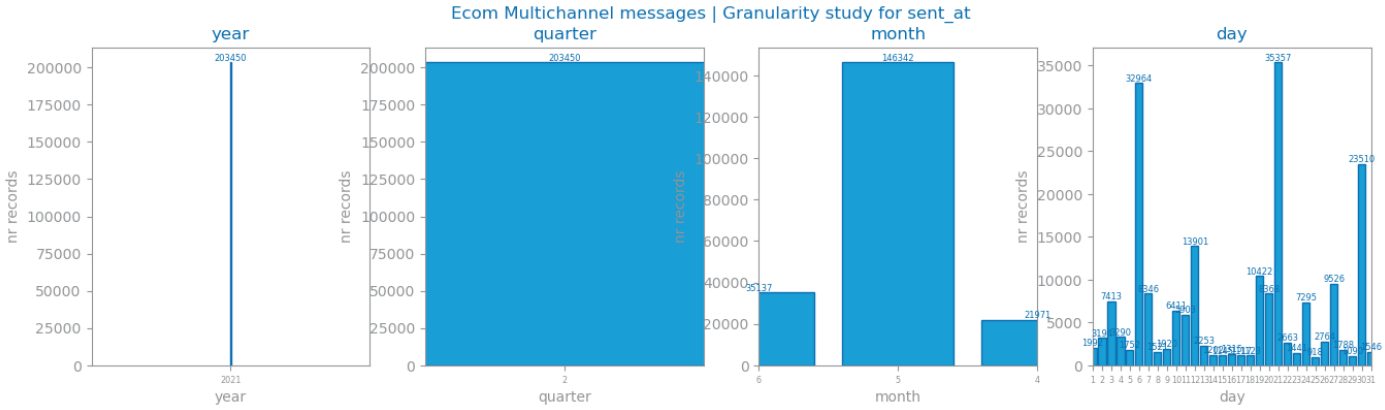


Figure 19 Granularity analysis for dataset 3

Data Sparsity

High correlation between variable groups (eg. date variables), show less data sparsity and therefore greater domain coverage. In dataset 1, price, activity in session, brand value, when combined with other variables, cover a large area for low values, revealing its outliers (activity in session x day). In dataset 2, sometimes geo region and geo city show the same value and different ranges (vs engagement time msec, datetime). In dataset 3, there is a high correlation between type and topic of campaign, and between campaigns with deadlines vs with discounts in the subject.

Figure 20 Sparsity analysis for dataset 1

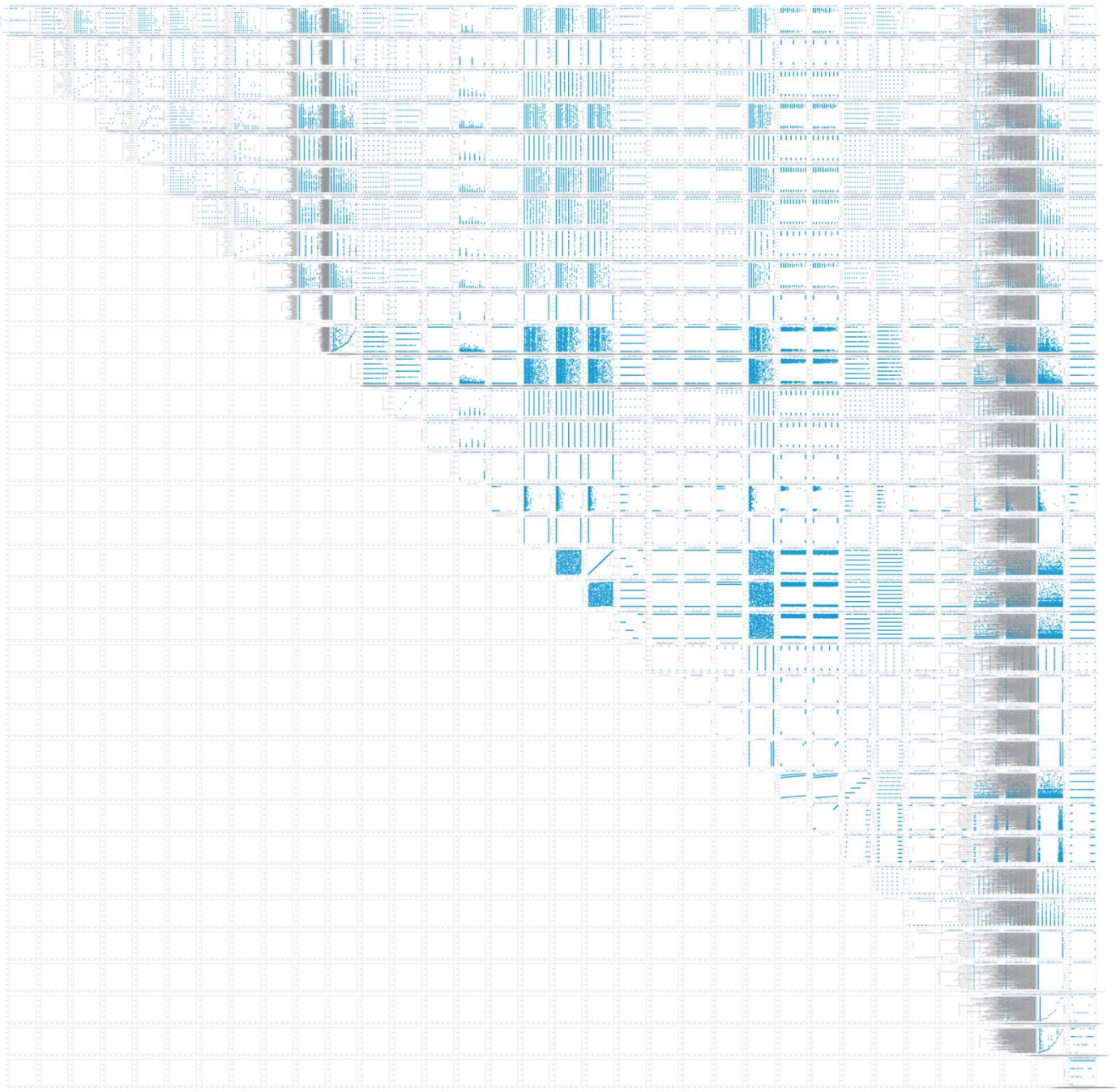


Figure 21 Sparsity analysis for dataset 2

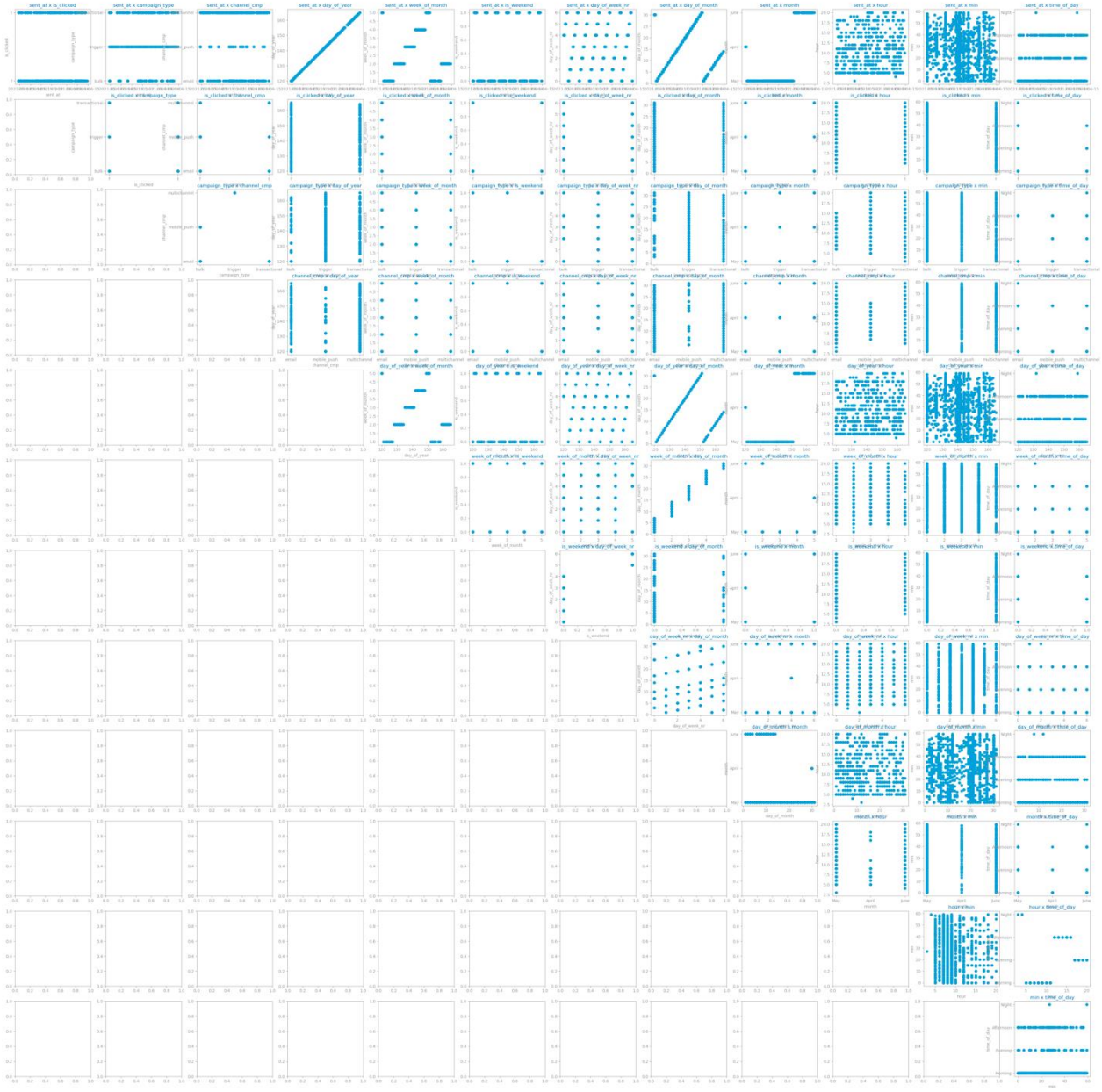


Figure 22 Sparsity analysis for dataset 3

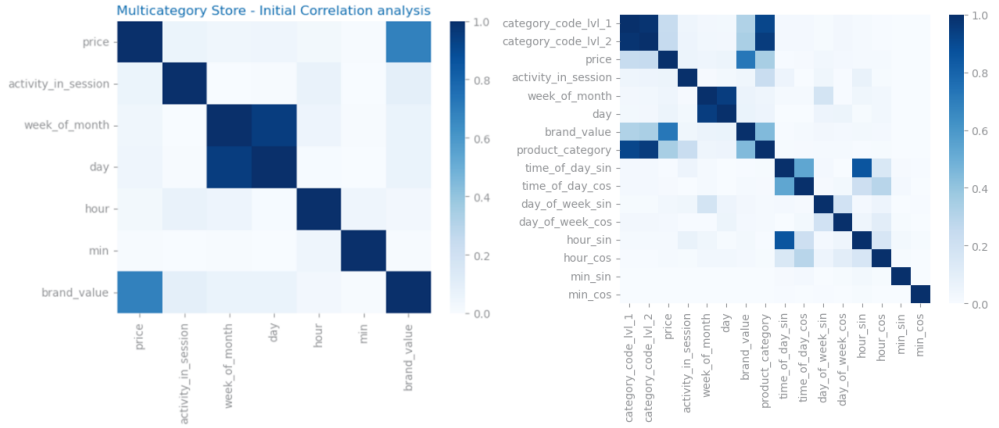


Figure 23 Correlation analysis before and after encoding for dataset 1

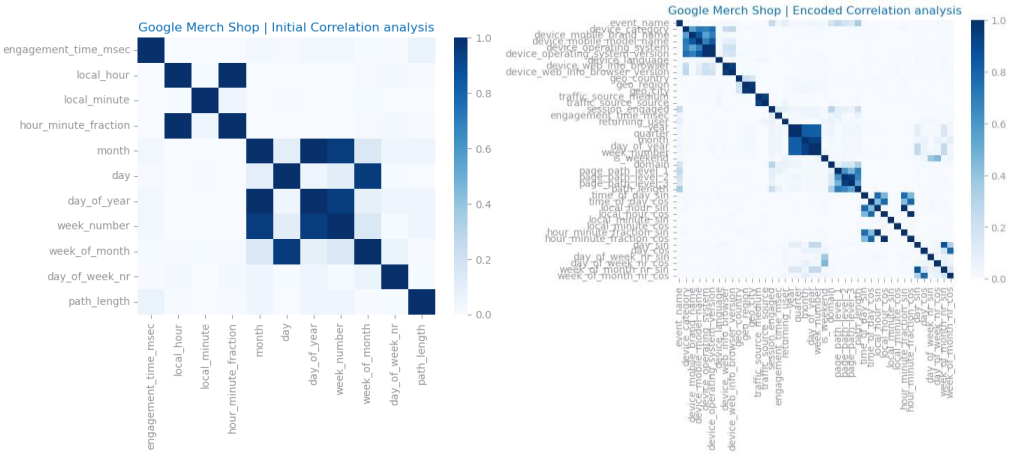


Figure 24 Correlation analysis before and after encoding for dataset 2

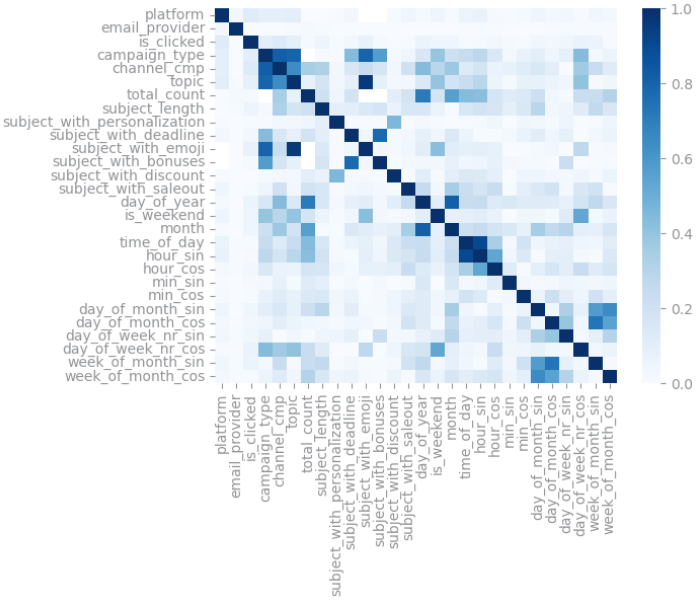


Figure 25 Correlation analysis after encoding for dataset 3

2 DATA PREPARATION

Variables Encoding

Dataset 1:

- **is weekend**: 0: weekday, 1: weekend
- **category code lvl 1**: Order products by similarity in hierarchy (e.g., electronics are closer to computers than to furniture)
- **category code lvl 2**: Order products by similarity at the 2nd category level, following category code lvl 1)
- **product category**: Combine product id with category code lvl 2 and order the output based on category code lvl 2.
- **brand value:** avg price of products added to cart by brand
- **cyclic transformation**: day of week, time of day, hour, min
    - **note**: week of month and day in our case are ordinal variables, with a 1month timespan

Dataset 2:

- **event name**: content -> page view -> ecommerce
- **device Variables**: Desktop vs Smartphone associations (eg.: Apple iOS Safari 14 vs PC Windows 10 Chrome 87). Logic is followed on the different variables. OS/browser versions ordered
- **country**: 1 US 2 Canada
- **region**, **City**: distance to Google HQ. Can change engagement from employees or fans
- **traffic medium**, **traffic source**: google direct grouped
- **page path level** 1,2,3: shop > ecommerce > misc
- **year**, **quarter**, **month**, **day of year**, **week number**: ordinal datetime
- **cyclic transform**: time of day, hour, min, hour min, day, day of week

Dataset 3:

- **campaign type** ranked based on definition and similarities
- **platform** according to size and functionalities
- **email provider** ranked based on popularity and common usage in Russia
- **channel campaigns** ranked based on effectiveness of channels in Russia to send campaigns
- **topic:** grouped into five groups and ranked based on similarities to the business case: order related, abandoned actions, user engagement, or promotional content
- **month** hierarchical
- **hour, min, week of month, day of month, day of week nr** transformed cyclic variables

Missing Value Imputation

Dataset 1 proceeded with Drop Null Strategy. Dataset 2 and 3 proceeded with Most Frequent Strategy. The decision is based on highest f2 (except Dataset 3 due to higher precision gain, low f2 loss). Dataset 3 also received more strategies due to the higher % of MV. Strategies:

Dataset 1:

- Most frequent
- Drop any MV

Dataset 2:

- Most frequent
- Drop columns & MV:
  - 1: drop device language, device operating system version (high % MV)
  - 2: remove any MV remaining

Dataset 3:

- Most frequent
- Drop any columns with MV
- Drop any MV
- Filling with KNN

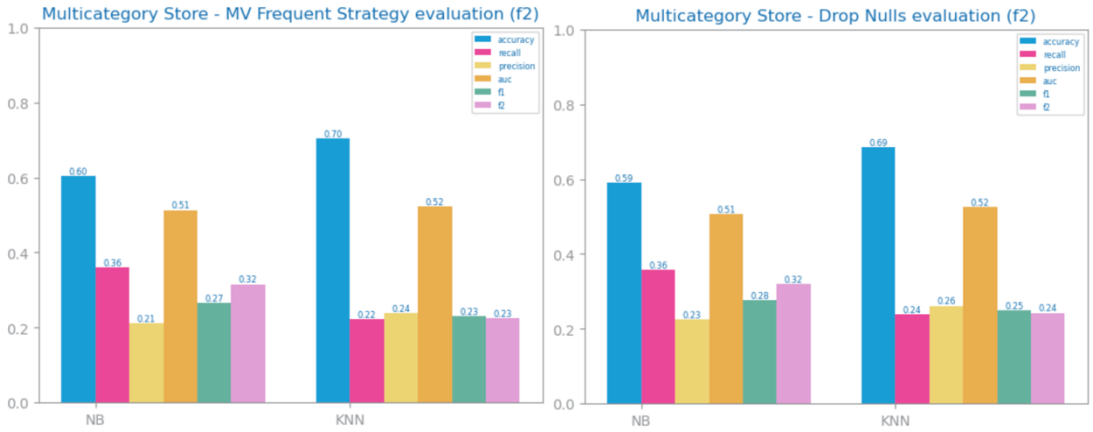


Figure 26 Missing values imputation results with different approaches for dataset 1

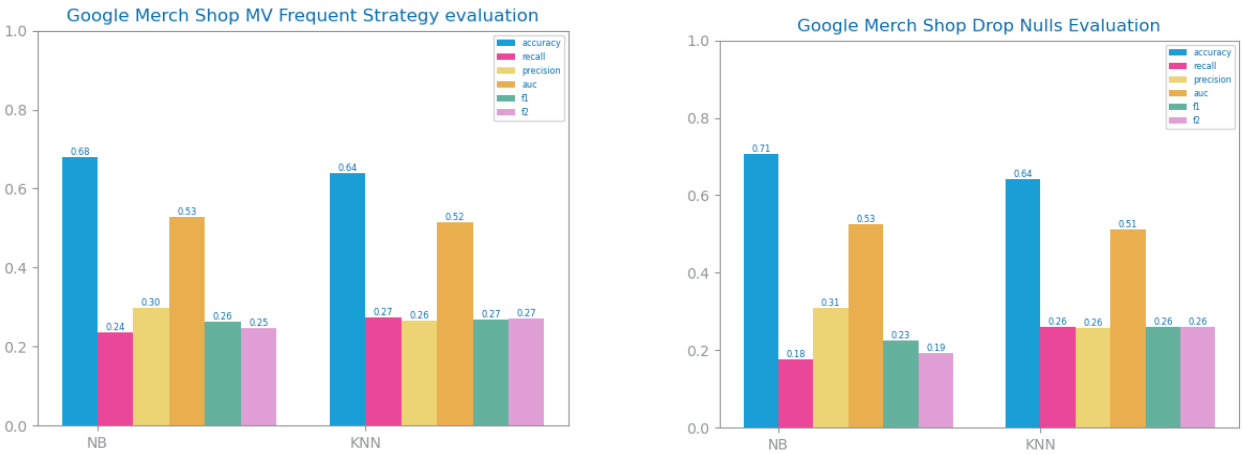


Figure 27 Missing values imputation results with different approaches for dataset 2

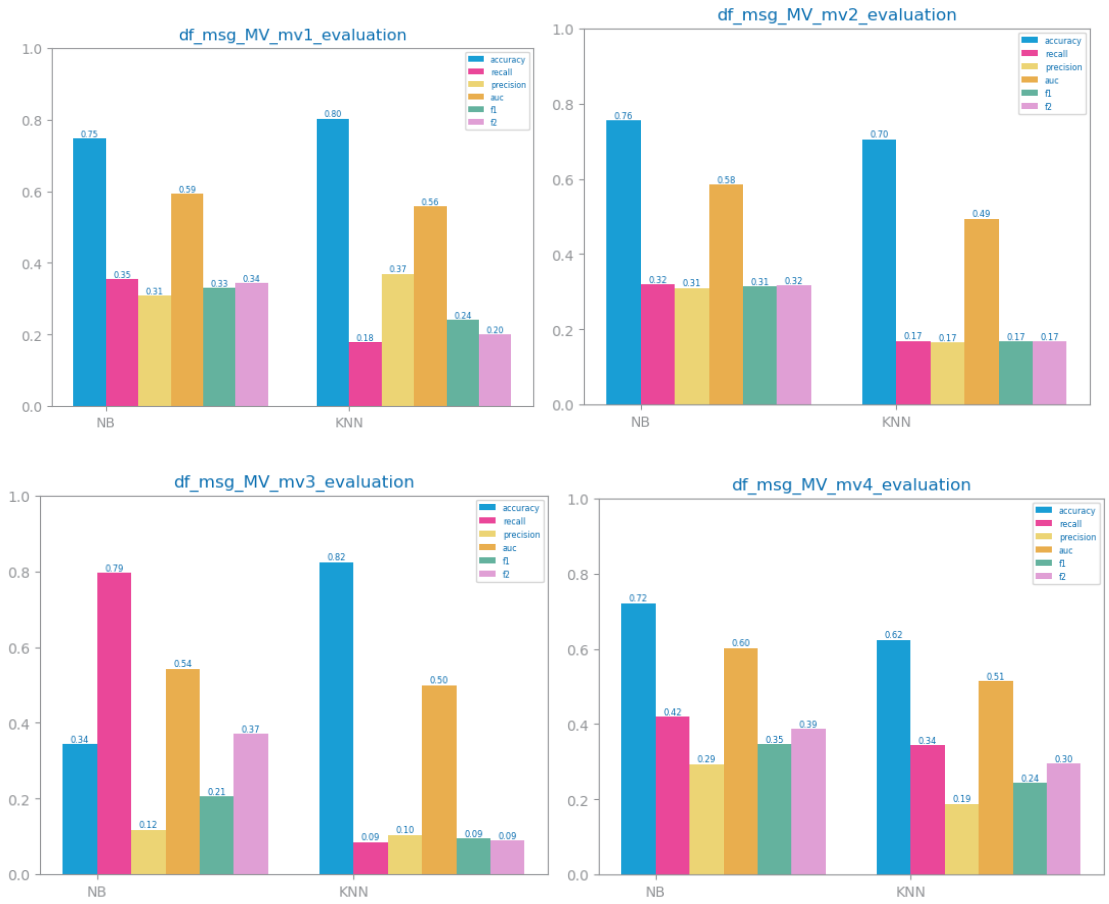


Figure 28 Missing values imputation results with different approaches for dataset 3: 1 – Most Frequent, 2 – Drop MV (rows), 3 – Drop MV (columns), 4 - KNN

Outliers Treatment

Dataset 1 and 2 proceeded with Outlier Drop Strategy. Dataset 3 proceeded with Outlier Truncating. Chosen strategies were based on higher f2. Strategies:

1. Outlier Truncation
2. Outlier Removal

Dataset 1: Outlier Removal had higher precision in both NB and KNN, with similar recall.

Dataset 2: Engagement time ms is the only eligible outlier treatment variable. Outlier Removal increased precision and recall.

Dataset 3: Truncating Outliers' strategy was chosen, as it presented better f2 results overall and more specifically in the KNN method.

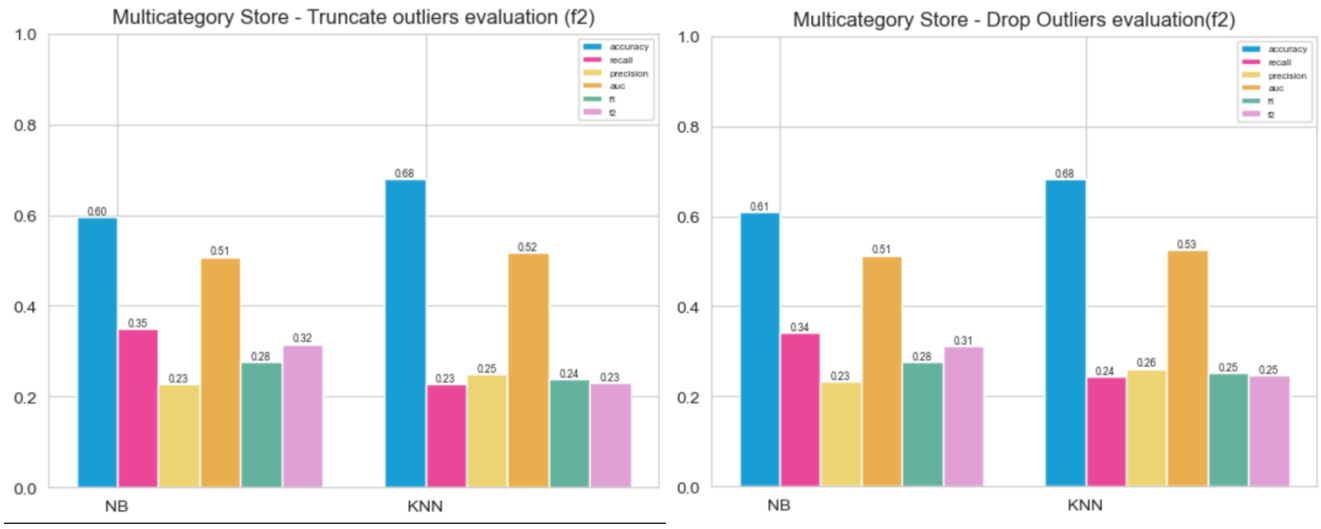


Figure 29 Outliers imputation results with different approaches for dataset 1

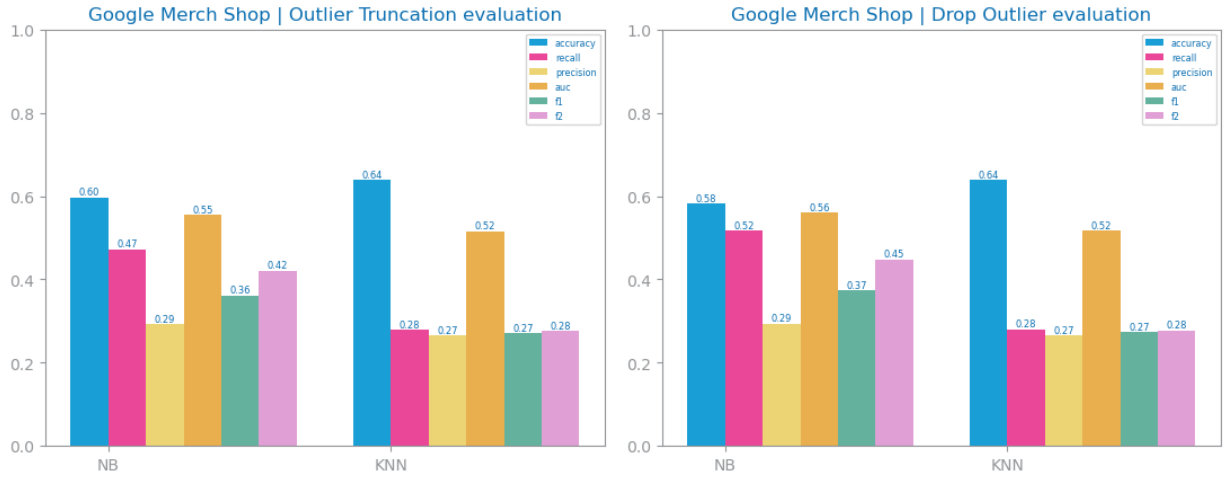


Figure 30 Outliers imputation results with different approaches for dataset 2

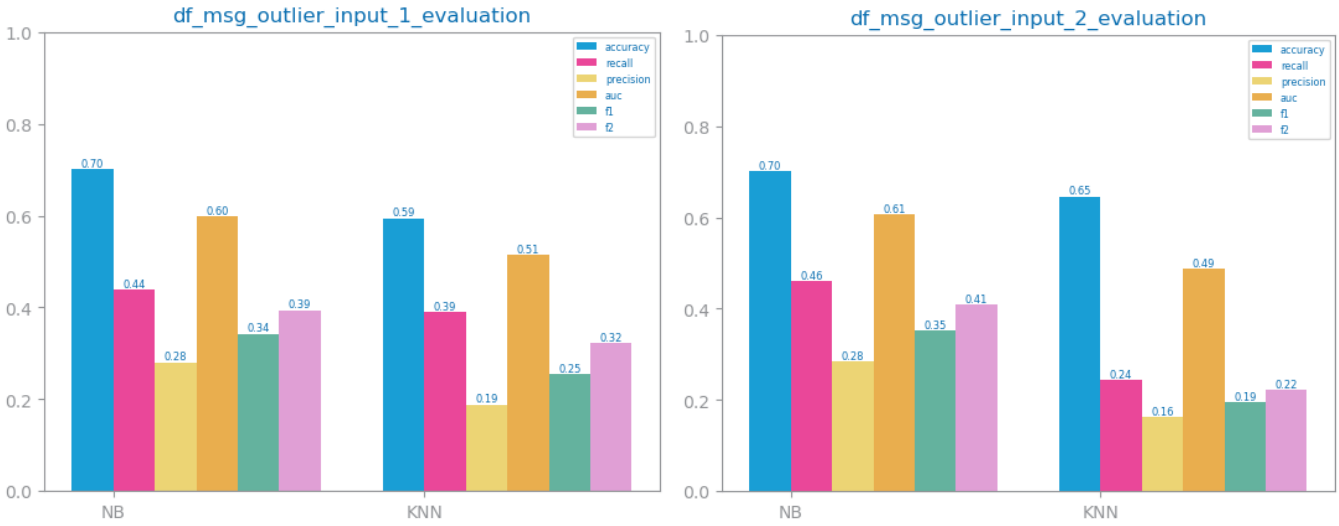


Figure 31 Outliers imputation results with different approaches for dataset 3: 1 – Outlier truncate, 2 – Outlier dropping

Scaling

Dataset 1 and 3 maintained Outlier dataset. Dataset 2 proceeded with Min-Max Scaler based on higher f2.

Dataset 1: KNN had better F2/precision in outliers' dataset.

Dataset 3: KNN had less recall and thus f2 VS outliers' dataset.

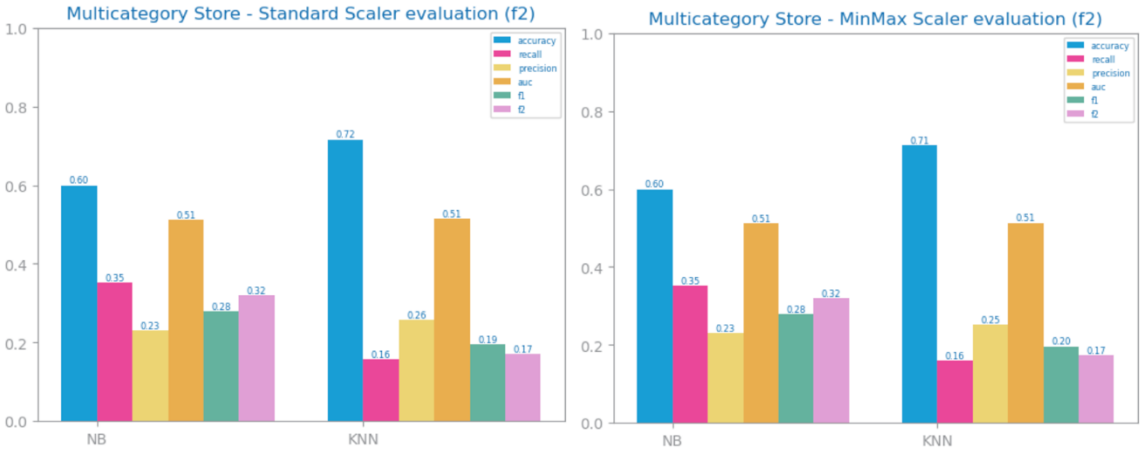


Figure 32 Scaling results with different approaches for dataset 1

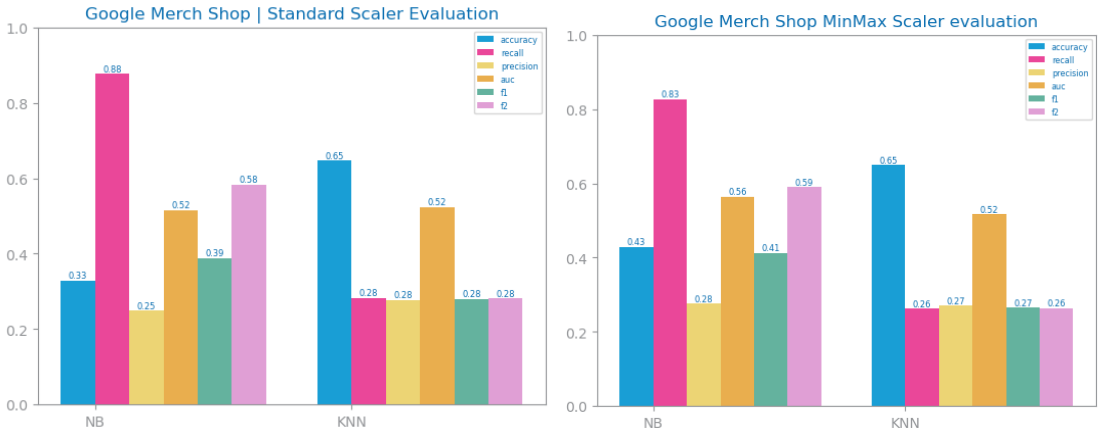


Figure 33 Scaling results with different approaches for dataset 2

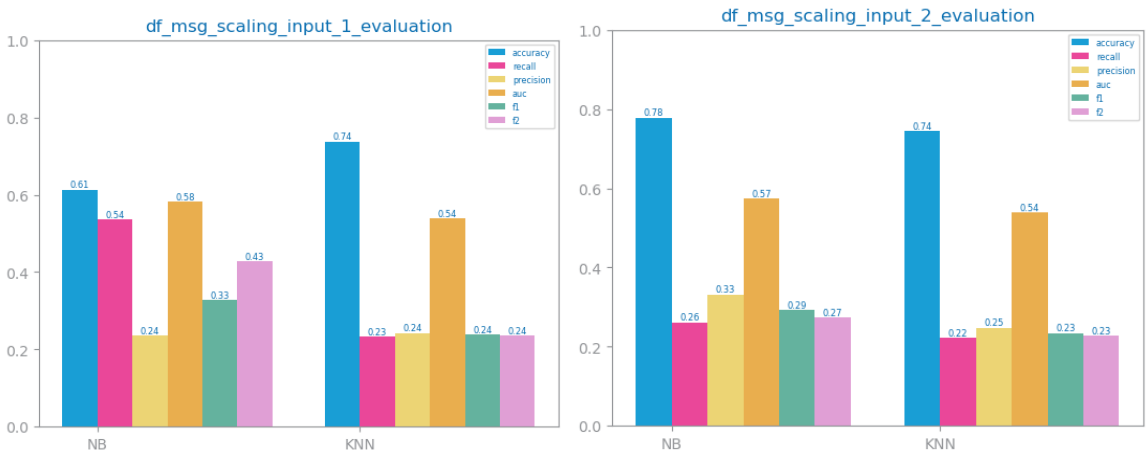


Figure 34 Scaling results with different approaches for dataset 3

Feature Selection

Dataset 1 chose irrelevant variables removal, and datasets 2 and 3 chose removing redundant variables. Chosen strategies were based on higher f2 except for dataset 2 and 3, where we chose based on higher precision. Although we aim to better f2, we tested that recall was near 100% after balancing the dataset without a precision optimized featured selection strategy. Therefore, we moved on with the strategy that offered higher precision. In all datasets some relevant variables for DT were maintained to improve diversity (Figure 61).

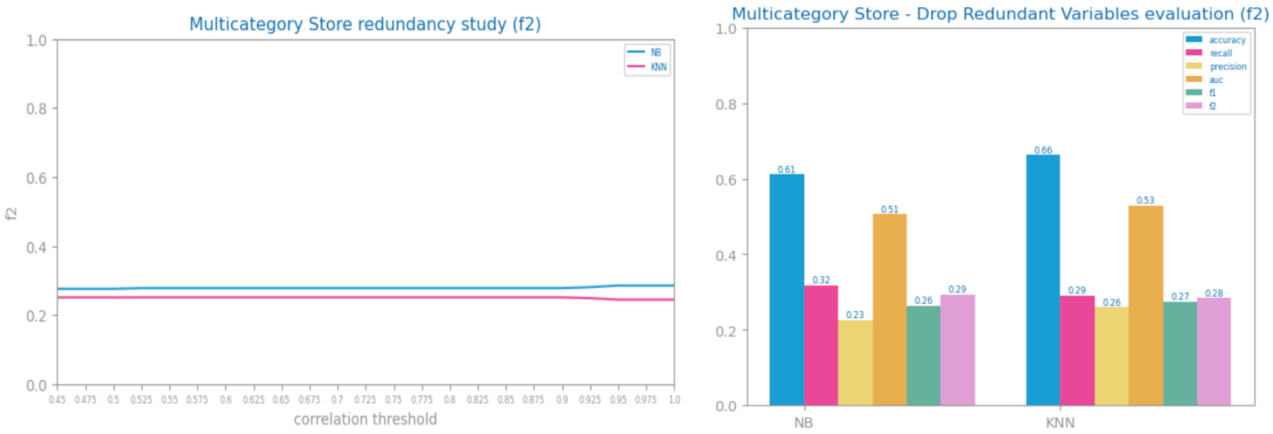


Figure 35 Feature selection of redundant variables results with different parameters for dataset 1

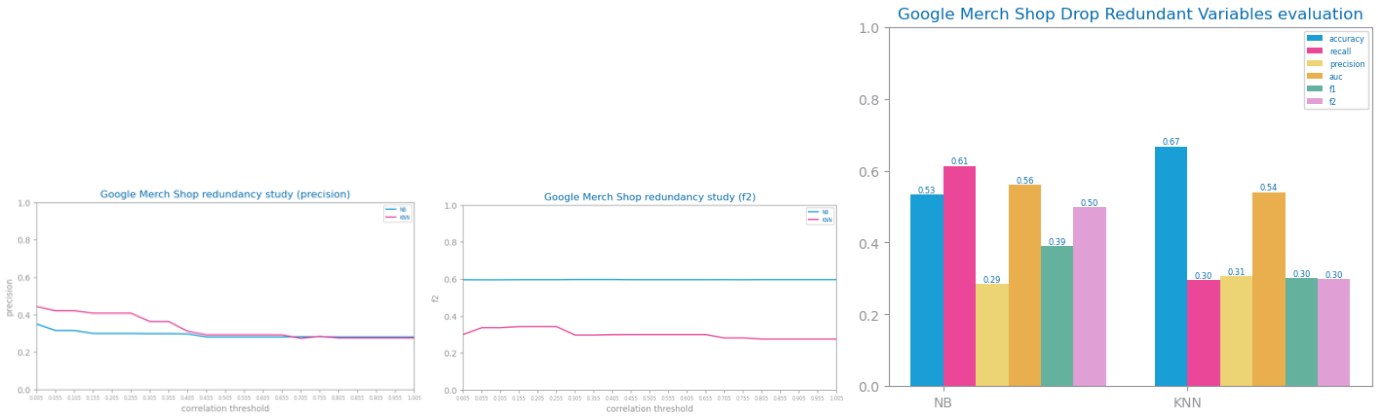


Figure 36 Feature selection of redundant variables results with different parameters for dataset 2

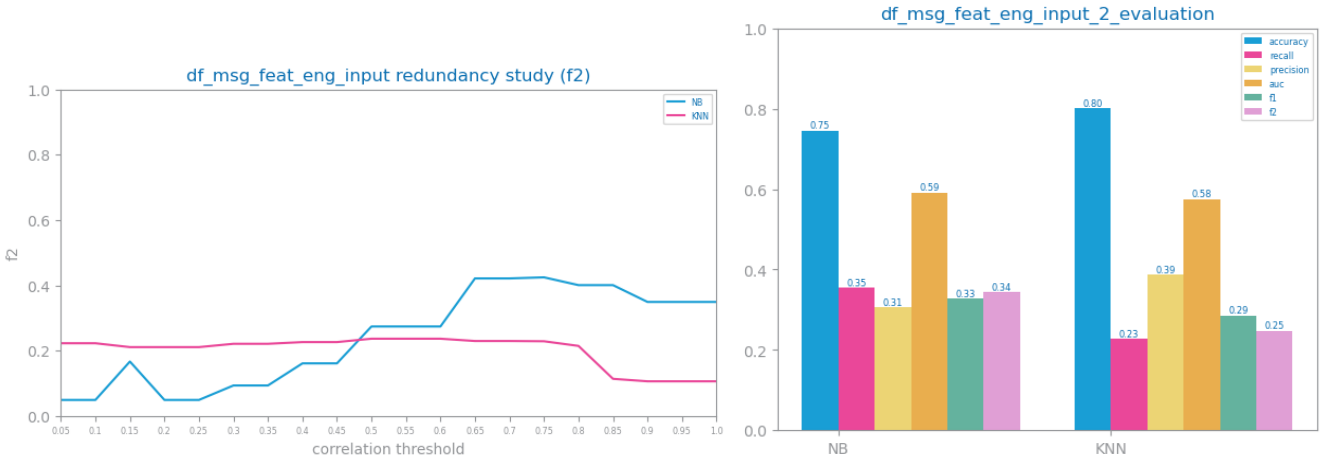
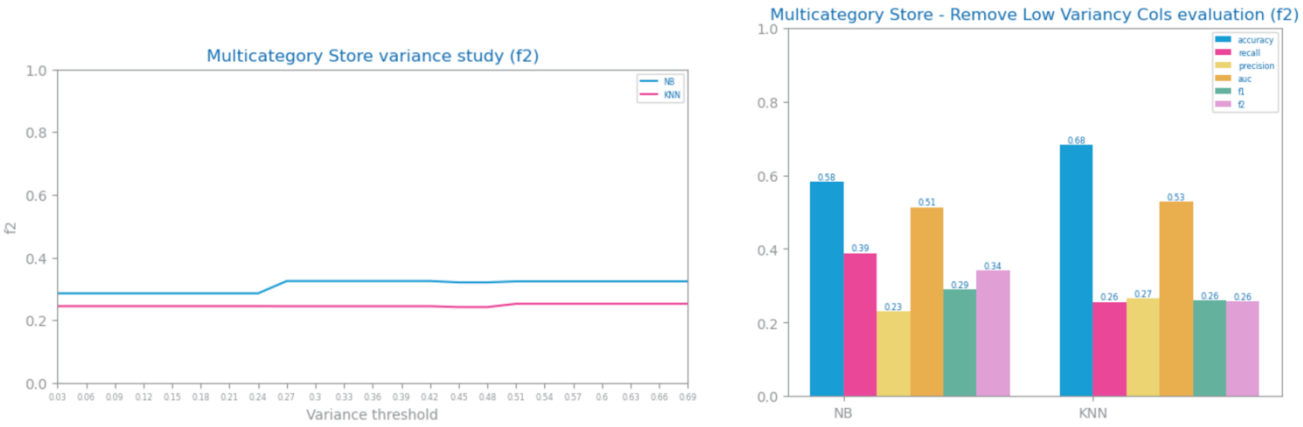


Figure 37 Feature selection of redundant variables results with different parameters for dataset 3



.

Figure 38 Feature selection of relevant variables results with different parameters for dataset 1 (variance study)

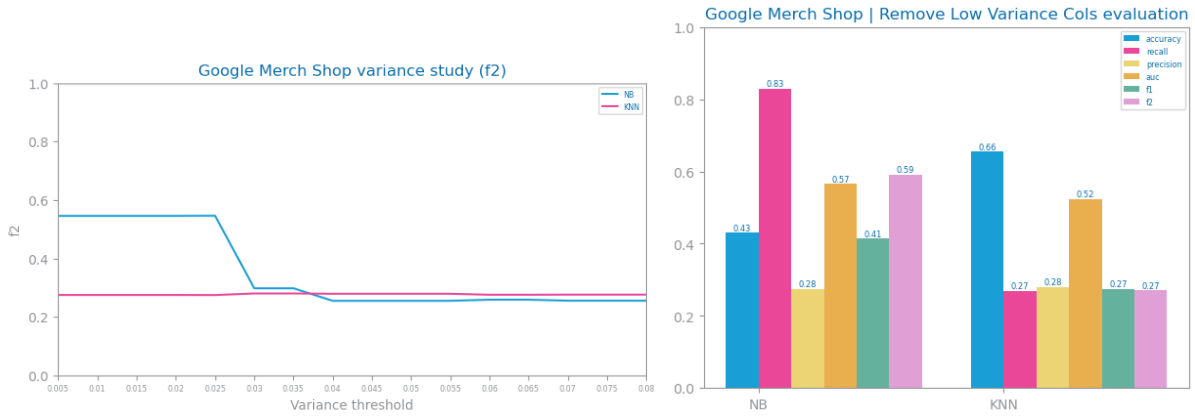


Figure 39 Feature selection of relevant variables results with different parameters for dataset 2 (variance study)

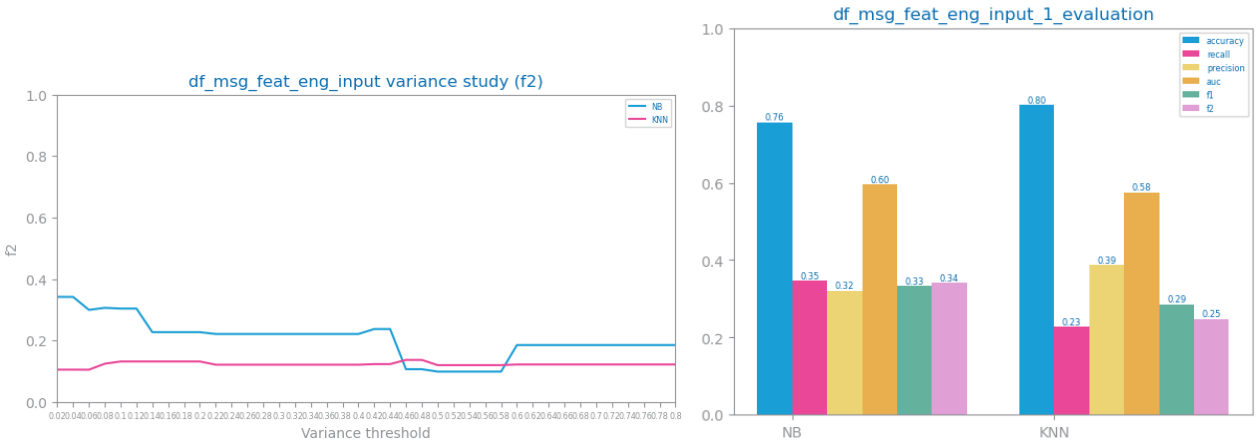


Figure 40 Feature selection of relevant variables results with different parameters for dataset 3 (variance study)

Balancing

Dataset 1 and 2 proceeded with SMOTE strategy. Dataset 3 proceeded with Undersampling. Chosen strategies were based on higher f2 similar to previous steps. However, in all balancing strategies and datasets we saw a much higher recall (~80% frequently), so we ended up choosing strategies that increased accuracy and precision for a more balanced approach, thus reducing too high of FP.

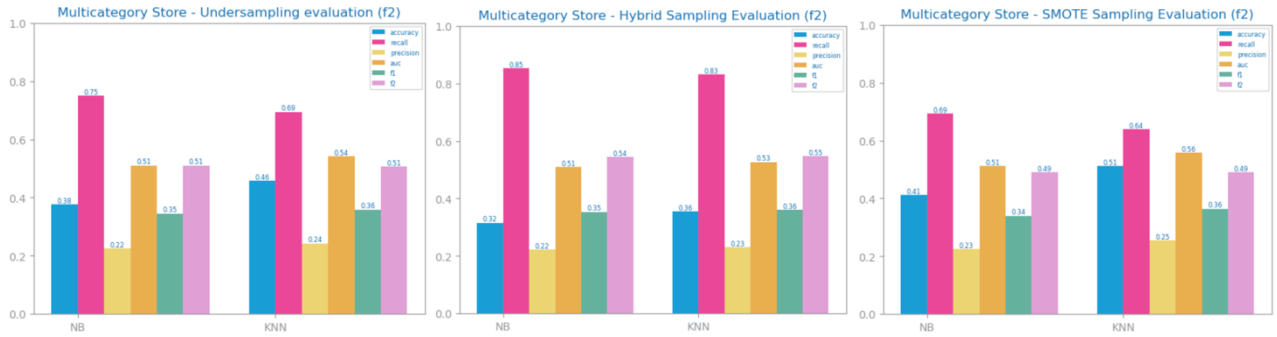


Figure 41 Balancing results with different approaches for dataset 1

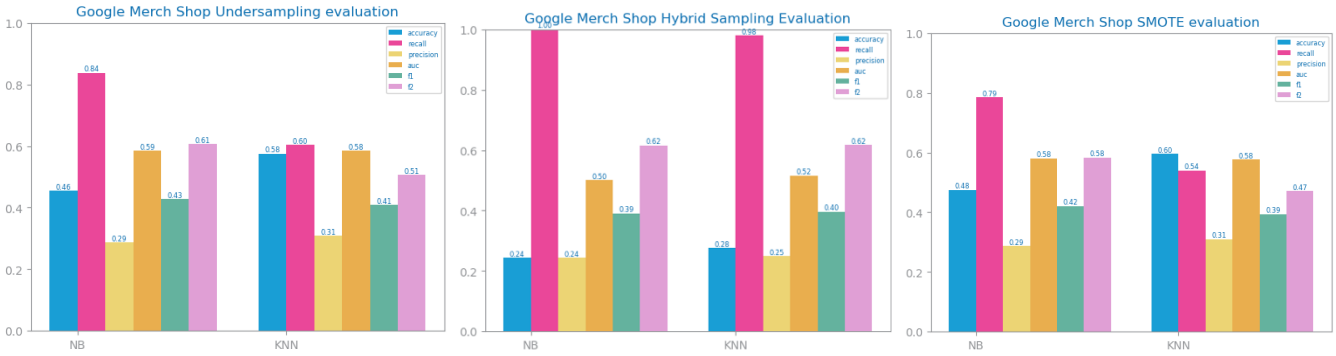


Figure 42 Balancing results with different approaches for dataset 2

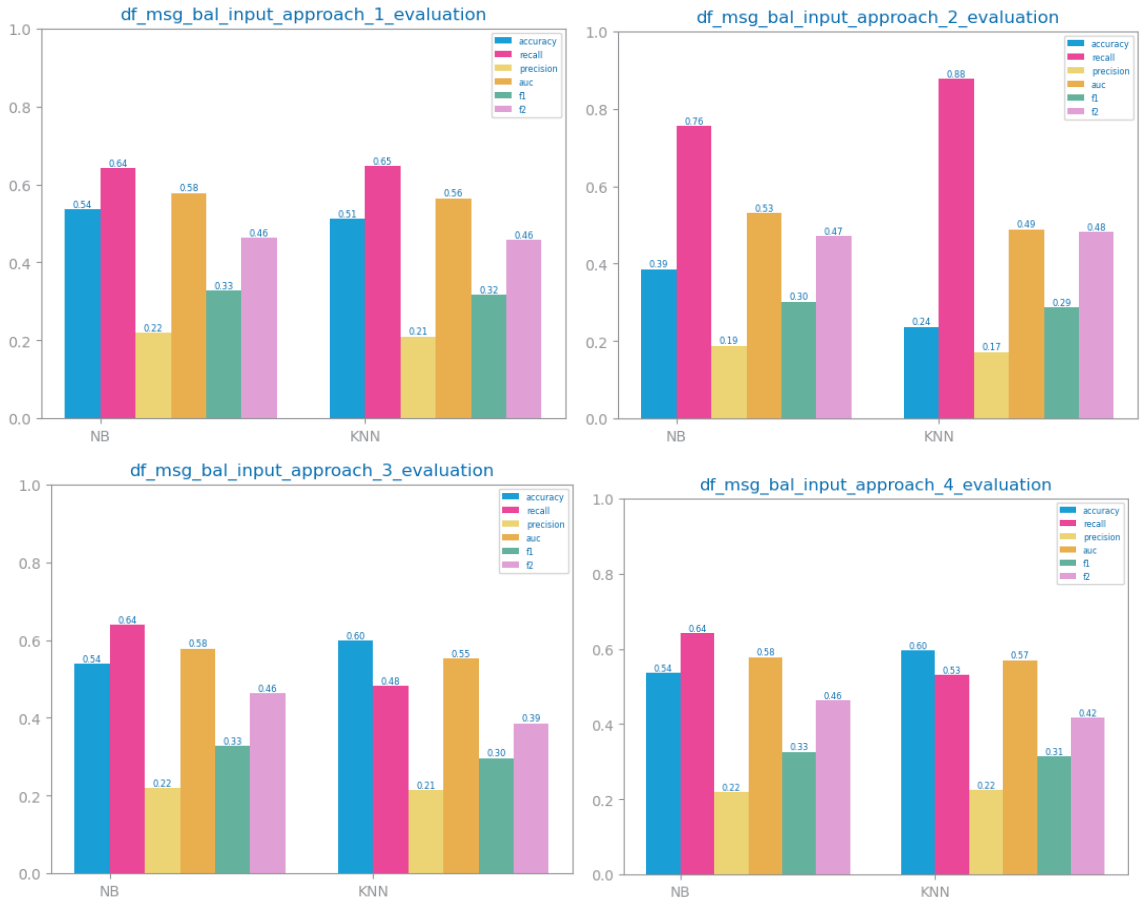


Figure 43 Balancing results with different approaches for dataset 3: 1 – Undersampling, 2 – Hybrid, 3 – SMOTE, 4 - Oversampling

3 MODELS' EVALUATION

Considering computational costs and timing, we used a sampled dataset (~7.5%) in heavier Models (KNN, RF, GB, MLP). Our goal is to maximize TP while still being able to minimize FP as much as possible. The focus on f2 across the three datasets will allow us to maximize marketing activation opportunities while aiming to ensure a not too large number of FP (to avoid unnecessary spending on campaigns targeted at users who are unlikely to buy). This is why we should always look for Precision, Recall and Accuracy when F2 optimized models' performance are not satisfying.

Naïve Bayes (NB)

We didn't use MultinomialNB because it doesn't handle variables with negative values. Dataset 1 had better results with Gaussian while datasets 2 and 3 achieved better f2 performance with Bernoulli. Precision does not change much between implementations for all datasets. We can afford to aim for a better f2 with more TP despite the increase in FP.

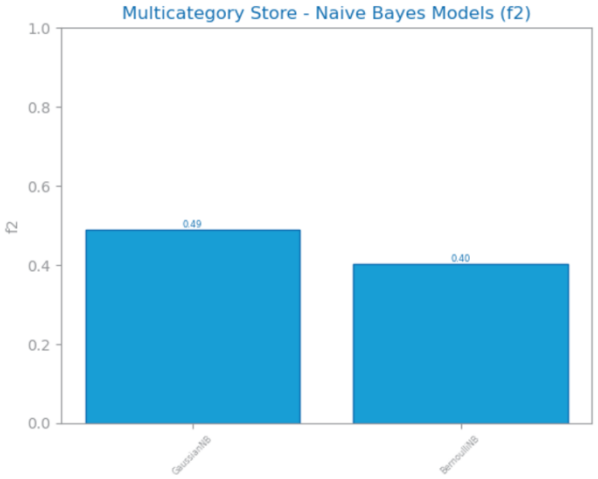


Figure 44 Naïve Bayes alternatives comparison for dataset 1

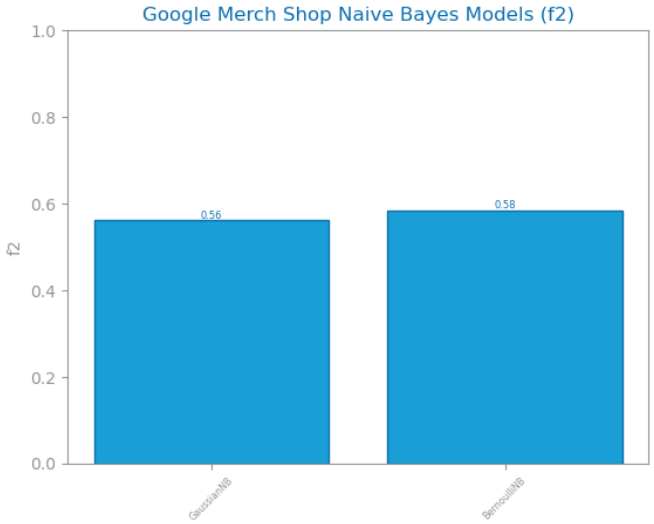


Figure 45 Naïve Bayes alternative comparison for dataset 2

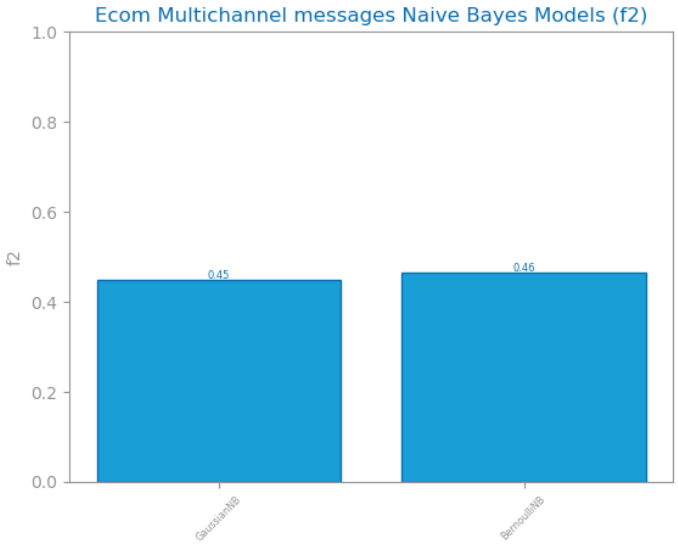


Figure 46 Naïve Bayes alternative comparison for dataset 3

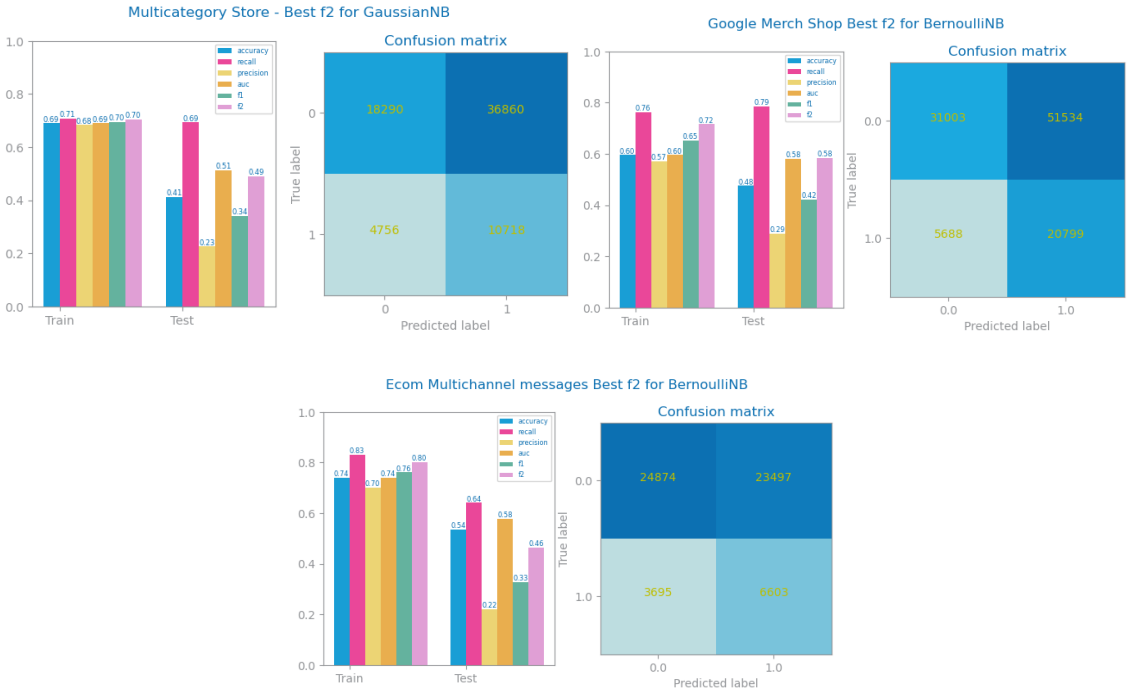


Figure 47 Naïve Bayes best model results for dataset 1 (left), dataset 2 (middle), and dataset 3 (right)

KNN (KNN)

For datasets 1 and 3, increasing the number of k-neighbors did not increase that much f2 score. For dataset 1, the best results were with k=3, while dataset 3 had better results at k=29. For dataset 2, k=91 provided the best f2 results with a higher number of neighbors. As for overfitting, datasets 1 and 2 train and test performance converged and are not under

overfitting, while dataset 3 is facing overfitting at the beginning between k=5 and 11 and eventually by increasing k-neighbors the performance curve converges.

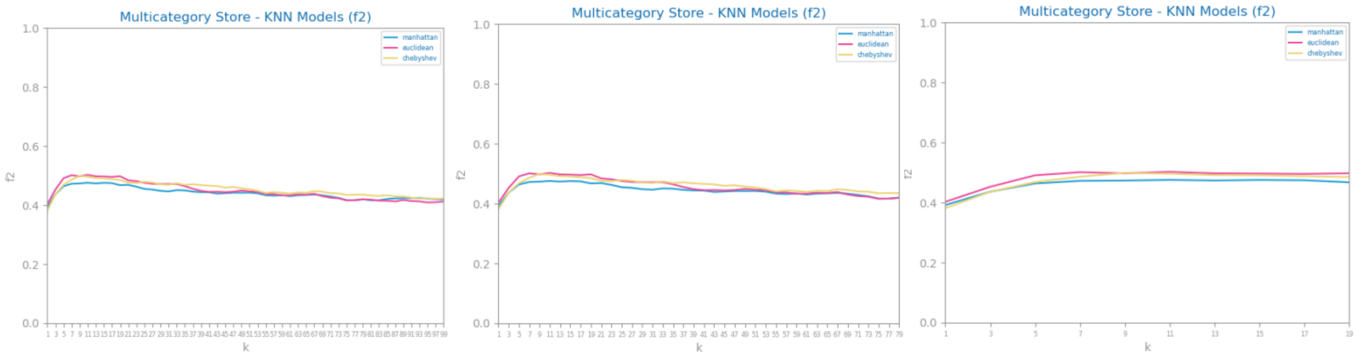


Figure 48 KNN different parameterisations comparison for dataset 1

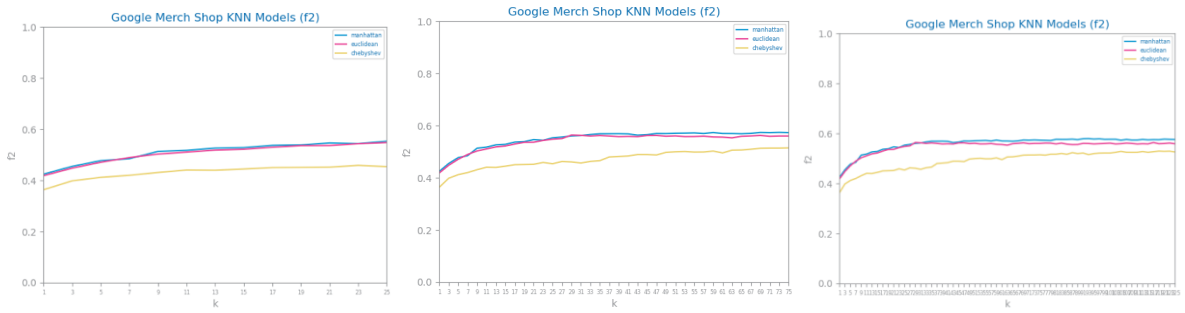


Figure 49 KNN different parameterisations comparison for dataset 2

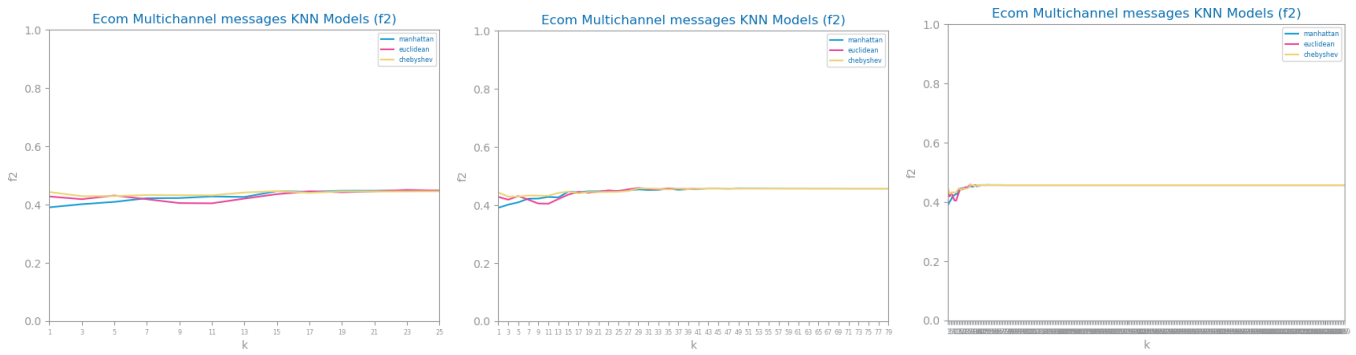


Figure 50 KNN different parameterisations comparison for dataset 3: k=25, k=80 and k=500

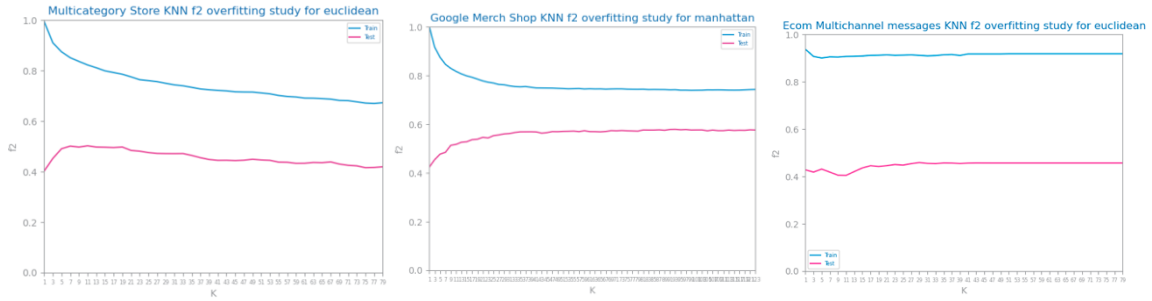


Figure 51 KNN overfitting analysis for dataset 1 (left), dataset 2 (middle), and dataset 3 (right)

Figure 52 KNN best model results for dataset 1 (left), dataset 2 (middle), and dataset 3 (right)

Decision Trees (DT)

Datasets 1 and 2 had better f2 results for d=6, and dataset 3 despite having better results for d=6, we reduced the max depth and got d=2 as the best model, as the loss of f2 performance was only 0.1, as a trade-off to having the tree understandable to the human eye. In all datasets, there are at least 2 variables with very high importance (Figure 61). Datasets 2 and 3 are in overfitting with two different trend lines and depths, as the later one has a steadier decline, while dataset 1's test performance fluctuates and slowly converges to train's performance but having overfitting around d=7.

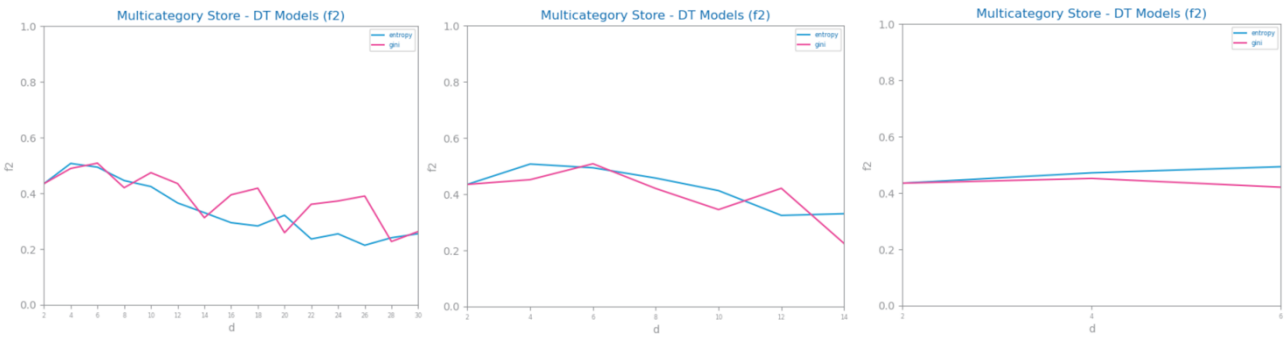


Figure 53 Decision Trees different parameterisations comparison for dataset 1

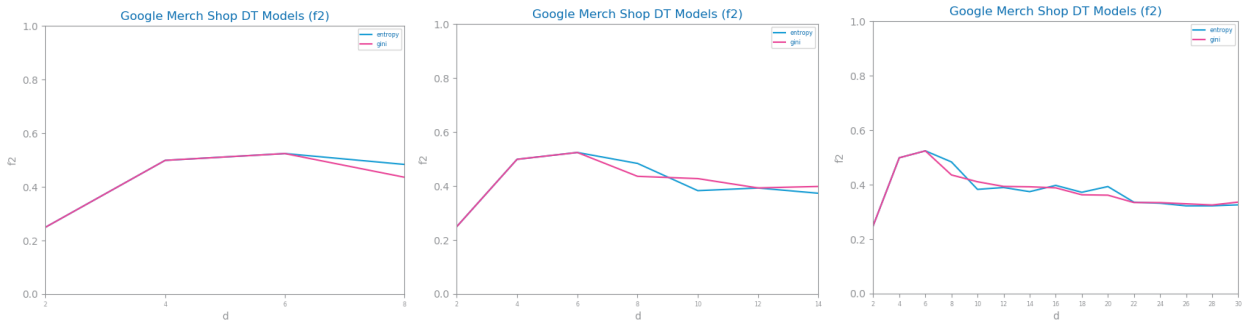


Figure 54 Decision Trees different parameterisations comparison for dataset 2

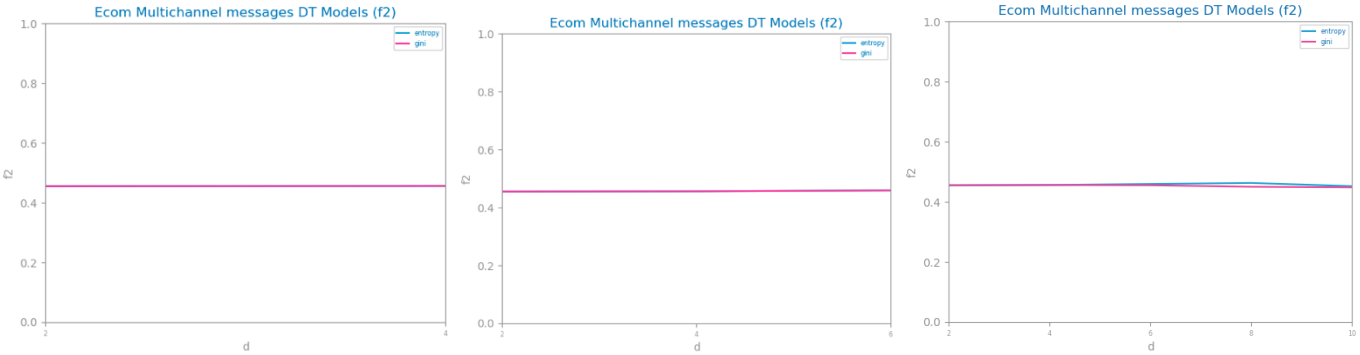


Figure 55 Decision Trees different parameterisations comparison for dataset 3: max depth = 4, 6, and 10

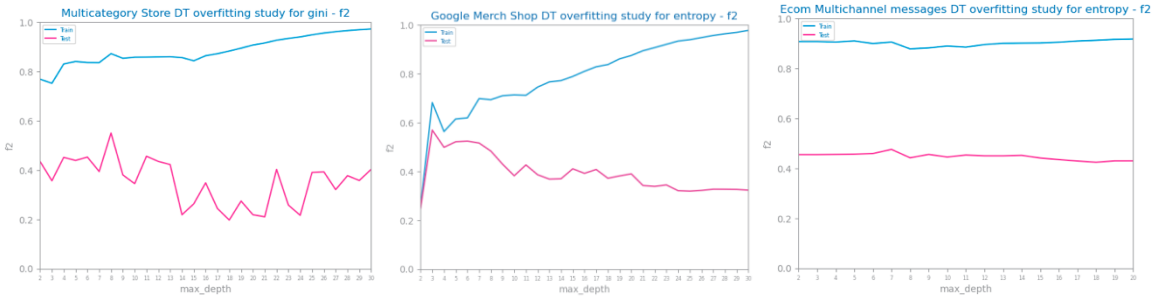


Figure 56 Decision Trees overfitting analysis for dataset 1 (left), dataset 2 (middle), and dataset 3 (right)

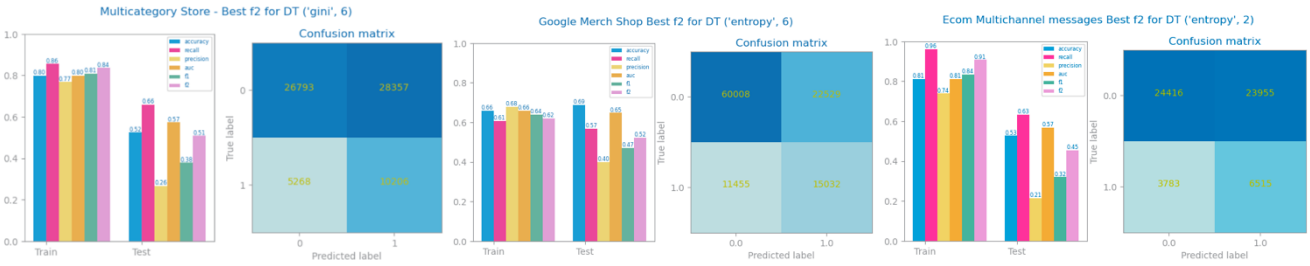


Figure 57 Decision trees best model results for dataset 1 (left), dataset 2 (middle), and dataset 3 (right)

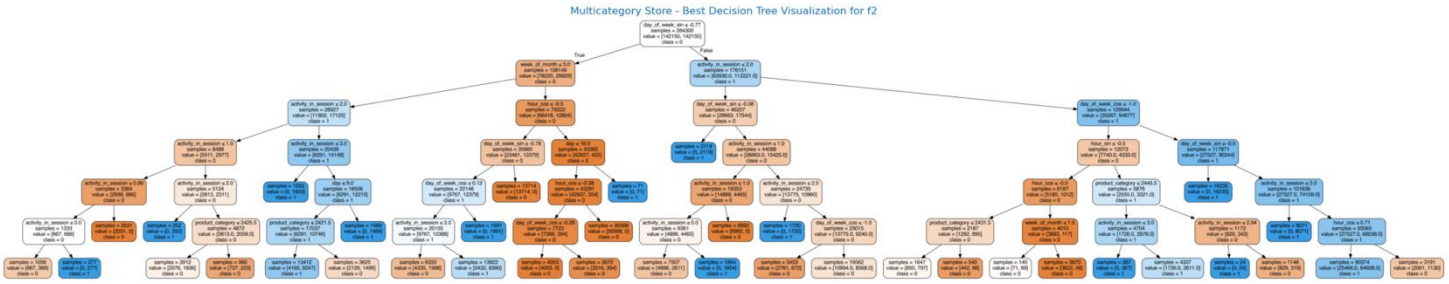


Figure 58 Best tree for dataset 1

Figure 59 Best trees for dataset 2

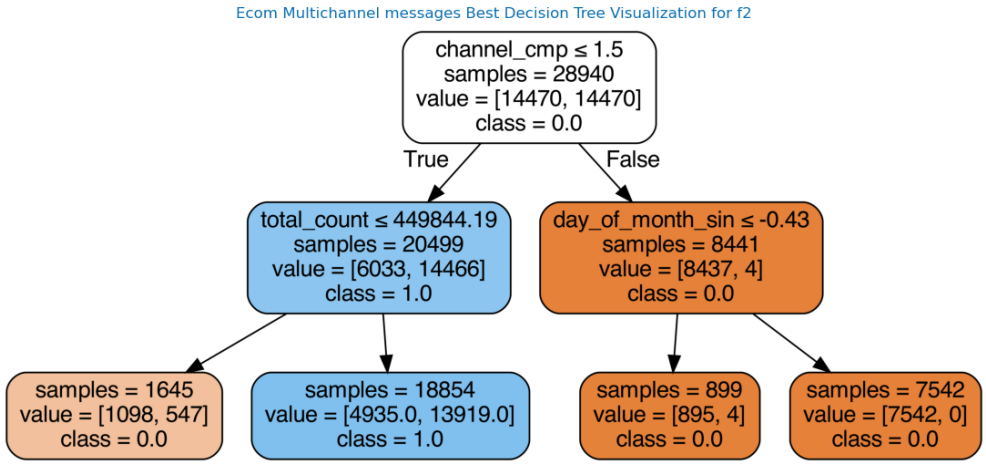


Figure 60 Best trees for dataset 3

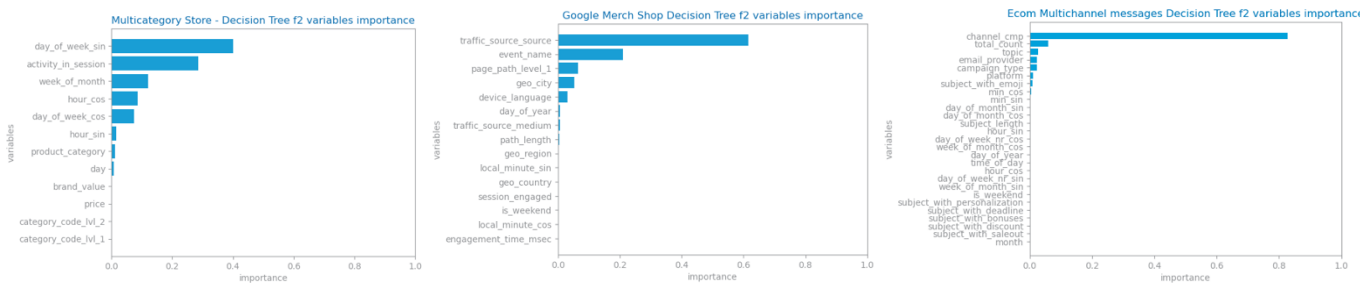


Figure 61 Decision Trees variables importance for dataset 1 (left), dataset 2 (middle), and dataset 3 (right)

Random Forests (RF)

Dataset 1 had the best f2 results (50%) with trees=600; d=2; f=0.1, like dataset 2 which had best results with f=0.1 and 750 trees, with 57% f2. Dataset 3 had best with lower max features, 50% f2 and f=0.001, 500 trees with d=2. Train/test performances fluctuate a lot, but overall keep the same trend, without clear overfitting for all 3 datasets for tested depths.

In dataset 1, time variables are most important, for dataset 2 engagement / traffic variables, and for dataset 3 campaign variables. We have different variables with different importance, but there is not a specific one that stands out.

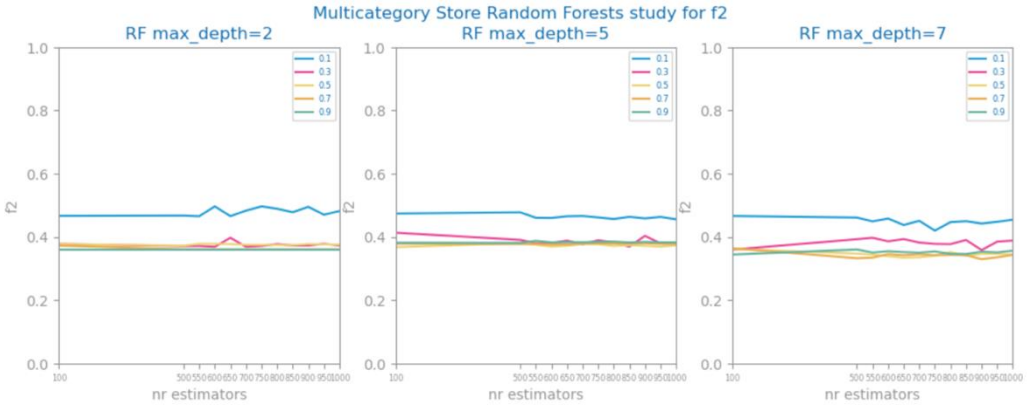


Figure 62 Random Forests different parameterisations comparison for dataset 1

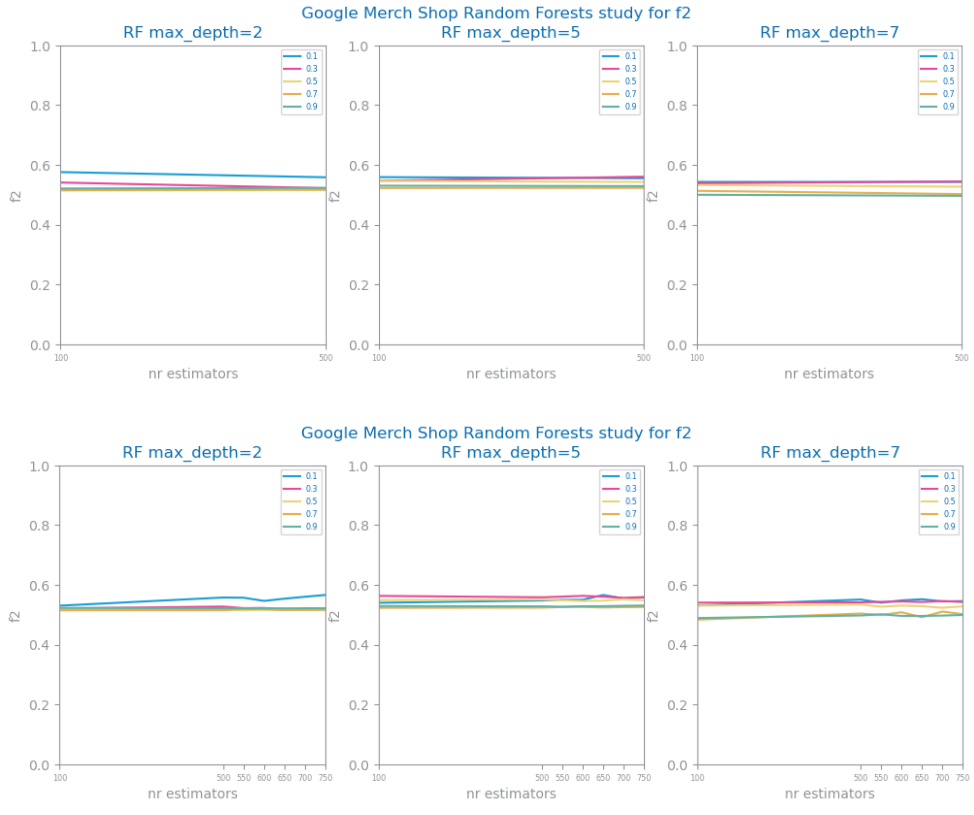
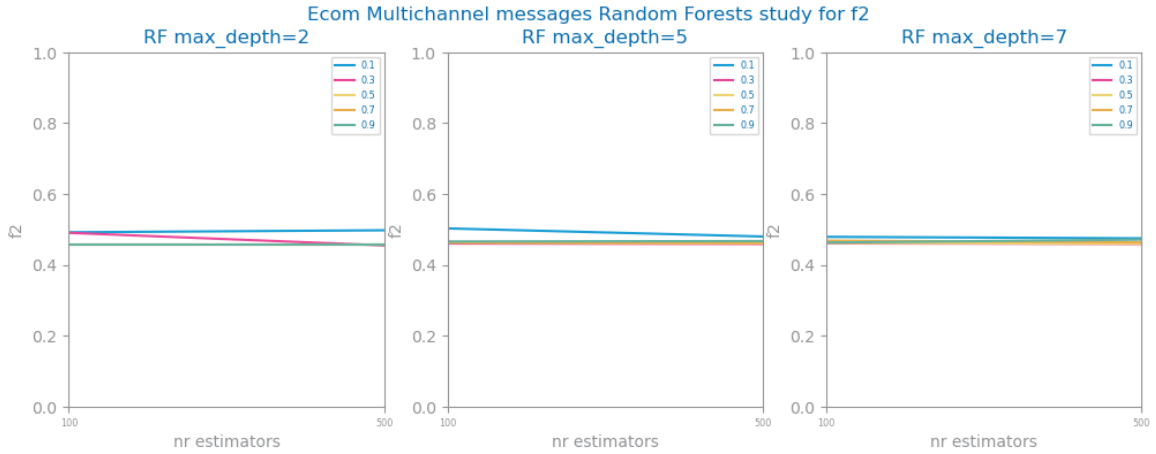
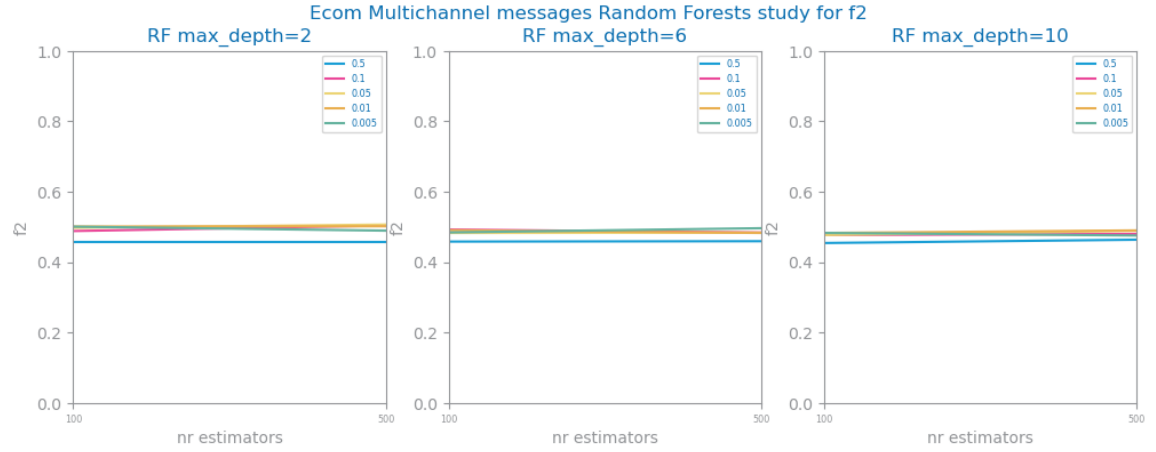
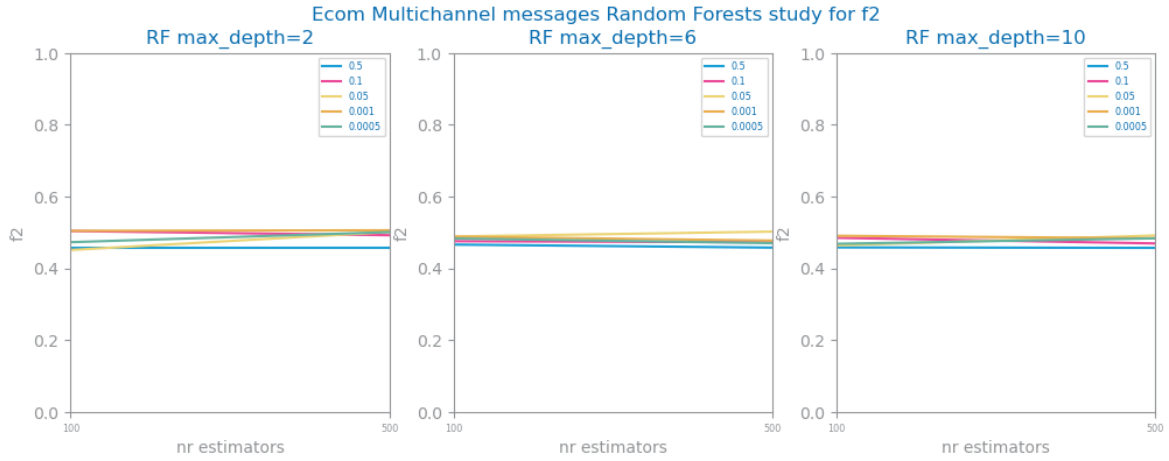


Figure 63 Random Forests different parameterisations comparison for dataset 2


Ecom Multichannel messages Random Forests study for f2
RF max_depth=2
RF max_depth=5
RF max_depth=7
f2
nr estimators
0.1
0.3
0.5
0.7
0.9


Ecom Multichannel messages Random Forests study for f2
RF max_depth=2
RF max_depth=6
RF max_depth=10
f2
nr estimators
0.5
0.1
0.05
0.01
0.005


Ecom Multichannel messages Random Forests study for f2
RF max_depth=2
RF max_depth=6
RF max_depth=10
f2
nr estimators
0.5
0.1
0.05
0.001
0.0005

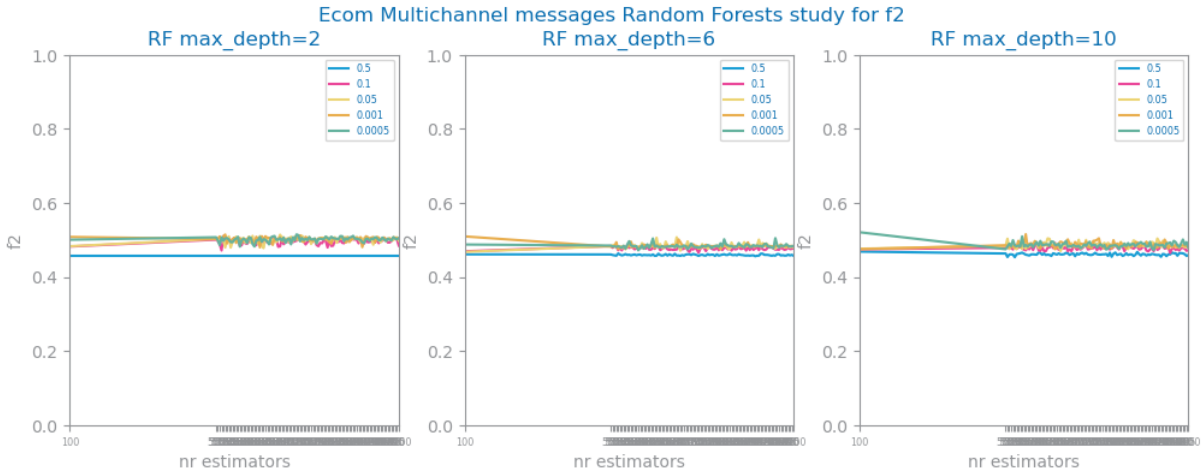


Figure 64 Random Forests different parameterisations comparison for dataset 3

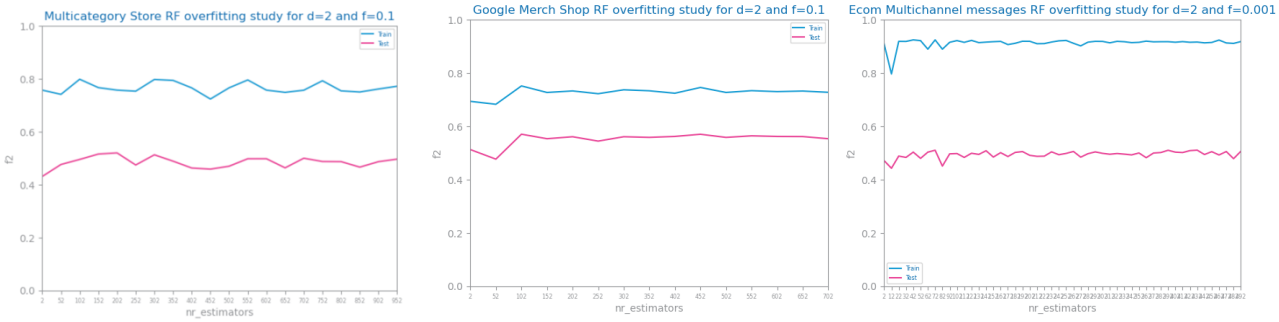


Figure 65 Random Forests overfitting analysis for dataset 1 (left), dataset 2 (middle), and dataset 3 (right)

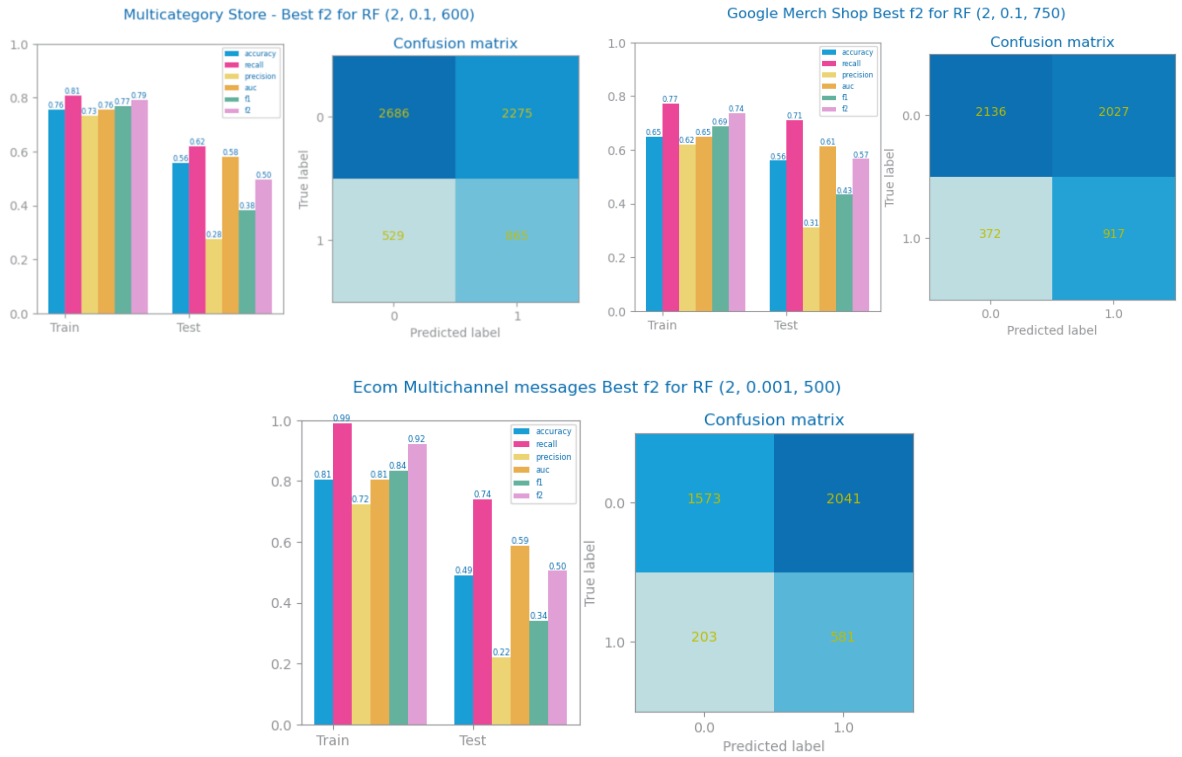


Figure 66 Random Forests best model results for dataset 1 (left), dataset 2 (middle), and dataset 3 (right)

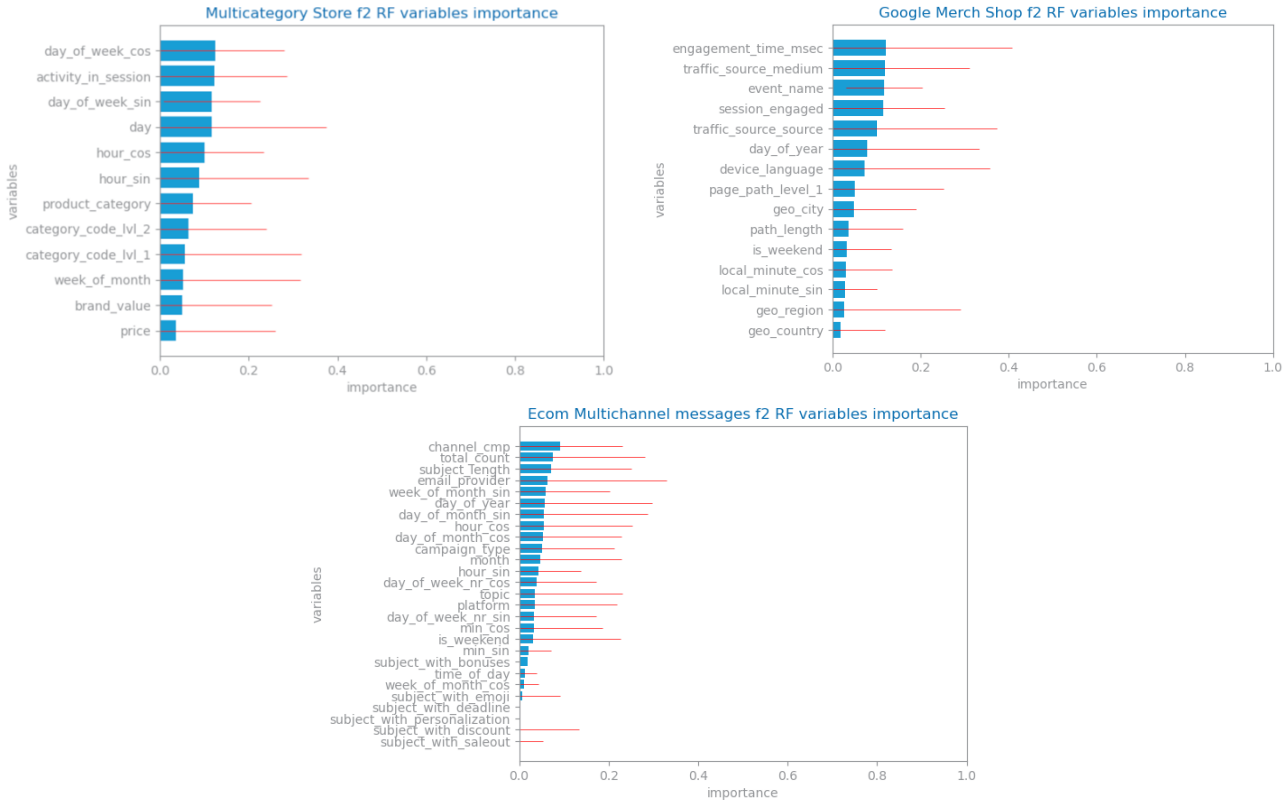


Figure 67 Random Forests variables importance for dataset 1 (left), dataset 2 (middle), and dataset 3 (right)

Gradient Boosting (GB)

For all datasets, GB performed best with a lower number of depths, although the 1st dataset performance was poor, which had best f2 (24%) with 100 trees, d=5, lr=0.9. Dataset 2 performed best (f2 42%) with a lower learning rate 0.1 with 100 trees, d=2, and dataset 3, best f2 (49%) with lr=0.9, with 100 trees and d=2. In dataset 1 and 2 we see overfitting, but the train and test's performance keep overall the same trendline in dataset 3, beginning to overfit if we increase model complexity. In dataset 3, the channel cmp stands out as the most important.

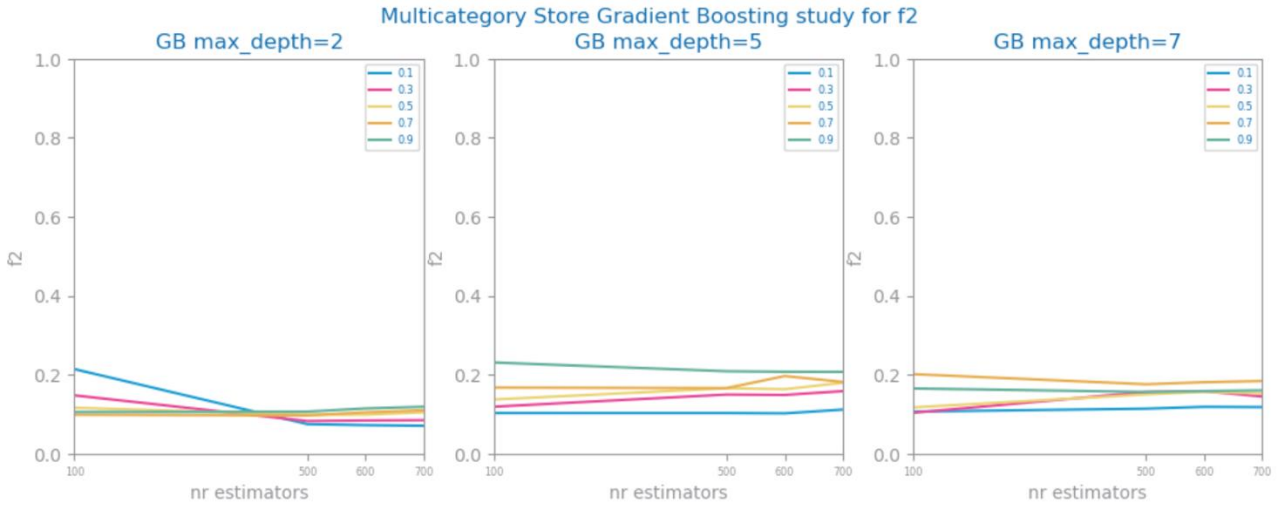


Figure 68 Gradient boosting different parameterisations comparison for dataset 1

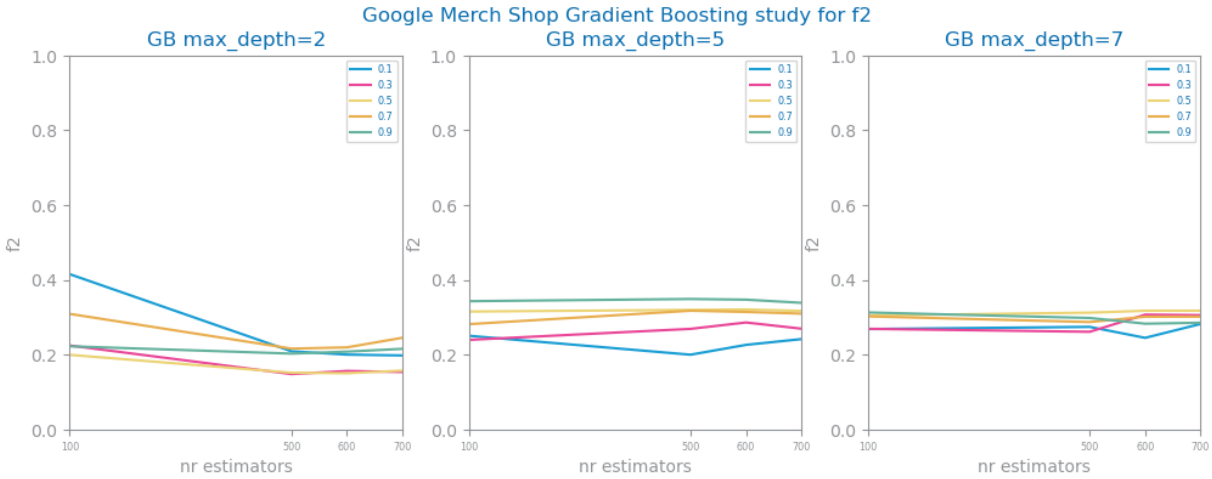
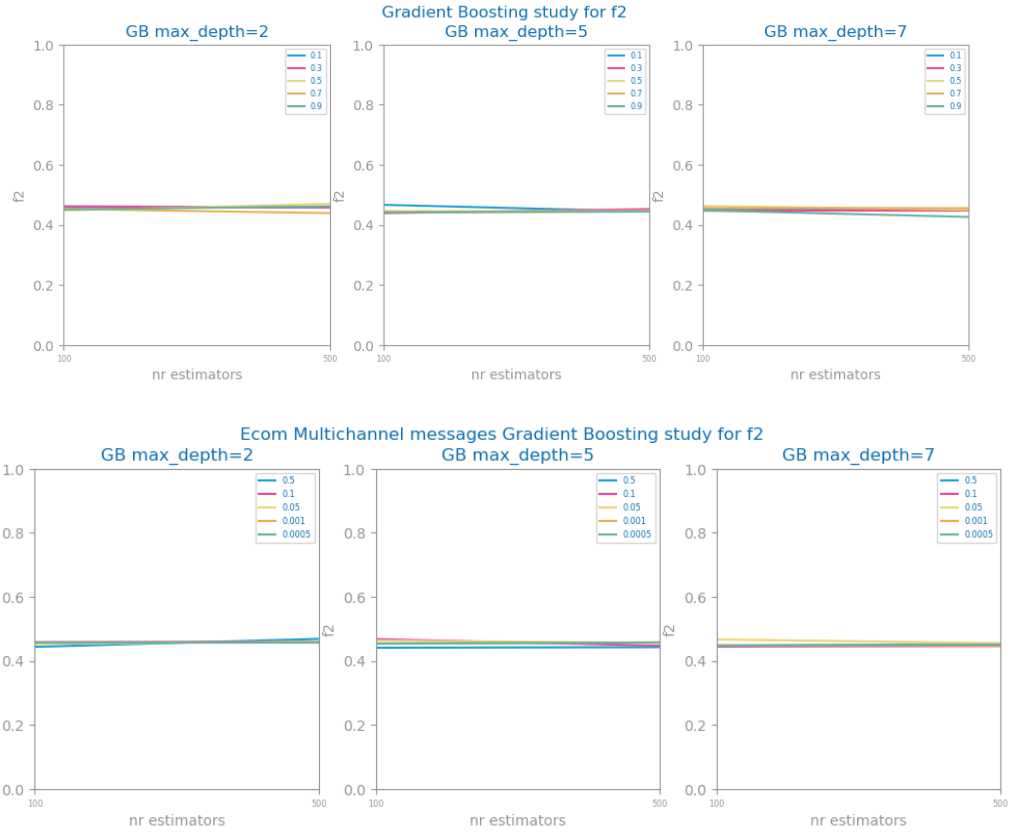


Figure 69 Gradient boosting different parameterisations comparison for dataset 2



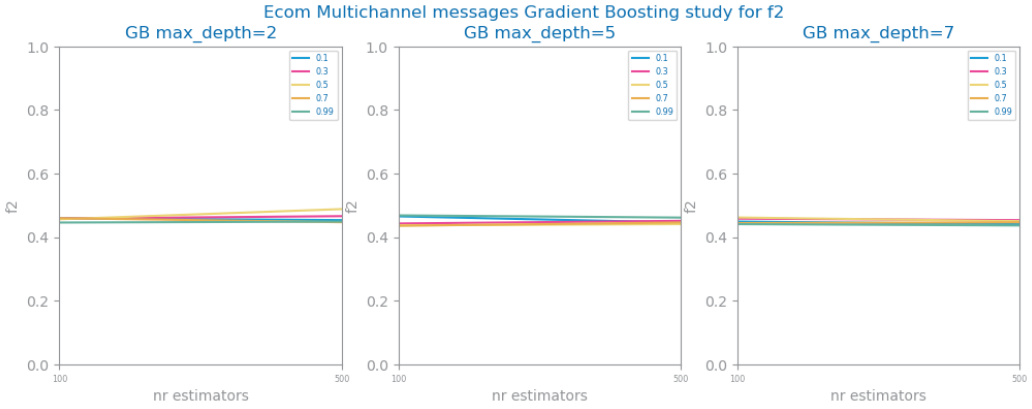


Figure 70 Gradient boosting different parameterisations comparison for dataset 3

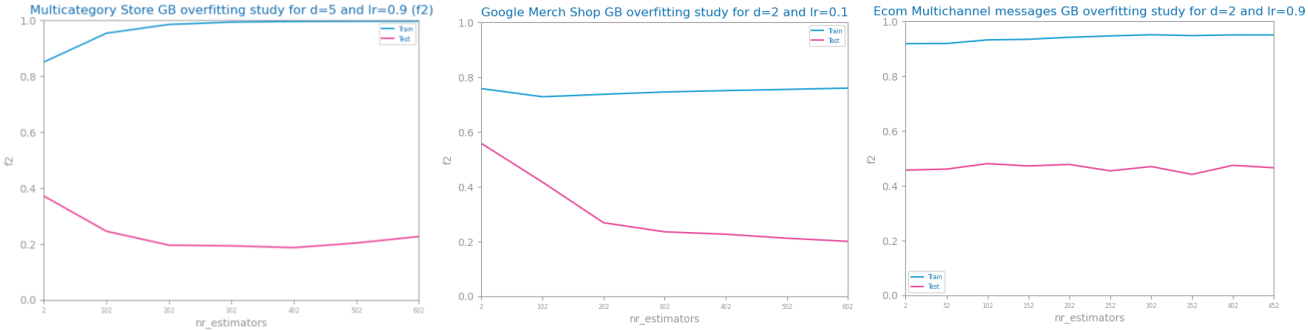


Figure 71 Gradient boosting overfitting analysis for dataset 1 (left), dataset 2 (middle), and dataset 3 (right)

Figure 72 Gradient boosting best model results for dataset 1 (left), dataset 2 (middle), and dataset 3 (right)

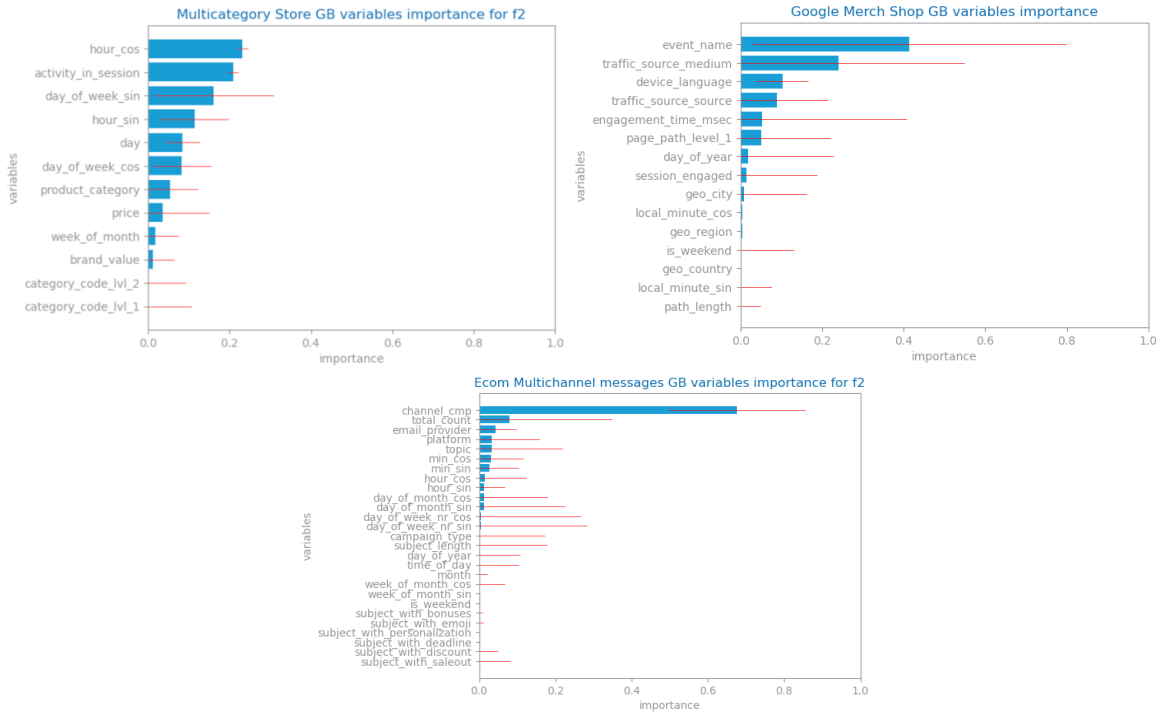


Figure 73 Gradient boosting variables importance for dataset 1 (left), dataset 2 (middle), and dataset 3 (right)

Multi-Layer Perceptrons (MLP)

In dataset 1 and 3, despite achieving higher values for f2 (58% and 52%), we have 100% recall and low precision for both datasets, which is not desirable. In dataset 2 similar happened, but with higher accuracy and precision, thus achieving less FP and FN. Unlike dataset 2, dataset 1 does not benefit from the increase of iterations (Figure 77). Several parameters were tested for dataset 3 but the model followed a bouncing pattern, which made the model training harder. There is no overfitting in the model's performance across all datasets, except dataset 2 between 1001-1201 iterations.

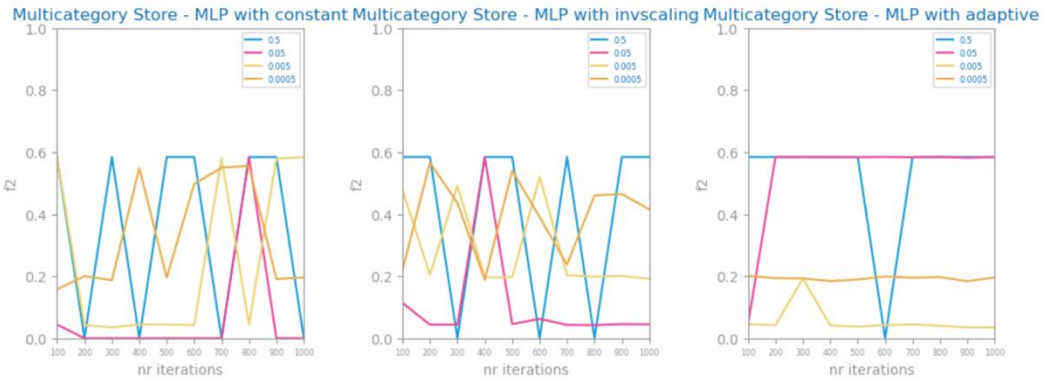


Figure 74 MLP different parameterisations comparison for dataset 1

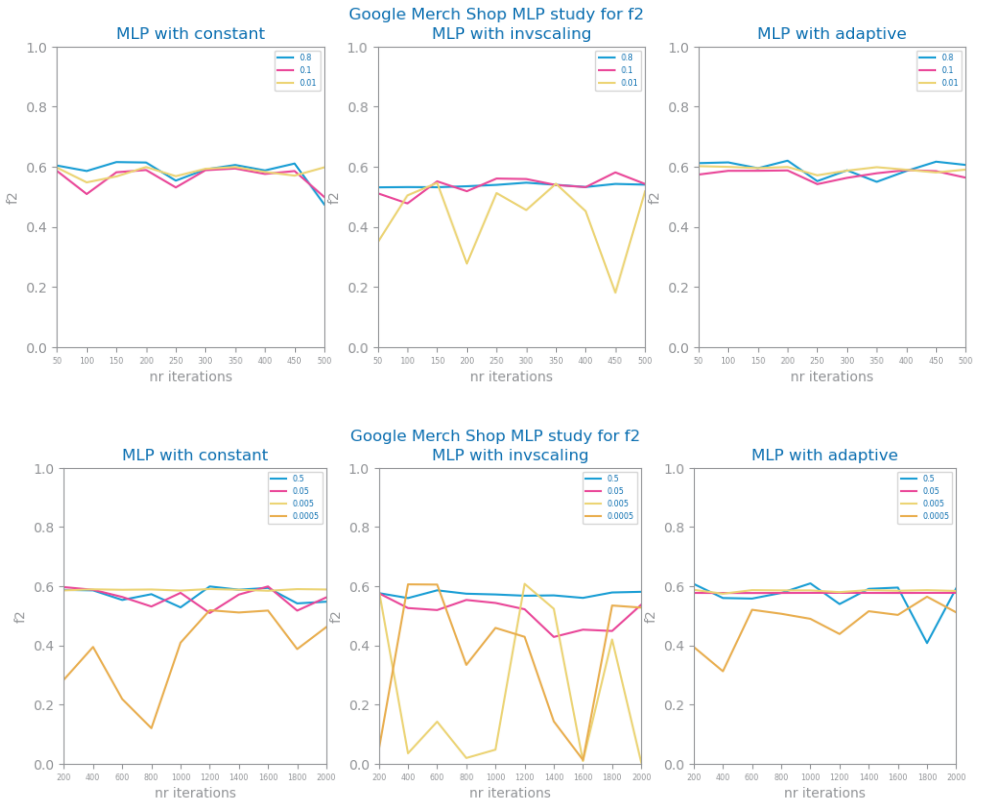
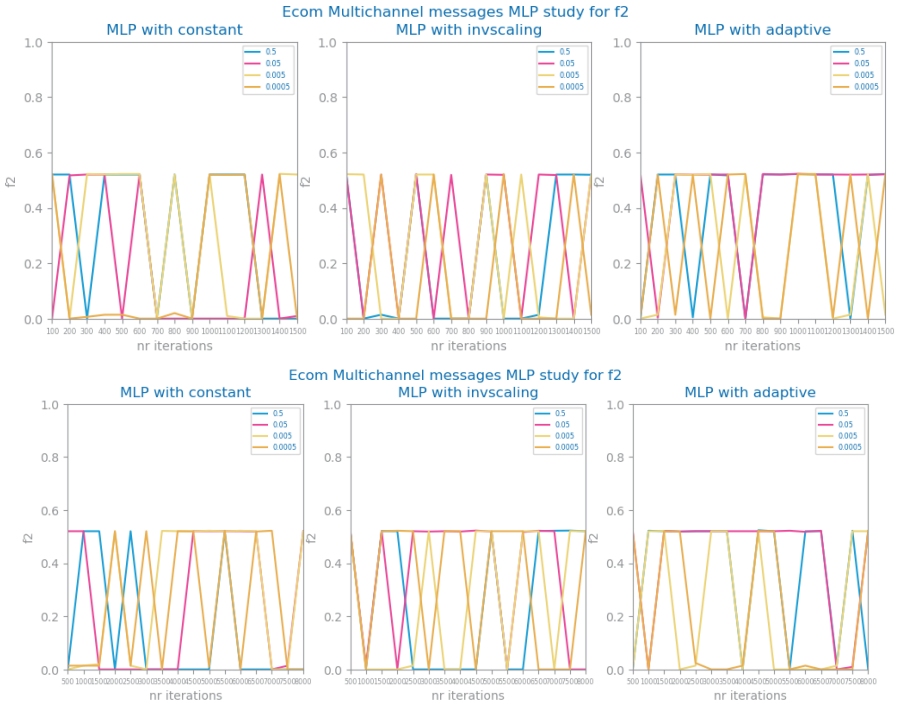


Figure 75 MLP different parameterisations comparison for dataset 2



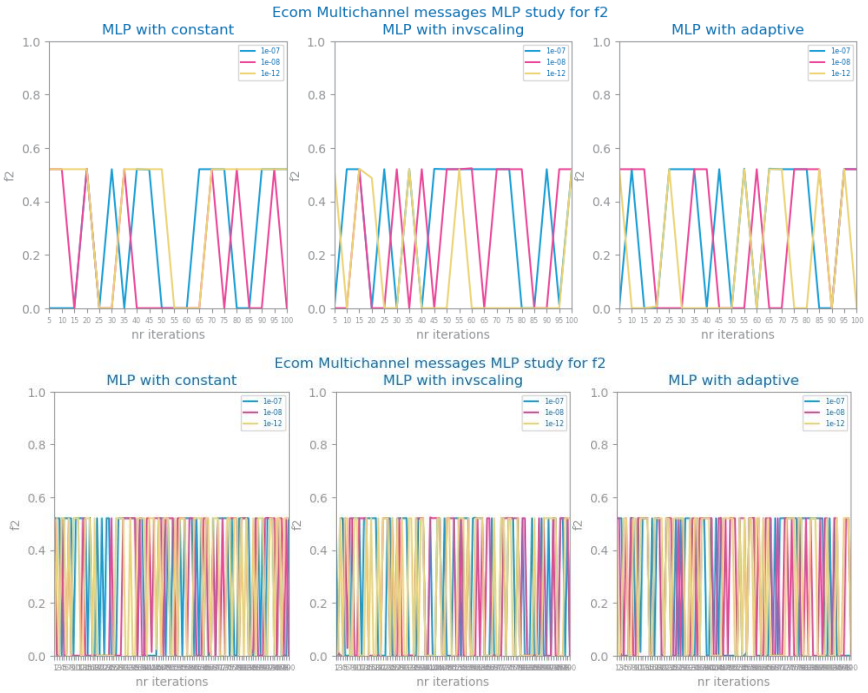


Figure 76 MLP different parameterisations comparison for dataset 3

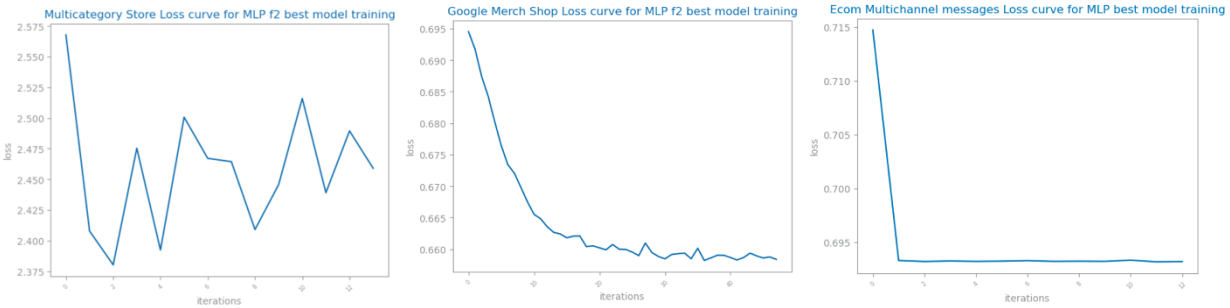


Figure 77 Loss curves for dataset 1 (left), dataset 2 (middle), and dataset 3 (right)

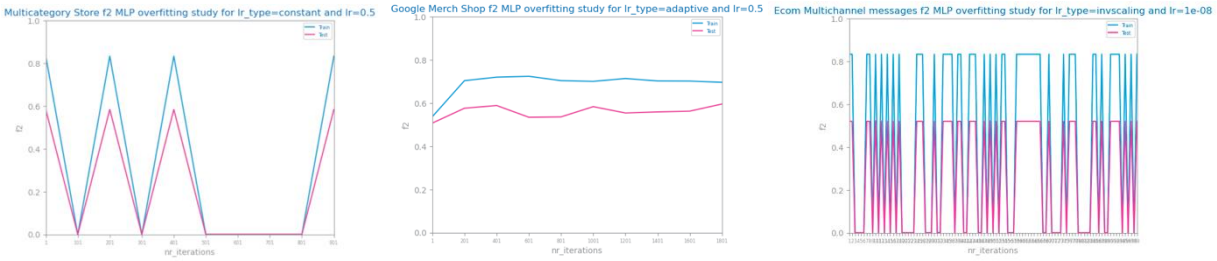


Figure 78 MLP overfitting analysis for dataset 1 (left), dataset 2 (middle), and dataset 3 (right)

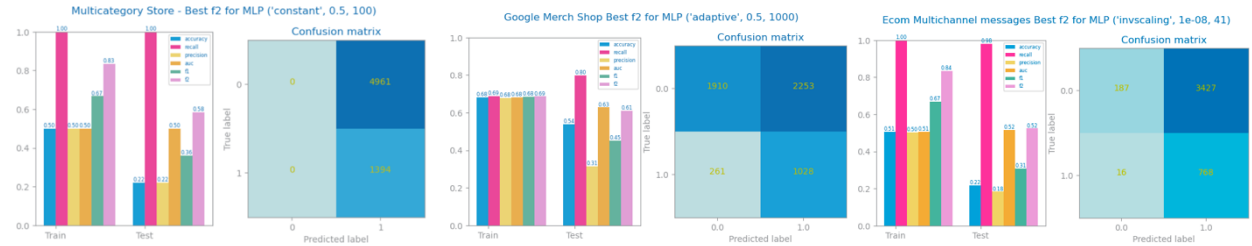


Figure 79 MLP best model results for dataset 1 (left), dataset 2 (middle), and dataset 3 (right)

4 CRITICAL ANALYSIS

On all datasets we used different algorithms (Table 1 Evaluation summary) that will allow us to do personalized marketing campaigns or digital experiences to targeted users.

For dataset 1, RF is considered the most useful, as it better predicts the number of TP, maintaining a low number of FP - better combination of recall (62%) and precision (28%). Time variables like day of week sin or day, are particularly important and can be used to discriminate the 'is purchase" class. This is likely due to a spike in website visits between 13/11 and 17/11 (Figure 80), possibly from a major marketing campaign or promotional event (Black Friday).

For dataset 2, DT is the best for identifying returning users, with traffic source and event name as the most important variables for the d=6 tree (beyond that it overfits). This % importance may allow us to decrease its complexity while maintaining good f2 with post-pruning. RF's f2 is also great but with lower accuracy, higher recall and different important variables.

For dataset 3, RF was the most useful (f2=50%) as it prioritizes identifying true users who clicked on the campaign. Although it captures some false clickers, RF offers the best recall-precision balance (74% - 22%), and FP's cost-risk is a less harmful trade-off in campaign messages. RF considers campaign channel, number of recipients and campaign subject length as the most important variables.

Model wise, GB has the worst f2 on all datasets, with especially low recall and high accuracy on datasets 1 and 2. While on dataset 1 and 2, other variables gained importance vs DT/RF ones, on dataset 3, channel cmp is the most important variable on DT, RF and GB which could explain its high recall in GB.

MLP achieved the highest f2 across all datasets. However, for datasets 1 and 3, recall is too high (~100%) while precision and accuracy are too low, making MLP useless for these cases. This issue arises because scaling was not applied to datasets 1 and 3, which is crucial for MLP since feature scales affect the learning process (Figure 74, Figure 76). In contrast, dataset 2 has the highest f2 (62%), was scaled, and achieved 80% recall and 31% precision. This suggests MLP could still be useful for dataset 2, although it had better accuracy with DT and RF.

All in all, RF is what best fits the characteristics of the 3 datasets, as it's where we obtain the best combination between recall and precision.

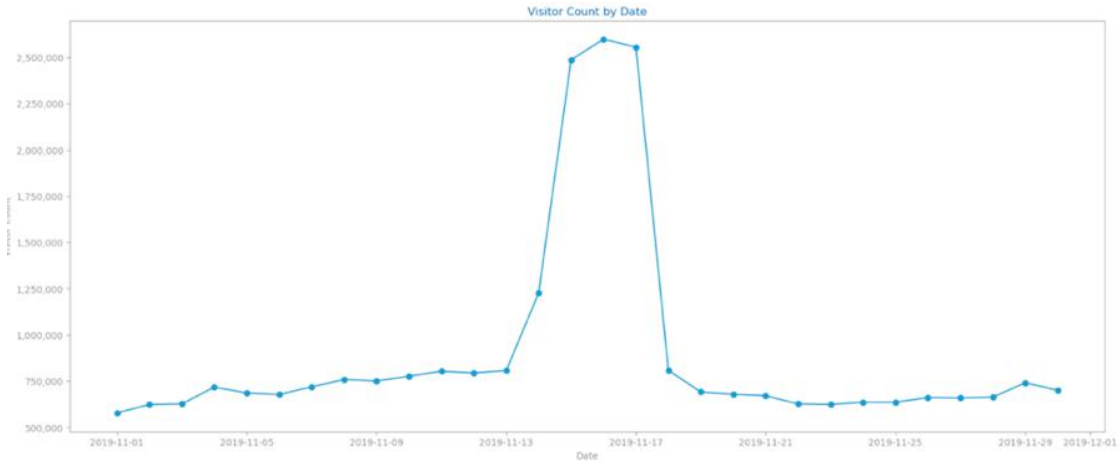


Figure 80 Visitor count by date in dataset 1

Table 1 Evaluation summary

Algorithm	Dataset	F2	Accuracy	Recall	Precision
NB	1	49%	41%	69%	23%
	2	58%	48%	79%	29%
	3	46%	54%	64%	22%
KNN	1	45%	52%	52%	25%
	2	58%	56%	73%	31%
	3	46%	53%	63%	22%
DT	1	51%	52%	66%	26%
	2	58%	69%	57%	31%
	3	45%	53%	63%	21%
RF	1	50%	56%	62%	28%
	2	57%	56%	71%	31%
	3	50%	49%	74%	22%
GB	1	24%	71%	23%	29%
	2	42%	73%	41%	43%
	3	49%	37%	80%	19%
MLP	1	58%	22%	100	22%
	2	62%	54%	80%	31%
	3	52%	22%	98%	18%