TÉCNICO LISBOA

DS 2023

Data Science Project

Team nr: 5	**Student 1 :** Eduardo Lança Lobo	**IST nr:** 99213
	Student 2 : Inês Ye Ji	**IST nr:** 99238
	Student 3 : Jiqi Wang	**IST nr:** 99241
	Student 4 : Raquel Filipa Marques Cardoso	**IST nr:** 99314

CLASSIFICATION

1 DATA PROFILING

While studying the variables in each dataset, we made the *Age* variable from D2 into numeric, since it was symbolic. We found an invalid entry in the *MonthlyBalance* variable from D2 and made it a MV.

Data Dimensionality

The dimensionality curse isn't present in either dataset, since both have a lot more records than variables. For D1 and D2 there are more than 9523 and 3571 records per dimension, respectively.

D1 has mainly binary/symbolic data, focusing on the physical or mental aspects of the patient. D2 has mainly numeric data, capturing various countable credit-related information about the person.

The variables with the most missing values for the datasets are *TetanusLast10Dup* (9%) - D1, *CreditMix* (20%) - D2.

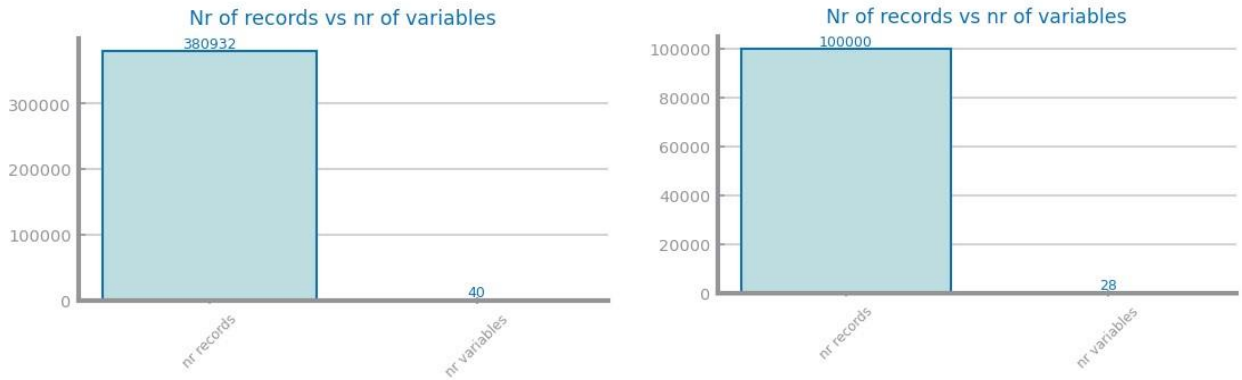


Figure 1 Nr Records x Nr variables for dataset 1 (left) and dataset 2 (right)

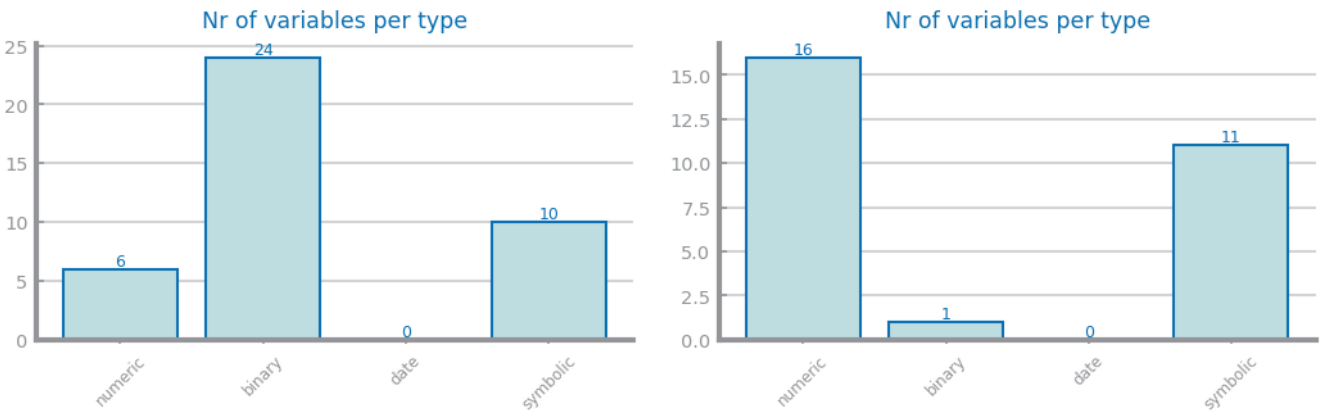


Figure 2 Nr variables per type for dataset 1 (left) and dataset 2 (right)

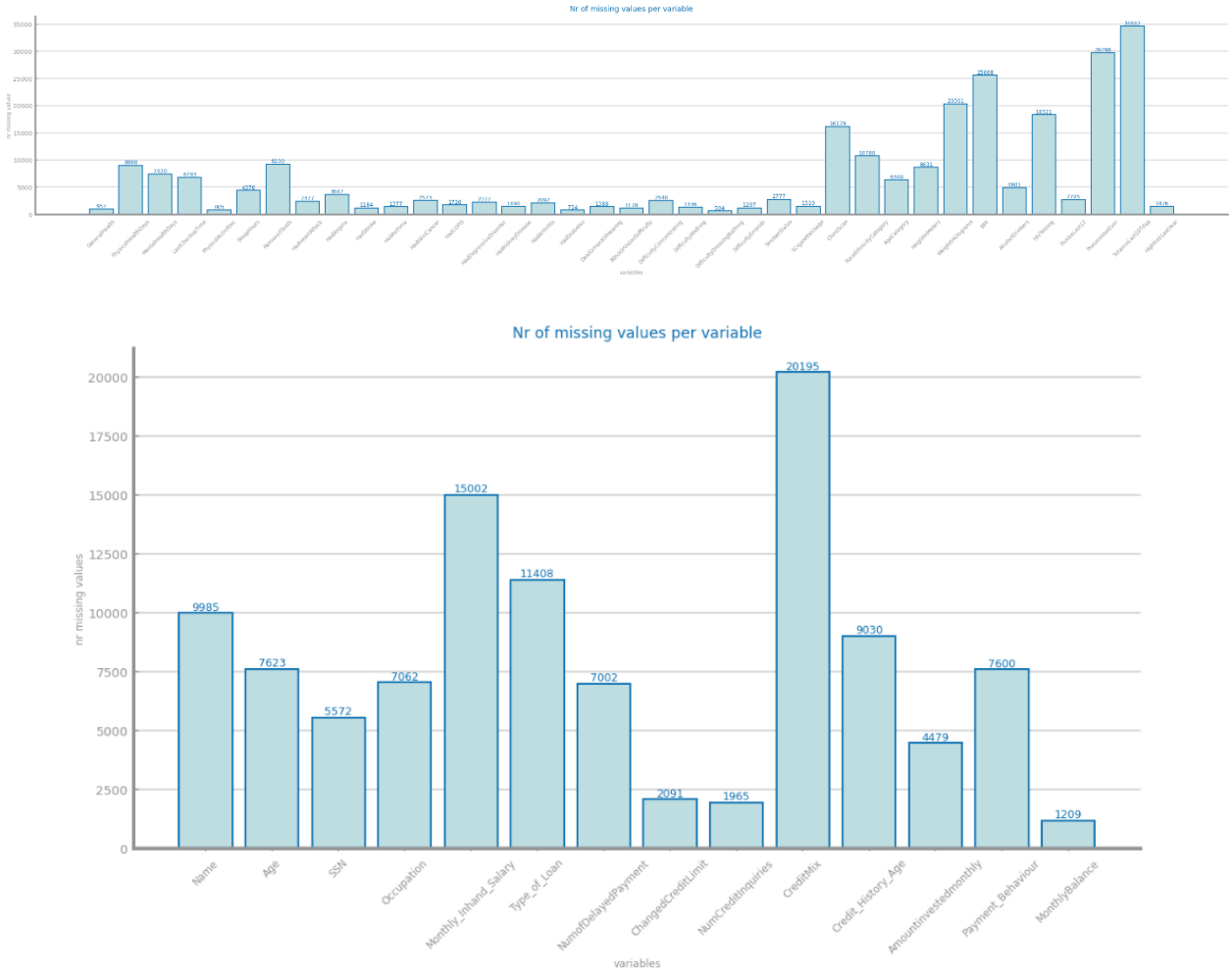


Figure 3 Nr missing values for dataset 1 (above) and dataset 2 (below)

Data Distribution

Both targets are binary, for D1 is *CovidPos* and for D2 is *Credit_Score*. Both are unbalanced, and the most important class is the minority in D1. There are some clear outliers and negative values in D2. We used STDEV=2 and IQR=1.5 for outliers in both D1 and D2.

In Fig3, D2 has similar distributions in all vars except the 2nd. In D1 4 vars fit **LogNormal** and 2 **Exp**. In D2, 9 vars fit **Exp** and the other 7 **LogNormal**.

The range varies a lot between vars in D1 and D2, some are up to 3 and others up to 10^7.

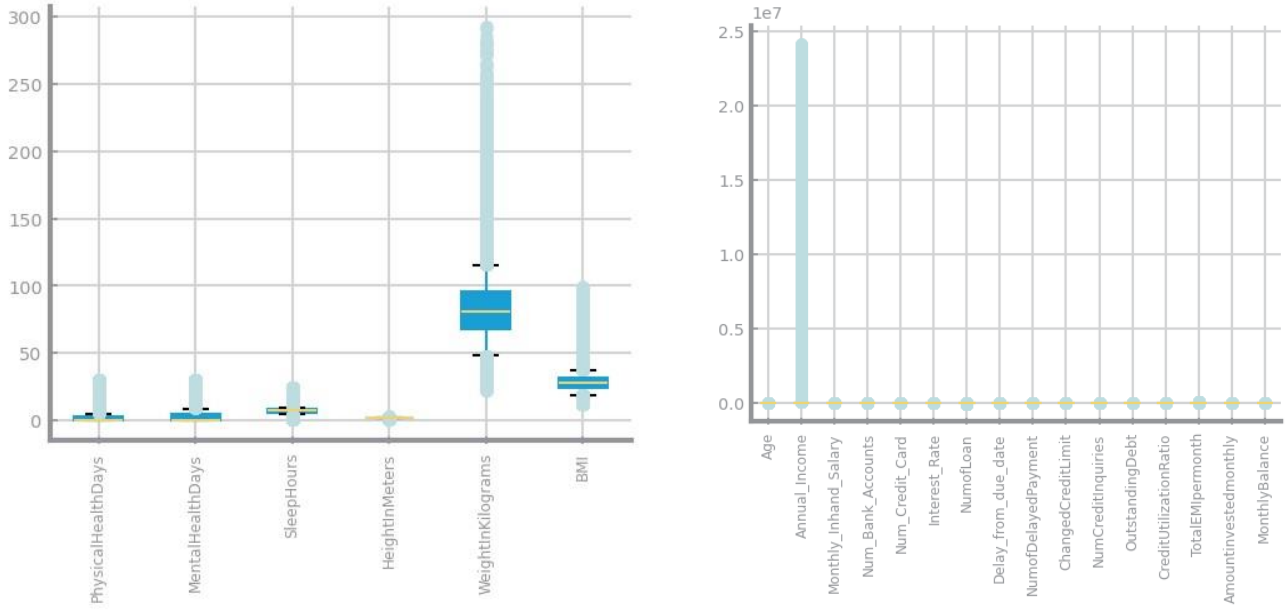


Figure 4 Global boxplots dataset 1 (left) and dataset 2 (right)

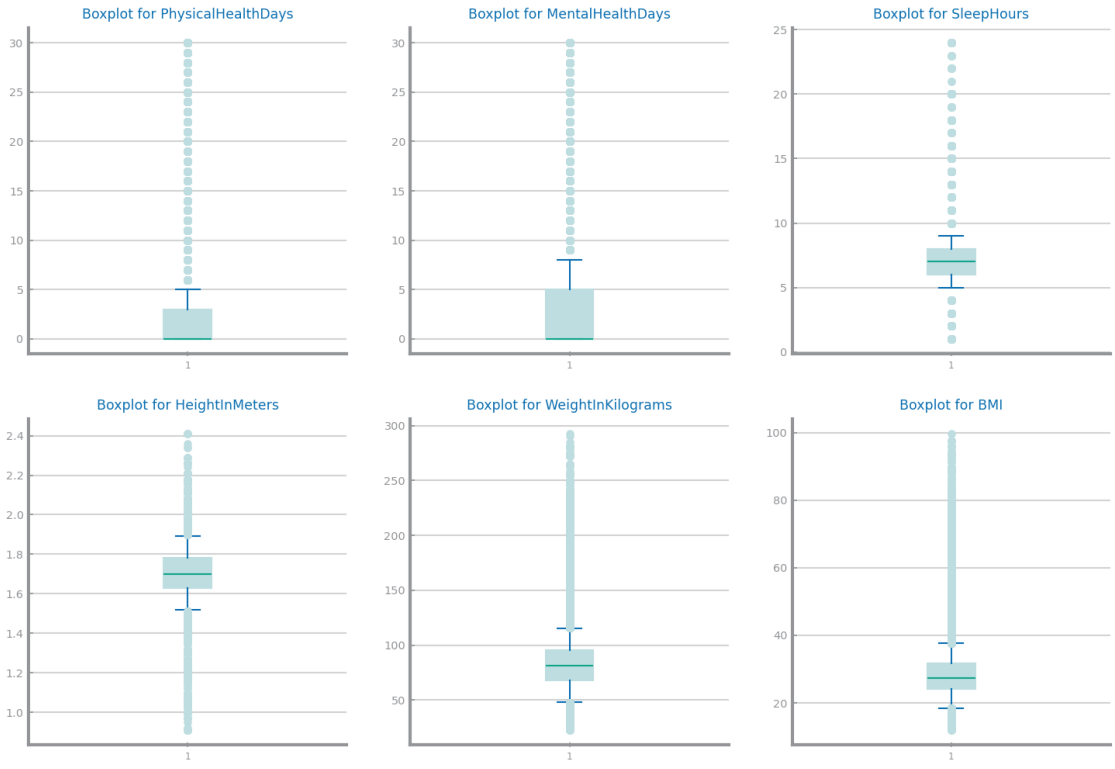


Figure 5 Single variable boxplots for dataset 1

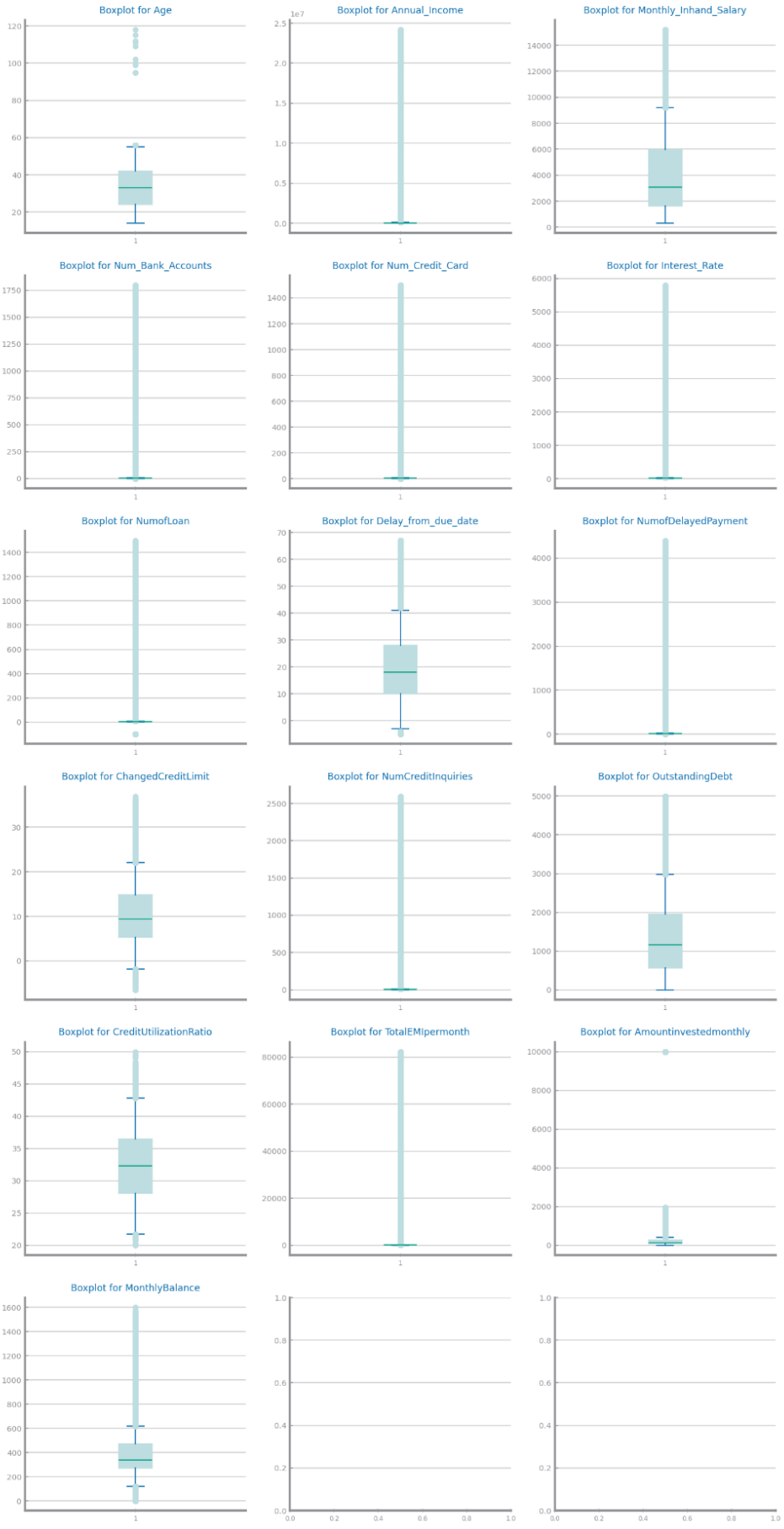


Figure 6 Single variable boxplots for dataset 2

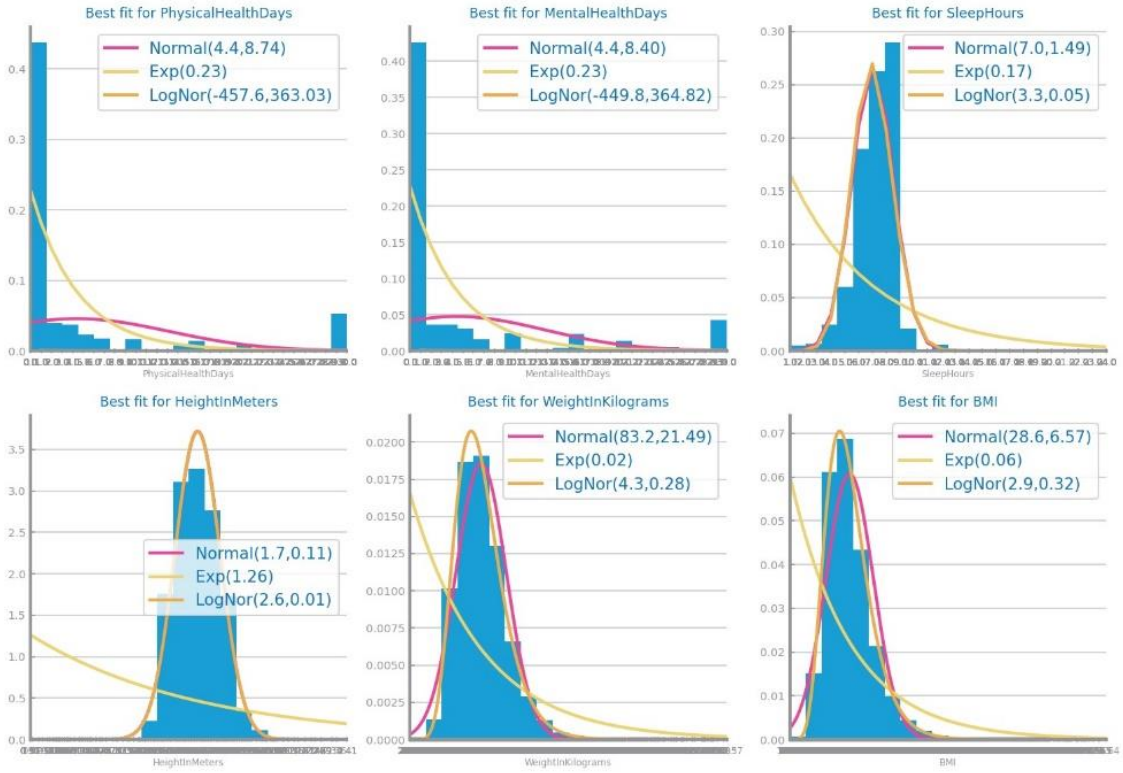


Figure 7 Histograms for dataset 1

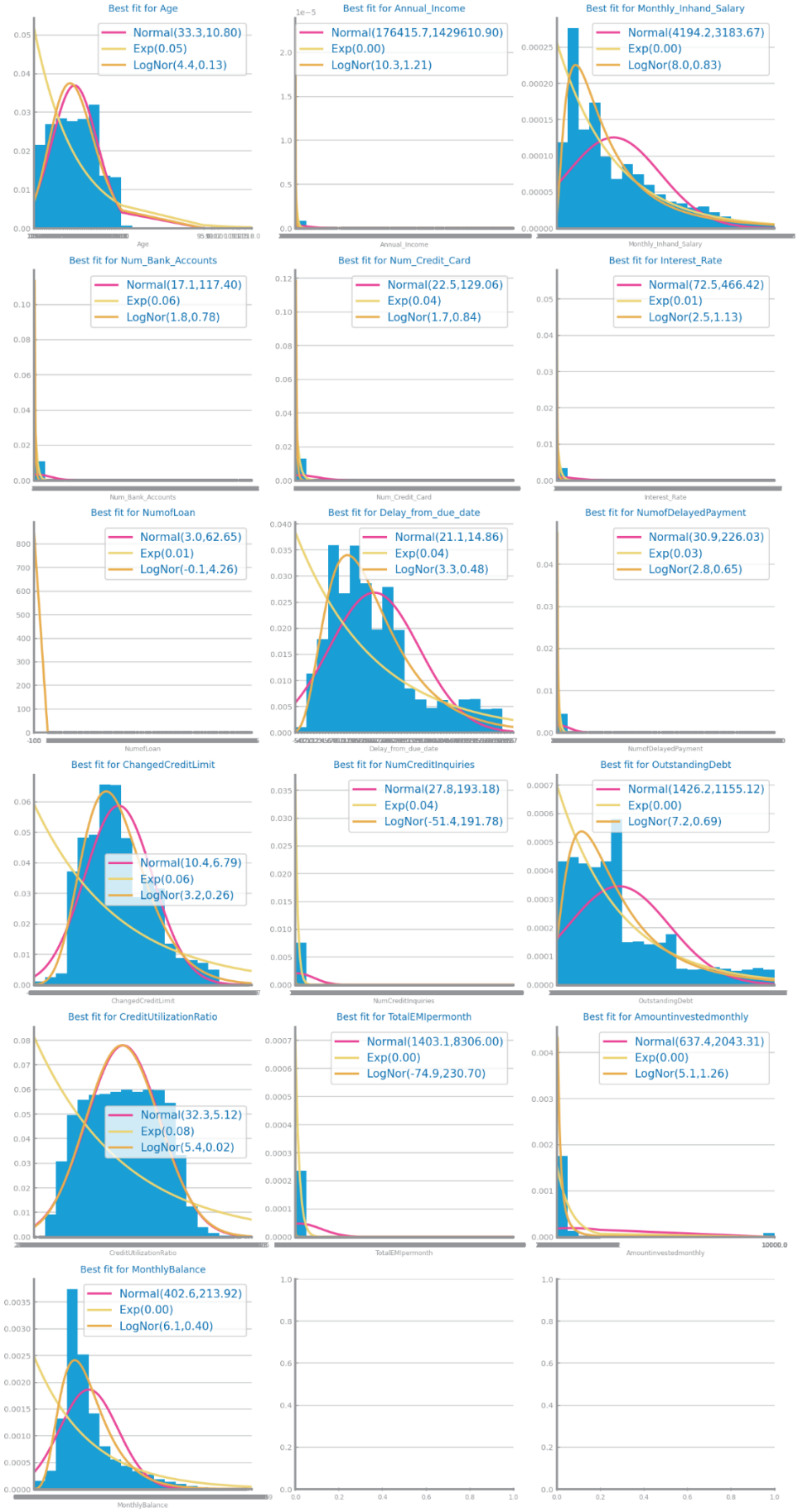


Figure 8 Histograms for dataset 2

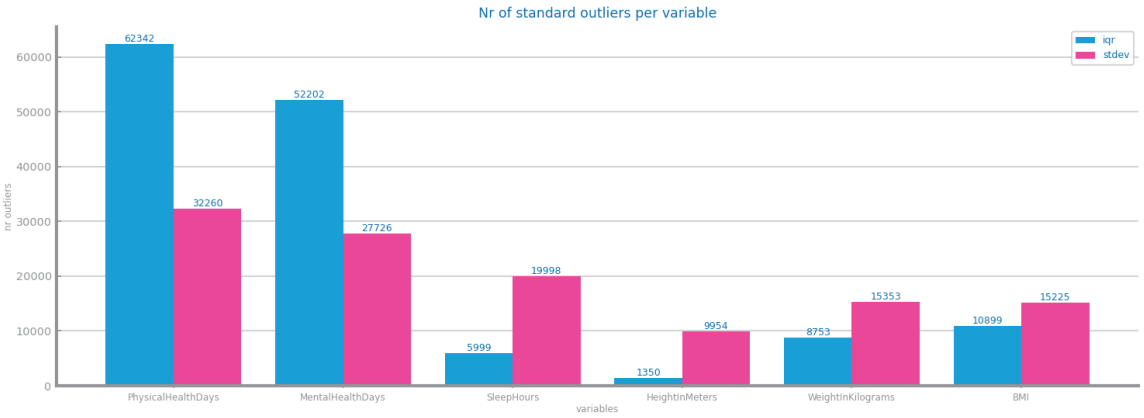


Figure 9 Outliers study dataset 1

Figure 10 Outliers study for dataset 2

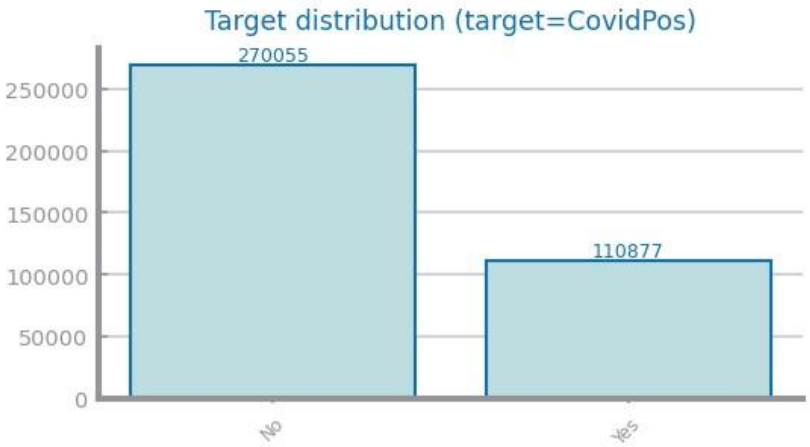


Figure 11 Class distribution for dataset 1

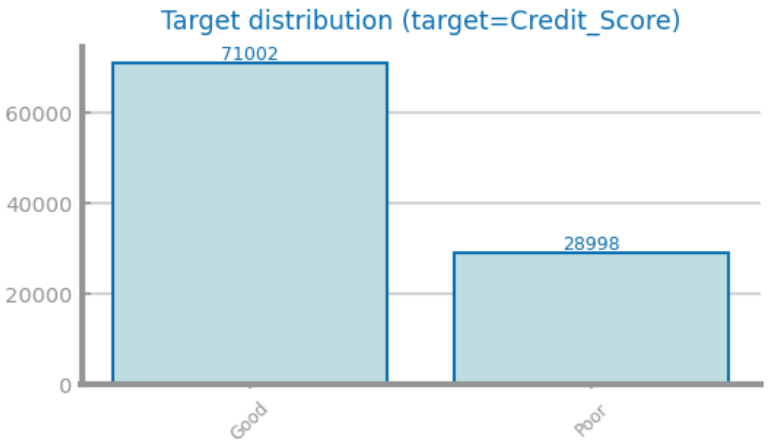


Figure 12 Class distribution for dataset 2

Data Granularity

In D1, we studied *SmokingStatus* and *E-cigarette Usage*, categorizing into **Former**, **Currently** and **Never Used**, and on **usage status**. There are more Smokers with *CovidPos*. For *States*, by Nominal GDP per Capita 2022, considering **low**: Puerto Rico to Wisconsin, **mid**: Ohio to Illinois, **high**: Washington to North Dakota.

In D2, we studied *Occupation* - considering Doctor to Teacher **STE**, Accountant to Entrepreneur **Economics**, Lawyer to Writer **SSH** and rest **Art**; and *Month*. There is no data in the 3rd trimester.

Figure 13 Granularity analysis for dataset 1

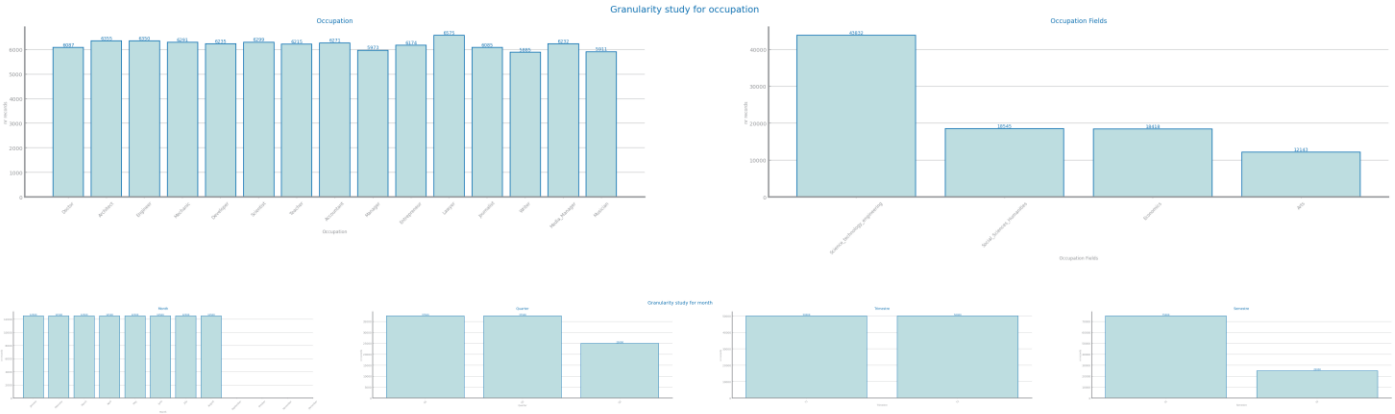


Figure 14 Granularity analysis for dataset 2

Data Sparsity

Both D1 and D2 have a notable number of non-zero elements, considering the available values of each var, therefore the datasets aren't sparse.

In D1, *BMI* and *WeightInKilograms* are highly correlated, *Sex* and *HeightInMeters* are also correlated.

In D2, *Monthly_Inhand_Salary and MonthlyBalance* are highly correlated, *Delay_from_due_date* and *CreditMix* are correlated, *CreditMix* and *OutstandingDebt* as well.

Besides these, most vars aren't correlated, therefore their information shouldn't be redundant.

Figure 15 Sparsity analysis for dataset 1

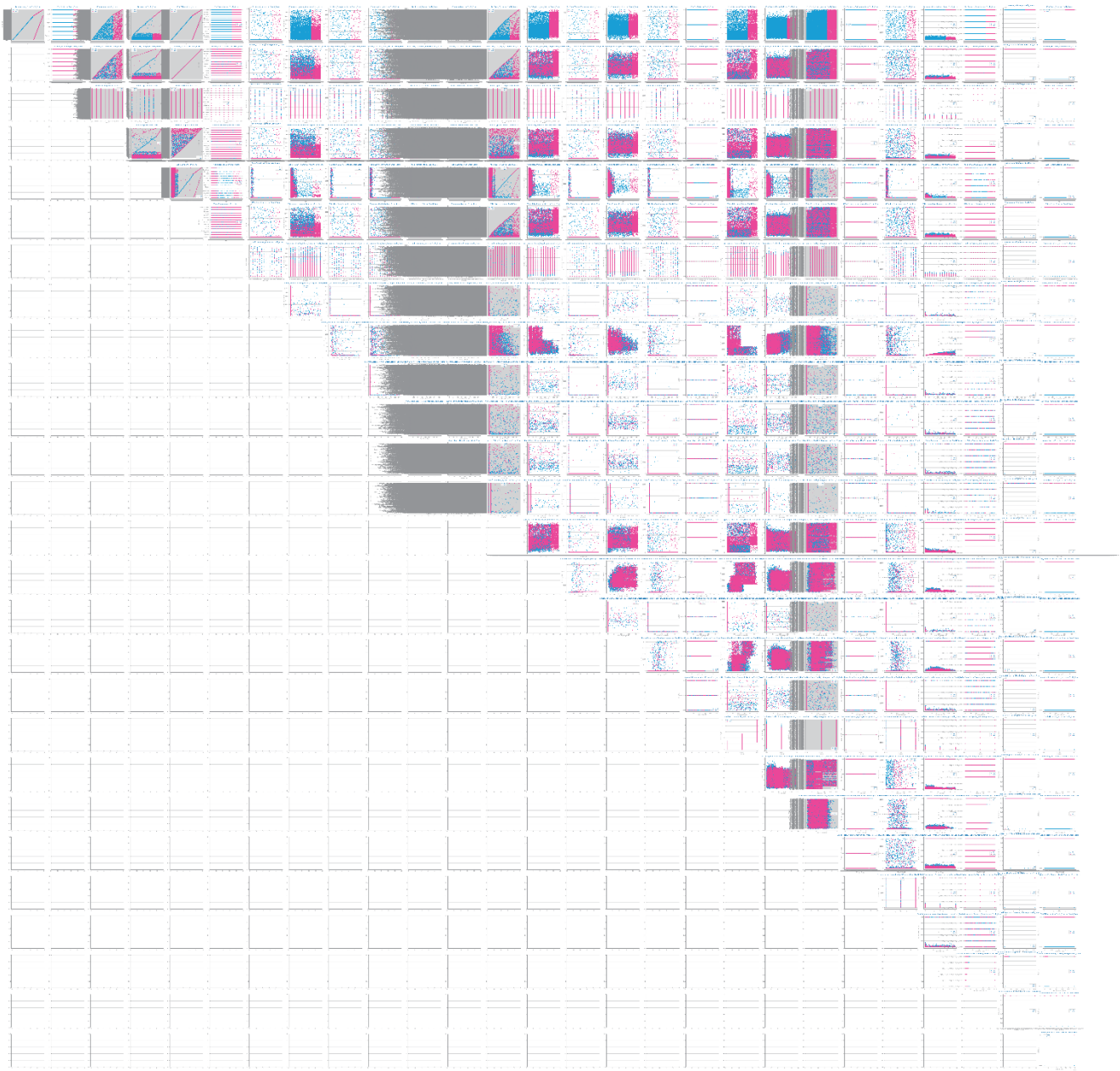


Figure 16 Sparsity analysis for dataset 2

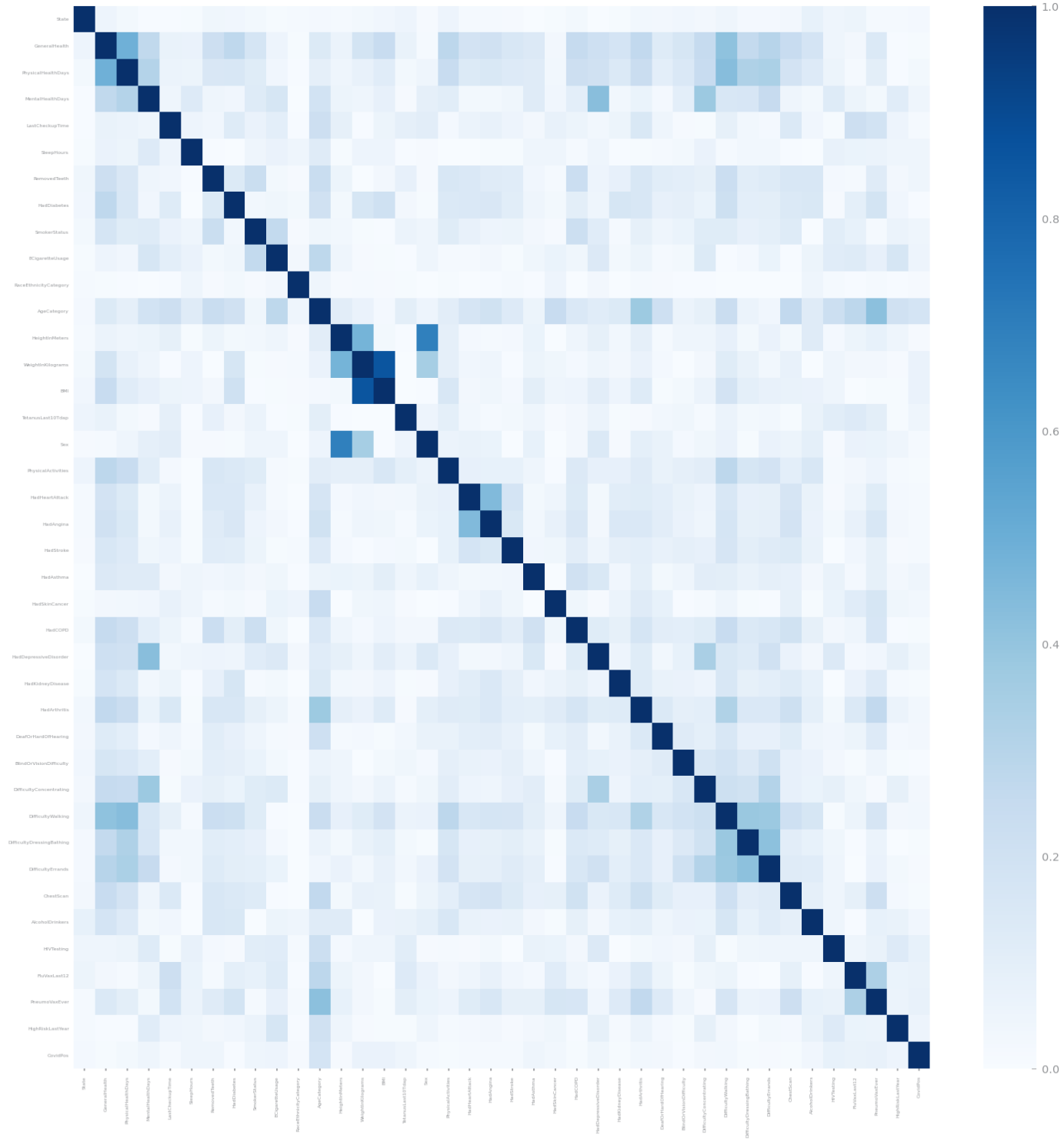


Figure 17 Correlation analysis for dataset 1

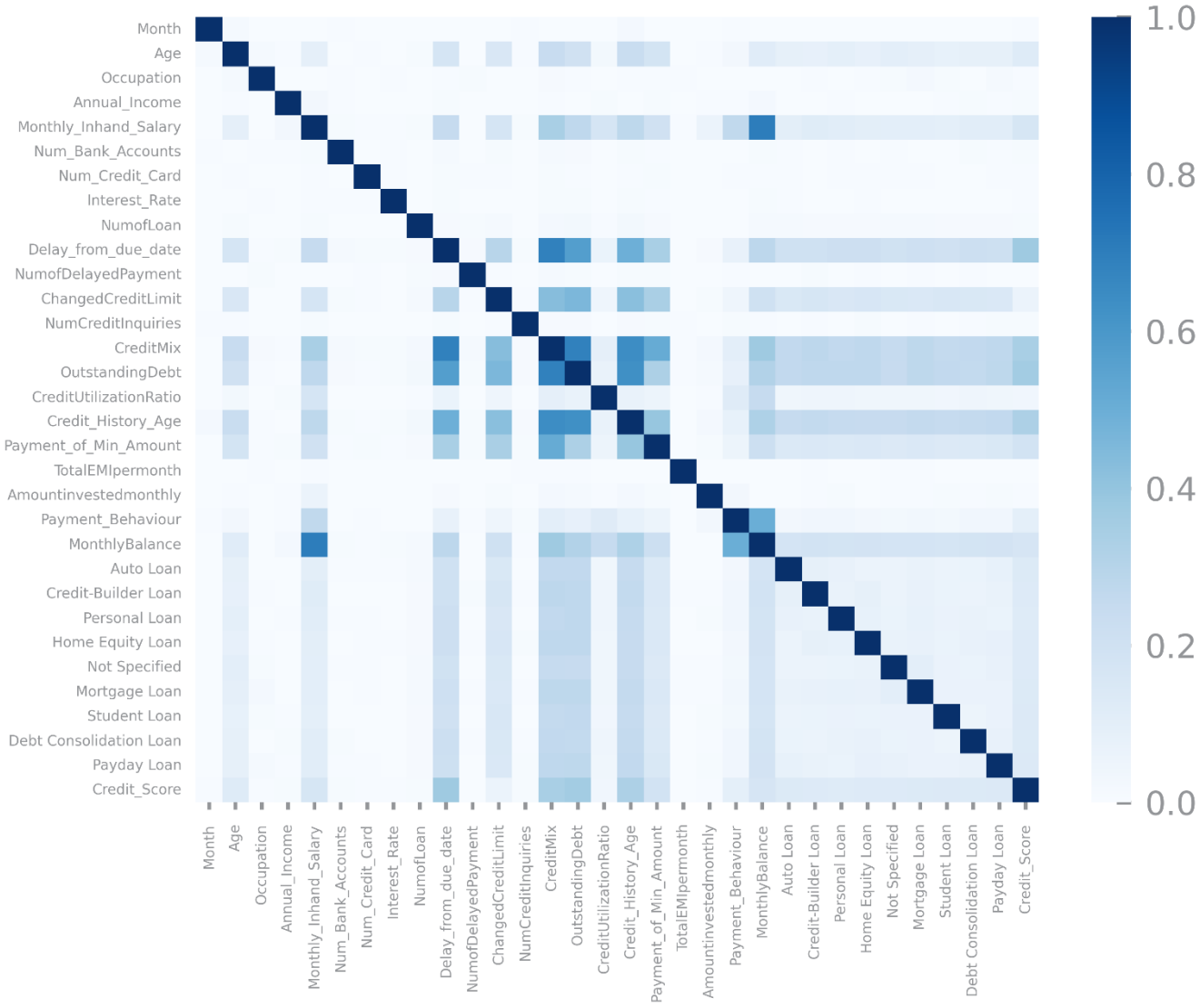


Figure 18 Correlation analysis for dataset 2

2 DATA PREPARATION

Before Variable Encoding, we made all existent negative values of D2 into missing values and capped the values in Age from 0-120.

Variables Encoding

In D1 the variable *ID* was dropped due its unique identifier nature, *State* was encoded using Nominal GDP per Capita in 2022 (in thousands of USD) by **OEBT**, and **OLE** was applied on all the remaining symbolic and binary vars.

In D2 we dropped unique identifiers and used **OEBT** for *Occupation*, *Payment_Behaviour*, *Credit_History_Age*. We used **OLE** on *Credit_Score*, *Month*, *CreditMix*.

In *Type_of_Loan*, each entry can hold more than one type of loan, so we created a var for each type, in which its value is the number of times the respective type of loan appears in *Type_of_Loan*. In the end of the process *Type_of_Loan* is dropped. This way we don't lose information and avoid creating excessive new vars by using dummification.

Variable/Encoded Values	**State Nominal GDP per capita (2022) in thousands of US Dollars**
State	State Name

Table 1 - Encoding of State variable in D1 – OEBT

Variable Name/Encoded Values	**0**	**1**
Sex	Female	Male
All the remaining binary variables	No	Yes

Table 2 - Encoding of binary variables in D1 - OLE

Variable/Encoded Values	**0**	**1**	**2**	**3**	**4**	**5**
GeneralHealth		Poor	Fair	Good	Very good	Excellent
LastCheckupTime		5 or more years ago	Within past 5 years (2 years but less than 5 years ago)	Within past 2 years (1 year but less than 2 years ago)	Within past year (anytime less than 12 months ago)	
RemovedTeeth		All	6 or more, but not all	1 to 5	None of them	
HadDiabetes		Yes	Yes, but only during pregnancy (female)	No, pre-diabetes or borderline diabetes	No	
SmokerStatus		Current smoker – now smokes every day	Current smoker – now smokes some days	Former smoker	Never smoked	
ECigaretteUsage		Use them every day	Use them some days	Not at all (right now)	Never used e-cigarettes in my entire life	
RaceEthnicityCategory	Hispanic	White only, Non-Hispanic	Black only, Non-Hispanic	Multiracial, Non-Hispanic	Other race only, Non-Hispanic	
TetanusLast10Tdap	No, did not receive any tetanus shot in the past 10 years	Yes, received tetanus shot but not sure what type	Yes, received tetanus shot, but not Tdap	Yes, received Tdap		

Table 3 - Encoding of symbolic variables in D1 - OLE

AgeCateogry	Encoding
Age 18 to 24	1
Age 25 to 29	2
Age 30 to 34	3
Age 35 to 39	4
Age 40 to 44	5
Age 45 to 49	6

Age 50 to 54	7
Age 55 to 59	8
Age 60 to 64	9
Age 65 to 69	10
Age 70 to 74	11
Age 75 to 79	12
Age 80 or older	13

Table 4 - Encoding of AgeCategory variable in D1 – OLE

Credit_History_Age	Encoding
XXX Years and YYY Months	(12 * XXX) + YYY

Table 5 - Encoding of Credit_History_Age variable in D2 – OEBT

Month	Encoding
Month name	Corresponding Month name integer

Table 6 - Encoding of Month variable in D2 – OLE

Variable/Encoded Values	**0**	**1**	**2**	**3**	**4**	**5**	**6**
Credit_Score	Poor	Good					
CreditMix	Bad	Standard	Good				
Payment_of_Min_Amount	No	Yes	NM				
Payment_Behaviour		Low_spent_Small_value_payments	Low_spent_Medium_value_payments	Low_spent_Large_value_payments	High_spent_Small_value_payments	High_spent_Medium_value_payments	High_spent_Large_value_payments

Table 7 - Encoding of binary and symbolic variables in D2 (OLE, OLE, OLE, OEBT respectively)

Occupation	Encoding
Doctor	1
Architect	2
Engineer	3
Mechanic	4
Developer	5
Scientist	6
Teacher	7
Accountant	8
Manager	9
Entrepreneur	10
Lawyer	11
Journalist	12
Writer	13
Media_Manager	14
Musician	15

Table 8 - Encoding of Occupation variable in D2 – OEBT

Type_of_Loan	Encoding
Enumeration of existent types of loan	Generation of new variables named after each different type of loan. Each new variable holds the number of existents loans of that type.

Table 9 - Encoding of Occupation variable in D2 – Feature Generation

Missing Value Imputation

In D1, AP1 drops vars and records with more than 5% and 30% MV respectively, then imputes the median on numeric vars and mode on symbolic/bool. AP2 drops vars and records with more than 10% MV.

In D2, AP1 drops vars and records with more than 30% and 10% MV respectively. AP2 drops vars and records with more than 15% and 10% MV and uses mean.

Results were similar but we chose AP2 on D1 and AP1 on D2 due to better performance, especially **Recall** for D1 and **Accuracy** for D2. No columns were dropped.

Figure 19 Missing values imputation results with different approaches for dataset 1

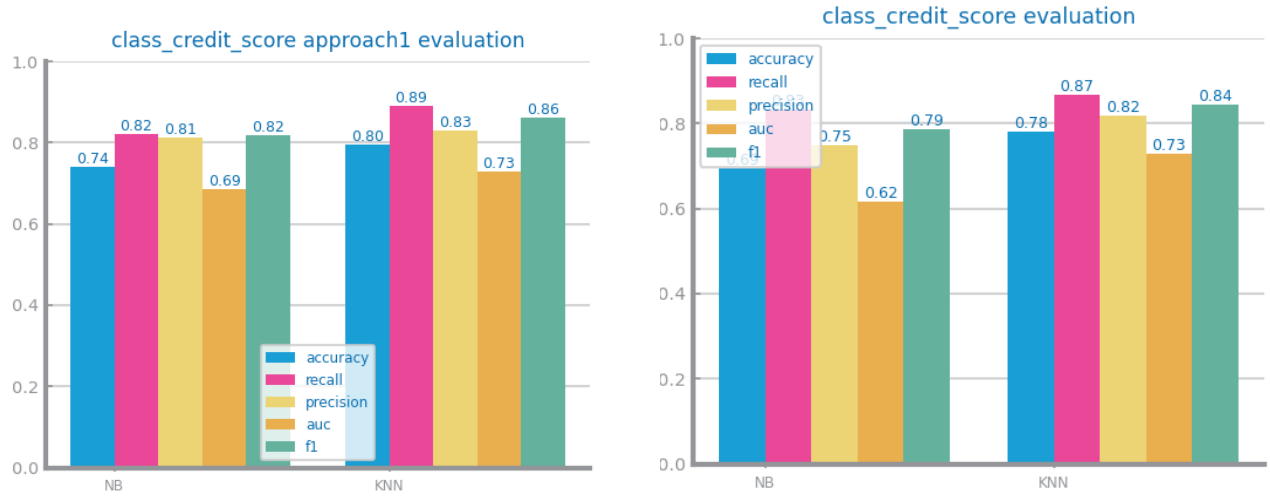


Figure 20 Missing values imputation results with different approaches for dataset 2

Outliers Treatment

We used a STDEV of 4 and 3 for D1 and D2, respectively, to decide the outliers. We evaluated approaches that drop, truncate or replace outliers with median, using NB and KNN.

We selected the replace outliers with the median approach for both datasets. Since D1 is a health dataset, dropping outliers would remove potentially important case studies of *Covid Pos*. In D2 the approach had the best **accuracy** and **F1-measure** scores, which we considered the most important in studying *Credit Scores*.

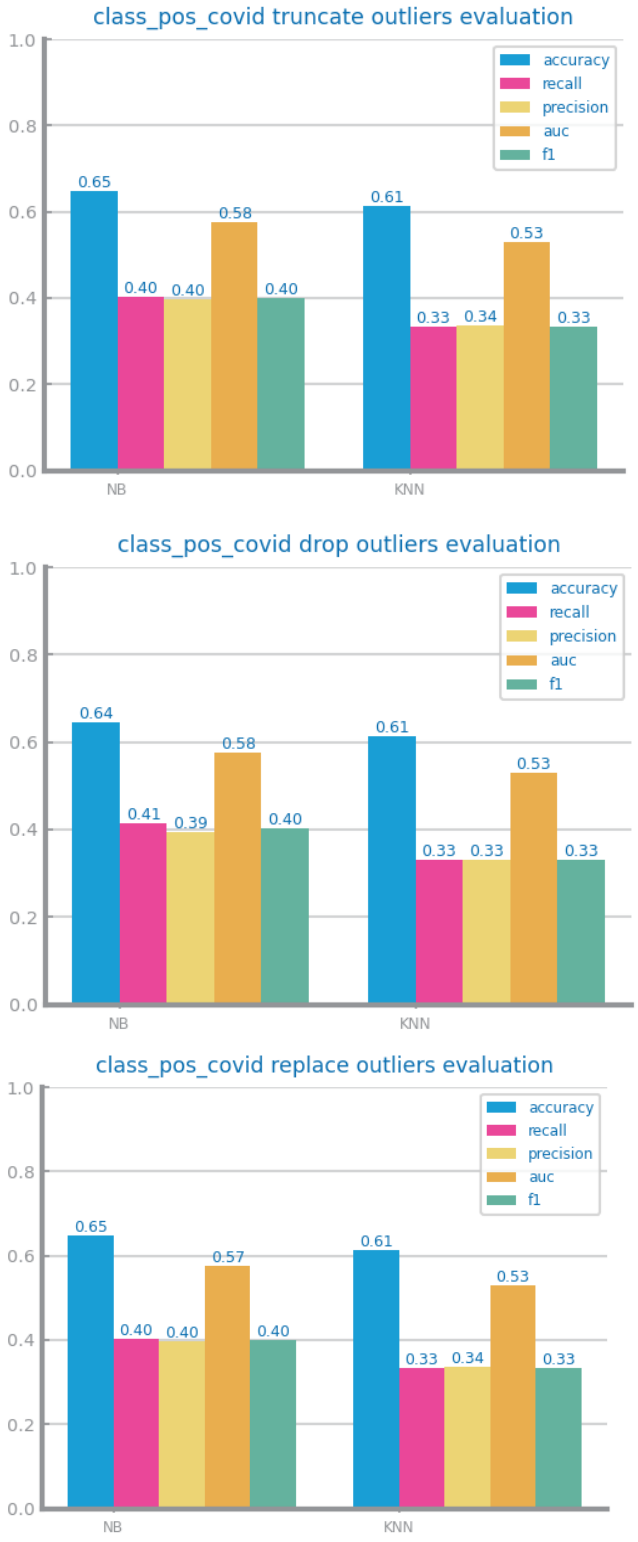


Figure 21 Outliers imputation results with different approaches for dataset 1

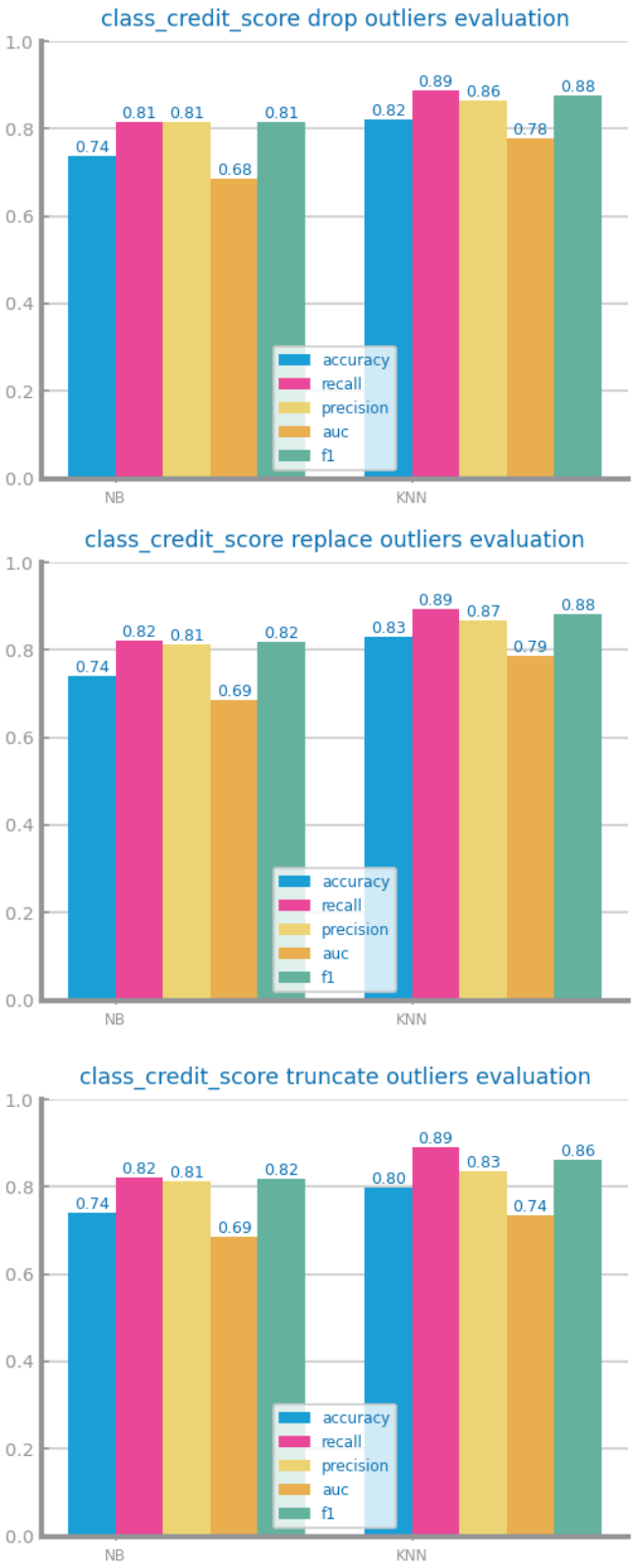


Figure 22 Outliers imputation results with different approaches for dataset 2

Scaling

Evaluated both Minmax and Zscore. Neither transformation improved the results, therefore we followed the paper's recommendation and didn't apply scaling in either dataset.

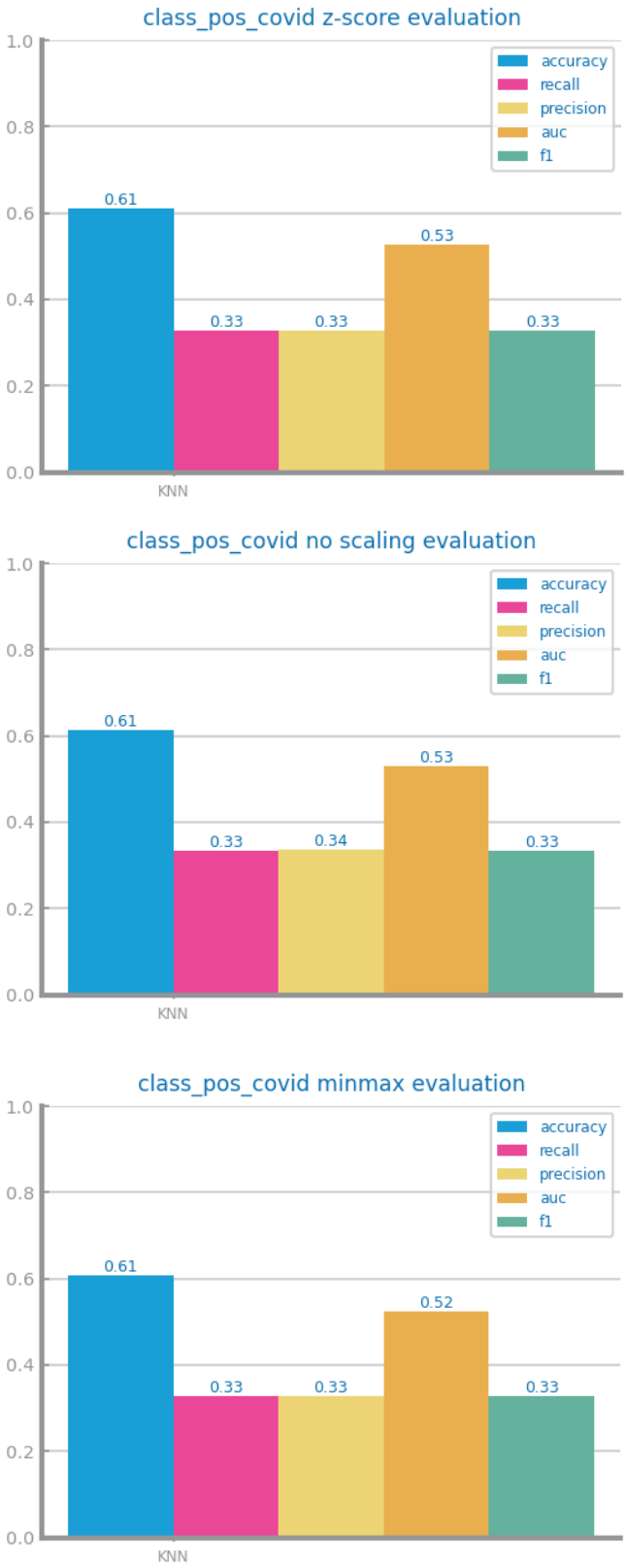
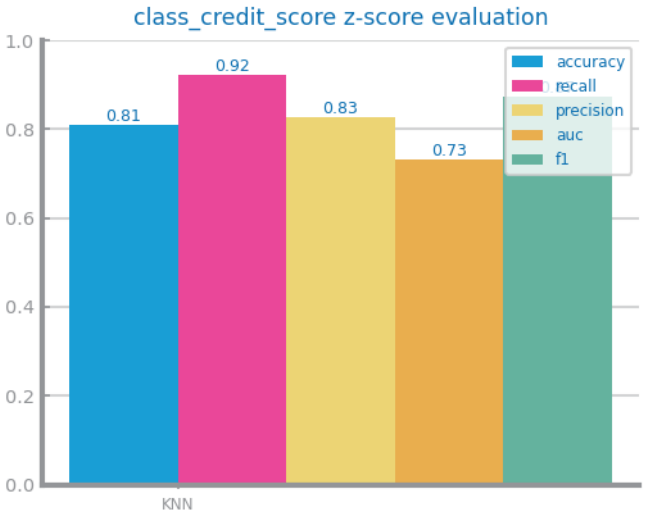
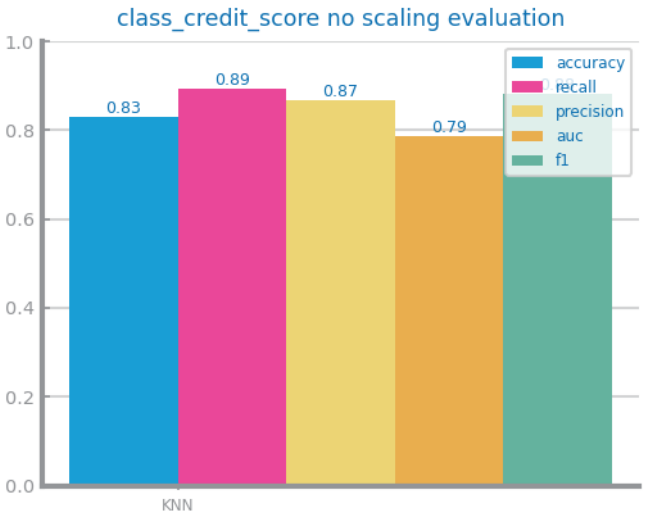


Figure 23 Scaling results with different approaches for dataset 1





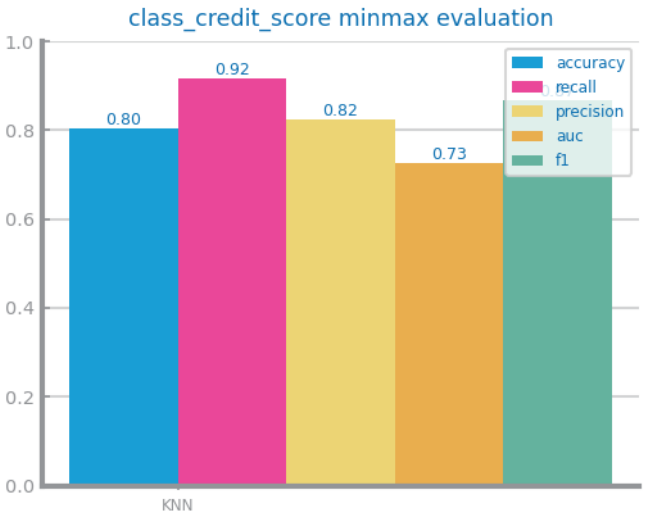


Figure 24 Scaling results with different approaches for dataset 2

Balancing

D1 has ~29% “Yes” entries and D2 has ~71% “Good” entries, making both unbalanced.

Undersampling was promising in theory, but it would lead to information loss, since the minority classes are too small. Even though oversampling performed well in both D1 and D2, SMOTE creates synthetic samples of Positive *CovidPos* and Bad *Credit_Score*, which are useful for future studying.

Because of that and better performance, we chose SMOTE for D1 and D2. D1 had a notable increase in performance with either method.

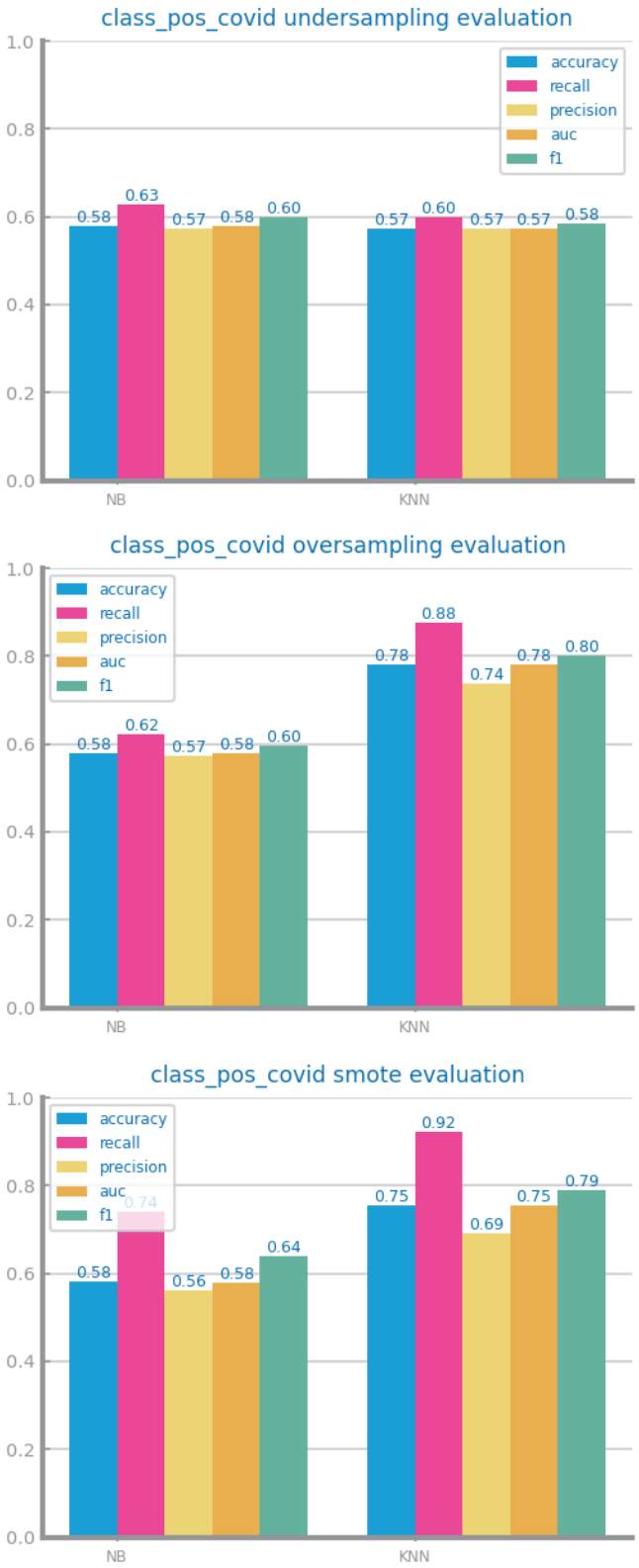


Figure 25 Balancing results with different approaches for dataset 1

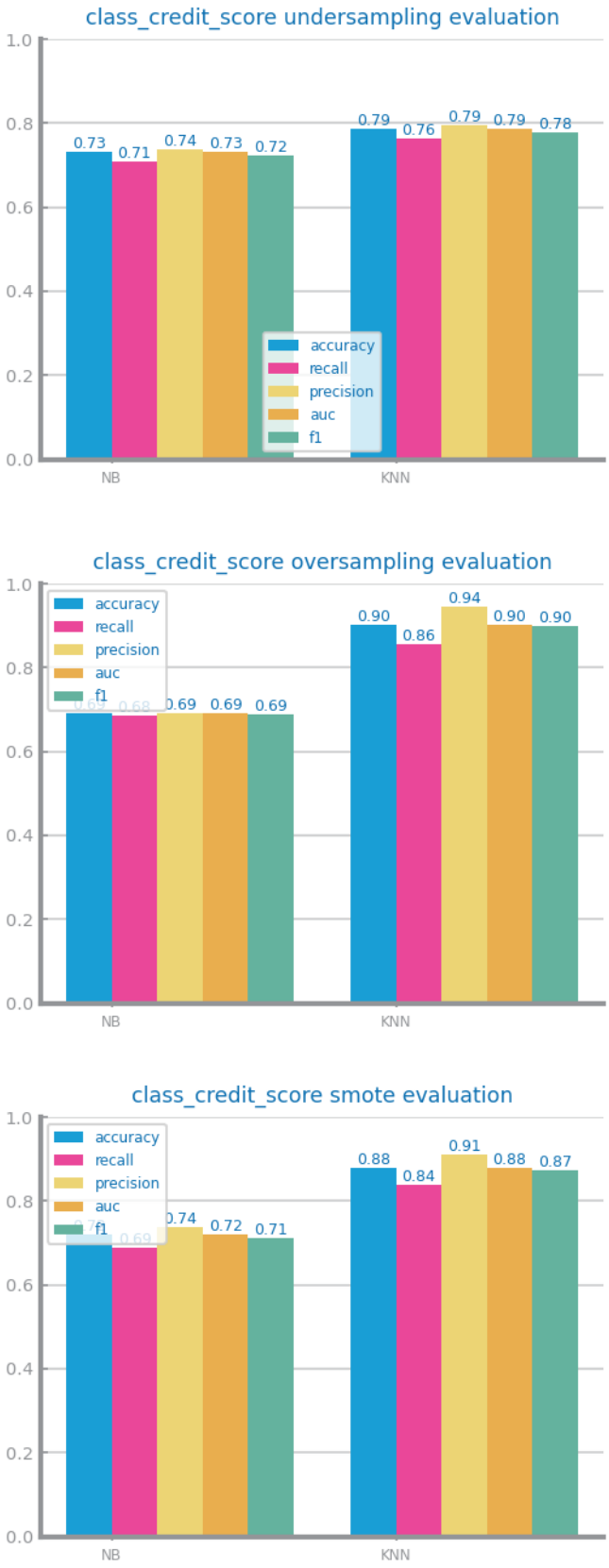


Figure 26 Balancing results with different approaches for dataset 2

Feature Selection

For redundancy, we used a threshold of 1. In D1, we have the best results in NB when discarding vars with correlation higher than 0.4. For D2 we have the best results in NB when discarding vars with a correlation lower than 0.4.

For variance, we used a threshold of 0.25. In D1 performance is better when not removing vars. For D2 performance is better when dropping vars with variance lower than 0.5.

We dropped the vars *PhysicalHealthDays* (redudant) and *HeightInMeters* (lowvar) in D1 and none in D2.

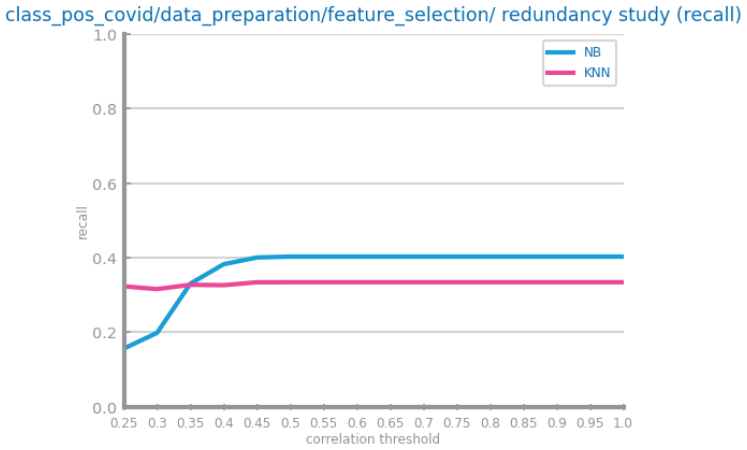


Figure 27 Feature selection of redundant variables results with different parameters for dataset 1

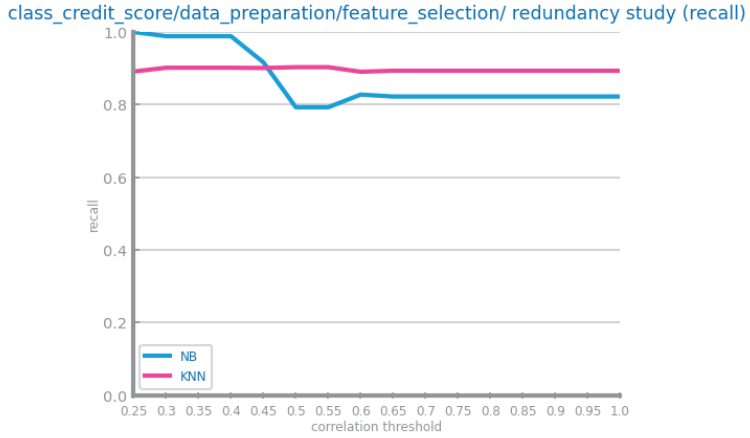


Figure 28 Feature selection of redundant variables results with different parameters for dataset 2

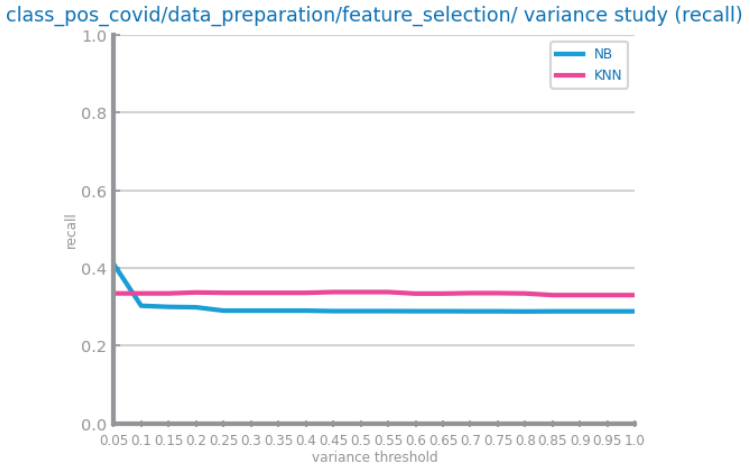


Figure 29 Feature selection of relevant variables results with different parameters for dataset 1 (variance study)

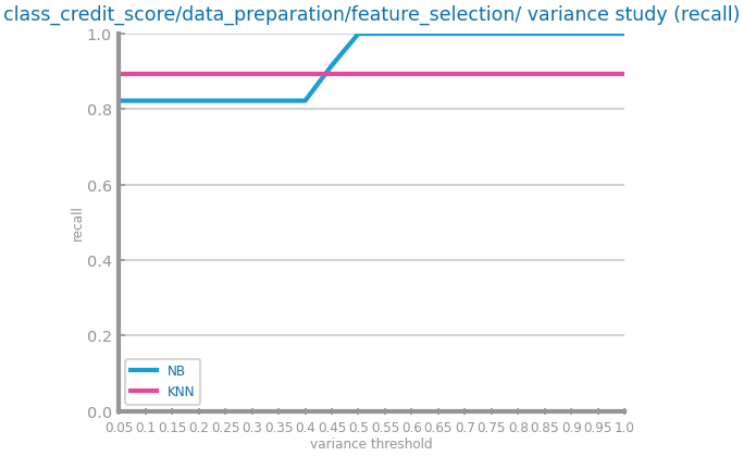


Figure 30 Feature selection of relevant variables results with different parameters for dataset 2 (variance study)

Feature Extraction

No scaling was applied in either dataset; therefore, no Feature Extraction was performed.

Additional Feature Generation

Performed on D2 in respect of *Type_of_Loan*, as part of variable encoding, as mentioned above. Positive impact on modelling results due to information being more accurately represented.

Type_of_Loan	Auto Loan	Credit-Builder Loan	Personal Loan	Home Equity Loan	Not Specified	Mortgage Loan	Student Loan	Debt Consolidation Loan	Payday Loan
Auto Loan, Credit-Builder Loan, Personal Loan, and Home Equity Loan	1	1	1	1	0	0	0	0	0
Auto Loan, Auto Loan, and Not Specified	2	0	0	0	1	0	0	0	0
Credit-Builder Loan, and Mortgage Loan	0	1	0	0	0	1	0	0	0
Personal Loan, Payday Loan, Student Loan, Auto Loan, Home Equity Loan, Student Loan, and Payday Loan	1	0	1	1	0	0	2	0	2

Table 10 - Examples of original values for Type_of_Loan and their result after Feature Generation

3 MODELS' EVALUATION

The datasets used to evaluate the models resulted from data preparation. A hold-out strategy was used, by splitting data into training and testing sets (67/33) always with the same seed.

In D1, we used recall since in a medical context it's very important that positive cases (they are the minority class as well) don't go undetected - recall reinforces that.

For D2, we used accuracy since the train set was balanced and it provides a thorough measure of overall model performance.

Naïve Bayes

In D1, Gaussian having the best recall might be due to the influence of the continuous variables that approximate a normal distribution being higher than its binary ones.

In D2, Bernoulli having the best accuracy might be due the new type of loan features created that resemble binary variables.

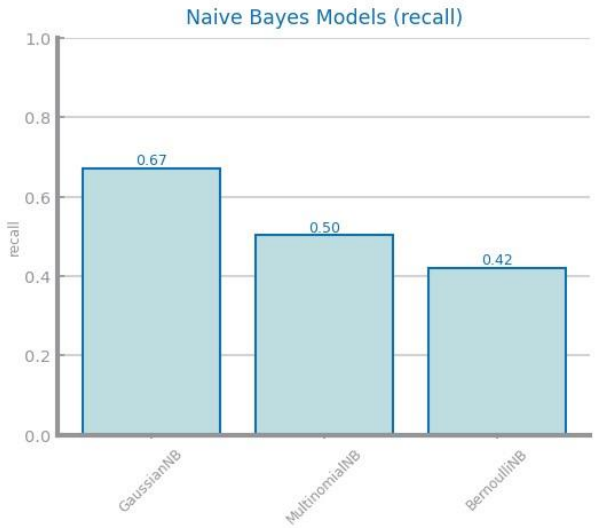


Figure 31 Naïve Bayes alternatives comparison for dataset 1

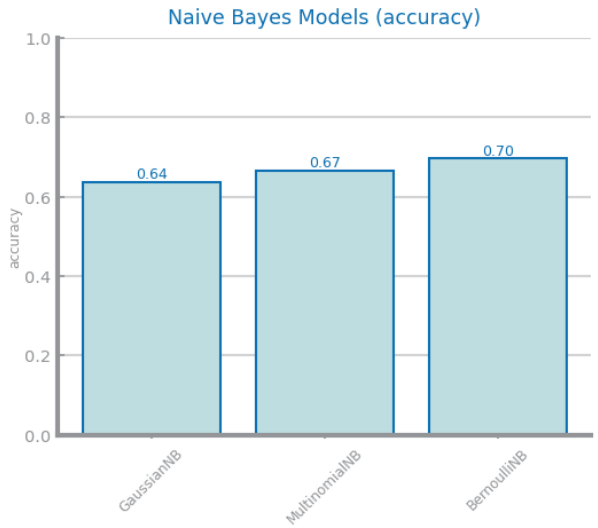
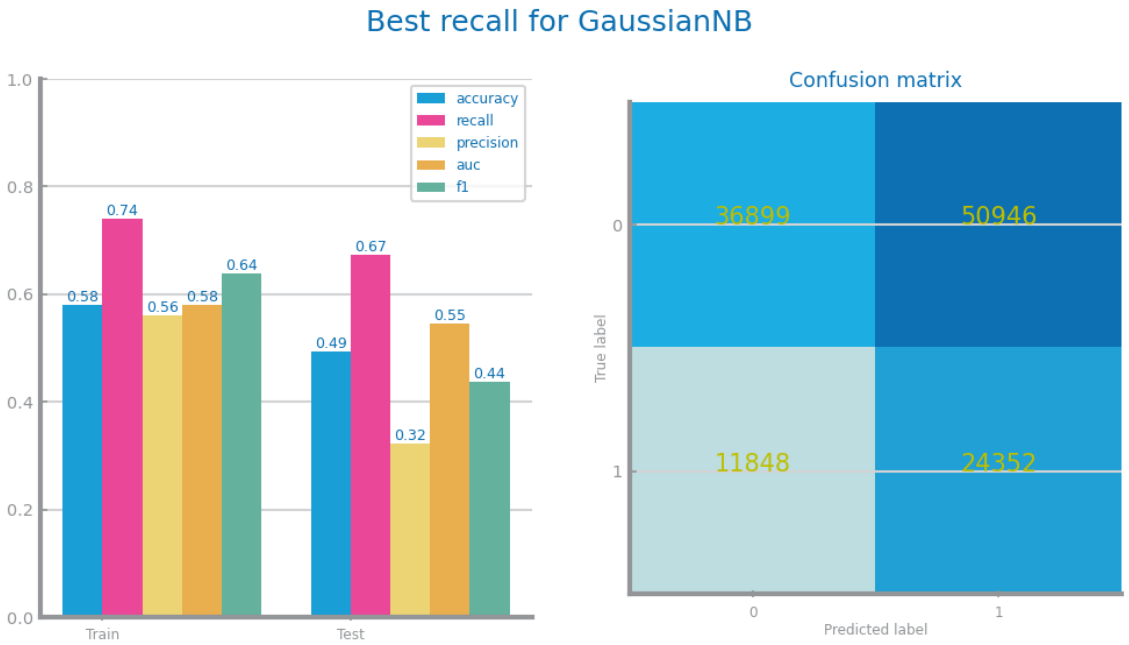


Figure 32 Naïve Bayes alternative comparison for dataset 2



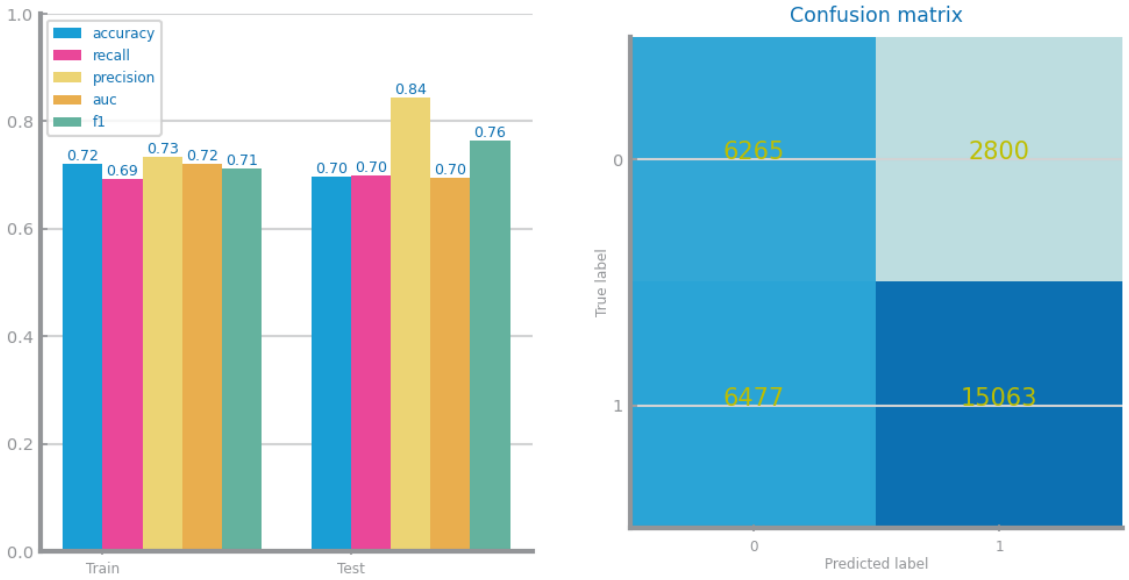


Figure 33 Naïve Bayes best model results for dataset 1 (above) and dataset 2 (below)

KNN

In D1, Euclidean had the best recall for any K tested (K=25 being the best), this is because the D1 has a lot of features measured in the same units and with similar importance. There is no overfitting since both curves keep the same trend.

In D2, the accuracies of the measures were similar, but Manhattan was the best with K=3, edging out due to its effectiveness when dealing with features of different scales in D2. It overfits for lower values of k, where train rises and test drops (from R-L).

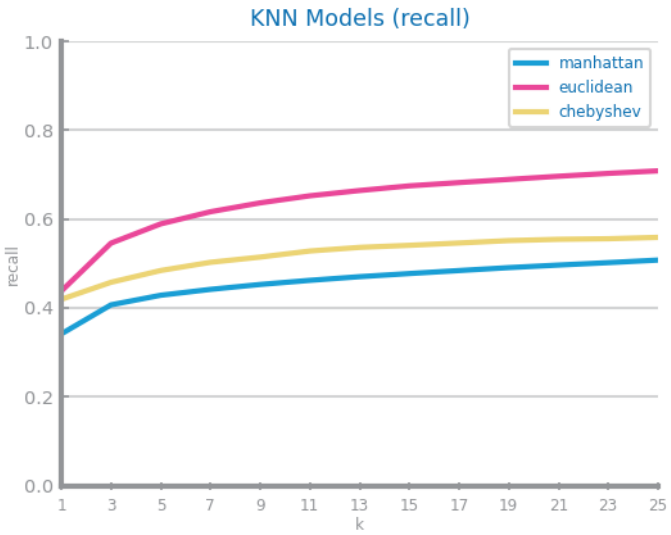


Figure 34 KNN different parameterizations comparison for dataset 1

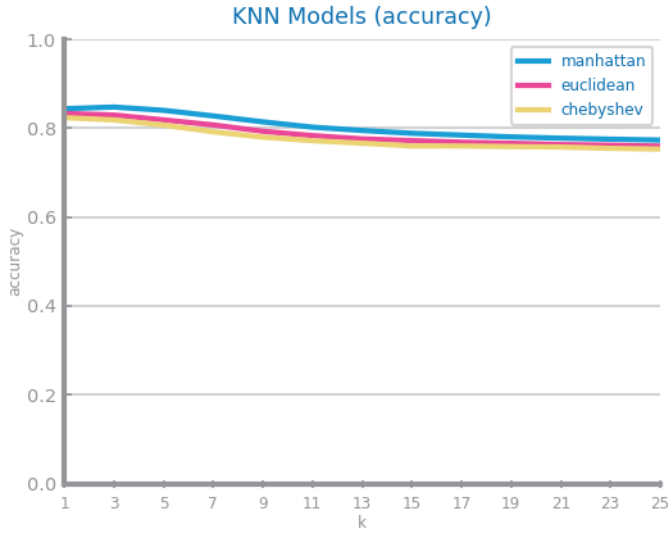


Figure 35 KNN different parameterizations comparison for dataset 2

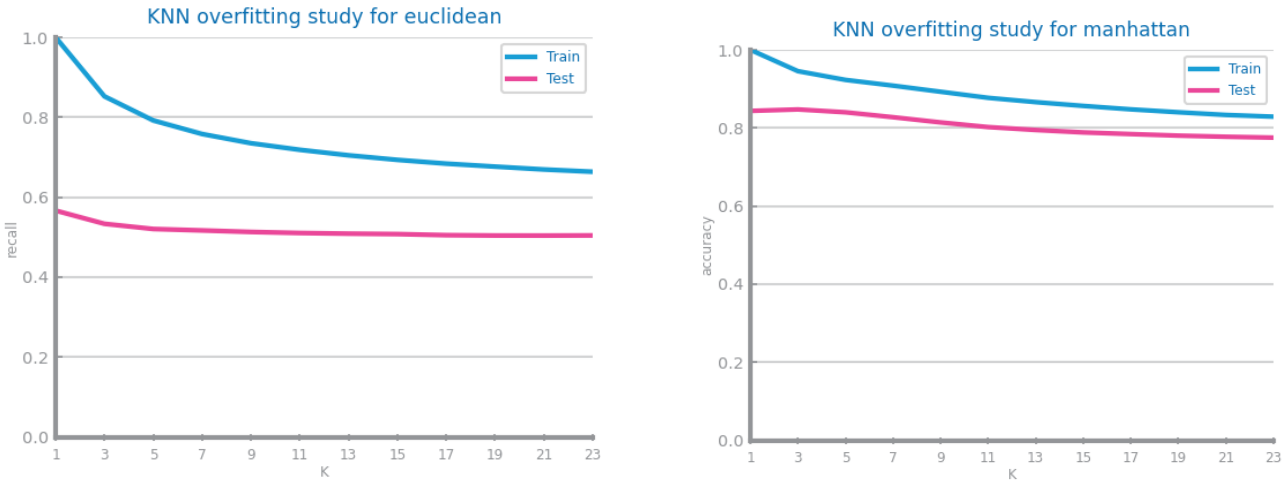
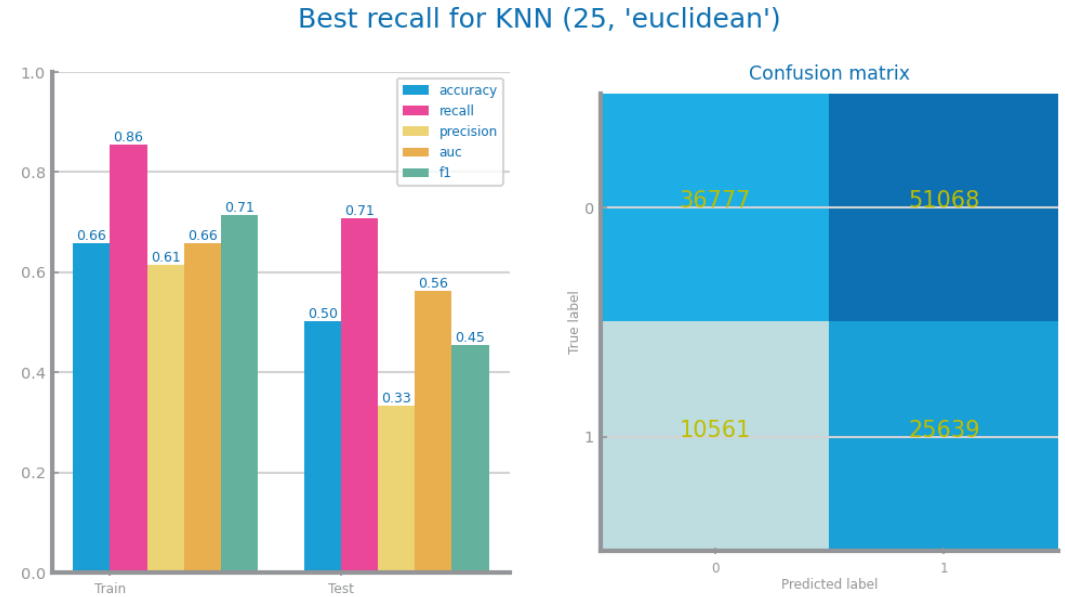


Figure 36 KNN overfitting analysis for dataset 1 (left) and dataset 2 (right)



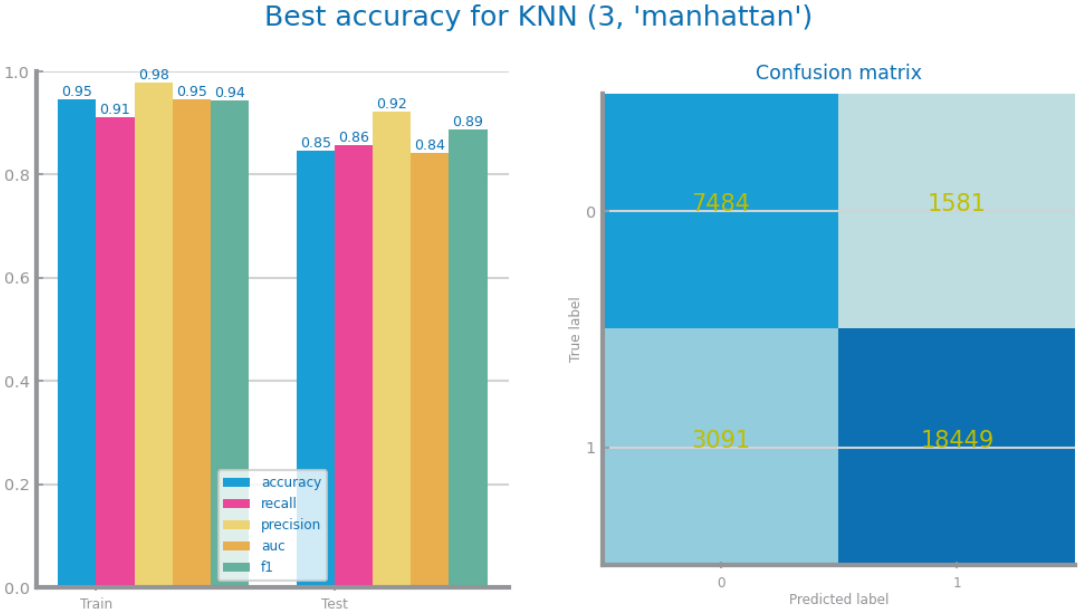


Figure 37 KNN best model results for dataset 1 (above) and dataset 2 (below)

Decision Trees

For both sets, we achieved the best results with *entropy* rather than *gini*. In D1, the best tree is with d=2 due to its higher recall while in D2 it is with d=8 due to its higher accuracy.

For the best tree, the most important variables in D1 are *AgeCategory* and *ChestScan*. In D2, they are *OutstandingDebt*, *CreditMix*, *Interest_Rate*, *Month*.

In D1, there is no overfitting while in D2 there is, as with the increasing depth (from d=15), the train curve increases while the test one is decreasing.

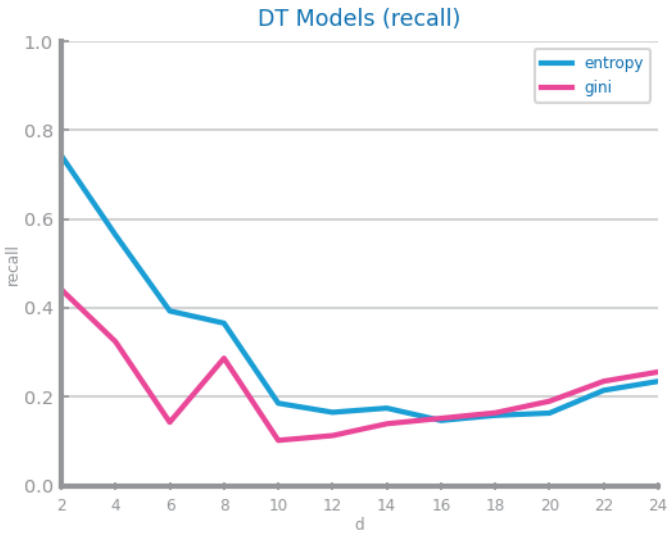


Figure 38 Decision Trees different parameterizations comparison for dataset 1

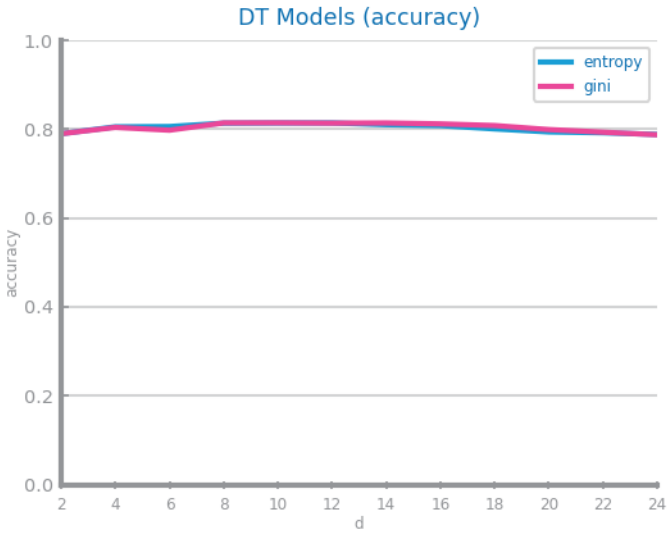


Figure 39 Decision Trees different parameterizations comparison for dataset 2

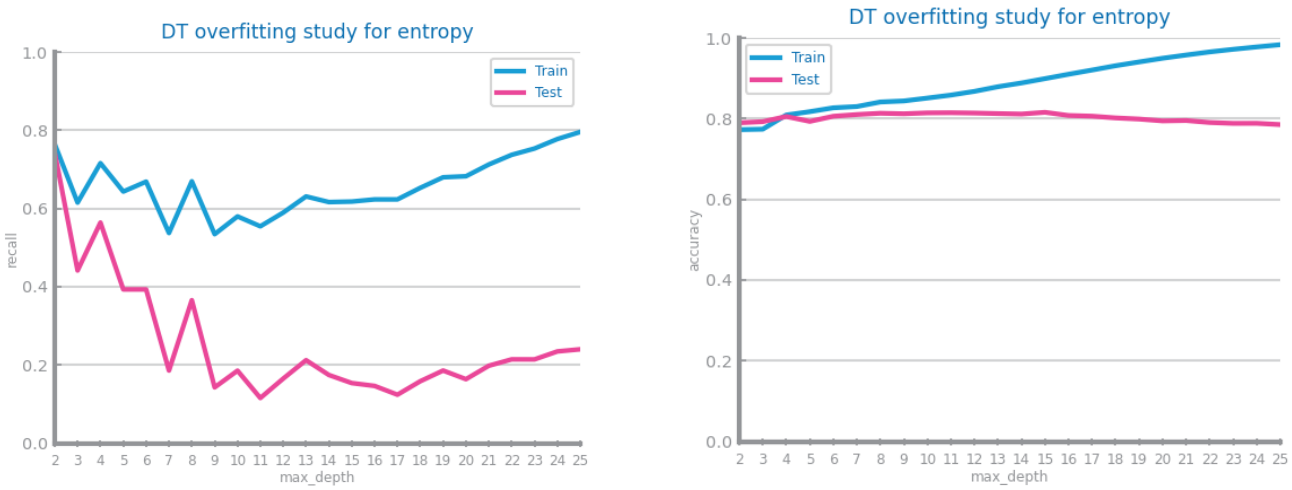
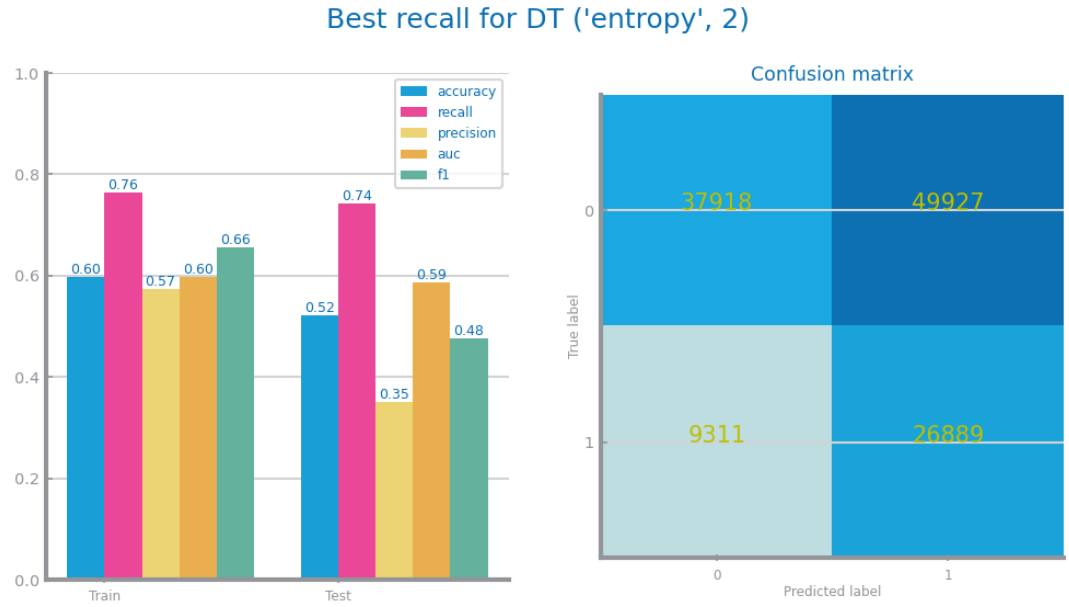


Figure 40 Decision Trees overfitting analysis for dataset 1 (left) and dataset 2 (right)



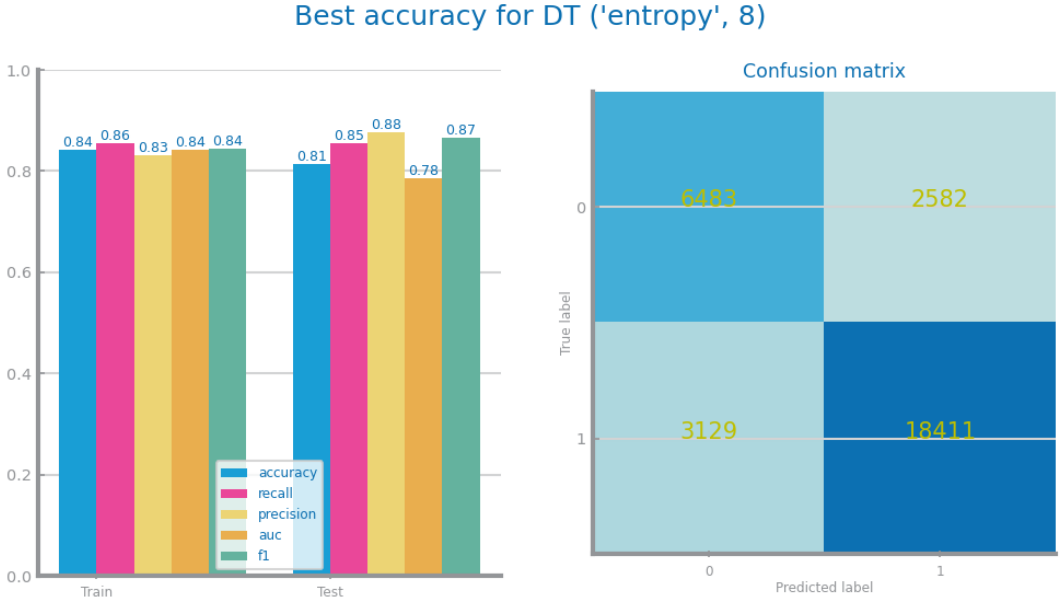


Figure 41 Decision trees best model results for dataset 1 (above) and dataset 2 (below)

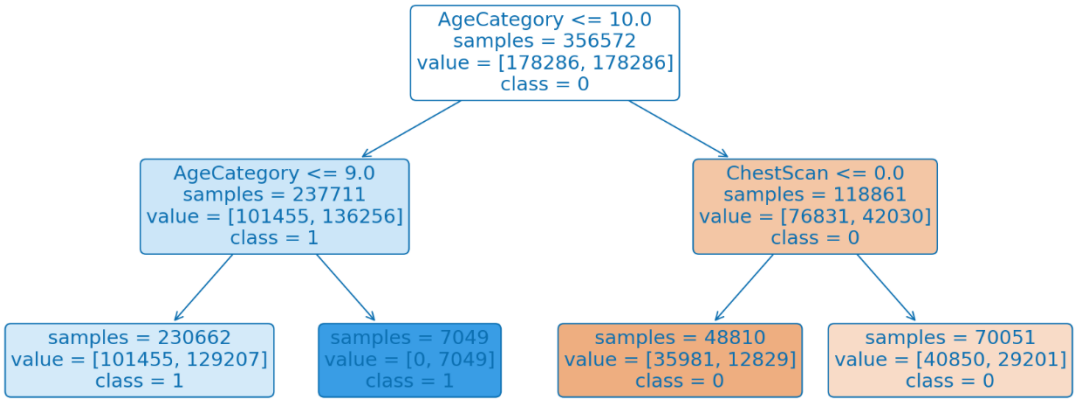


Figure 42 Best tree for dataset 1

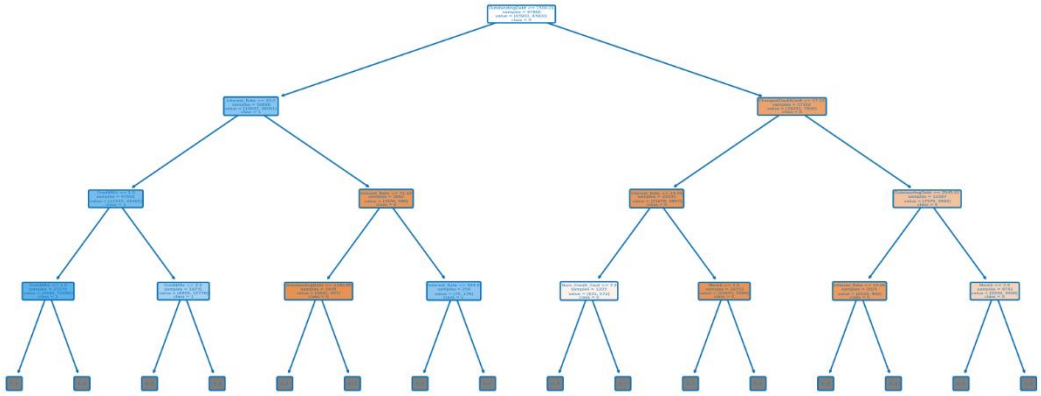


Figure 43 Best trees for dataset 2

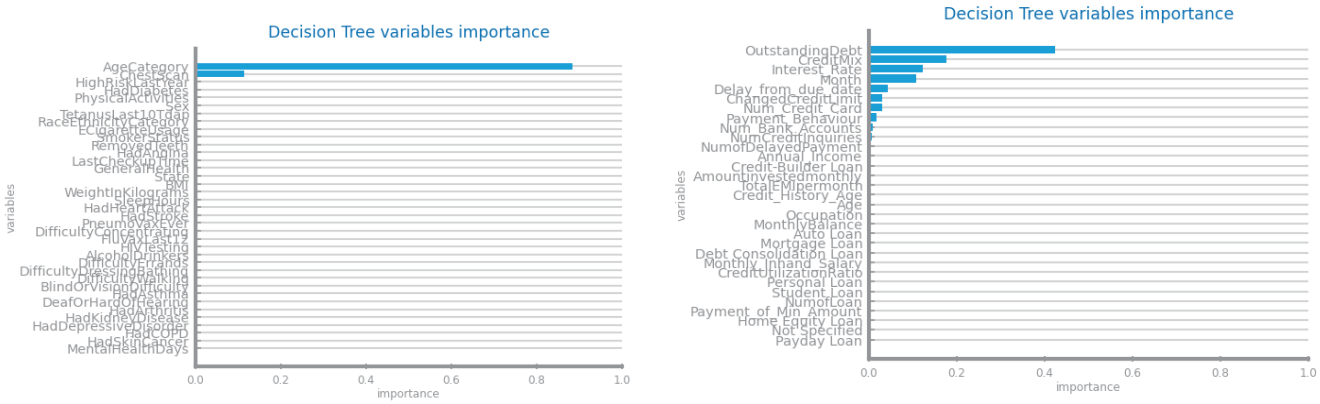


Figure 44 Decision trees variables importance for dataset 1 (left) and dataset 2 (right)

Random Forests

In D1 the higher the depth and the number of features used, the worse the results. In D2 the accuracies were similar for any of the parameters.

For both sets, the best model had 50% feature usage, 100 estimators but D1 with d=2 and D2 with d=7.

The most important variables from D1 are *AgeCategory*, *FluVaxLast12*, *HIVTesting*. In D2, they are *OutstandingDebt*, *Interest_Rate*, *CreditMix*.

There is no overfitting, which is not surprising since combining different models allows for better generalization.

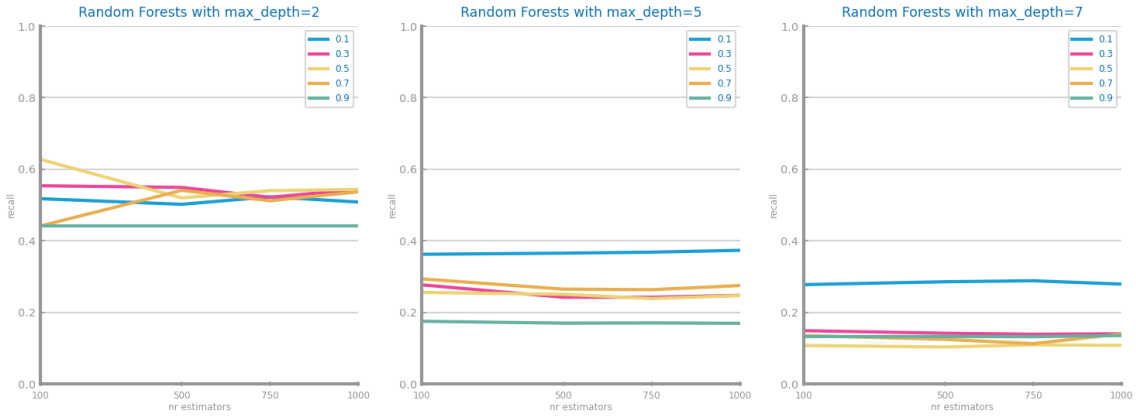


Figure 45 Random Forests different parameterizations comparison for dataset 1

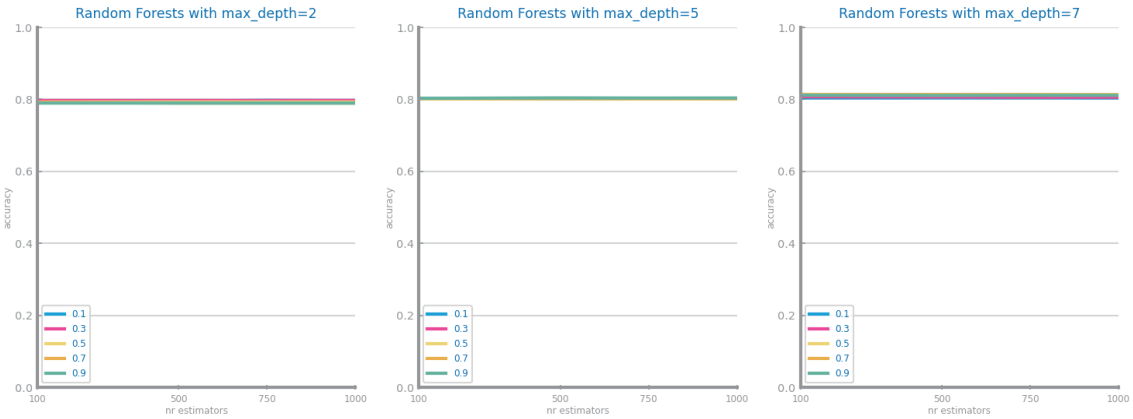


Figure 46 Random Forests different parameterizations comparison for dataset 2

Figure 47 Random Forests overfitting analysis for dataset 1 (left) and dataset 2 (right)

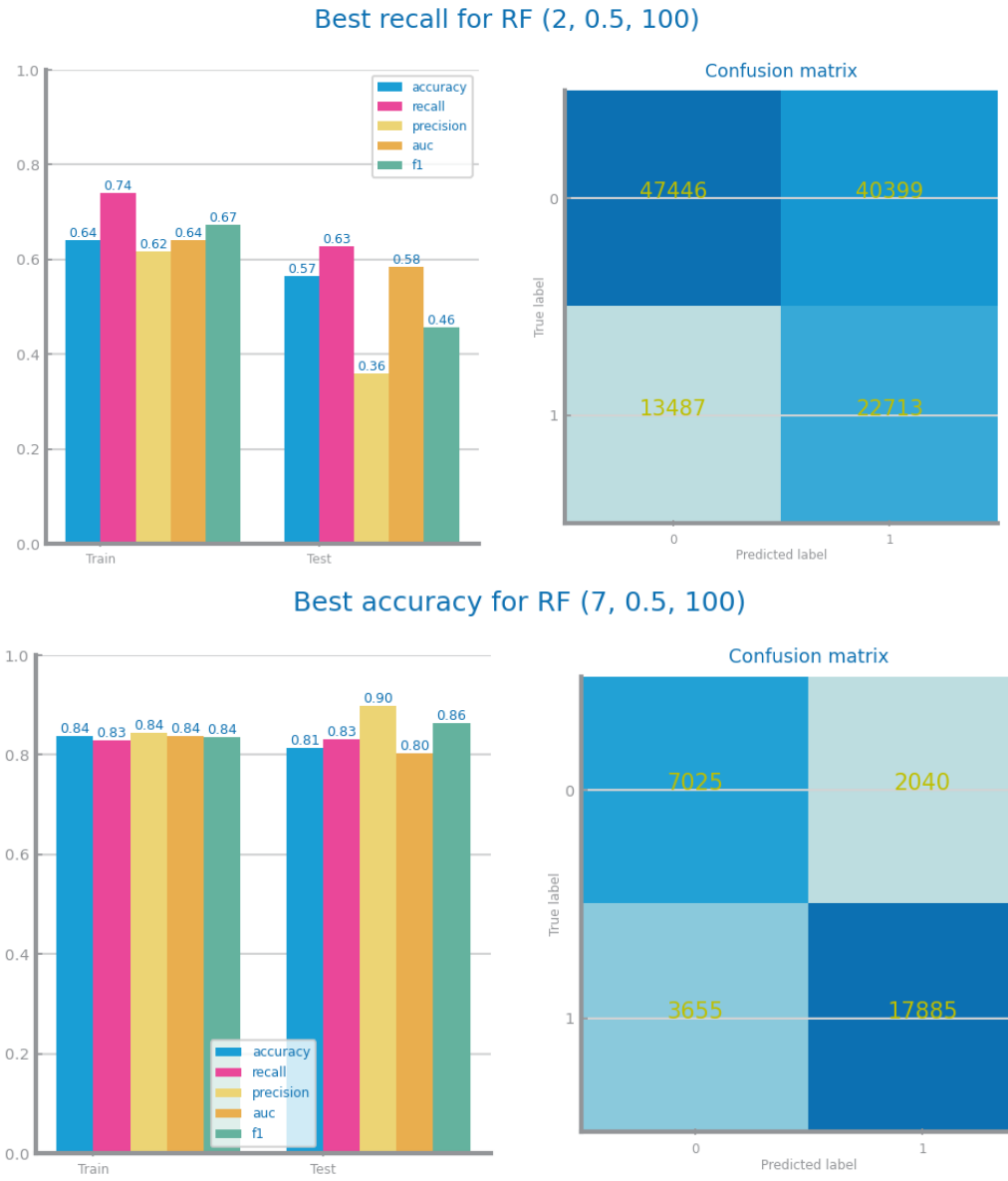


Figure 48 Random Forests best model results for dataset 1 (above) and dataset 2 (below)

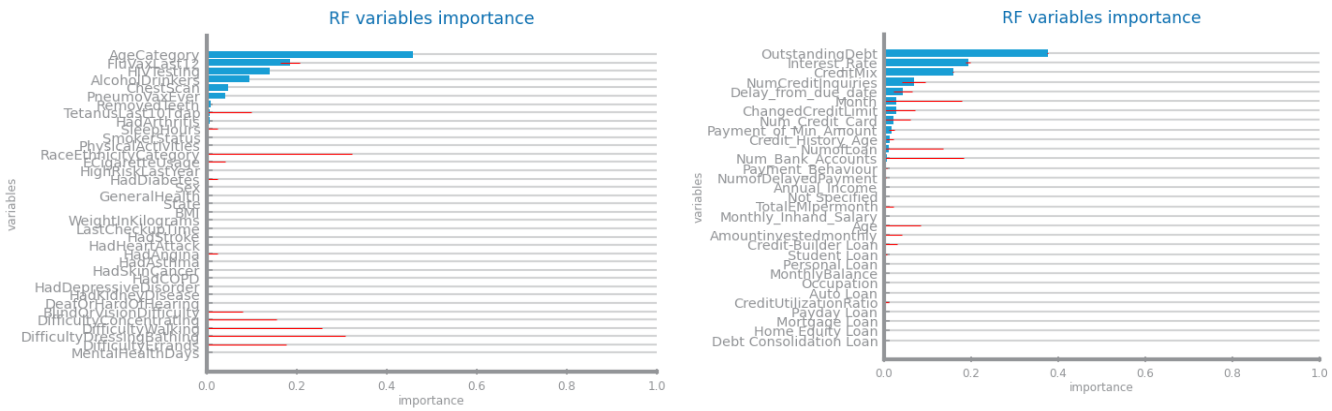


Figure 49 Random Forests variables importance for dataset 1 (left) and dataset 2 (right)

Gradient Boosting

In D1, the recall results are low for any params, from around 1-30%. Best params are d=7, 0.9 lr and 1000 estimators.

In D2, the accuracy slightly increases with increasing depth and estimators but generally the results are similar with any params. Best params are d=7, 0.3 lr and 1000 estimators.

Most important vars in D1 are *AgeCategory*, *AlcoholDrinkers*, *ChestScan*. In D2, *OutstandingDebt*, *CreditMix*, *Interest_Rate*.

No overfitting in D1 and D2 as the train/test curves keep the same trend.

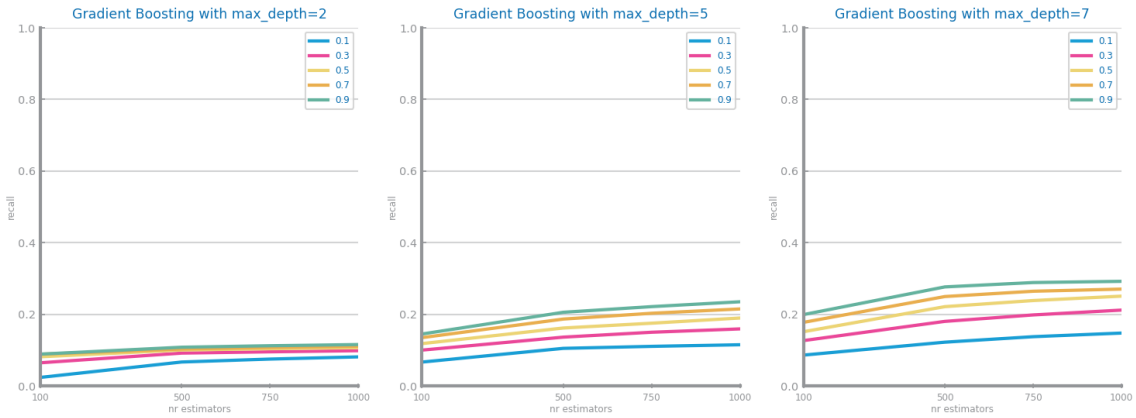


Figure 50 Gradient boosting different parameterizations comparison for dataset 1

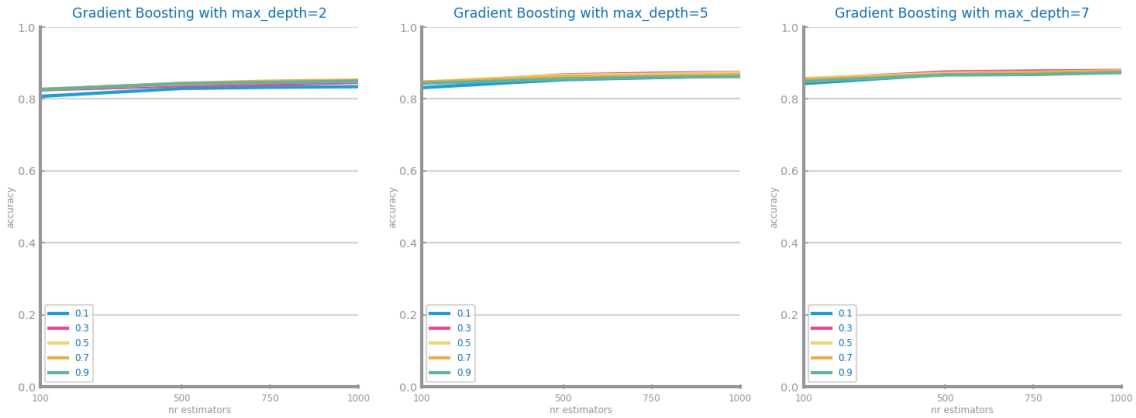


Figure 51 Gradient boosting different parameterizations comparison for dataset 2

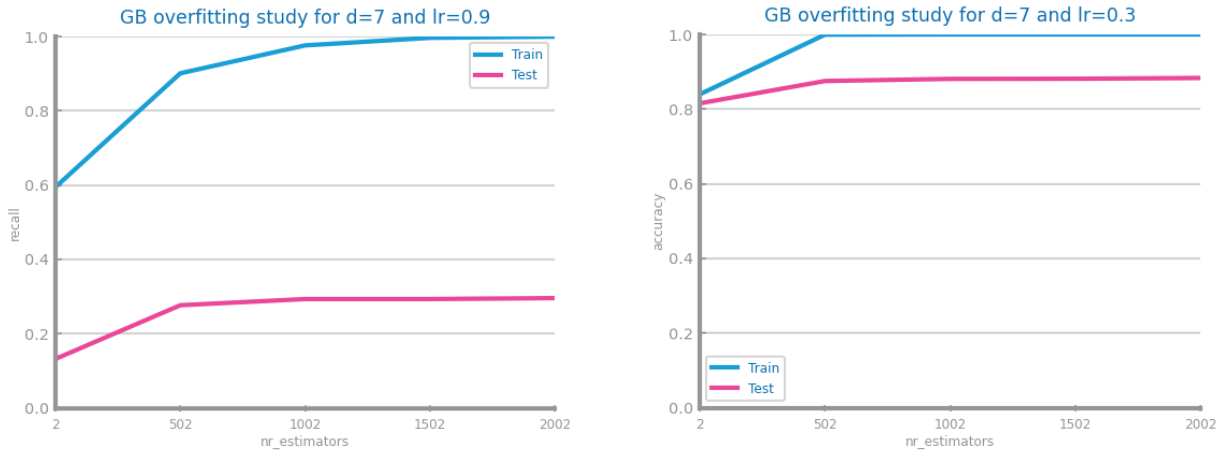


Figure 52 Gradient boosting overfitting analysis for dataset 1 (left) and dataset 2 (right)

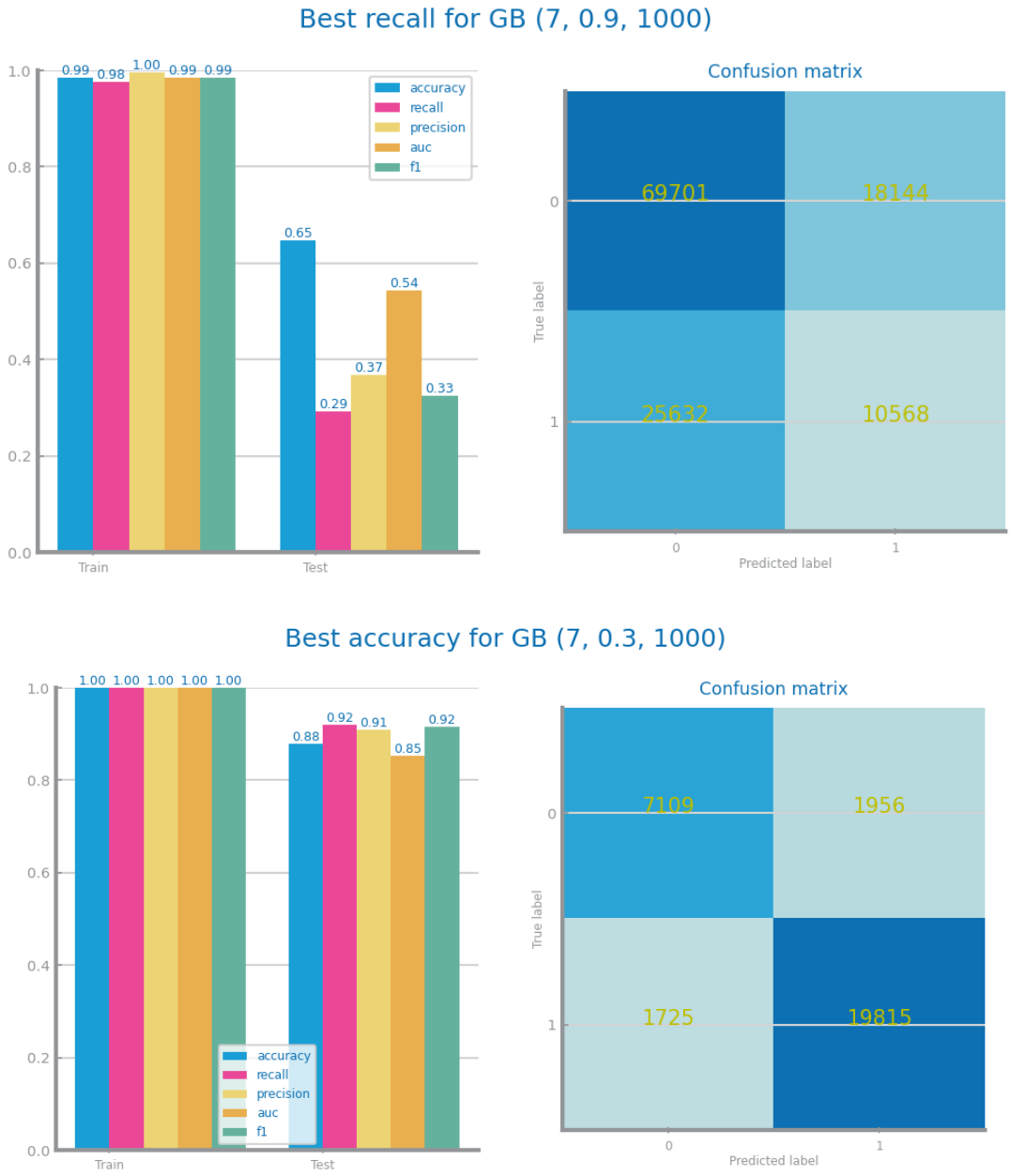


Figure 53 Gradient boosting best model results for dataset 1 (above) and dataset 2 (below)

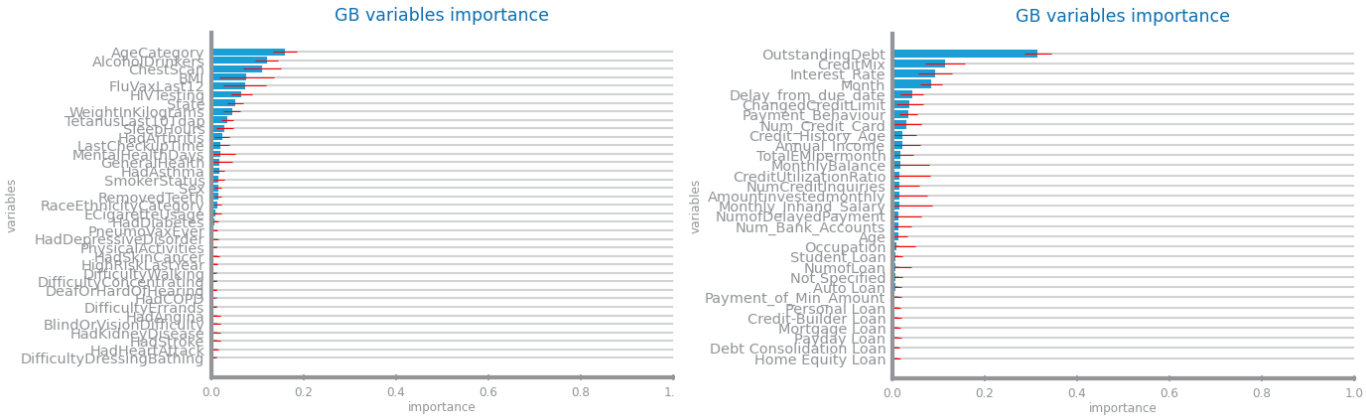


Figure 54 Gradient boosting variables importance for dataset 1 (left) and dataset 2 (right)

Multi-Layer Perceptrons

In D1 initially recall was used but as seen in Fig. 61, everything is positive, so it serves no purpose. Instead, we used F1. The best model is 0.005 LR, 1250 iterations and *constant* update.

In D2 the best model is 0.0005 LR, 500 iterations and *invscaling*.

In general, the higher the LR the worse it performs (high LR can overshoot optimal values).

No overfitting in D1 since train/test keep the same trend and D2 has it after 500th iteration, as train is constant while test increases, then decreases.

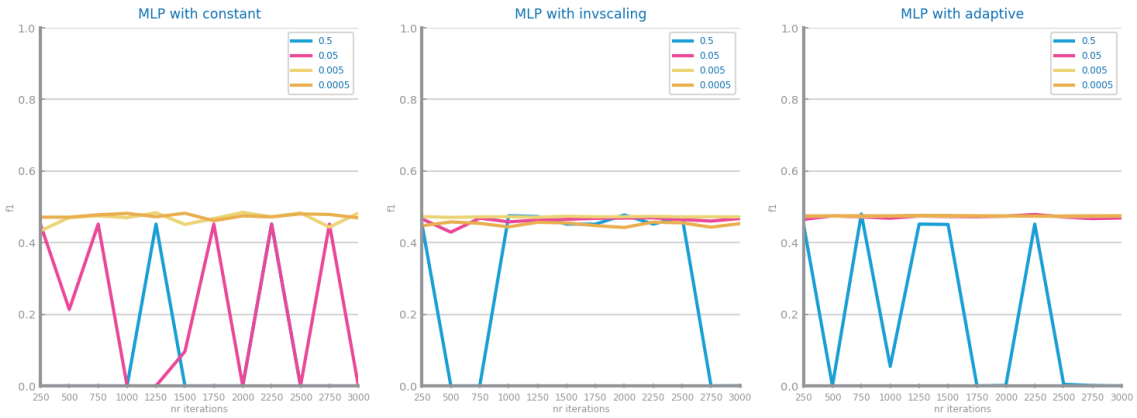


Figure 55 MLP different parameterizations comparison for dataset 1

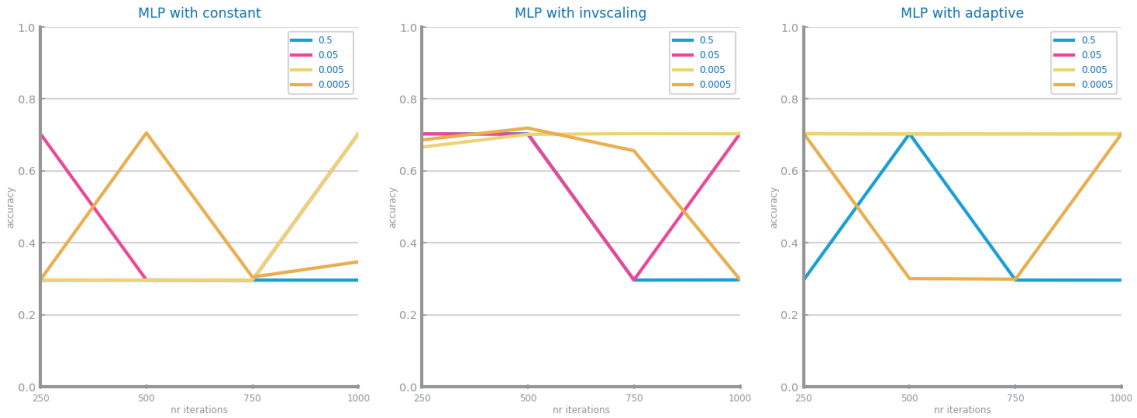
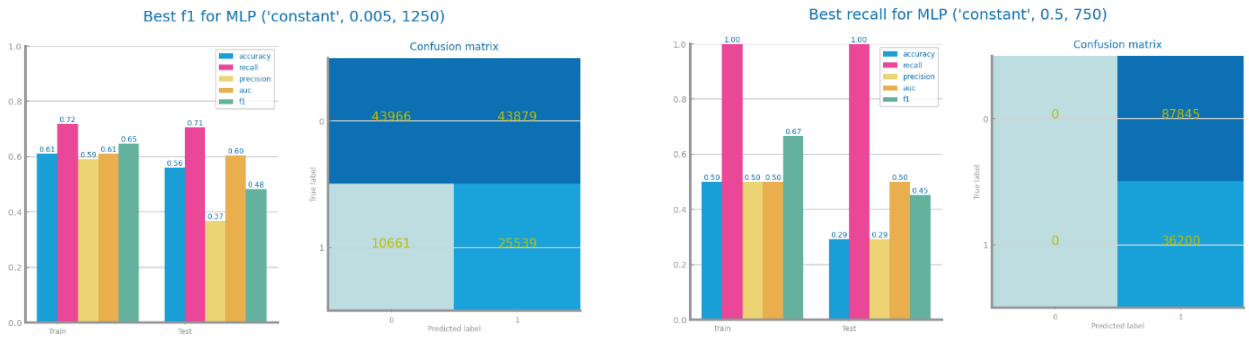


Figure 56 MLP different parameterizations comparison for dataset 2

Figure 57 MLP overfitting analysis for dataset 1 (left) and dataset 2 (right)



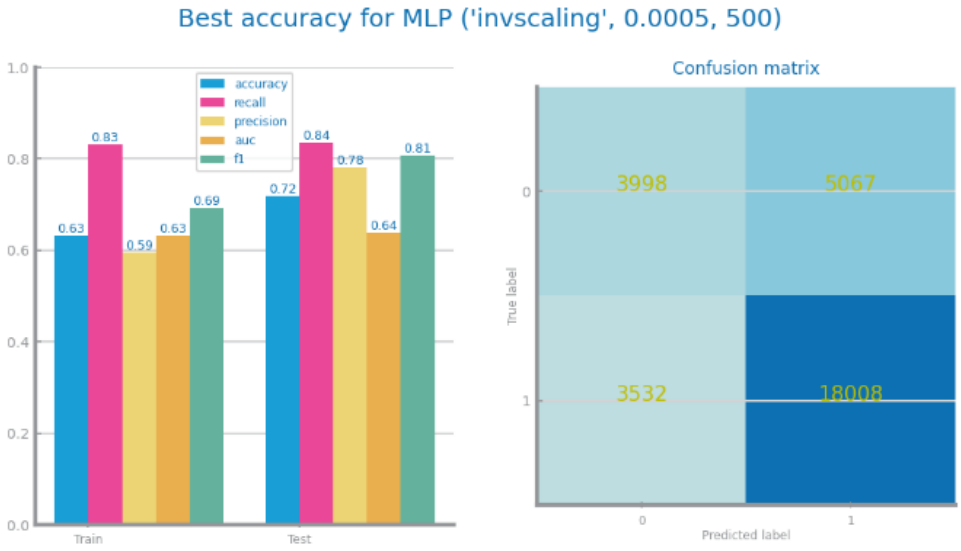


Figure 58 MLP best model results for dataset 1 (above) and dataset 2 (below)

4 CRITICAL ANALYSIS

Invalid entries that impacted histograms and created false outliers were removed. Neg values were altered into MV, but some vars could've benefited from being set to 0 instead of MVI with mean.

We dropped all unique identifiers to benefit our analysis but, in D2 the same person can have many entries, so keeping a unique identifier could've provided insights on how personal evolution impacts Credit Score. For example, it could help assess if score oscillations have different thresholds for different people. It could also make MVI more accurate, since there are cases like "Annk", that has 30 years in almost every entry and 7580 in one.

Scaling wasn't applied in any dataset, benefiting **NB** (conditionally independent features) and **Tree-Based Models** (decisions based on splits in the data). Data preparation methods improved both datasets.

In D1, most common values for vars won't change with different targets, which can weaken its diagnostic capacity. More relevant symptoms like loss of smell or taste could be added. *RaceEthnicityCategory* and *State* are statistically relevant (ex: Non-White people having less Pos can be related to healthcare inequity access) but aren't so relevant in prediction. Issues linked to human error in diagnosing and its consequences (FN/FP) are also recognized, prompting focus on **Recall** in D1 to favor predicting positives, at the expense of some FP. In D2 **Accuracy** was selected as it provides a thorough measure of overall model performance.

DT was the best model in D1, with a Recall of **0.74**, and it's in concordance with the scaling choice stated above. In **MLP**, F1 was chosen as metric to ensure a balanced evaluation, since it classified everything as Pos when using Recall (making its value the max).

In D2, **GB** was the best, with an Accuracy of **0.88.** The models performed well overall though overfit occurred in some. D1 and D2's most impactful vars are *AgeCategory* and *OutstandingDebt* overall. D2's best model has good test accuracy, however D1 could be better.

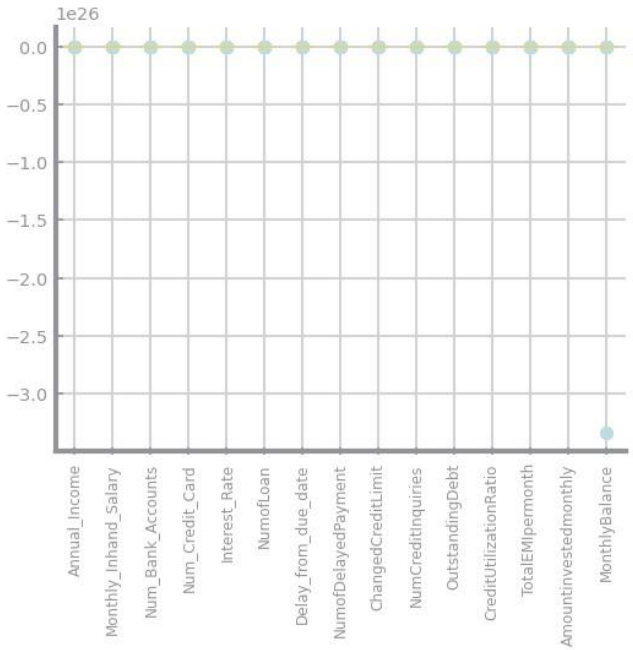


Figure 59 - Global Boxplots of dataset 2 before removing an extreme outlier in Monthly Balance

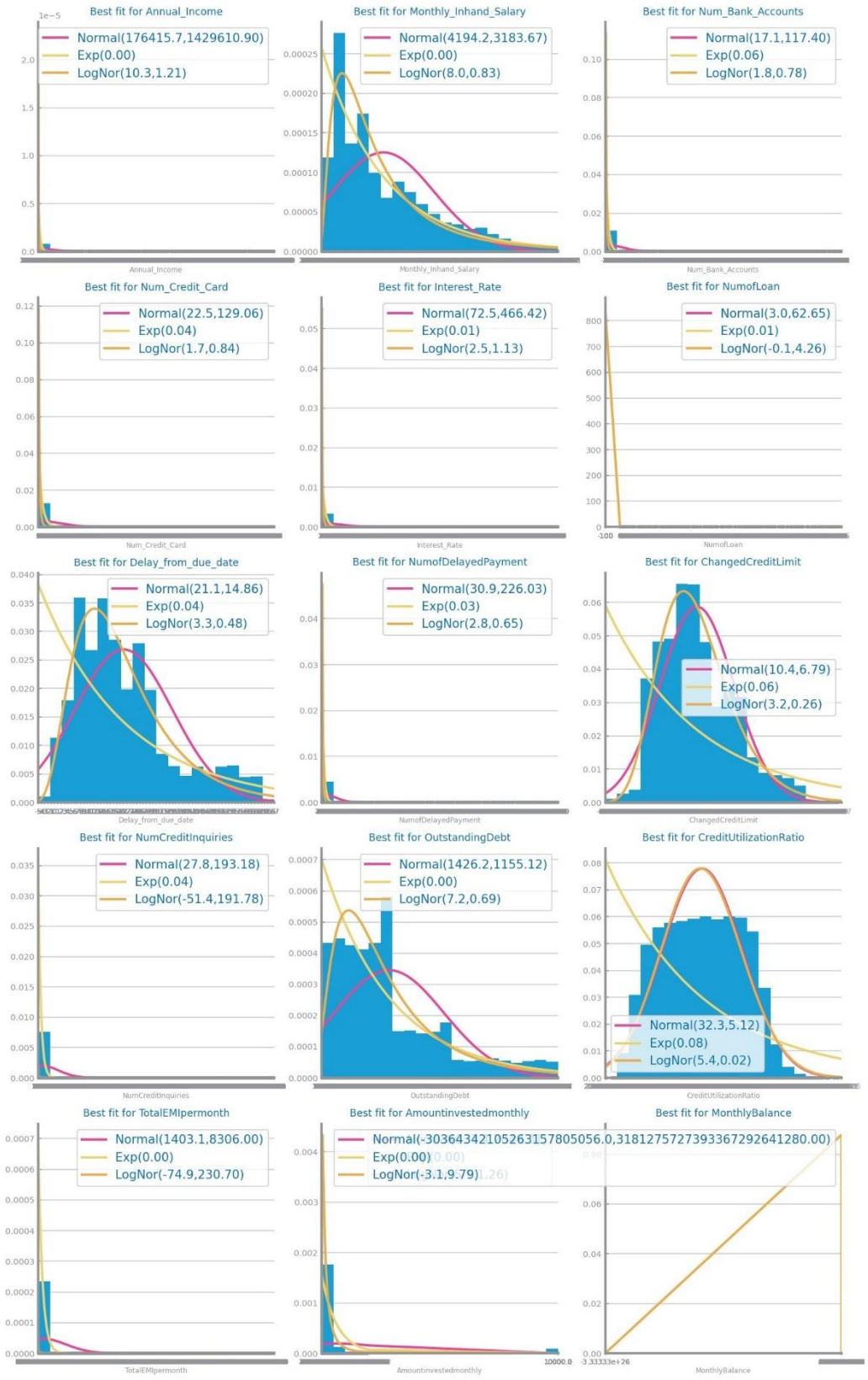


Figure 60 – Histograms of dataset 2 before removing an extreme outlier in Monthly Balance

Variable name	**CovidPos = 1 - Yes**	**CovidPos = 0 - No**
MentalHealthDays	0.000	0.000
SleepHours	7.000	7.000
WeightInKilograms	81.650	81.190
BMI	27.500	27.440
State	71.729 - Kansas	71.729 - Kansas
GeneralHealth	3.000 - Good	3.000 - Good
LastCheckupTime	4.000 - Within past year (anytime less than 12 months ago)	4.000 - Within past year (anytime less than 12 months ago)
RemovedTeeth	4.000 - None of them	4.000 - None of them
HadDiabetes	4.000 - No	4.000 - No
SmokerStatus	4.000 – Never smoked	4.000 – Never smoked
ECigaretteUsage	4.000 – Never used e-cigarettes in my entire life	4.000 – Never used e-cigarettes in my entire life
RaceEthnicityCategory	1.000 - White only, Non-Hispanic	1.000 - White only, Non-Hispanic
AgeCategory	7 - Age 50 to 54	9.000 - Age 60 to 64
TetanusLast10Tdap	1.000 - Yes, received tetanus shot but not sure what type	1.000 - Yes, received tetanus shot but not sure what type
Sex	0.000 - Female	0.000 - Female
PhysicalActivities	1.000 - Yes	1.000 - Yes
HadHeartAttack	0.000 - No	0.000 - No
HadAngina	0.000 - No	0.000 - No
HadStroke	0.000 - No	0.000 - No
HadAsthma	0.000 - No	0.000 - No
HadSkinCancer	0.000 - No	0.000 - No
HadCOPD	0.000 - No	0.000 - No
HadDepressiveDisorder	0.000 - No	0.000 - No
HadKidneyDisease	0.000 - No	0.000 - No
HadArthritis	0.000 - No	0.000 - No
DeafOrHardOfHearing	0.000 - No	0.000 - No
BlindOrVisionDifficulty	0.000 - No	0.000 - No
DifficultyConcentrating	0.000 - No	0.000 - No
DifficultyWalking	0.000 - No	0.000 - No
DifficultyDressingBathing	0.000 - No	0.000 - No
DifficultyErrands	0.000 - No	0.000 - No
ChestScan	0.000 - No	0.000 - No
AlcoholDrinkers	1.000 - Yes	1.000 - Yes
HIVTesting	0.000 - No	0.000 - No
FluVaxLast12	0.000 - No	1.000 - Yes
PneumoVaxEver	0.000 - No	0.000 - No
HighRiskLastYear	0.000 - No	0.000 - No

Table 11 – Median values of dataset 1 (after feature selection, before balancing) for each variable depending on the target value

Variable name	**Credit_Score = 1 - Good**	**Credit_Score = 0 - Poor**
Age	33.000	32.000
Annual_Income	38985.960000	32179.720000
Monthly_Inhand_Salary	3076.995833	3076.995833
Num_Bank_Accounts	5	7
Num_Credit_Card	5	7
Interest_Rate	11	21
NumofLoan	3	5
Delay_from_due_date	15	27
NumofDelayedPayment	13	16
ChangedCreditLimit	9.58	9.61
NumCreditInquiries	4	8
OutstandingDebt	926.62	1941.73
CreditUtilizationRatio	32.404108	32.000423
TotalEMIpermonth	67.818216	74.733349
Amountinvestedmonthly	135.169234	123.898725
MonthlyBalance	344.051277	299.748275
Month	4 - April	5 - May
Occupation	8 - Accountant	8 - Accountant
CreditMix	1 - Standard	1 - Standard
Credit_History_Age	233 months – 19 years and 5 months	168 months = 14 years
Payment_of_Min_Amount	1 - Yes	1 - Yes
Payment_Behaviour	3 - Low_spent_Large_value_payments	2 - Low_spent_Medium_value_payments
Auto Loan	0	0
Credit-Builder Loan	0	0
Personal Loan	0	0
Home Equity Loan	0	0
Not Specified	0	0
Mortgage Loan	0	0
Student Loan	0	0
Debt Consolidation Loan	0	0
Payday Loan	0	0

Table 12 – Median values of dataset 2 (after feature selection, before balancing) for each variable depending on the target value

5 DATA PROFILING

Data Dimensionality and Granularity

In S1, the most atomic granularity was weekly, while in S2 was quarter-hourly. The other granularities chosen were monthly and quarterly (S1), and hourly and daily (S2). S1 has 199 records and S2 has 2976.

Higher granularities in S1 only smooth the curve, since the data is cumulative. In S2 the data is not cumulative, and we lose detail and get limited perspective in the daily granularity.

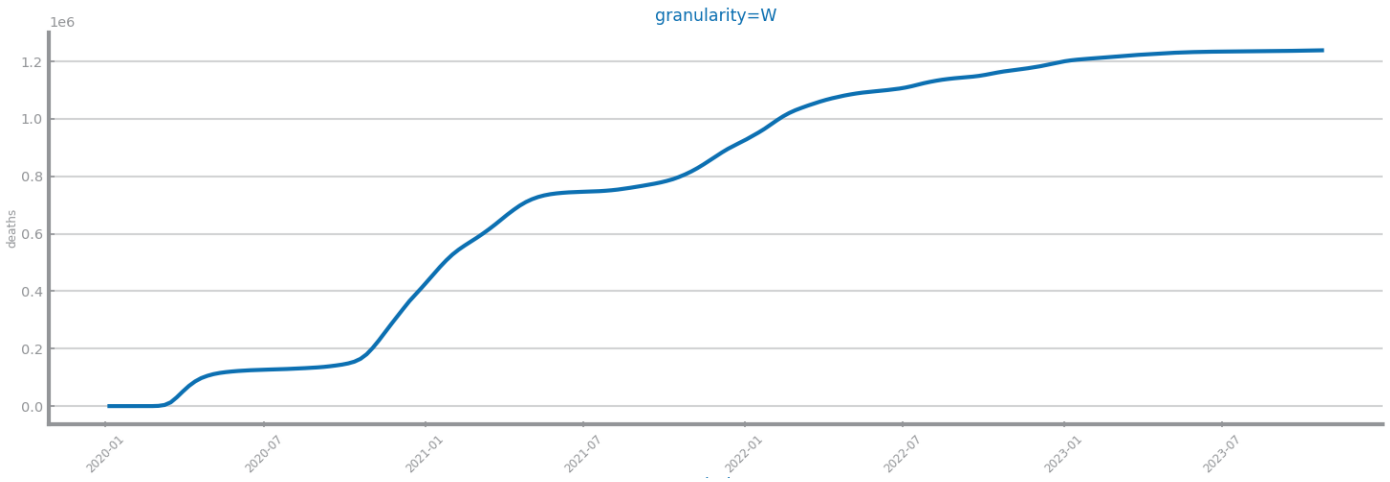


Figure 61 Original time series 1 (the most atomic detail)

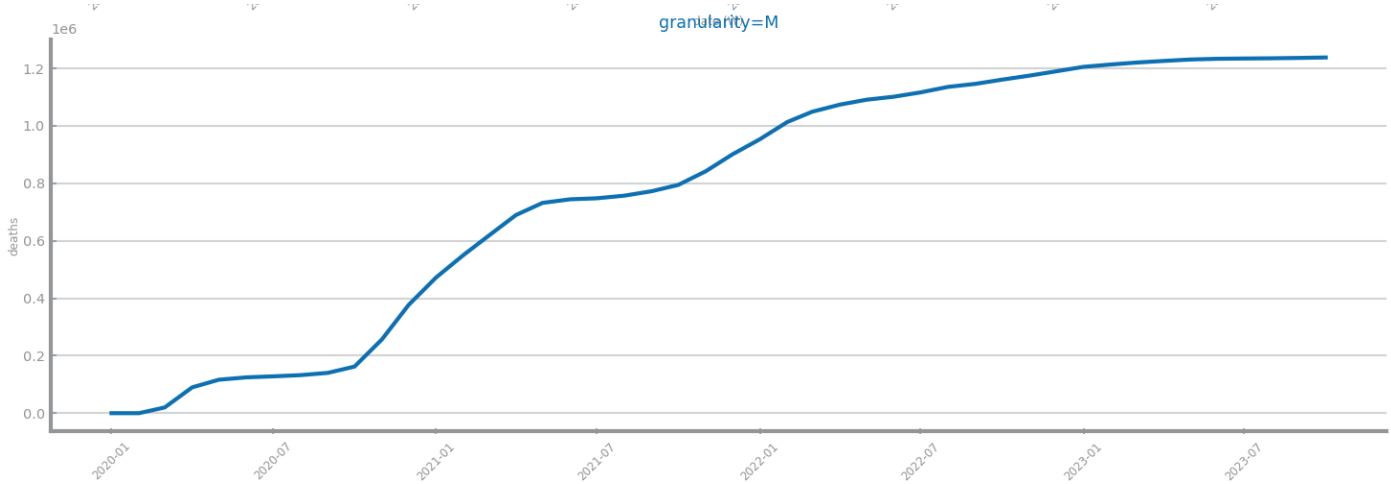


Figure 62 Time series 1 at the second chosen granularity

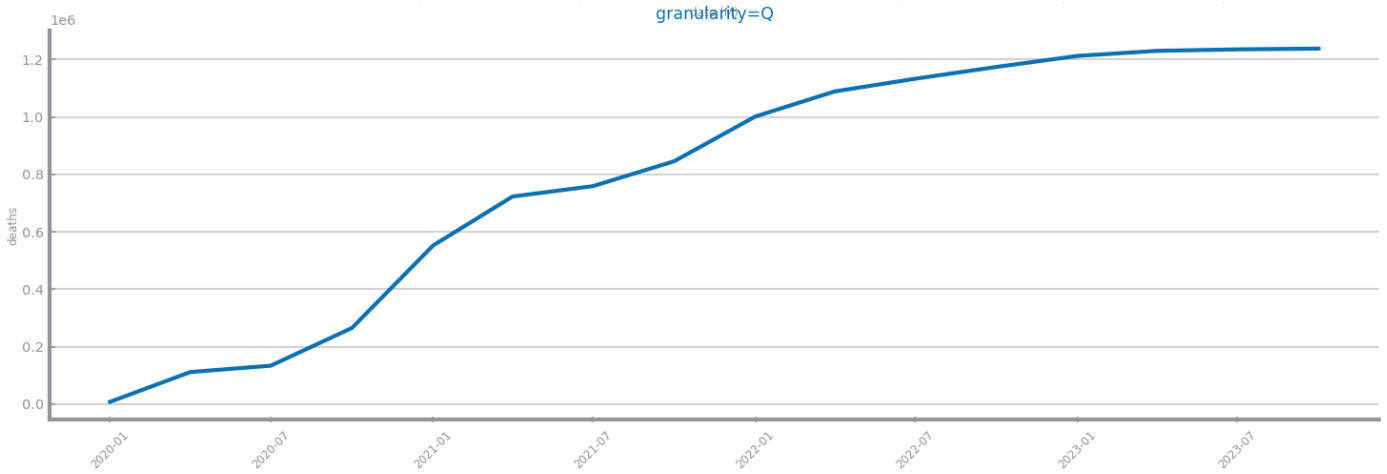


Figure 63 Time series 1 at the third chosen granularity

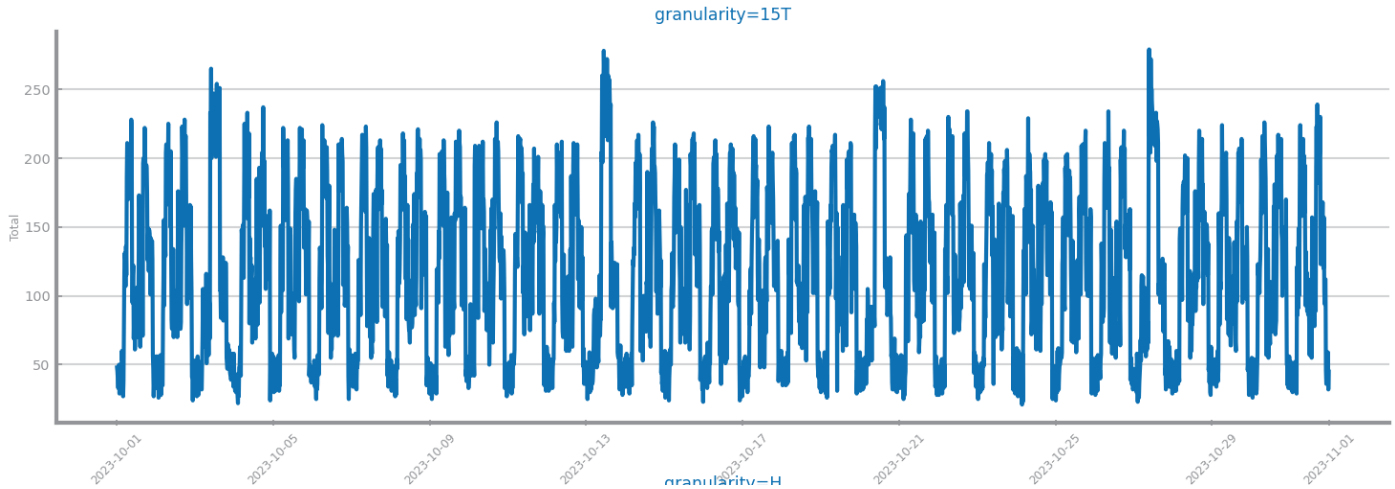


Figure 64 Original time series 2 (the most atomic detail)

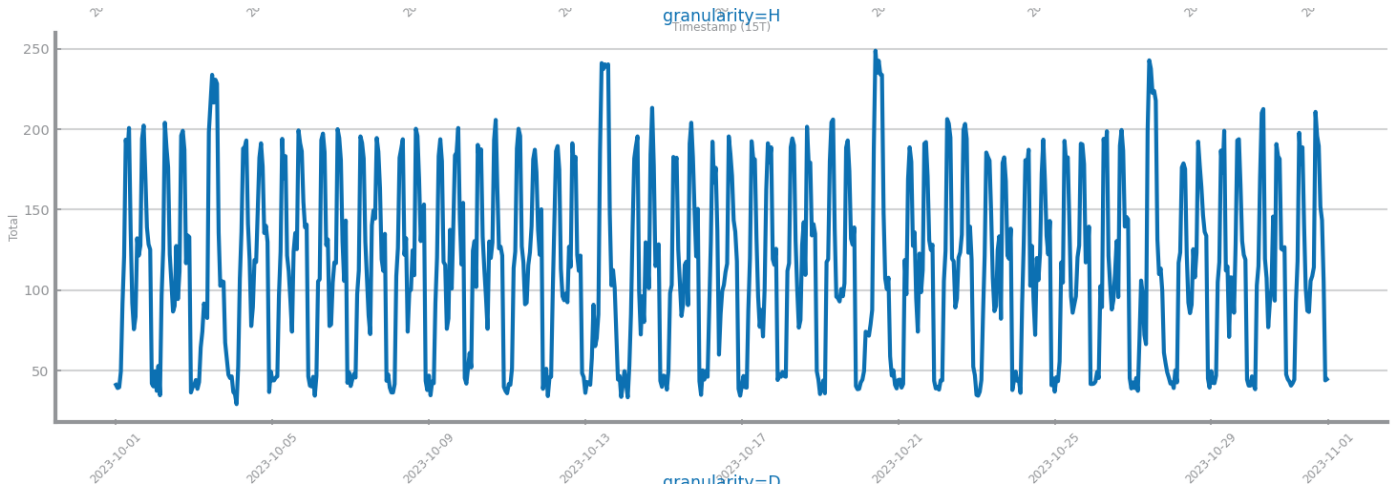


Figure 65 Time series 2 at the second chosen granularity

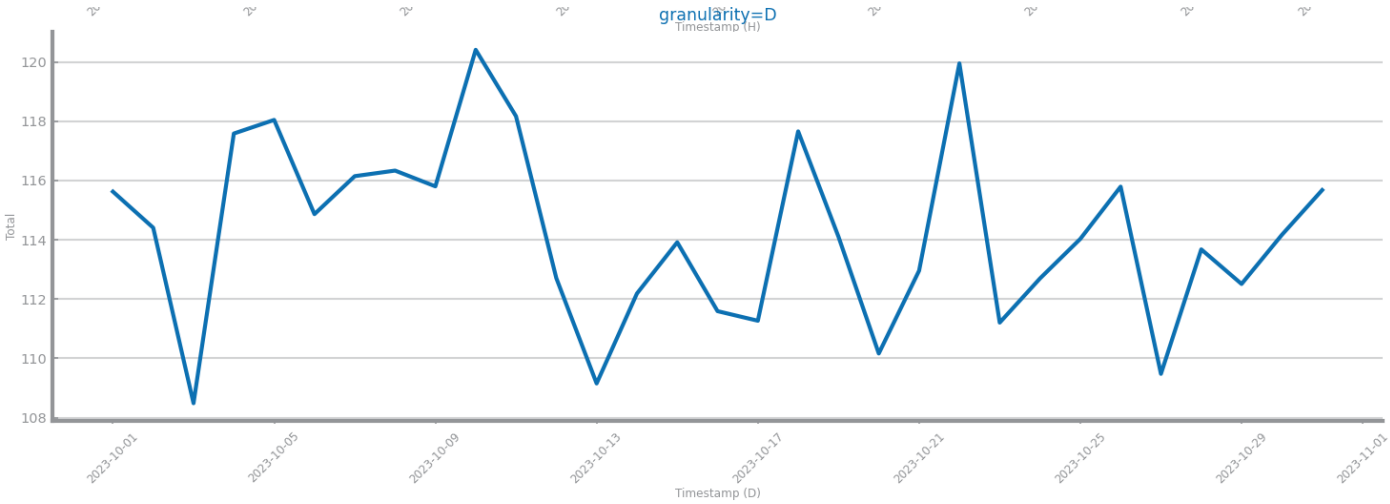
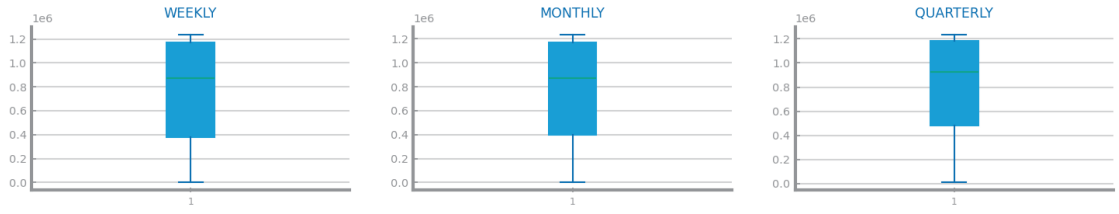


Figure 66 Time series 2 at the third chosen granularity

Data Distribution

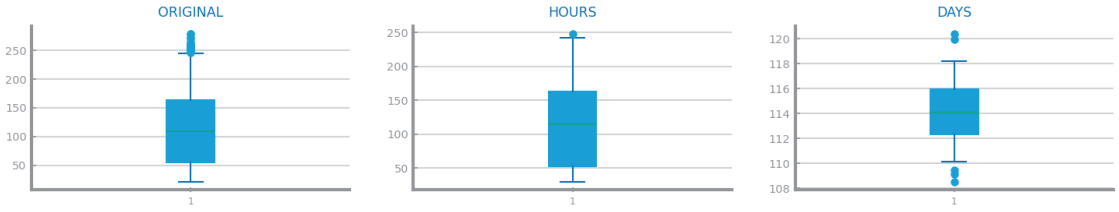
S1 follows a multimodal distribution for all granularities. S2 follows an (original) inverse cumulative, (second) multimodal, (third) normal distribution.

Both series have significant autocorrelations but not seasonal.



WEEKLY		MONTHLY		QUARTERLY	
count	1.990000e+02	count	4.600000e+01	count	1.600000e+01
mean	7.743876e+05	mean	7.750457e+05	mean	7.944554e+05
std	4.356312e+05	std	4.398647e+05	std	4.483506e+05
min	0.000000e+00	min	2.500000e-01	min	7.682846e+03
25%	3.775975e+05	25%	4.007008e+05	25%	4.803667e+05
50%	8.691990e+05	50%	8.714809e+05	50%	9.235195e+05
75%	1.172049e+06	75%	1.171477e+06	75%	1.183809e+06
max	1.238650e+06	max	1.238175e+06	max	1.238175e+06
Name: deaths, dtype: float64		Name: deaths, dtype: float64		Name: deaths, dtype: float64	

Figure 67 Boxplots for time series 1 at different granularities



ORIGINAL		HOURS		DAYS	
count	2976.000000	count	744.000000	count	31.000000
mean	114.218414	mean	114.218414	mean	114.218414
std	60.190627	std	56.144258	std	3.043082
min	21.000000	min	29.250000	min	108.489583
25%	55.000000	25%	52.687500	25%	112.348958
50%	109.000000	50%	114.875000	50%	114.104167
75%	164.000000	75%	162.687500	75%	115.973958
max	279.000000	max	248.750000	max	120.406250
Name: Total, dtype: float64		Name: Total, dtype: float64		Name: Total, dtype: float64	

Figure 68 Boxplots for time series 2 at different granularities

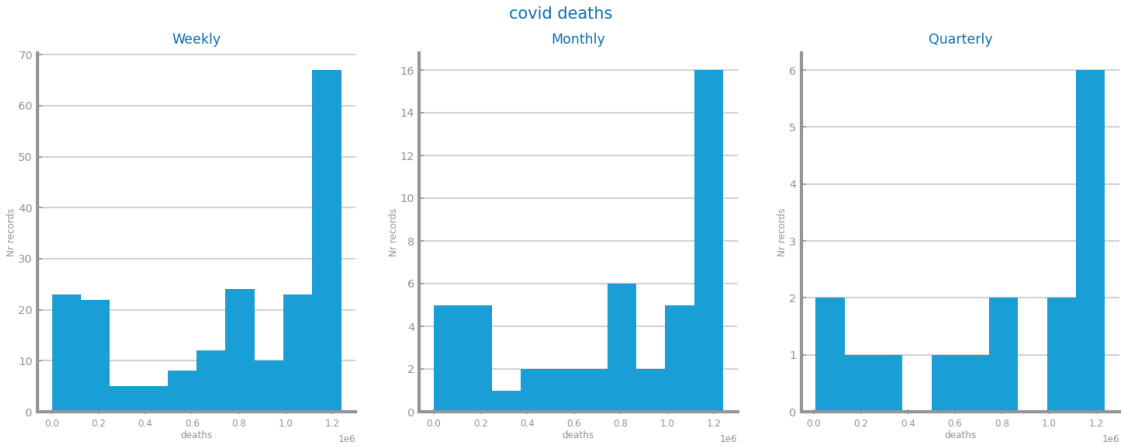


Figure 69 Histograms for time series 1 at different granularities

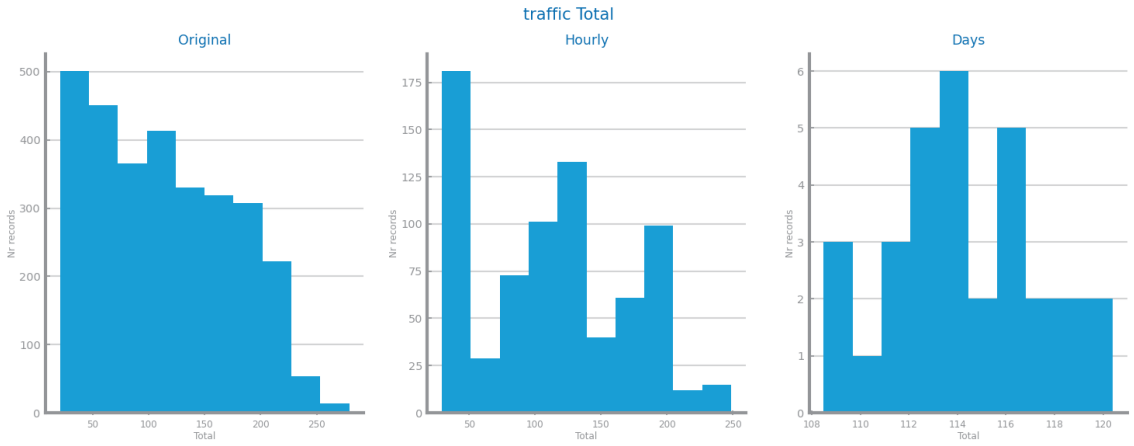


Figure 70 Histograms for time series 2 at different granularities

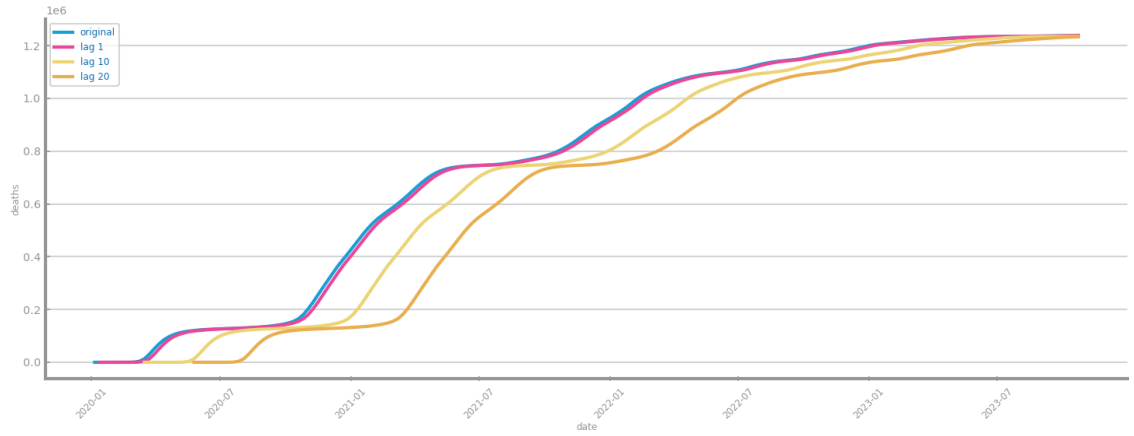


Figure 71 Autocorrelation lag-plots for original time series 1

Figure 72 Autocorrelation lag-plots for original time series 2

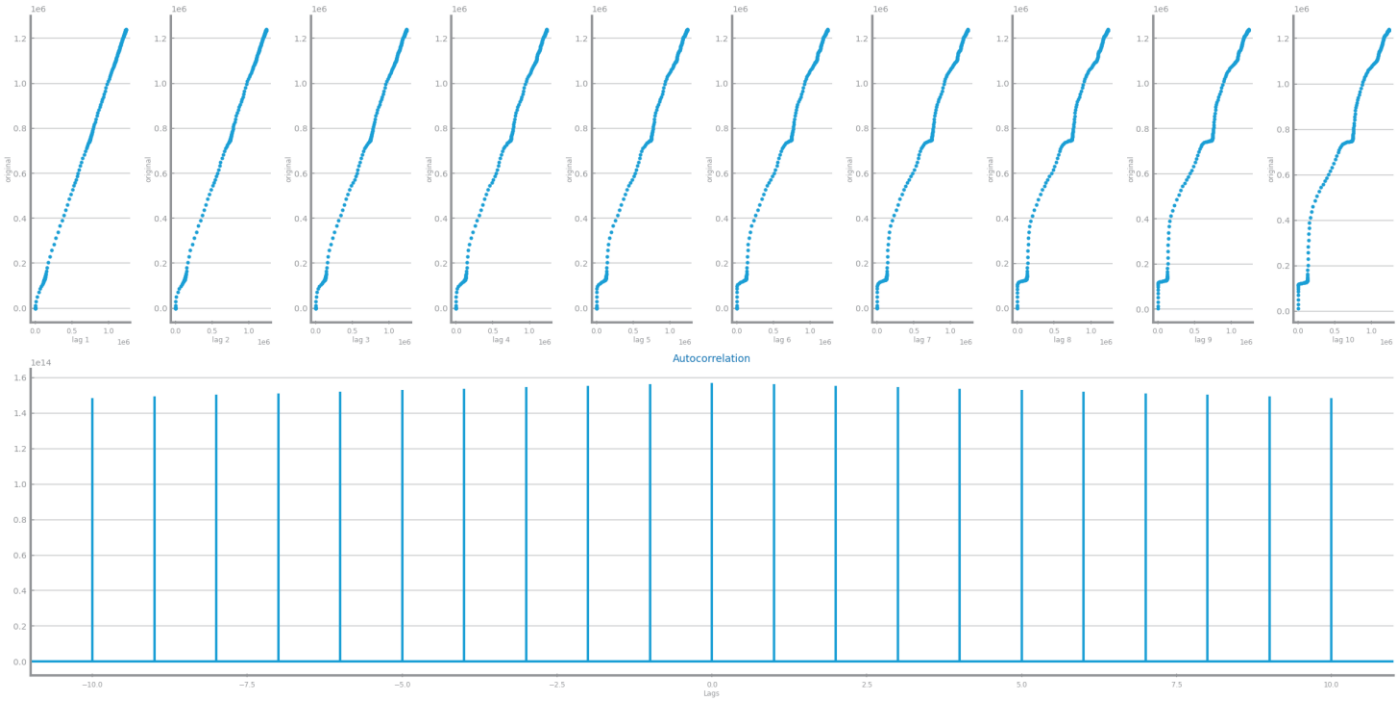


Figure 73 Autocorrelation correlogram for original time series 1

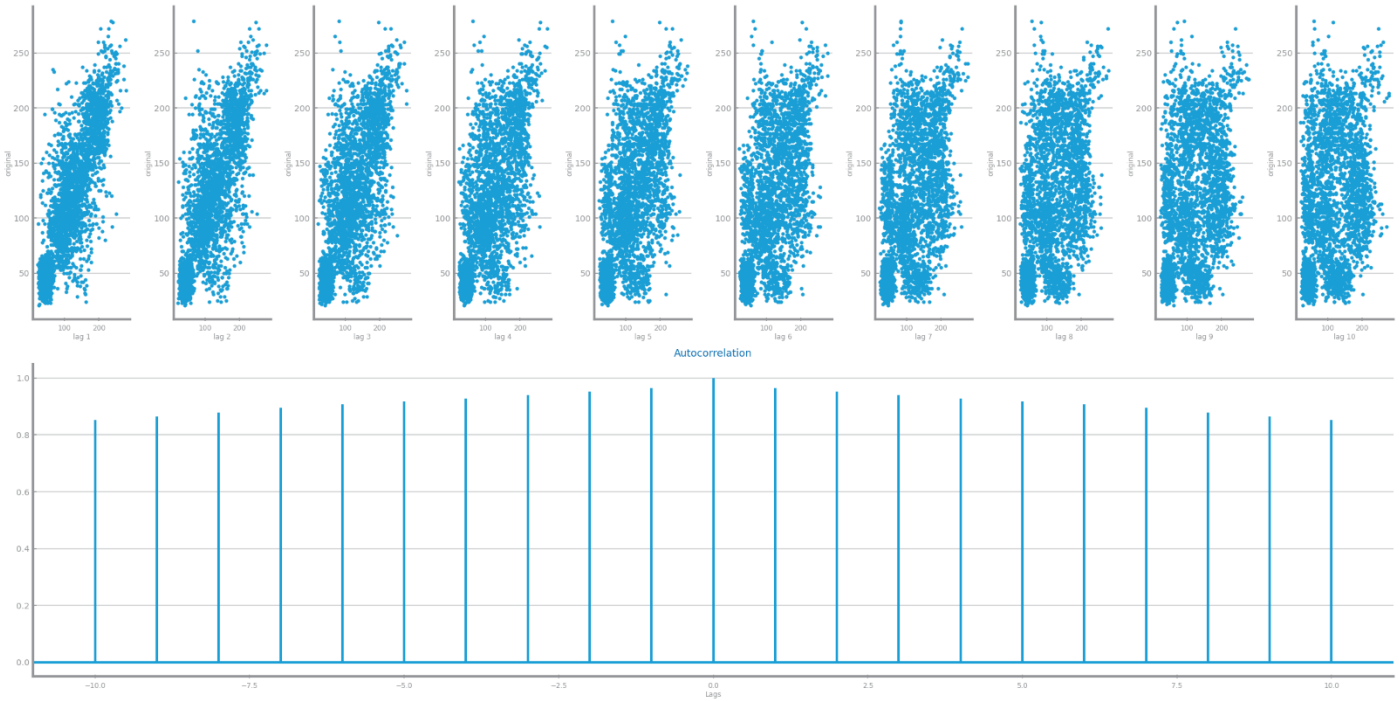


Figure 74 Autocorrelation correlogram for original time series 2

Data Stationarity

Per Augmented Dickey-Fuller test, S2 is stationary in its original and second granularity, while S1 is only stationary in its second one.

S2's mean variance oscillates in all granularities, unlike S1's, which trends upwards.

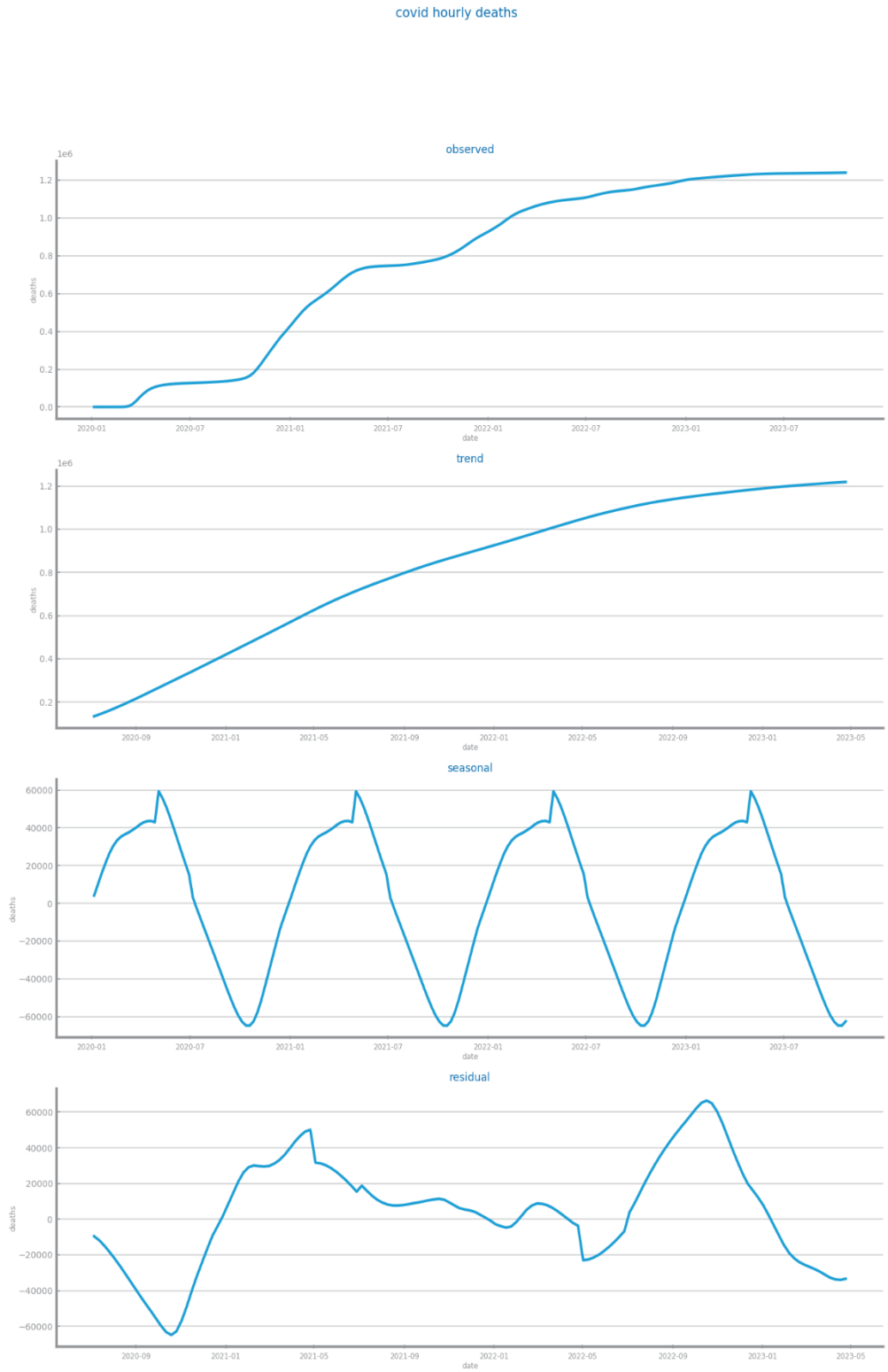
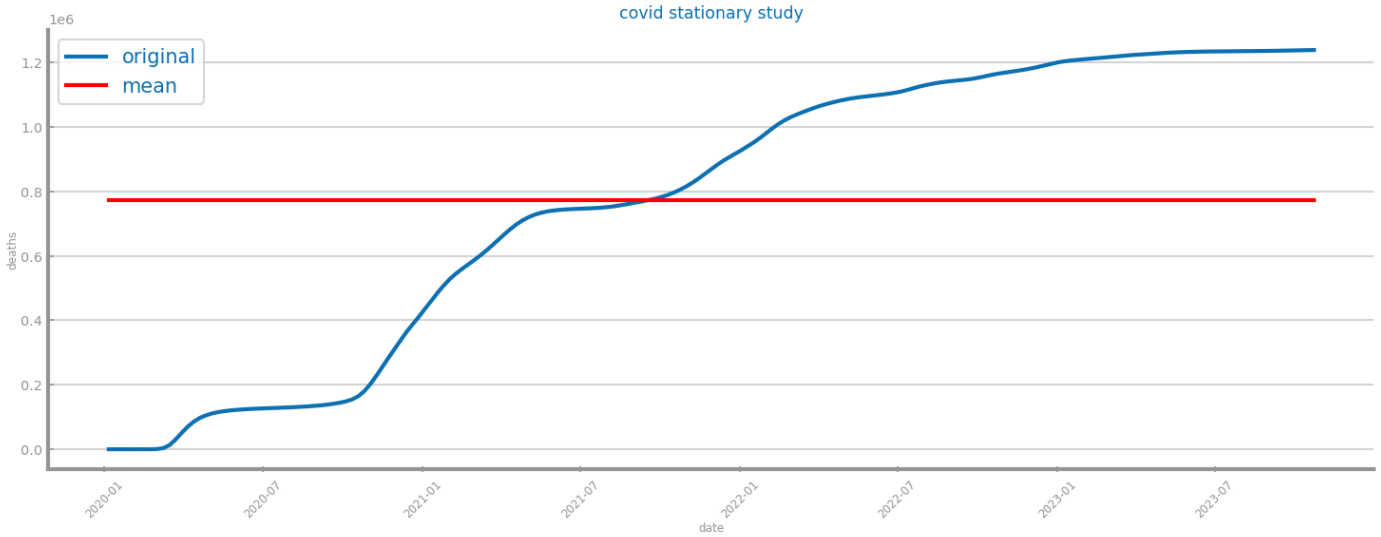
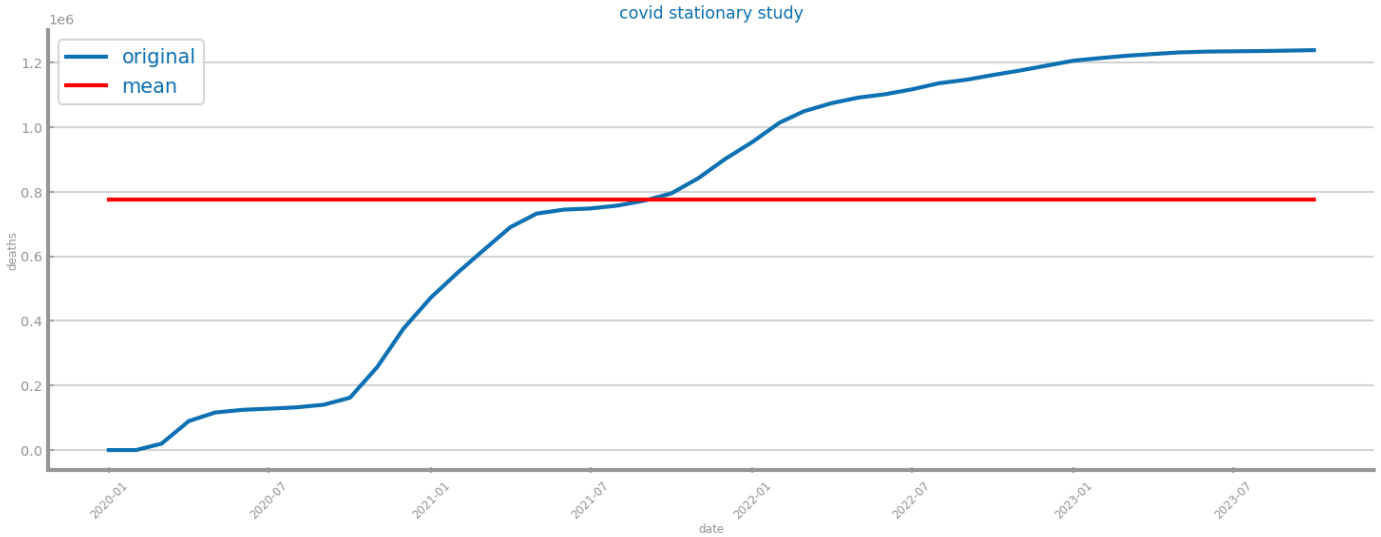
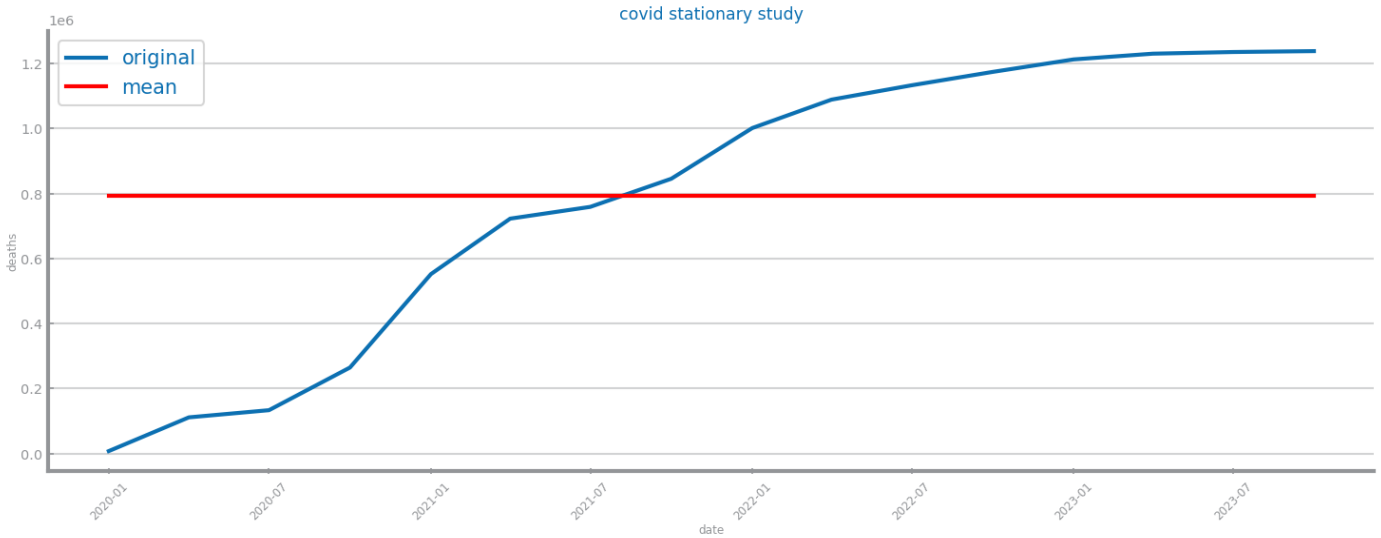


Figure 75 Components study for time series 1


covid stationary study
1e6
original
mean
deaths
date


covid stationary study
1e6
original
mean
deaths
date


covid stationary study
1e6
original
mean
deaths
date

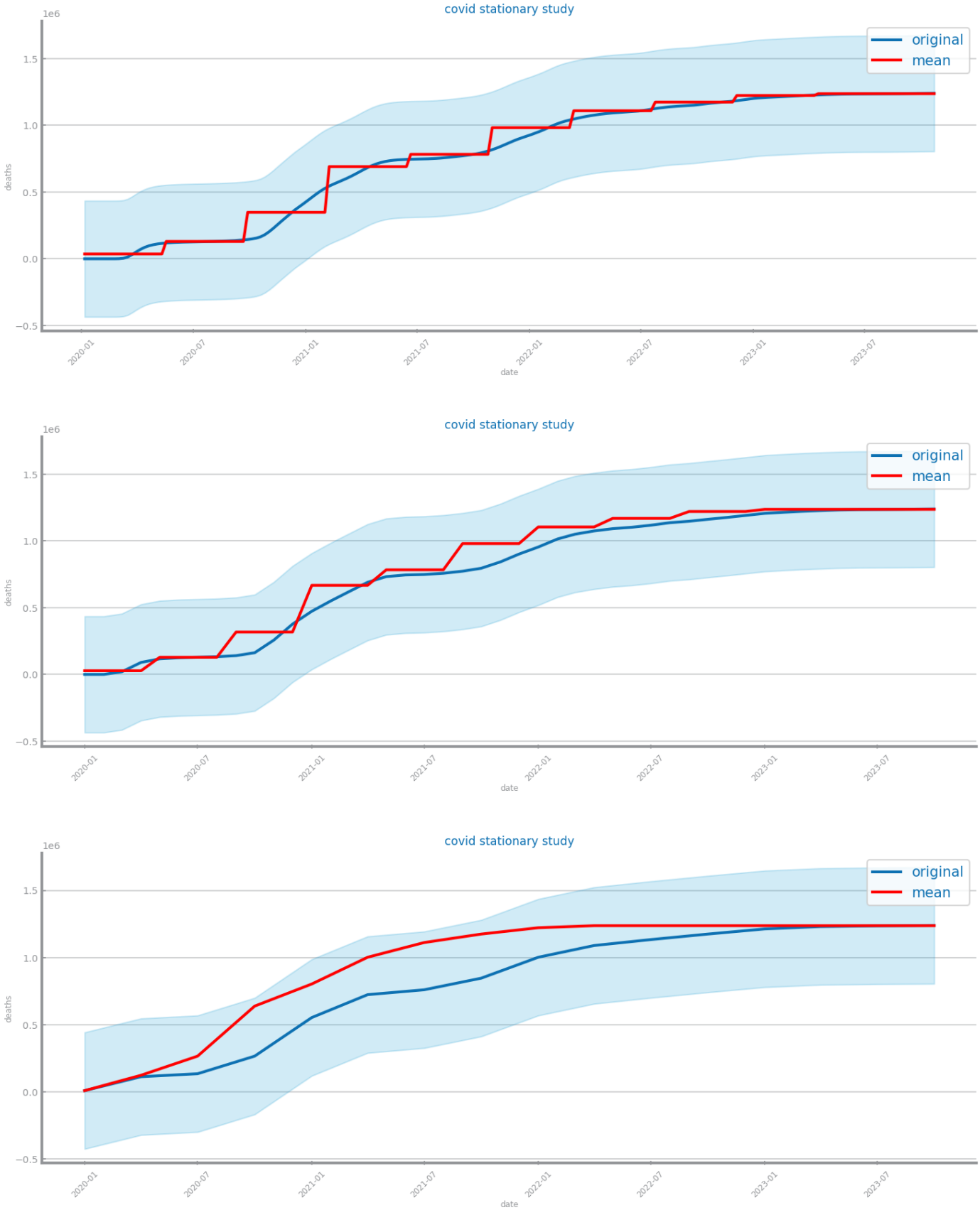


Figure 76 Stationarity study for time series 1

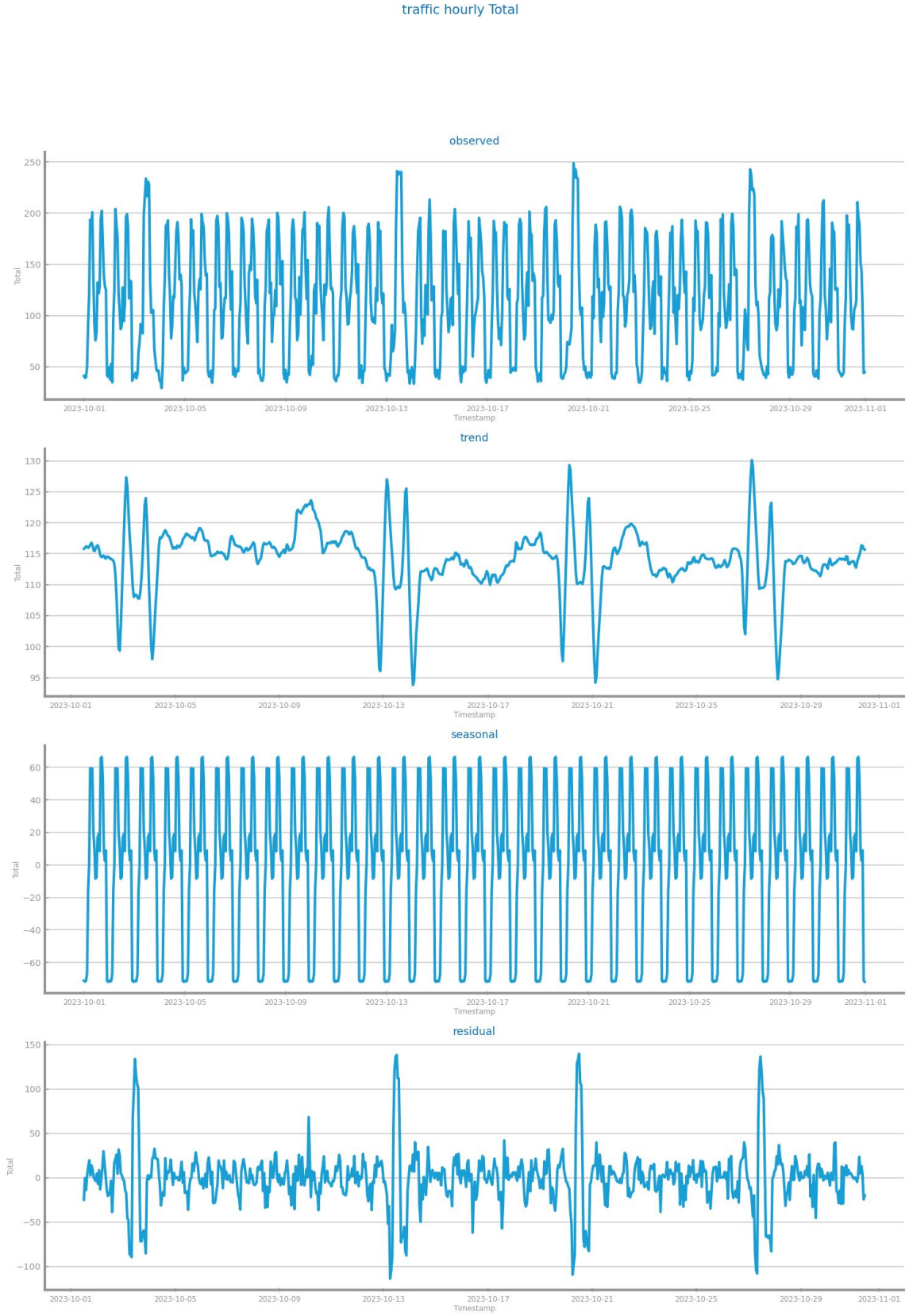
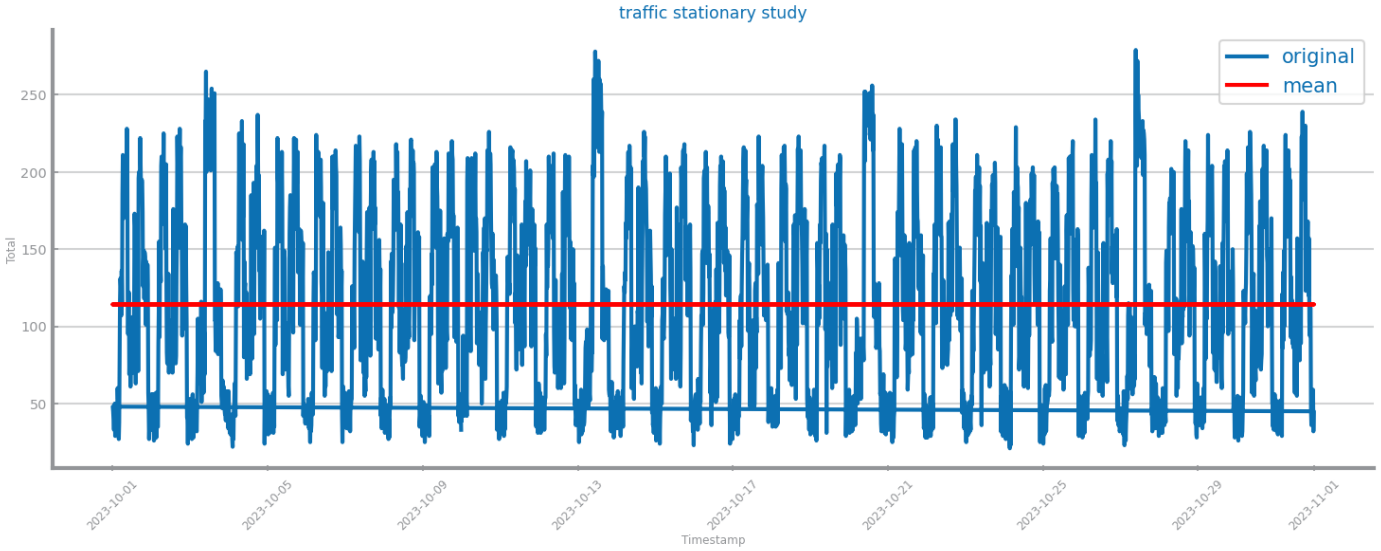
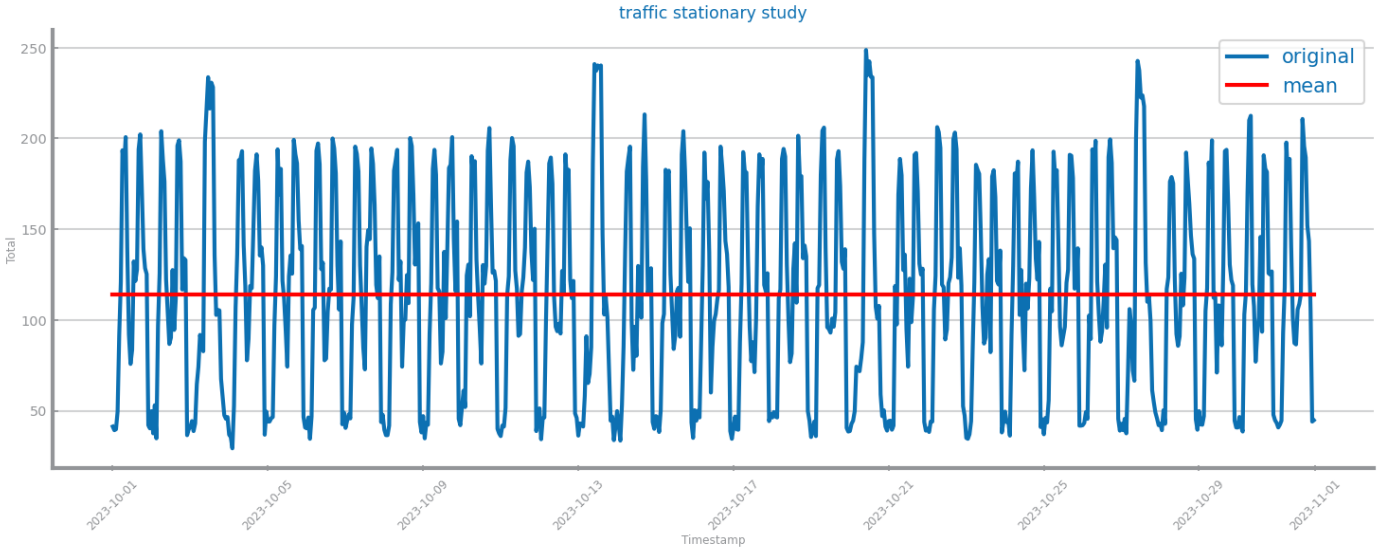
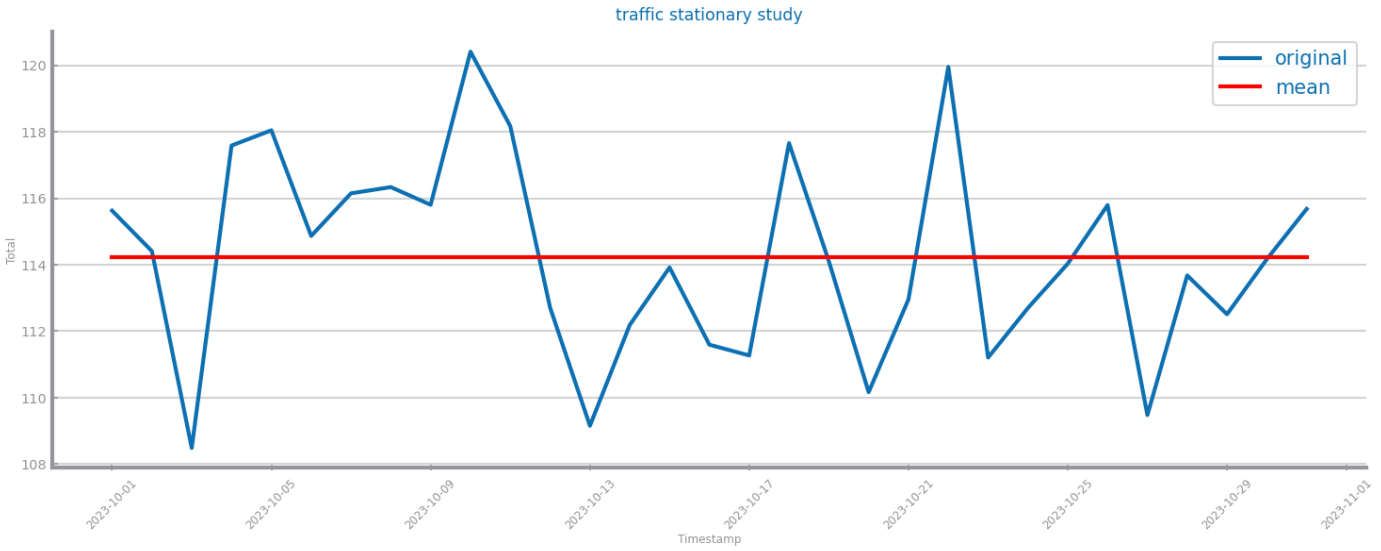


Figure 77 Components study for time series 2


traffic stationary study
original
mean
Total
Timestamp


traffic stationary study
original
mean
Total
Timestamp


traffic stationary study
original
mean
Total
Timestamp

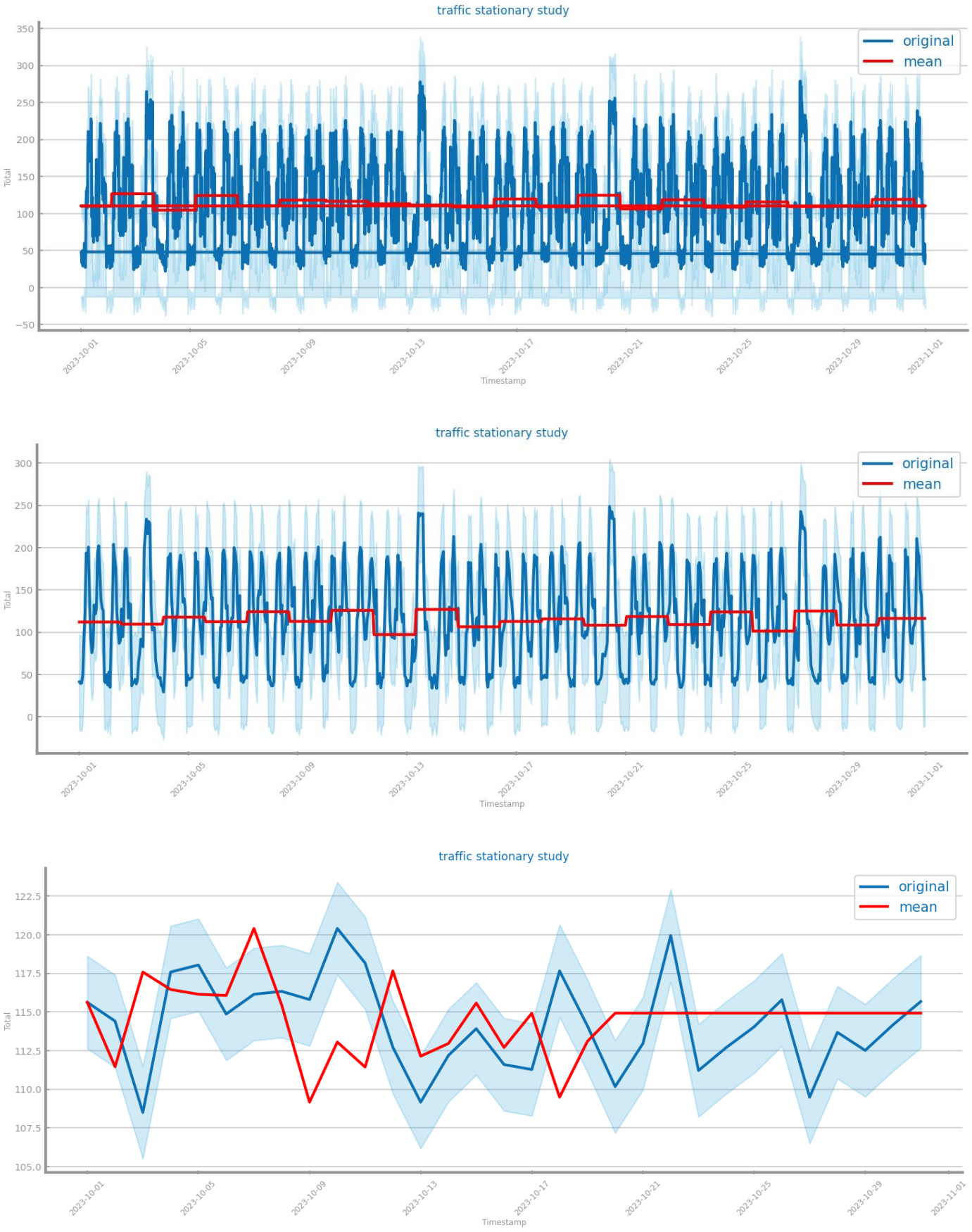


Figure 78 Stationarity study for time series 2

6 DATA TRANSFORMATION

Aggregation

Data aggregated given the 3 most atomic granularities.

In S1, the chosen granularity was the original (best performance), and in S2 was hourly (second) - although the 3rd presented better results, due to loss of information.

Transformations applied to the entire series to maintain dimensionality.

Figure 79 Forecasting plots after different aggregations on time series 1 (weekly, monthly, quarterly)

Figure 80 Forecasting results after different aggregations on time series 1 (weekly, monthly, quarterly)

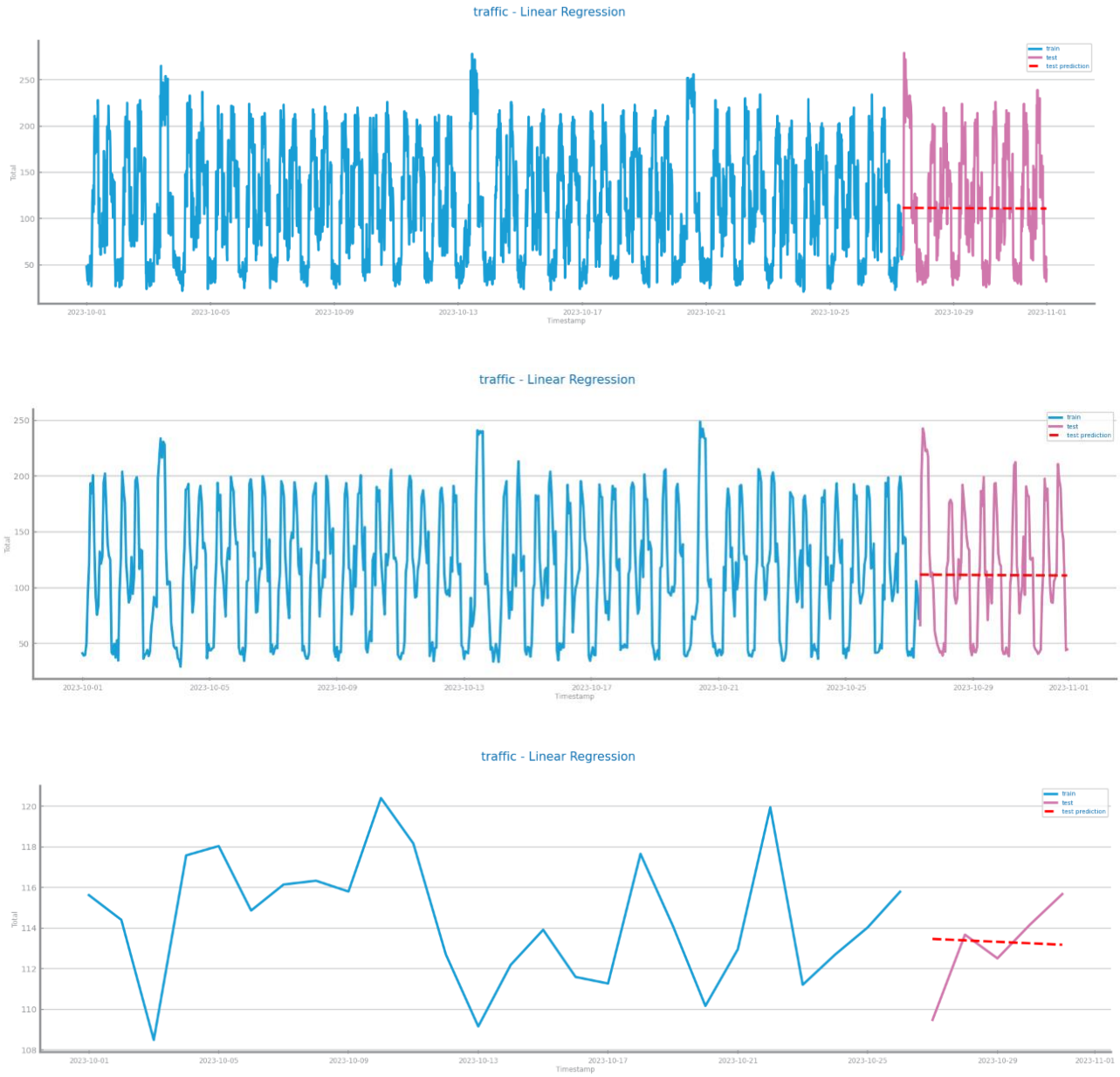


Figure 81 Forecasting plots after different aggregations on time series 2 (quarter-hourly, hourly, daily)

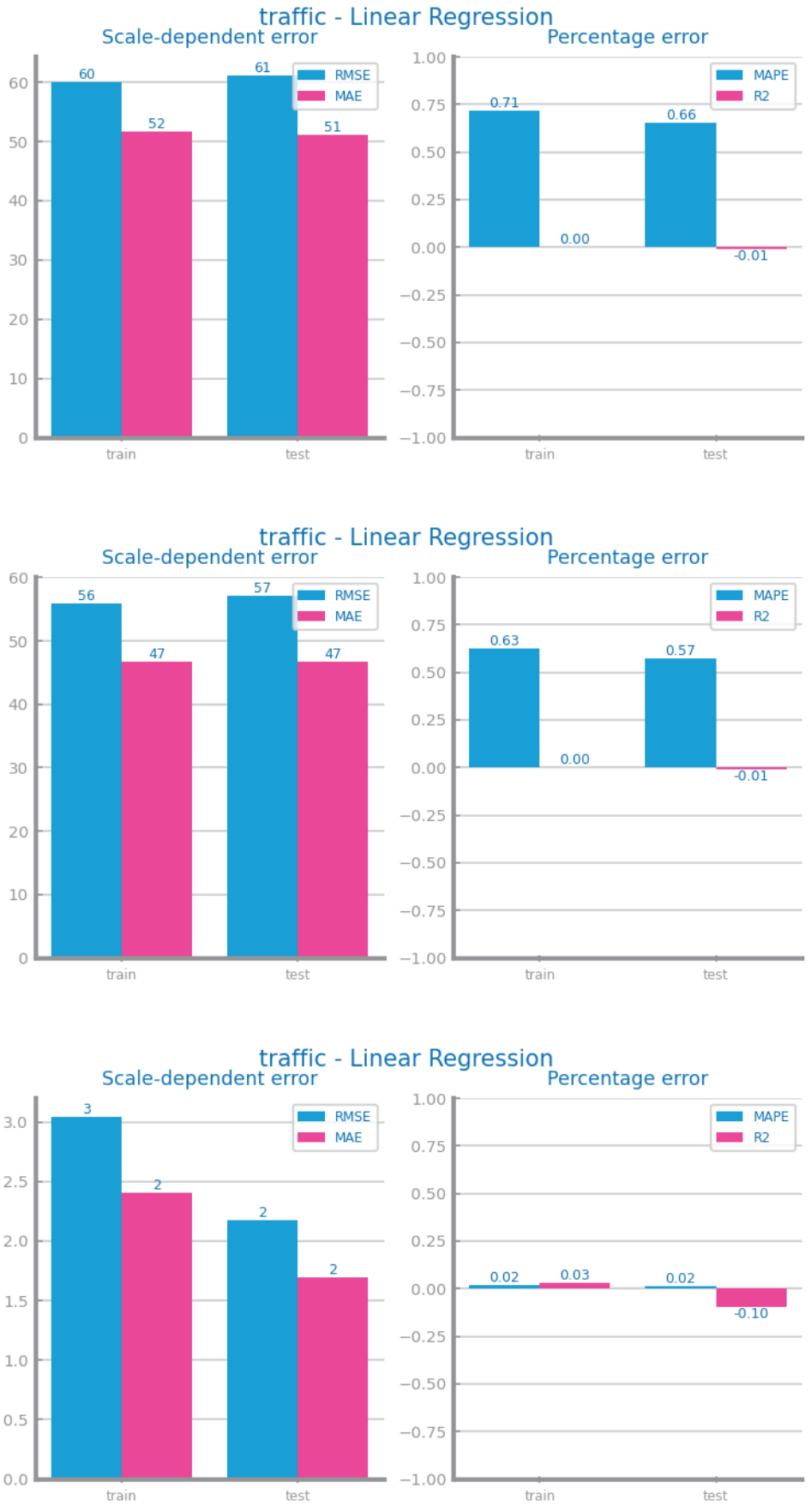


Figure 82 Forecasting results after different aggregations on time series 2 (quarter-hourly, hourly, daily)

Smoothing

For S1, win sizes of 20, 50, 75. Although the best performance was 75, we chose 20 since the # of records is low and smoothing too much would lead to a smaller train set than recommended.

For S2, win sizes of 10, 30, 75. Performances were similar but 10 was chosen (to not lose sharpness on the data).

Figure 83 Forecasting plots after different smoothing parameterizations on time series 1 (20, 50, 75)

Figure 84 Forecasting results after different smoothing parameterizations on time series 1 (20, 50, 75)

Figure 85 Forecasting plots after different smoothing parameterizations on time series 2 (10, 30, 75)

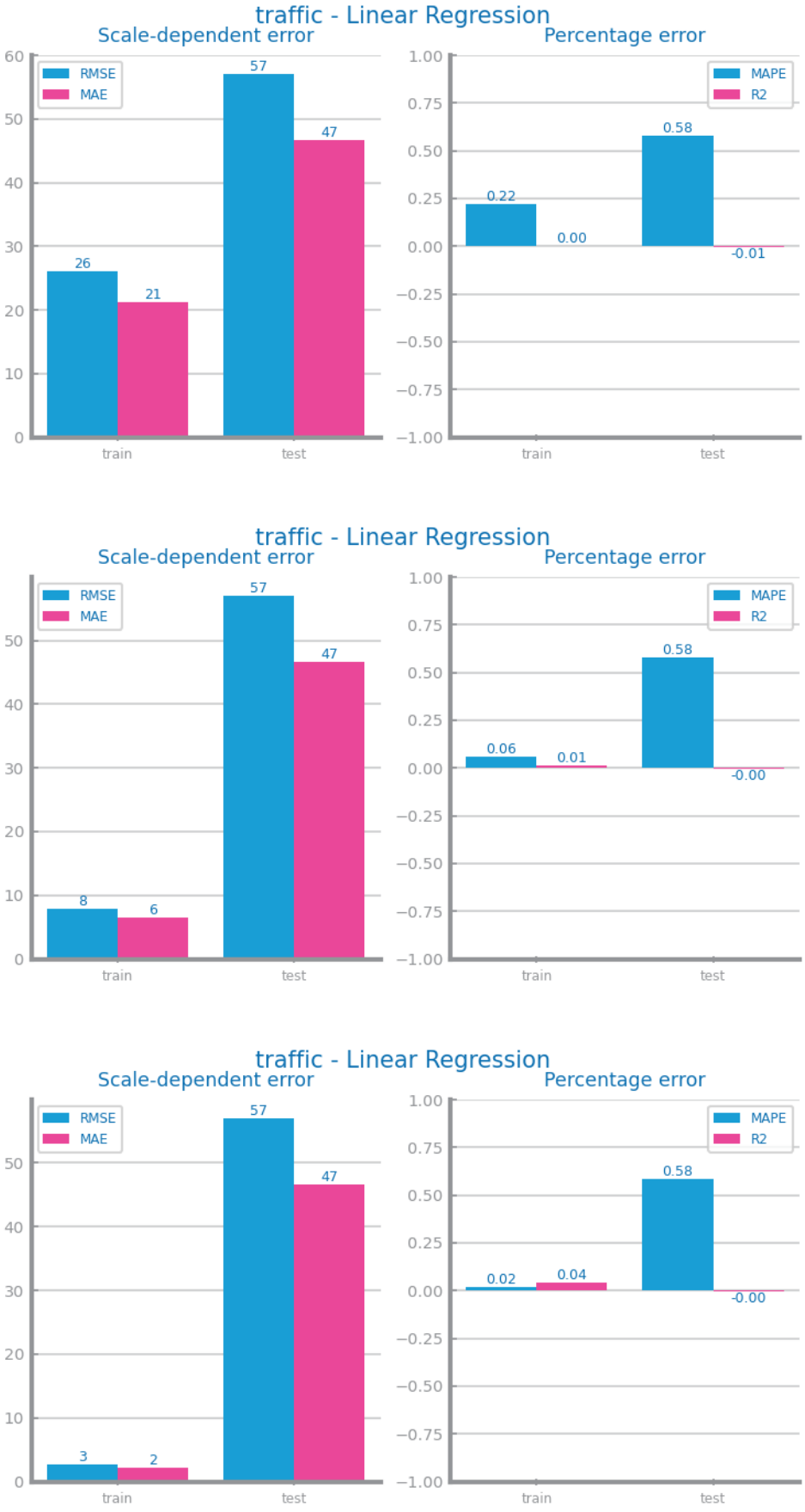


Figure 86 Forecasting results after different smoothing parameterizations on time series 2 (10, 30, 75)

Differentiation

Differentiation applied to train and test since we are changing the data space.

Performances were bad since the model was trained with smooth and differentiated data and the test set only has differentiation applied to it.

No differentiation was chosen.

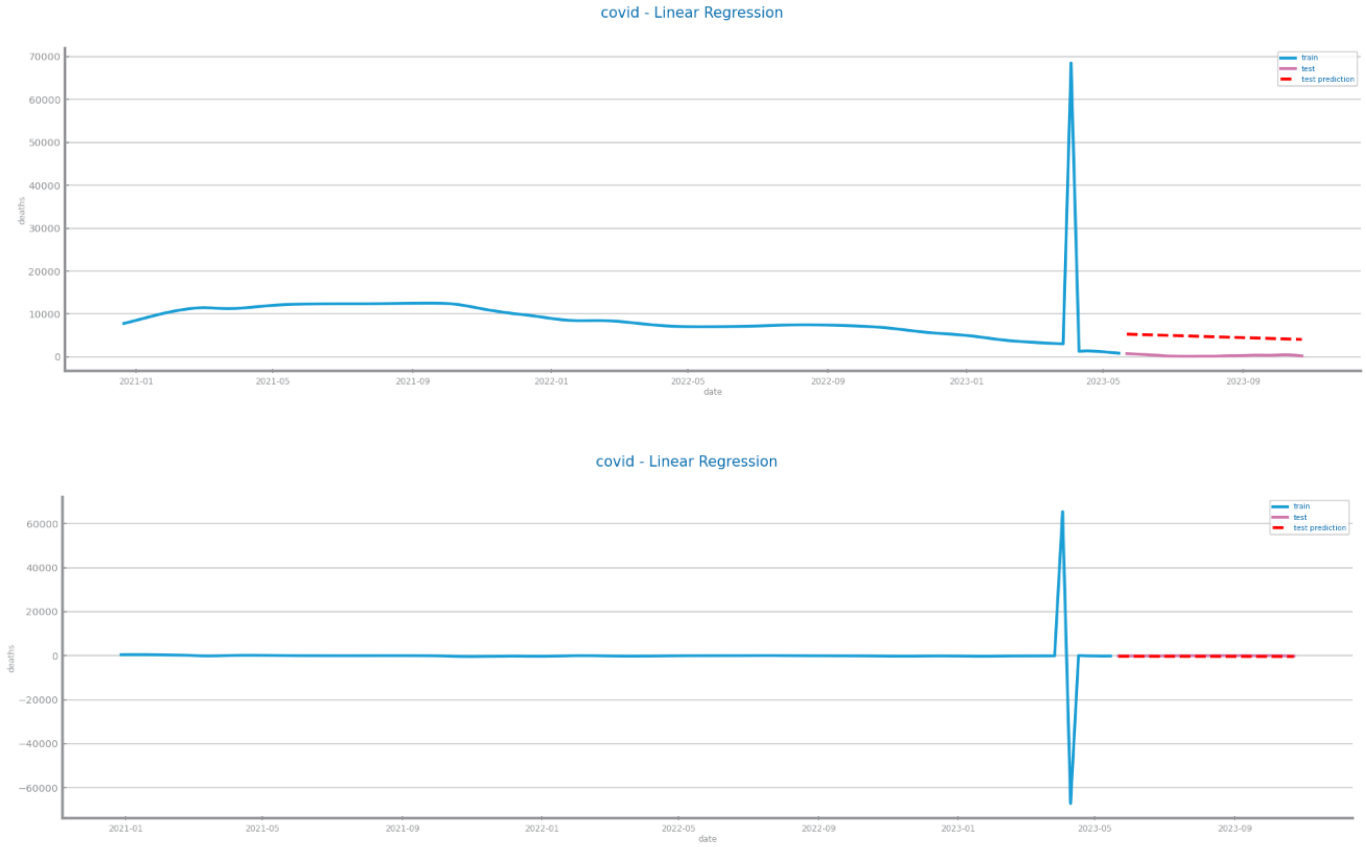
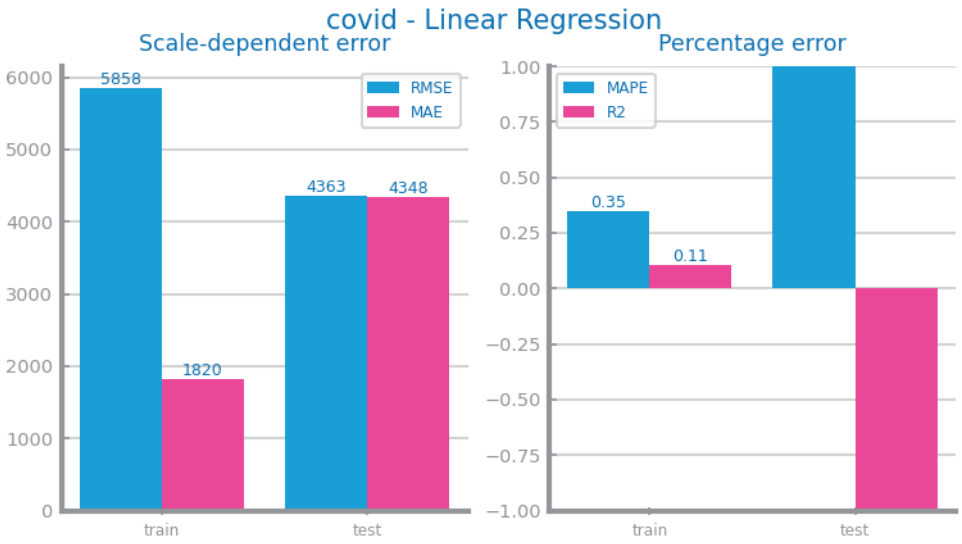


Figure 87 Forecasting plots after first and second differentiation of time series 1 (first, second)



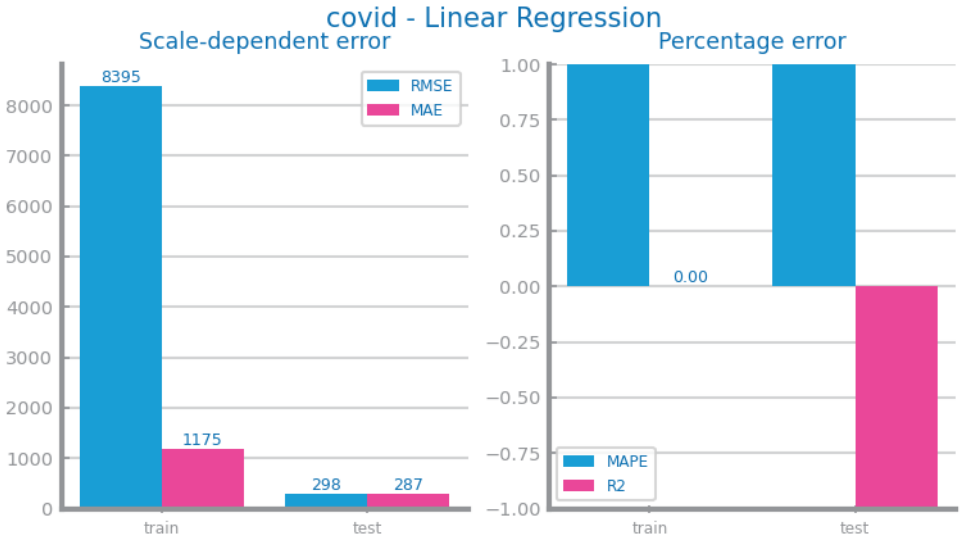


Figure 88 Forecasting results after first and second differentiation of time series 1 (first, second)

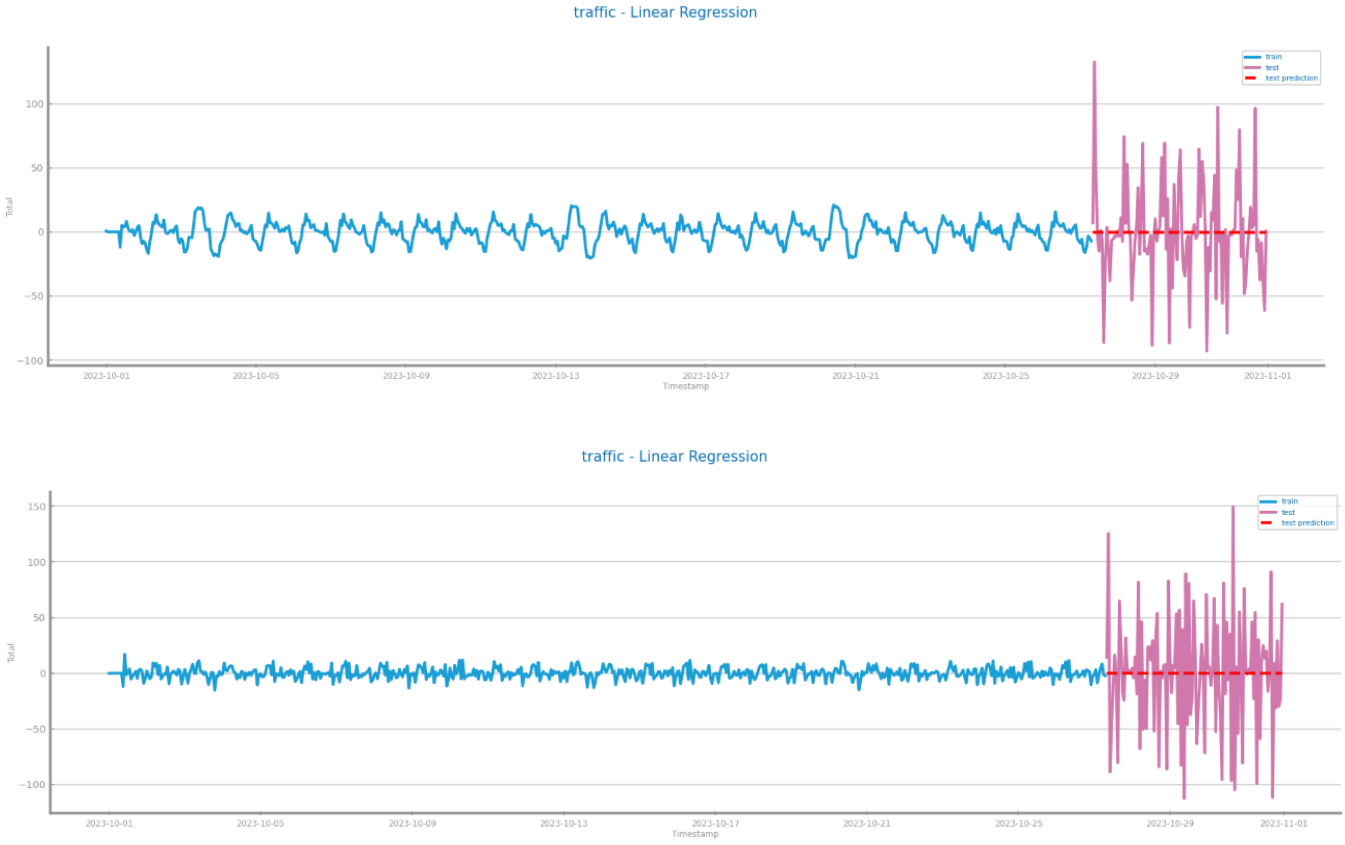


Figure 89 Forecasting plots after first and second differentiation of time series 2 (first, second)

Figure 90 Forecasting results after first and second differentiation of time series 2 (first, second)

Other transformations

No other transformations were applied.

7 MODELS' EVALUATION

For S1, the only transformation applied was smoothing the train set with window size 20, while for S2, we aggregated hourly (second granularity) and smooth the train with window size 10.

We will be minimizing MAPE for both series.

To enhance the model evaluation process, we have chosen to consistently split the data using a fixed seed.

Simple Average Model

SAM uses the mean value of a data set to predict. Since it's such a simple model, its' very poor performance is expected, especially on S1 since it is not stationary.

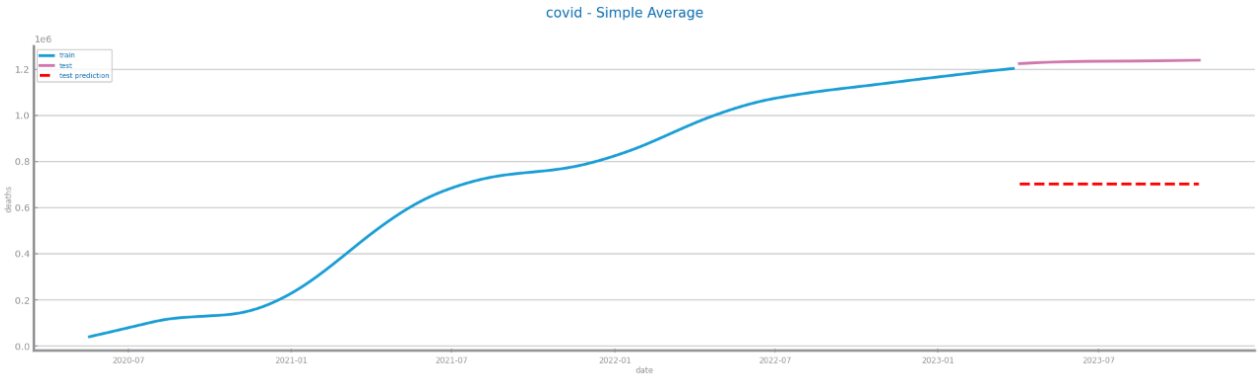


Figure 91 Forecasting plots obtained with Simple Average model over time series 1

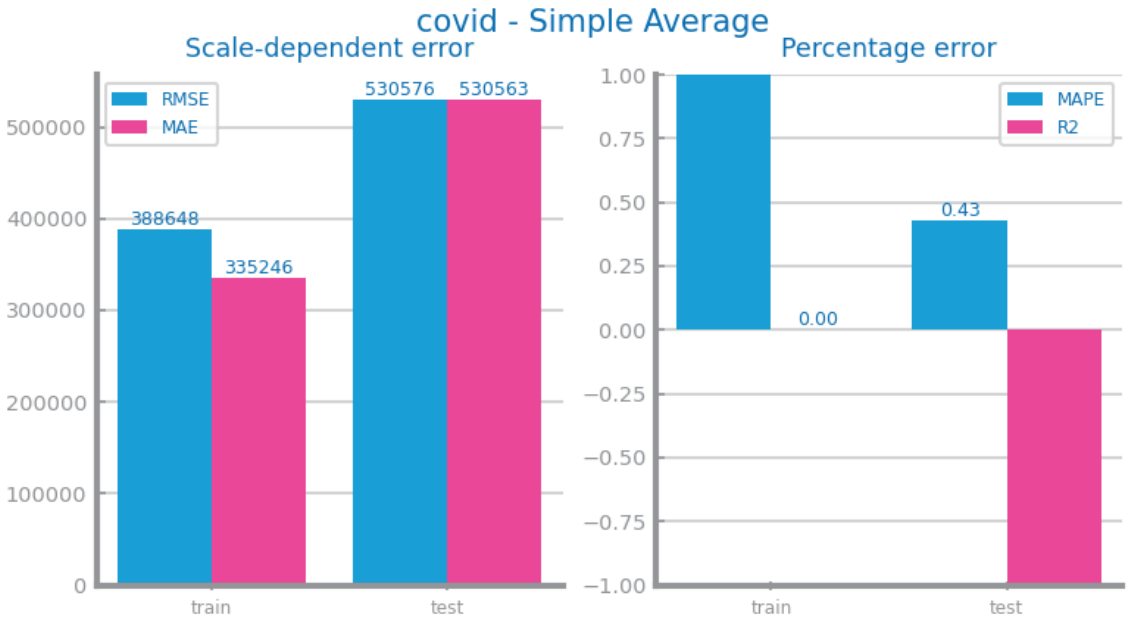


Figure 92 Forecasting results obtained with Simple Average model over time series 1

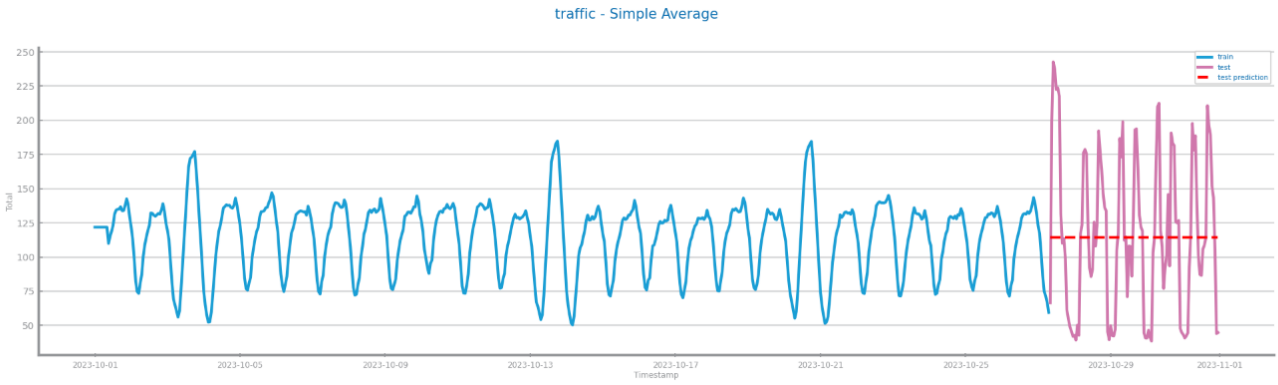


Figure 93 Forecasting plots obtained with Simple Average model over time series 2

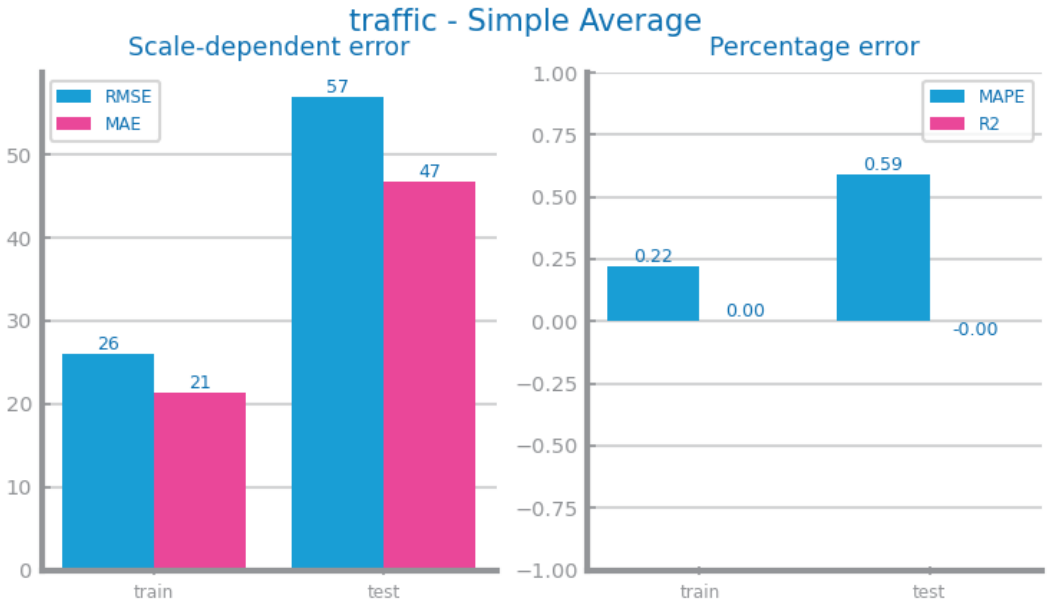


Figure 94 Forecasting results obtained with Simple Average model over time series 2

Persistence Model

PM predicts the outcome to be the same as the last value seen. It has 2 implementations: Realist and Optimistic.

Optimistic performs well since it forecasts based on the recent past – it can predict just one step ahead.

Realistic predicts that the test is equal to the last value of the train (straight line), thus it's poor performance.

Realist model performs well for S1 and not so good for S2.

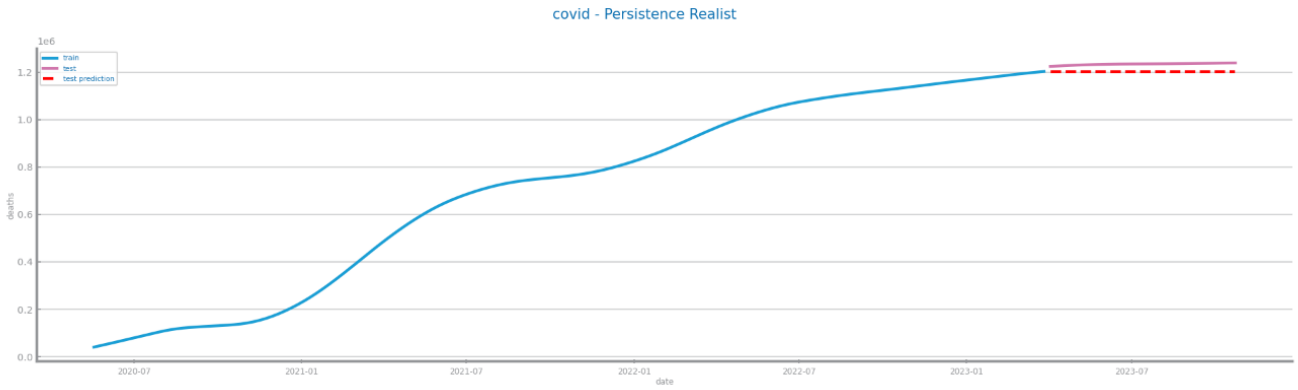


Figure 95 Forecasting plots obtained with Persistence model Realist over time series 1

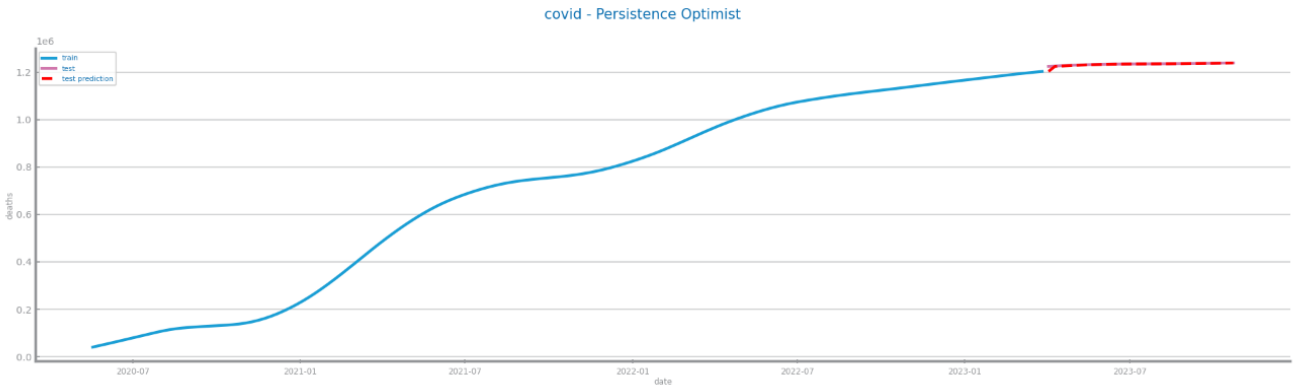
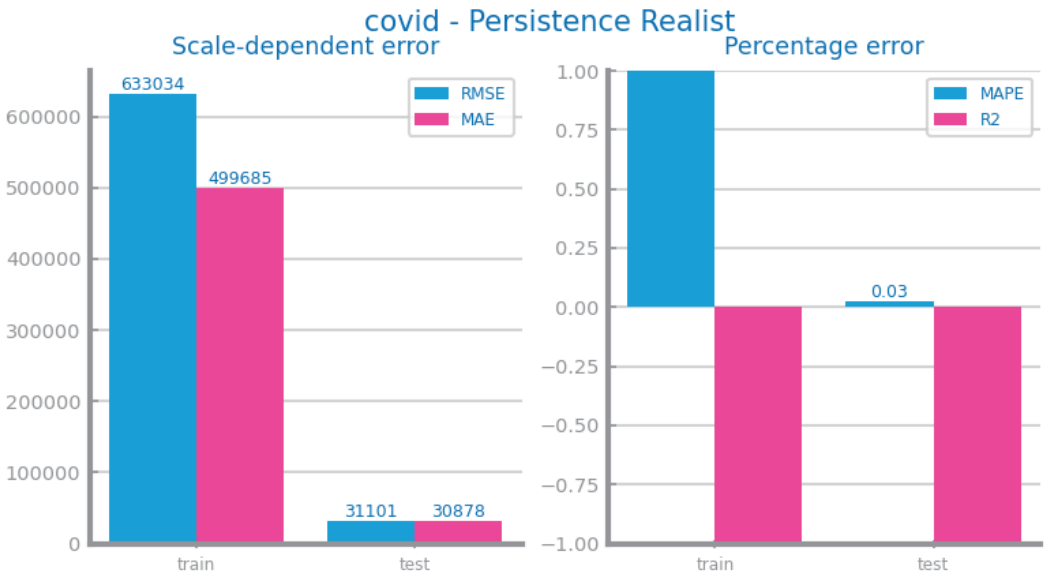


Figure 96 Forecasting plots obtained with Persistence model Optimist over time series 1



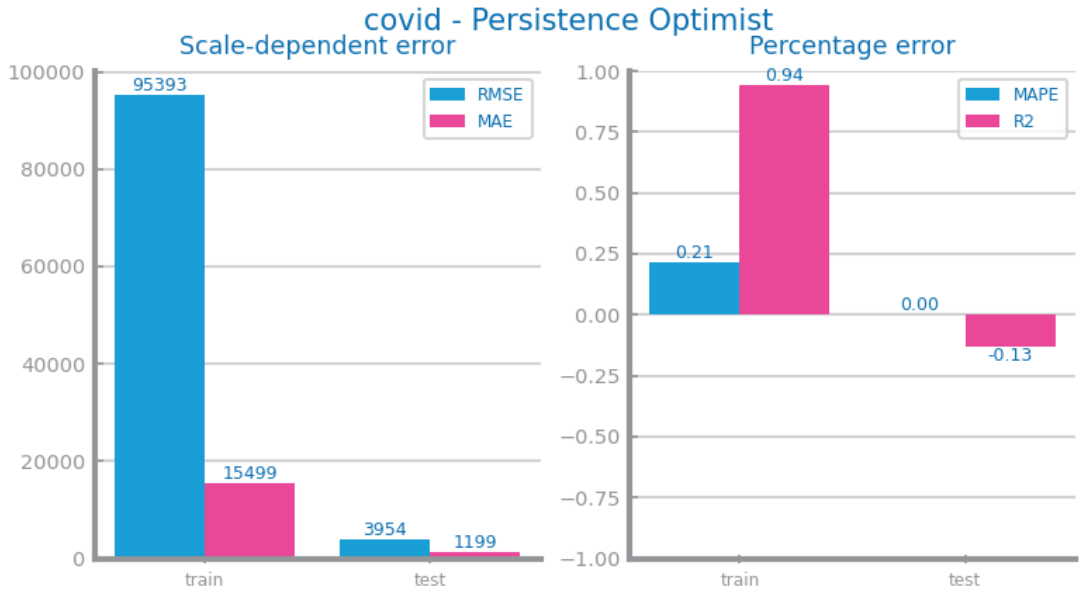


Figure 97 Forecasting results obtained with Persistence model in both situations over time series 1

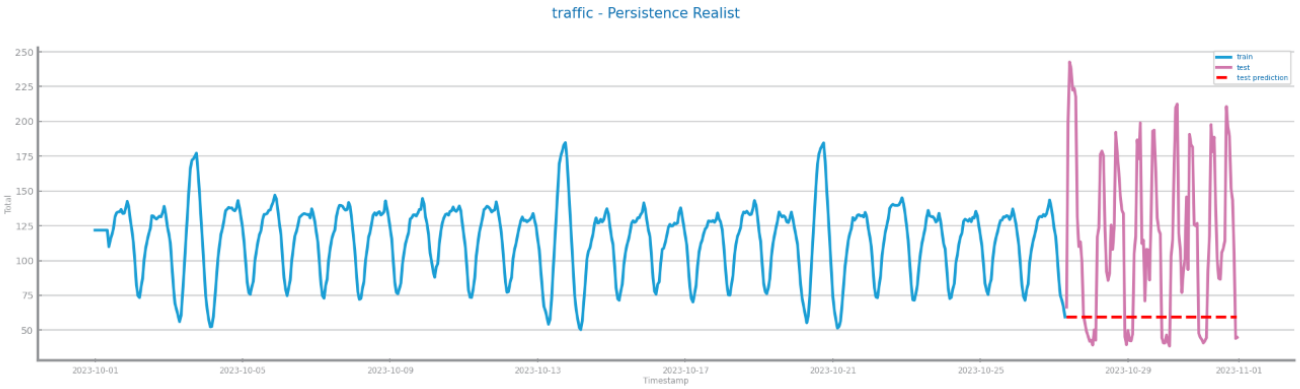


Figure 98 Forecasting plots obtained with Persistence model (long term) over time series 2

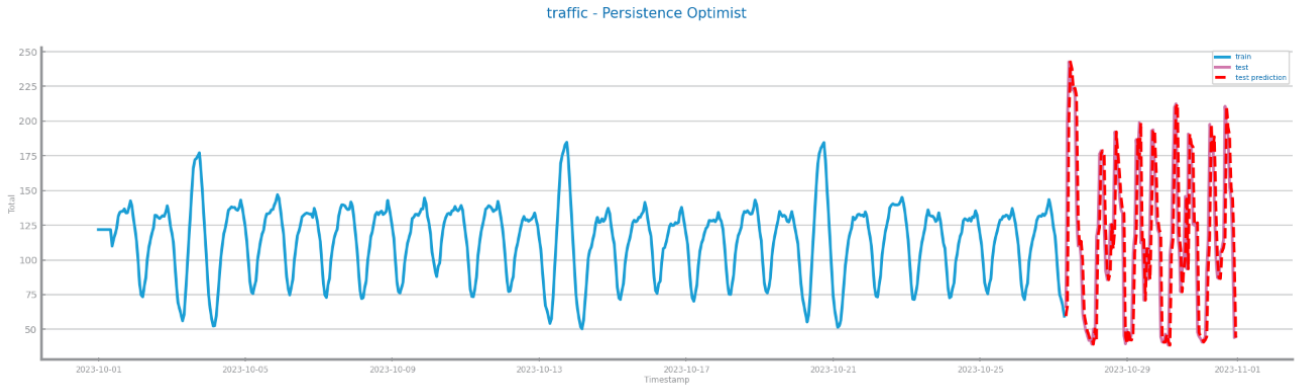


Figure 99 Forecasting plots obtained with Persistence model (next point) over time series 2

Figure 100 Forecasting results obtained with Persistence model in both situation over time series 2

Rolling Mean Model

For S1, we studied with win sizes of 3, 5, 10, 15, 20, 25, 30, 40, 50 and 100. The best win size was 3 (first size) with MAPE measure.

S2 studies the same win sizes except 100. The best win size was 3 with MAPE measure, even though it was not good.

As the win size increases, the model's responsiveness to short-term fluctuations in the data diminishes, while its sensitivity to long-term trends and patterns becomes more pronounced.

Even though S2 is more repetitive, it presents worse results than S1.

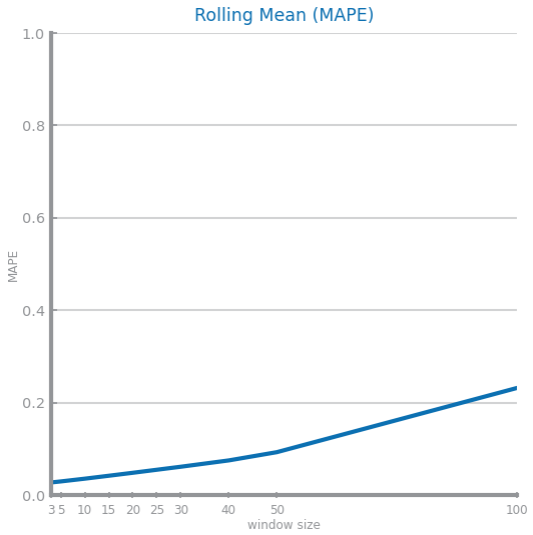


Figure 101 Forecasting study over different parameterizations of the rolling mean algorithm over time series 1

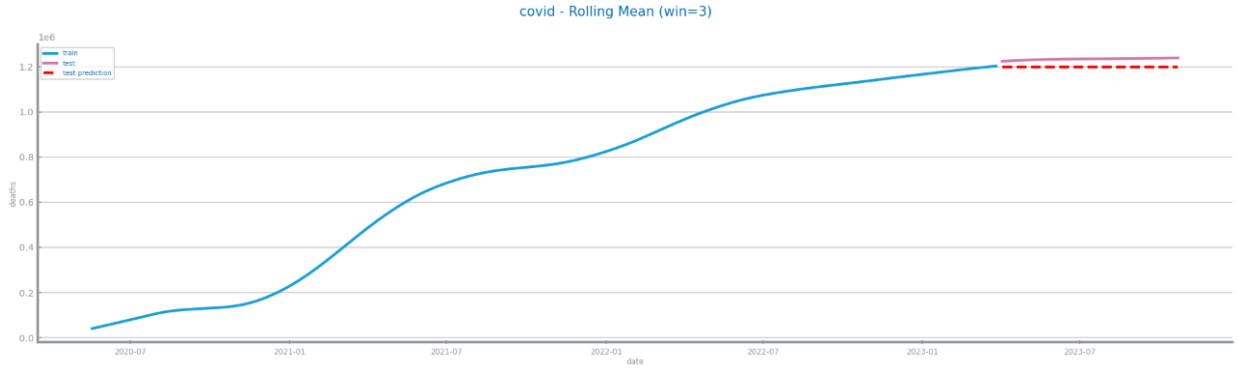


Figure 102 Forecasting plots obtained with the best parameterization of rolling mean algorithm, over time series 1

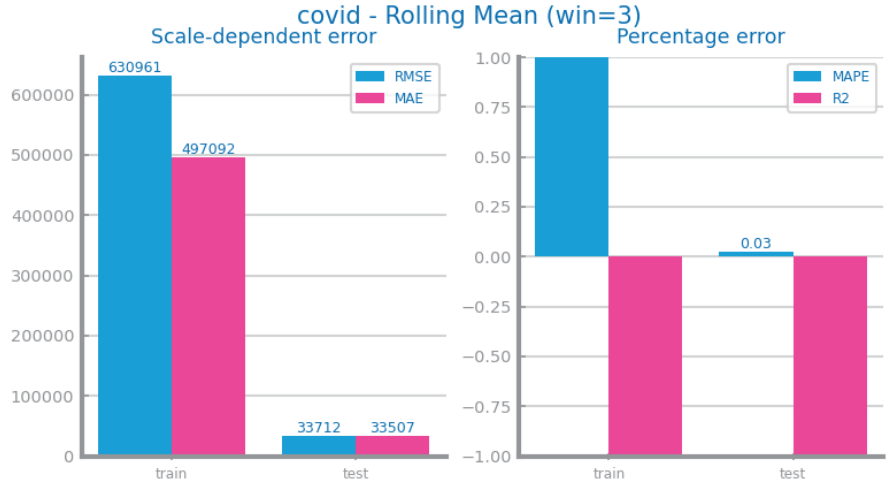


Figure 103 Forecasting results obtained with the best parameterization of rolling mean algorithm, over time series 1

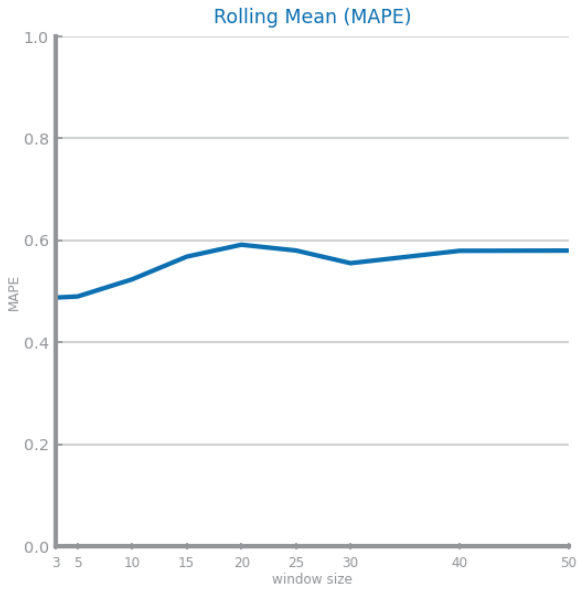


Figure 104 Forecasting study over different parameterizations of the rolling mean algorithm over time series 2

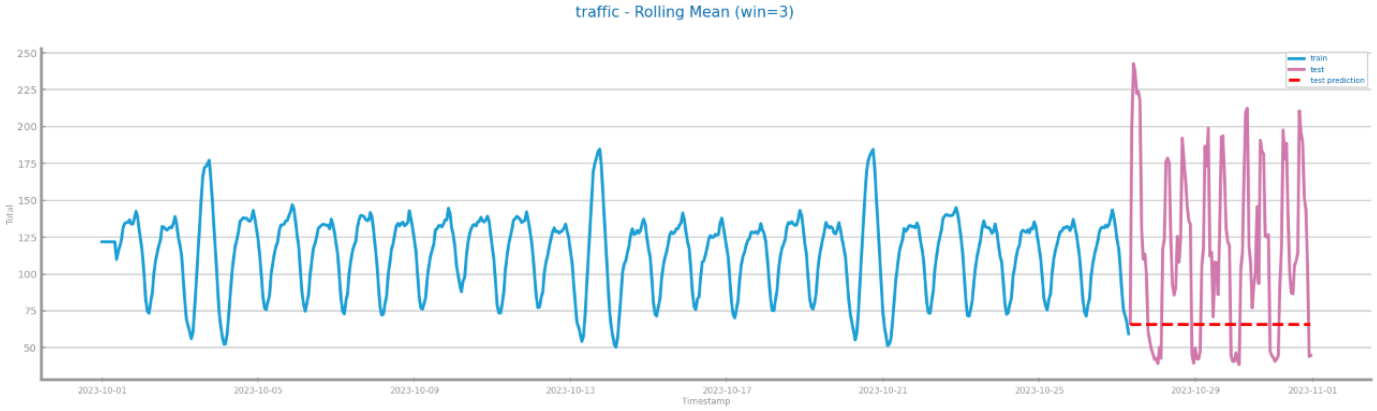


Figure 105 Forecasting plots obtained with the best parameterization of rolling mean algorithm, over time series 2

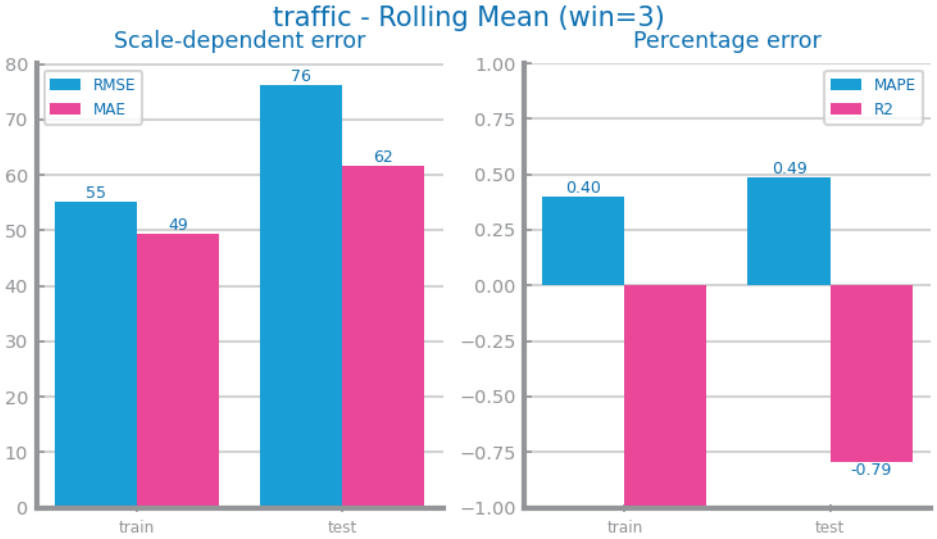


Figure 106 Forecasting results obtained with the best parameterization of rolling mean algorithm, over time series 2

ARIMA Model

ARIMA assumes that the data is stationary and thrives on temporal dependencies. For S2, the plot indicates that the model has predicted a consistent value, likely due to its reliance on temporal dependencies, which may not always be accurate. Despite this, the model has shown good performance compared to previous regressors, even for S1, which is not stationary.

The variations of the params p, d, q didn't really impact the results of our series.

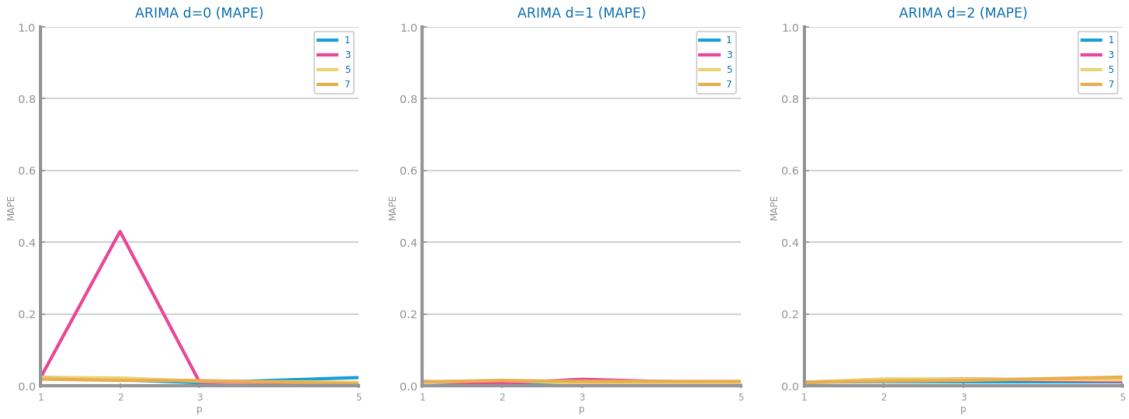


Figure 107 Forecasting study over different parameterizations of the ARIMA algorithm over time series 1

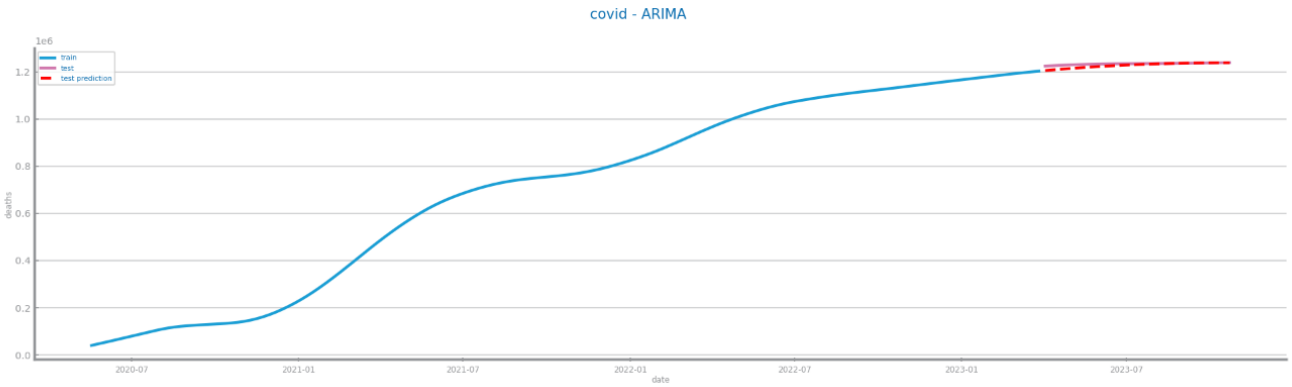


Figure 108 Forecasting plots obtained with the best parameterization of ARIMA algorithm, over time series 1

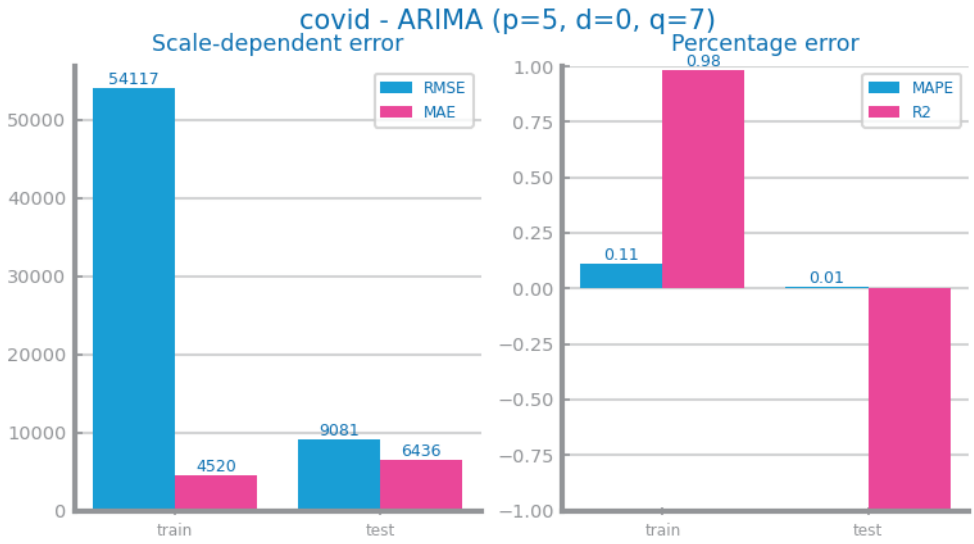


Figure 109 Forecasting results obtained with the best parameterization of ARIMA algorithm, over time series 1

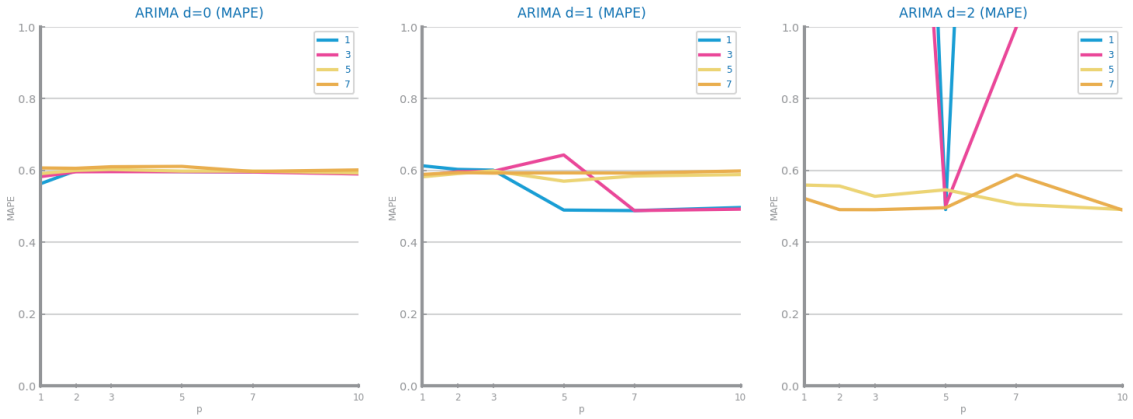


Figure 110 Forecasting study over different parameterizations of the ARIMA algorithm over time series 2

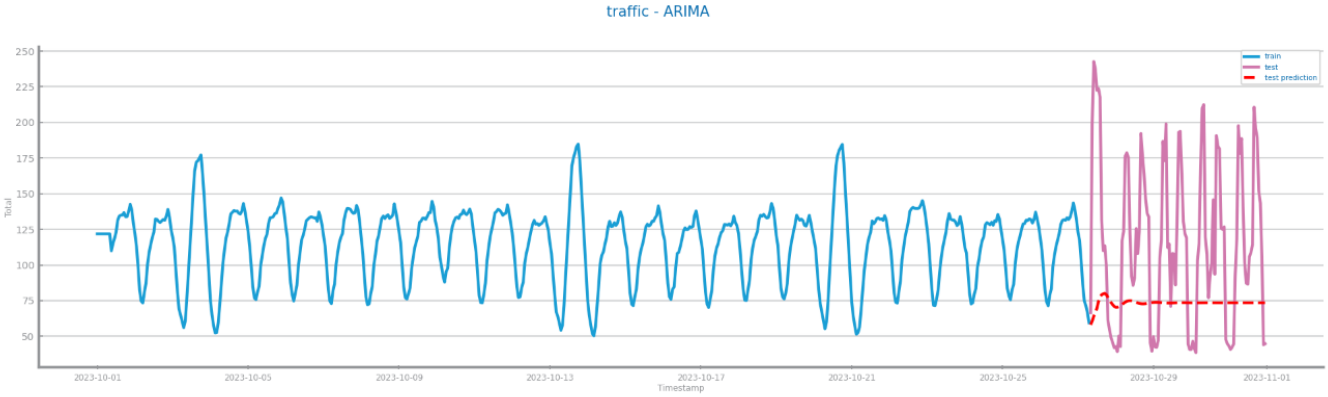


Figure 111 Forecasting plots obtained with the best parameterization of ARIMA algorithm, over time series 2

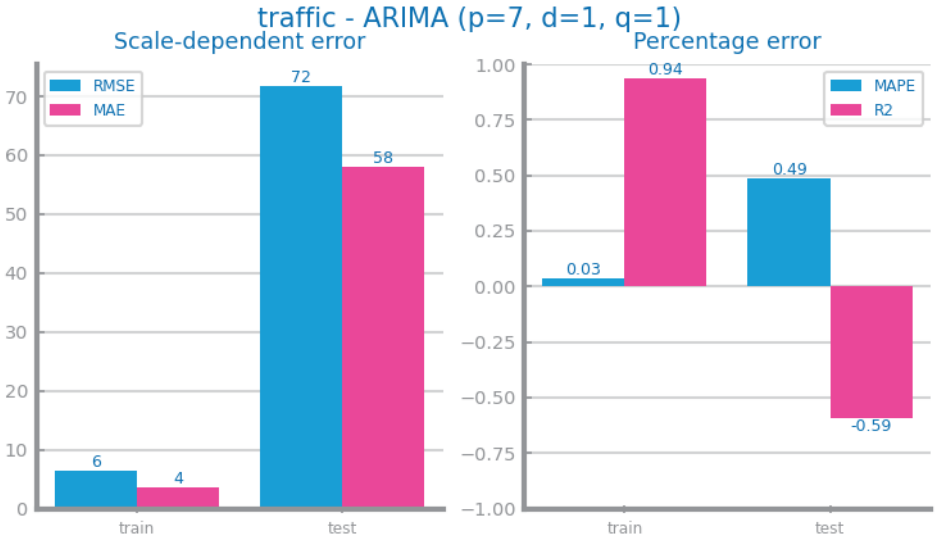


Figure 112 Forecasting results obtained with the best parameterization of ARIMA algorithm, over time series 2

LSTMs Model

LSTM is the first model to forecast S2 accurately but unexpectedly predicts 0 for variable S1, which indicates a potential coding error. A higher # of episodes leads to better results.

This is one of the most powerful RNN to do forecasting, especially when the data has long-term trends, so even though it is more effective for handling non-stationary data, the results are as expected for S2.

As the nr of episodes increases, the model's performance is better until it stabilizes, reaching its peak.

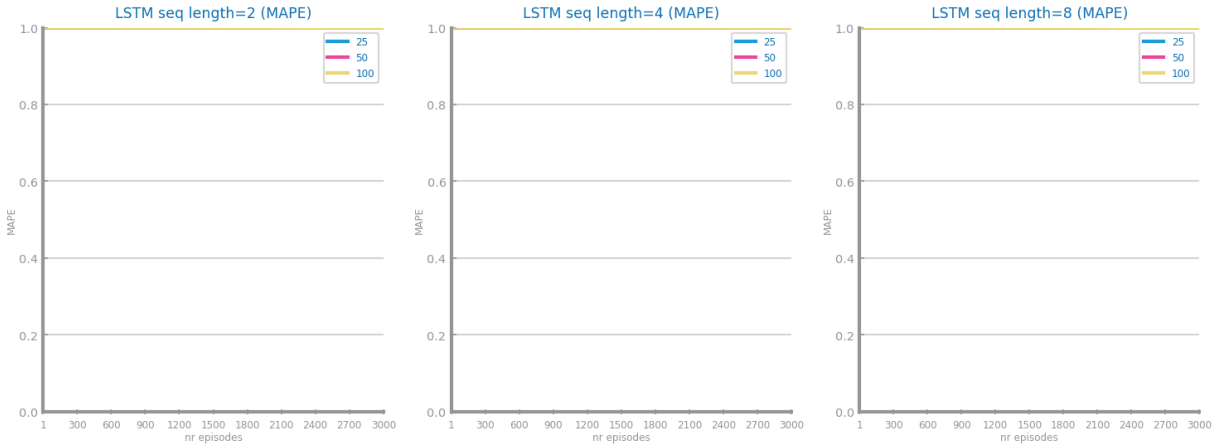


Figure 113 Forecasting study over different parameterizations of LSTMs over time series 1

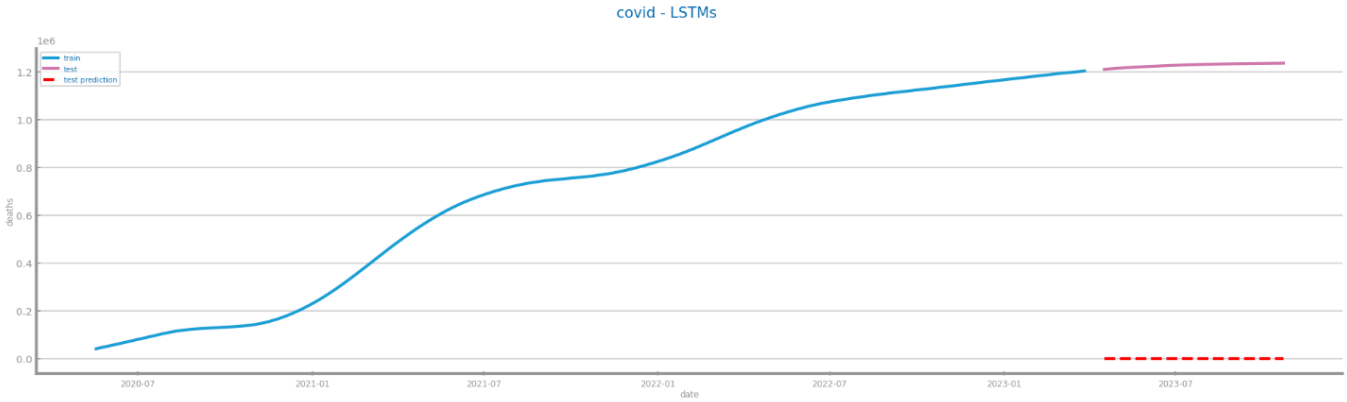


Figure 114 Forecasting plots obtained with the best parameterization of LSTMs, over time series 1

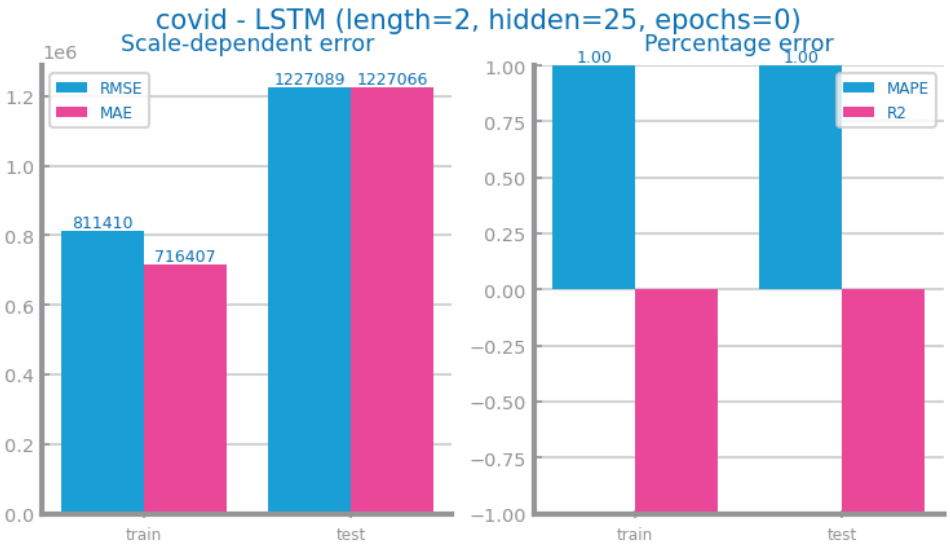


Figure 115 Forecasting results obtained with the best parameterization of LSTMs, over time series 1

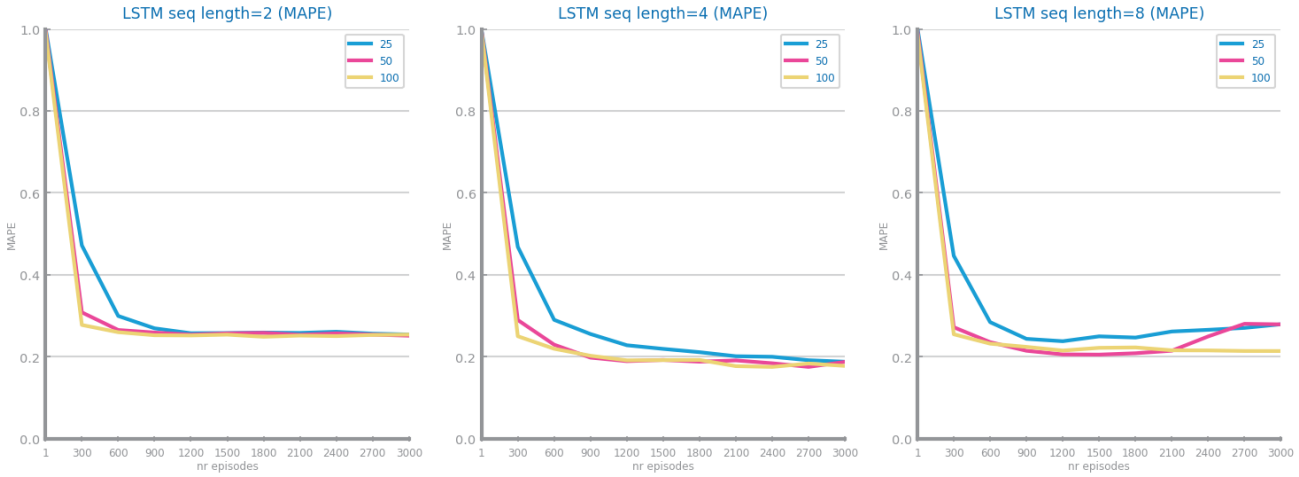


Figure 116 Forecasting study over different parameterizations of the LSTMs over time series 2

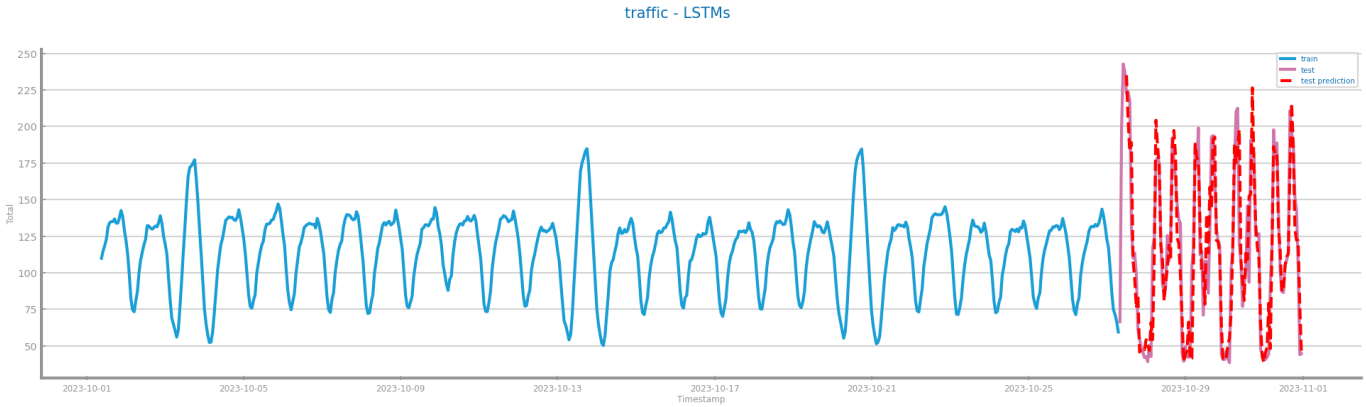


Figure 117 Forecasting plots obtained with the best parameterization of LSTMs, over time series 2

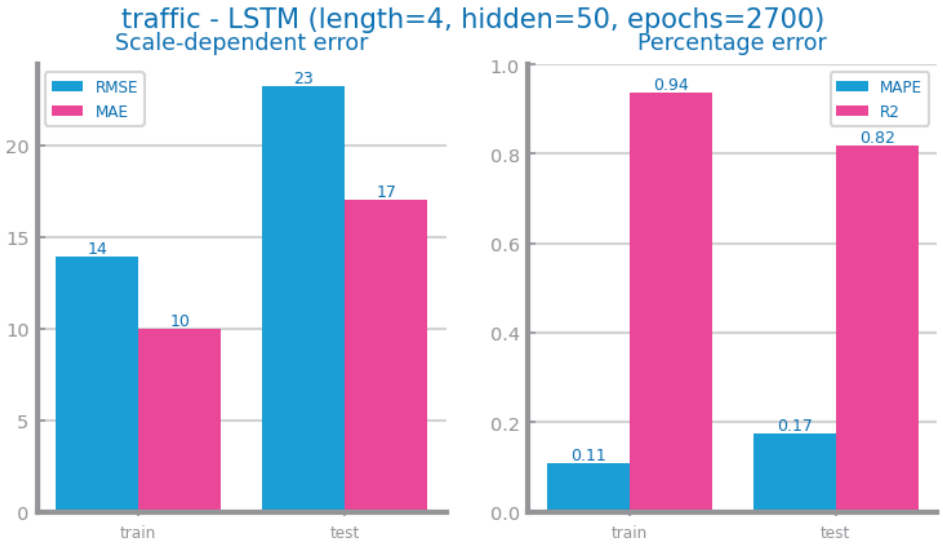


Figure 118 Forecasting results obtained with the best parameterization of LSTMs, over time series 2

8 CRITICAL ANALYSIS

S1 didn't improve much from the transformations, especially because we didn't incorporate smoothing with higher values. Whereas S2, which had worse scores initially, improved to some degree. Smoothing S2, even if it was only by 10, decreased **Linear Regression's** performance by 1%, so we shouldn't have done it.

We could've have aggregated different granularities to get even more insight into our series. For example, aggregating a granularity of 8 hours in S2 could've given us the specific hours that most people enter and leave their works.

The measure used was MAPE, since R2 (especially for S1) was not the most appropriate measure (was always negative: -100%). This indicates that predicting the mean value outperformed the model's predicted value. This decision impacted on the study plots, especially in **ARIMA**, where the variations of the parameters had little impact on the model.

The simpler models such as **Simple Average**, **Persistence Model** and **Rolling Mean** performed worse than the complex ones, which makes sense for S2.

For S1, all models, except for **LSTM**, performed well due to the simplicity of the series – it's cumulative, therefore the values only go up. In contrast, S2, which is stationary (time-dependent) and seasonal, was a hard series to predict. Only **LSTM** exhibited good results in forecasting S2 accurately, with negligible differences between models.

All forecasted values were expected, except for **LSTM**. In S1, it may be due to small data size, although it's more likely to be some unknown error, since it predicted an unreasonable value. In S2, **LSTM** was the best performing model, despite the model being more effective for non-stationary and scaled data.

For S1, the best model is **ARIMA**, with 1% **MAPE**. For S2, the best model is **LSTM**, with 17% **MAPE** (and 82% R2 score). Both best models seem to be adequate.