TÉCNICO LISBOA

# DS 2024

# Data Science Project

| | | |
|---|---|---|
| **Team nr:** 9 | **Student 1 :** Diogo Paixão | **IST nr:** 113214 |
| | **Student 2 :** Fábio Ribeiro | **IST nr:** 98933 |
| | **Student 3 :** João Teixeira | **IST nr:** 113227 |
| | **Student 4 :** Miguel Agostinho | **IST nr:** 115185 |

## CLASSIFICATION

## 1 DATA PROFILING

In dataset 1, before data profiling, in the variable AGE_GROUP we transformed some wrong values to missing values and performed the suggested transformation for the target (JURISDICTION_CODE) to binarize it (NY, nonNY).

### *Data Dimensionality*

Dataset 1: There are 4968684 records, 18 variables (9 numeric, 3 binary, 1 date and 5 symbolic), with a lot of missing values in several variables (especially 3). Some steps later on will be computationally-intensive due to the huge amount of records, so we sampled the dataset to 100000 random records after profiling, since it must have the same properties.

Dataset 2: There are 3672 records, 87 variables, (17 numeric and 1 binary). The low amount of records may present challenges later on. This dataset doesn't have missing values.

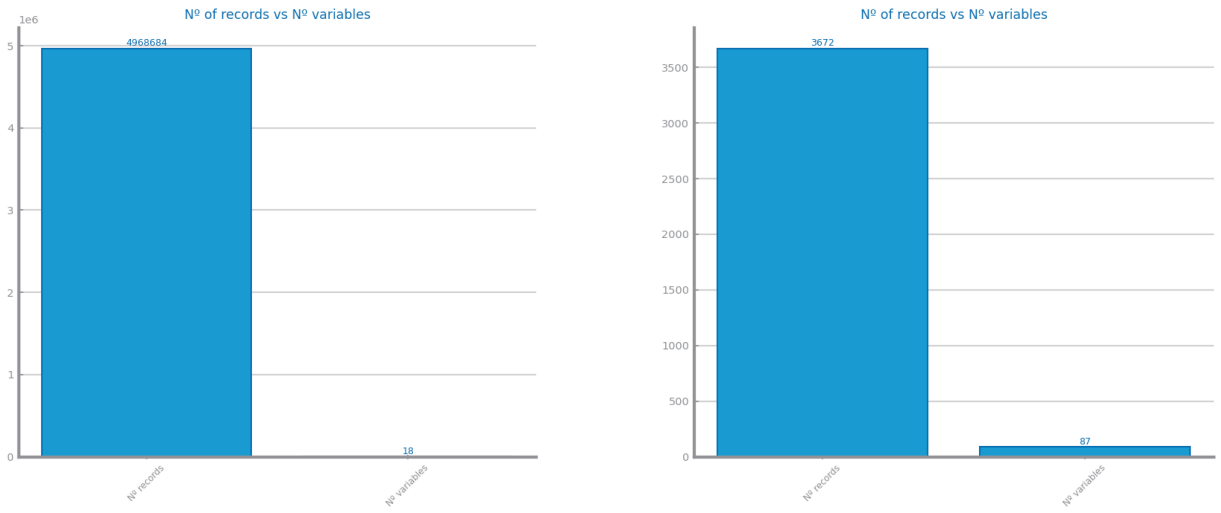



*Figure 1 Nr Records x Nr variables for dataset 1 (left) and dataset 2 (right)*

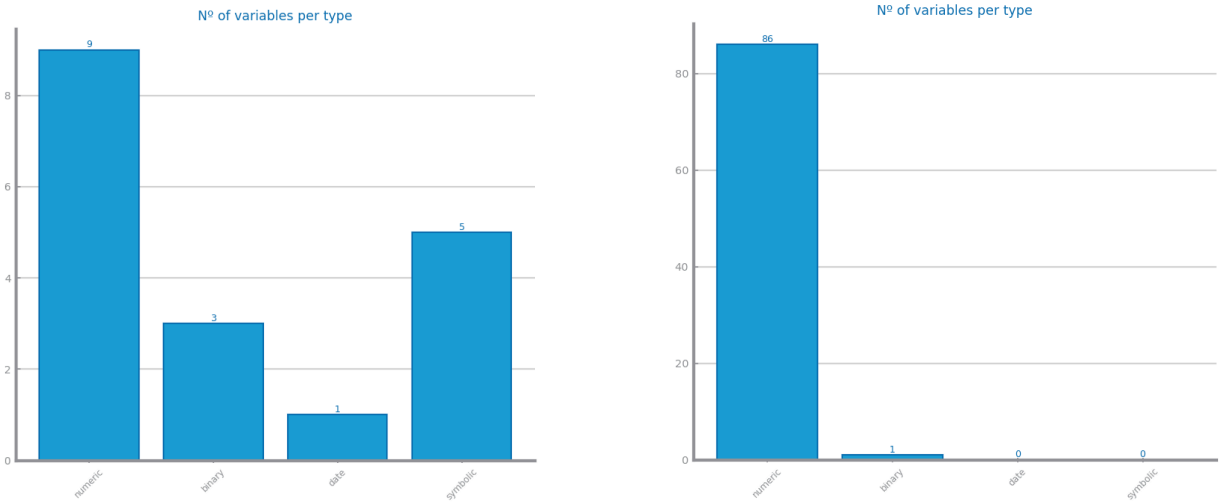



*Figure 2 Nr variables per type for dataset 1 (left) and dataset 2 (right)*

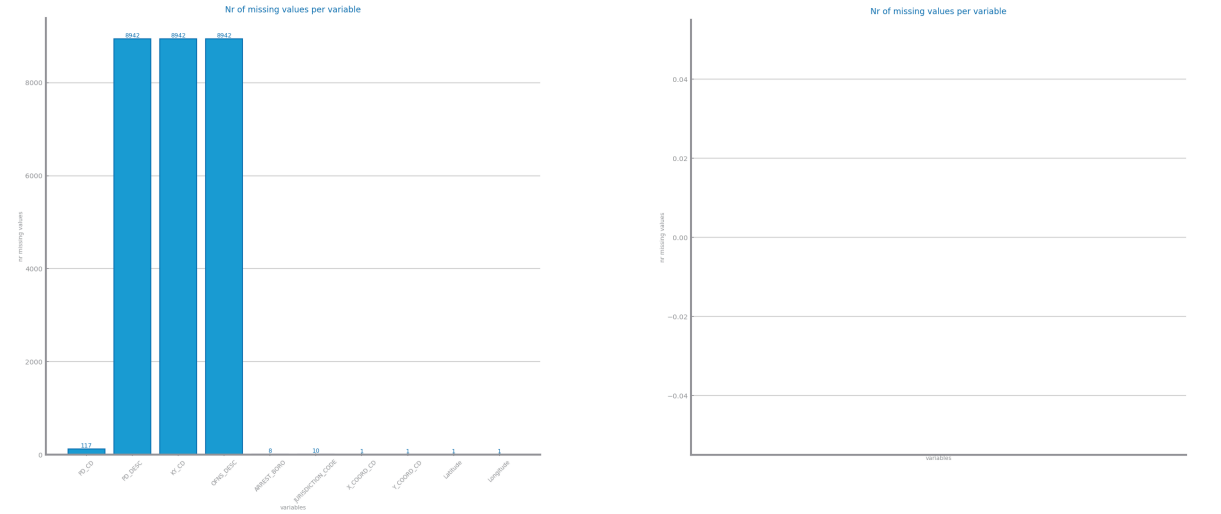



*Figure 3 Nr missing values for dataset 1 (left) and dataset 2 (right)*

### *Data Distribution*

Through Figs. 4, 5 and 6, we can observe clear outliers in some variables and different scales, in both datasets. In Figs. 7 and 8, we can see variables with all types of distributions (normal, exponential and log-normal). In Figs. 9 and 10, with nstdev=2 and IQR=1.5, we can observe a decent quantity of outliers in both datasets. Finally, in Figs. 11 and 12, we can notice a high imbalance for the target variables of both datasets, which **lead us to choose the recall as the metric to optimize and use in the next phases**, for both datasets.

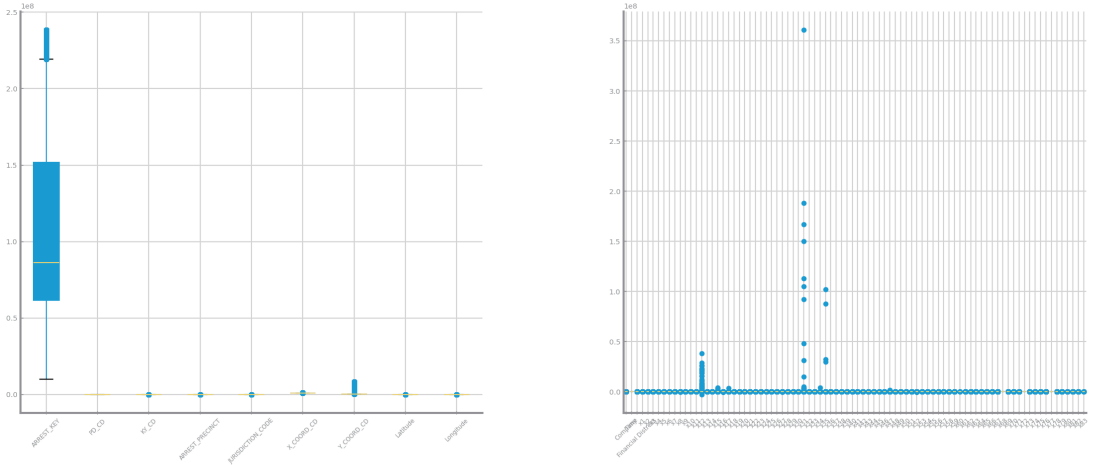


*Figure 4 Global boxplots dataset 1 (left) and dataset 2 (right)*

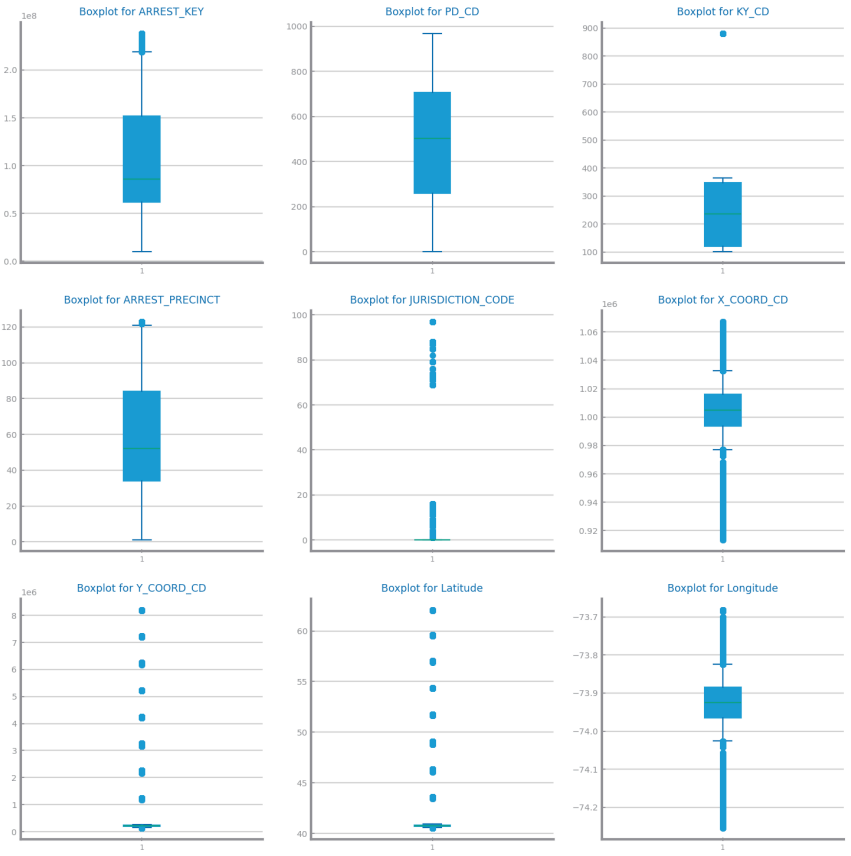



*Figure 5 Single variable boxplots for dataset 1*

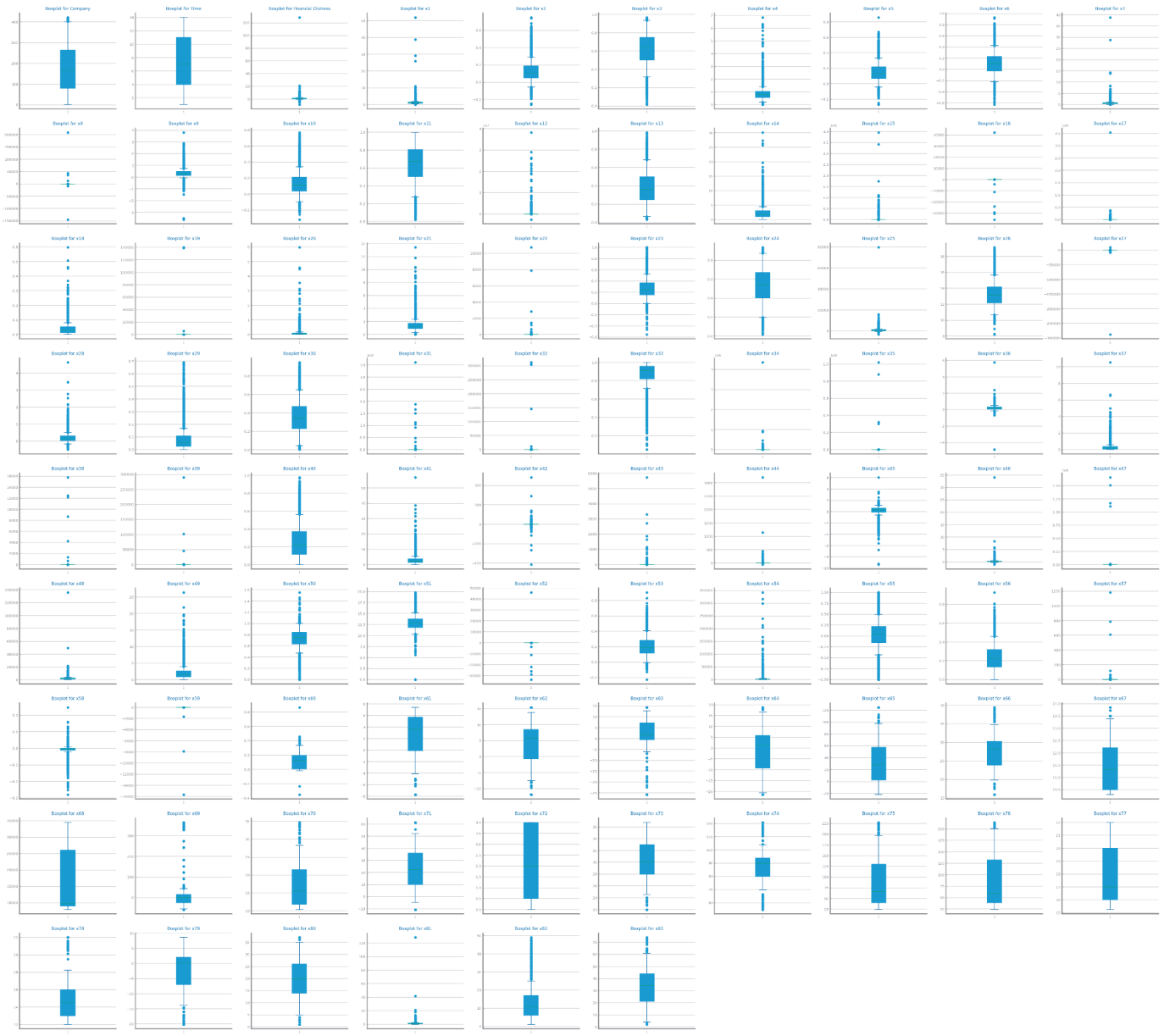


*Figure 6 Single variable boxplots s for dataset 2*

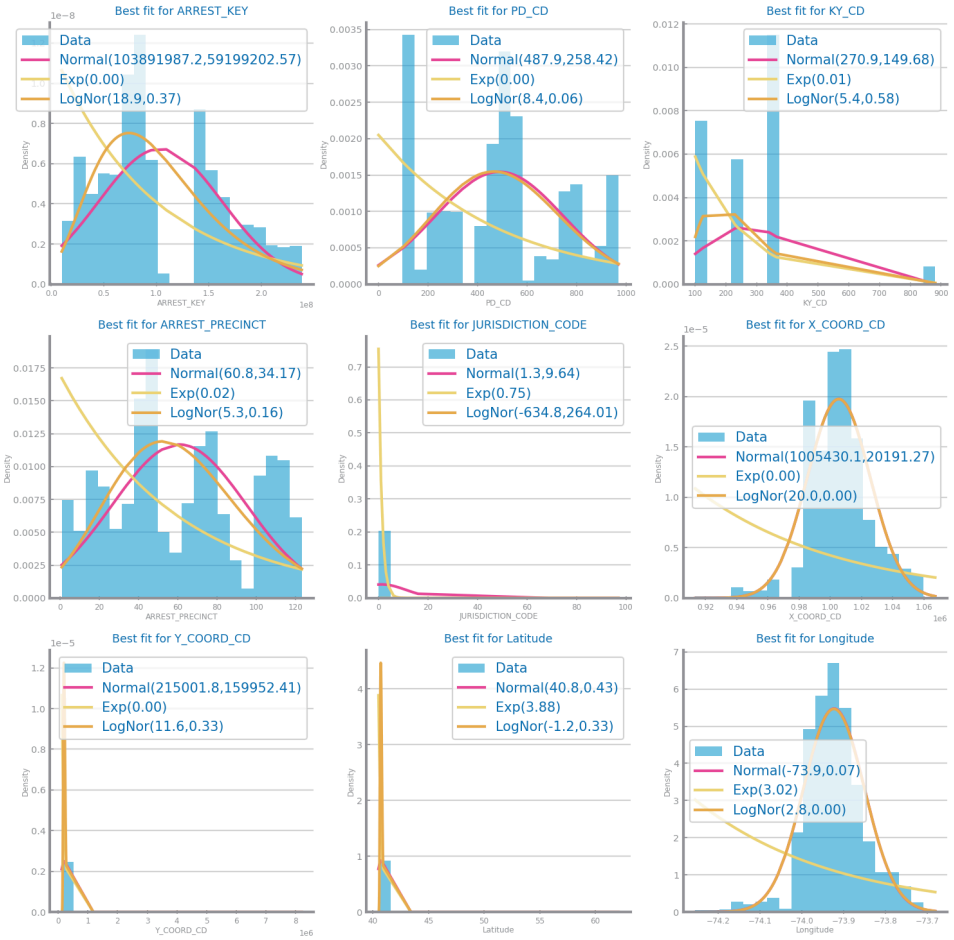



*Figure 7 Histograms for dataset 1 (with distributions is enough)*

*Figure 8 Histograms for dataset 2 (with distributions is enough)*

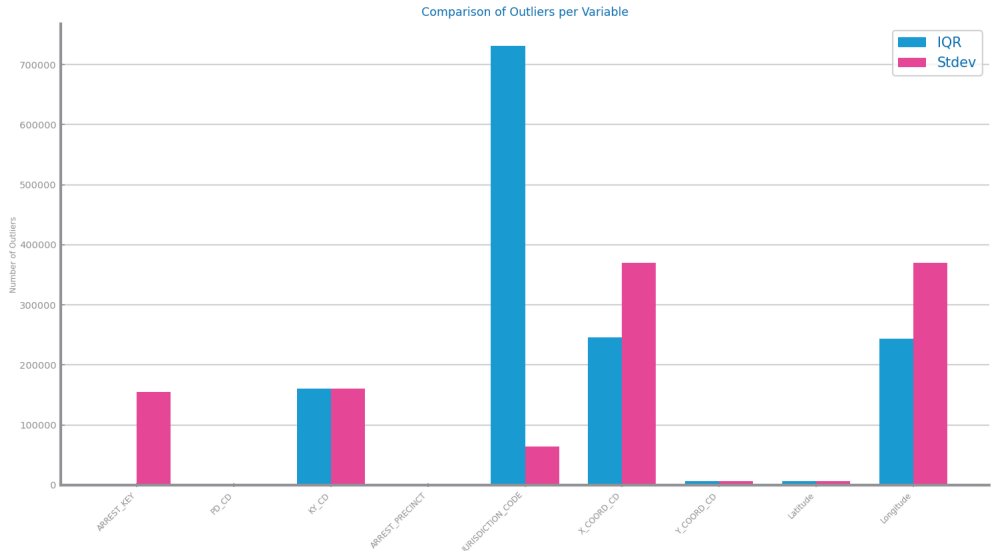



*Figure 9 Outliers study dataset 1*

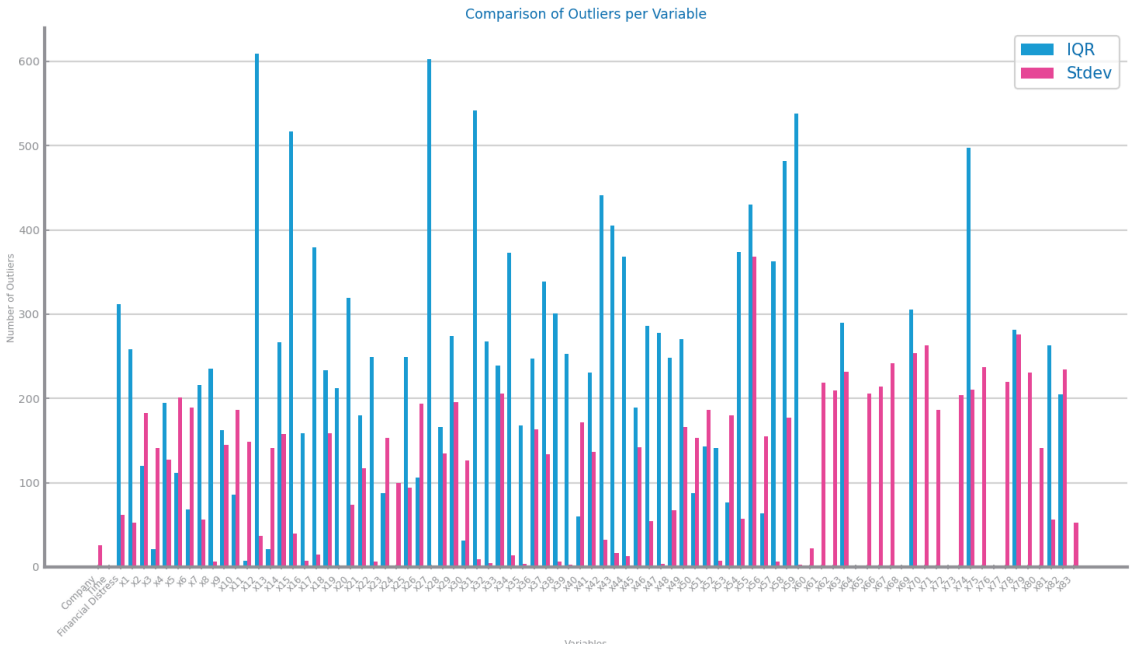



*Figure 10 Outliers study for dataset 2*

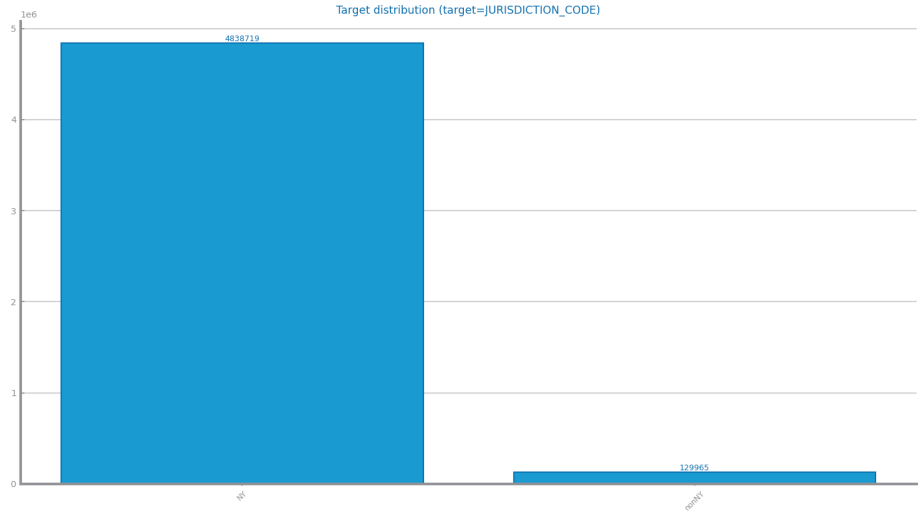



*Figure 11 Class distribution for dataset 1*

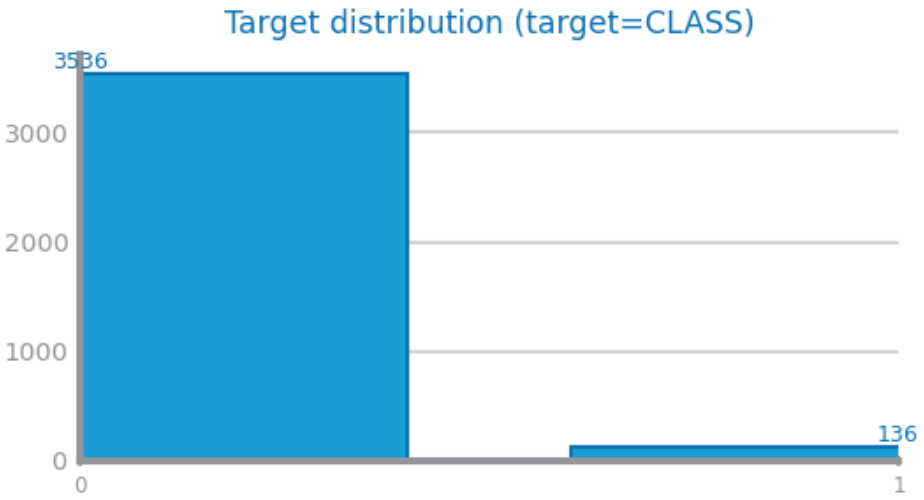
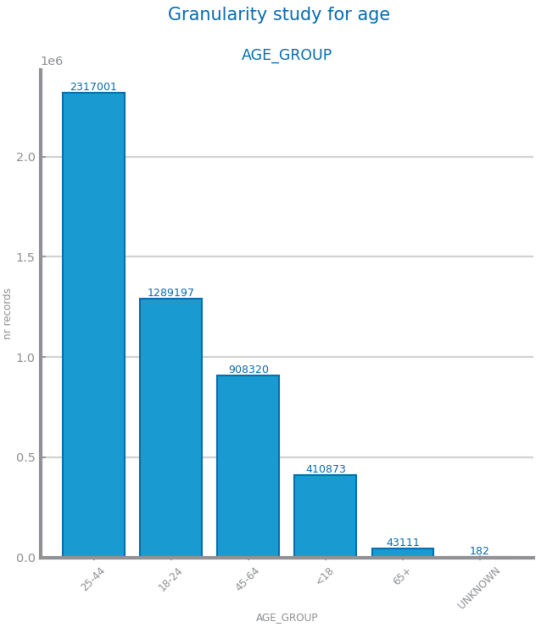


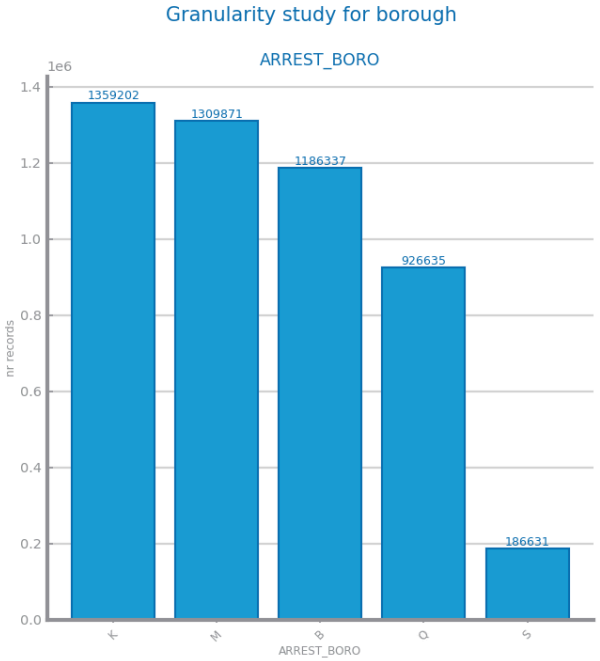

*Figure 12 Class distribution for dataset 2*

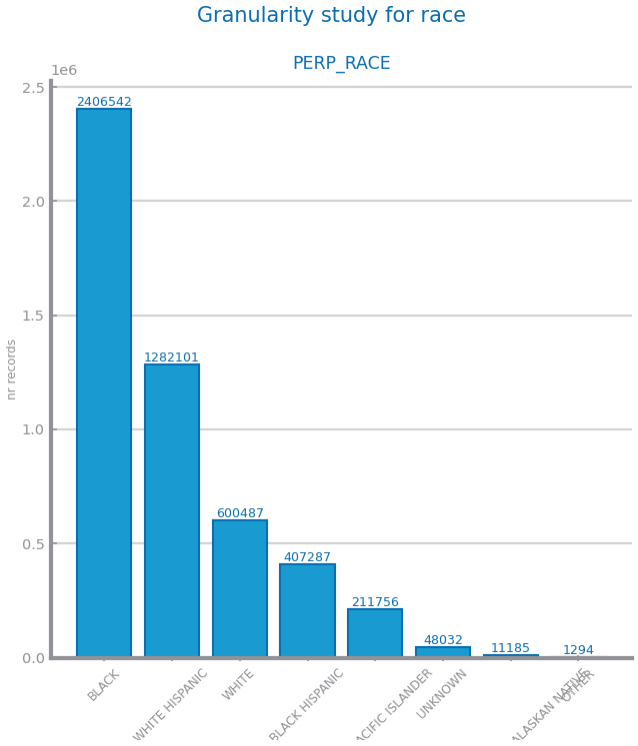
### *Data Granularity*

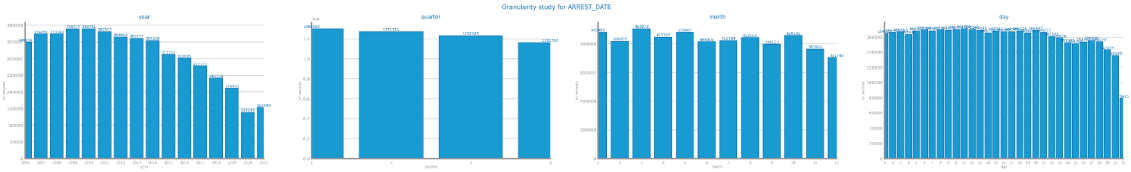
Dataset 1: We studied the granularity for age (AGE_GROUP), borough (ARREST_BORO), race (PERP_RACE), date (ARREST_DATE), law code (OFNS_DESC, PD_DESC and LAW_CODE) and location (Latitude and Longitude). From the law code granularity study, we concluded that it would be beneficial to try to reduce the high cardinality of these variables in the variable encoding phase.

Dataset 2: not applicable, since all variables are numeric.


Granularity study for age
AGE_GROUP
1e6
2317001
1289197
908320
410873
43111
182
nr records
25-44
18-24
45-64
<18
65+
UNKNOWN
AGE_GROUP


Granularity study for borough
ARREST_BORO
1e6
1359202
1309871
1186337
926635
186631
nr records
K
M
B
Q
S
ARREST_BORO


Granularity study for race
PERP_RACE
1e6
2406542
1282101
600487
407287
211756
48032
11185
1294
nr records
BLACK
WHITE HISPANIC
WHITE
BLACK HISPANIC
UNKNOWN
OTHER


Granularity study for ARREST_DATE

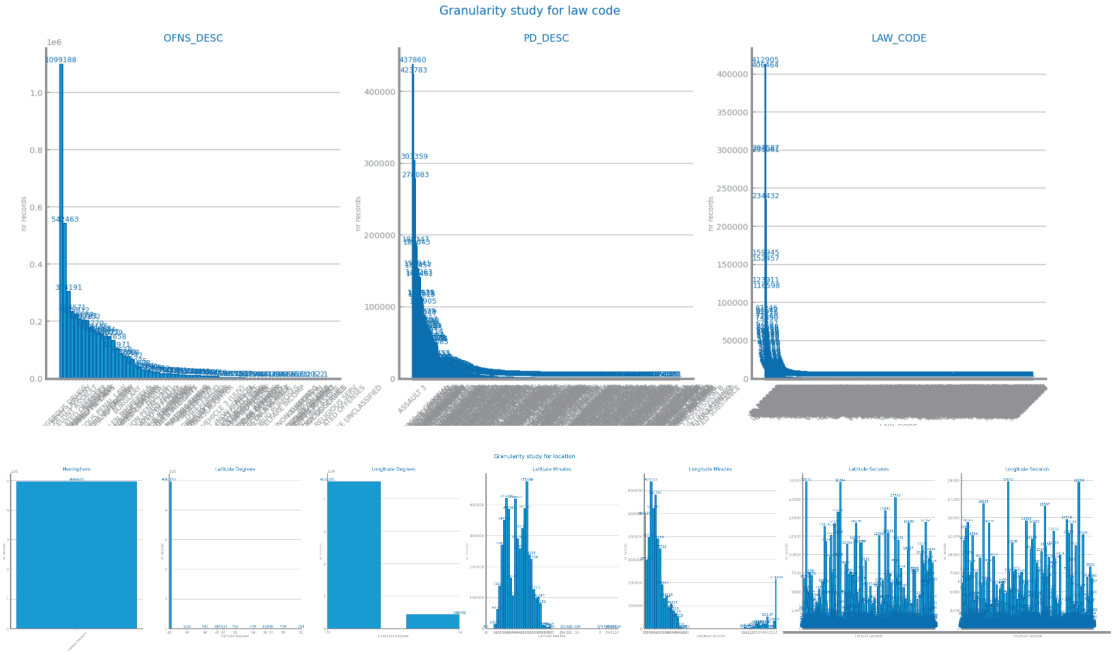



*Figure 13 Granularity analysis for dataset 1*

## *Data Sparsity*

Dataset 1: We can observe high correlations between some variables, such as Latitude/X_COORD, Longitude/Y_COORD, ARREST_BORO/ARREST_PRECINT. These correlations are expected, considering the domain knowledge.

Dataset 2: Even though we do not have any domain knowledge regarding the variables of this dataset, we can see several high correlations scattered around the dataset.

*Figure 14 Sparsity analysis for dataset 1*

class_financial distress_sparsity_study.png

*Figure 15 Sparsity analysis for dataset 2*

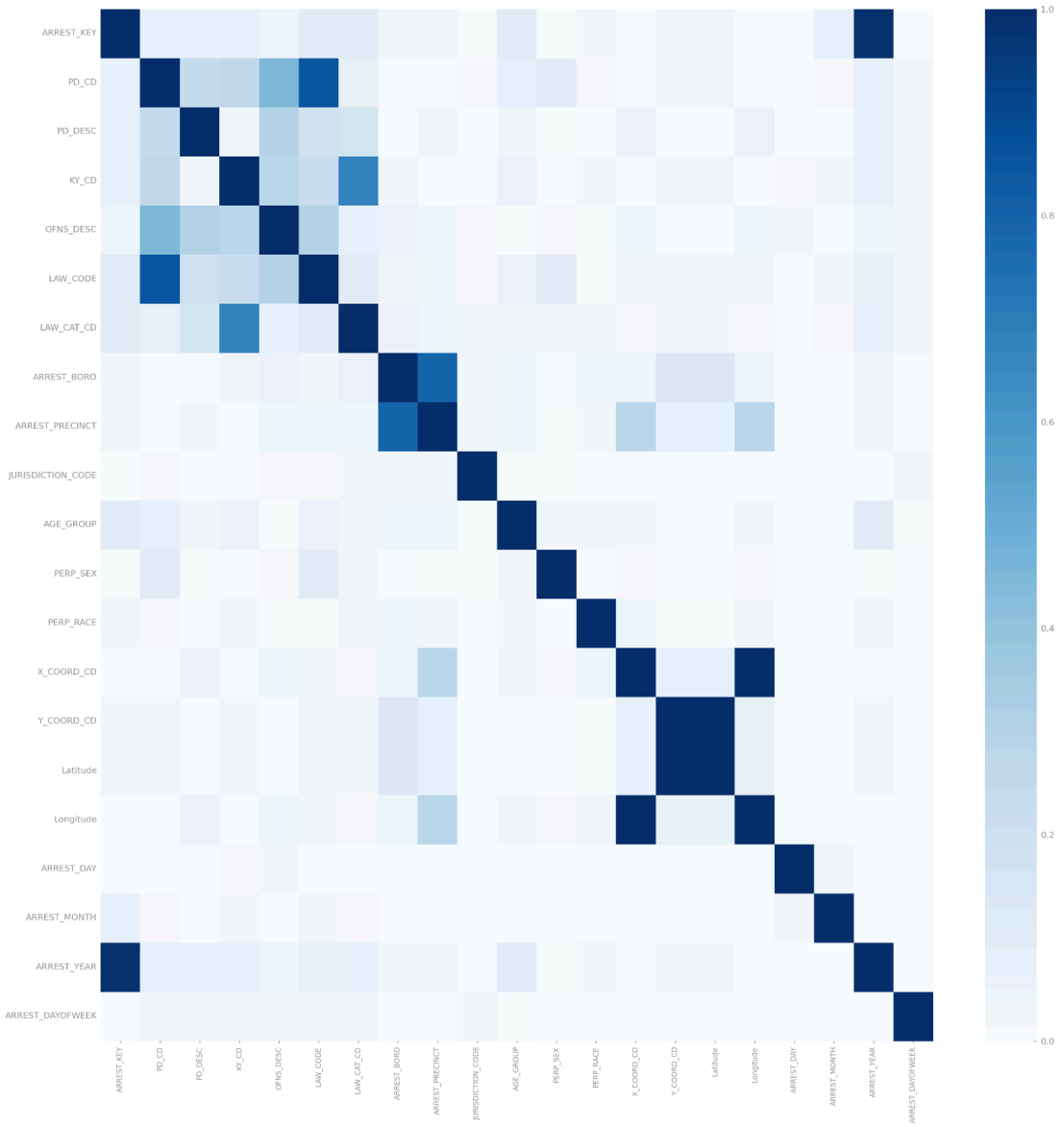



*Figure 16 Correlation analysis for dataset 1*

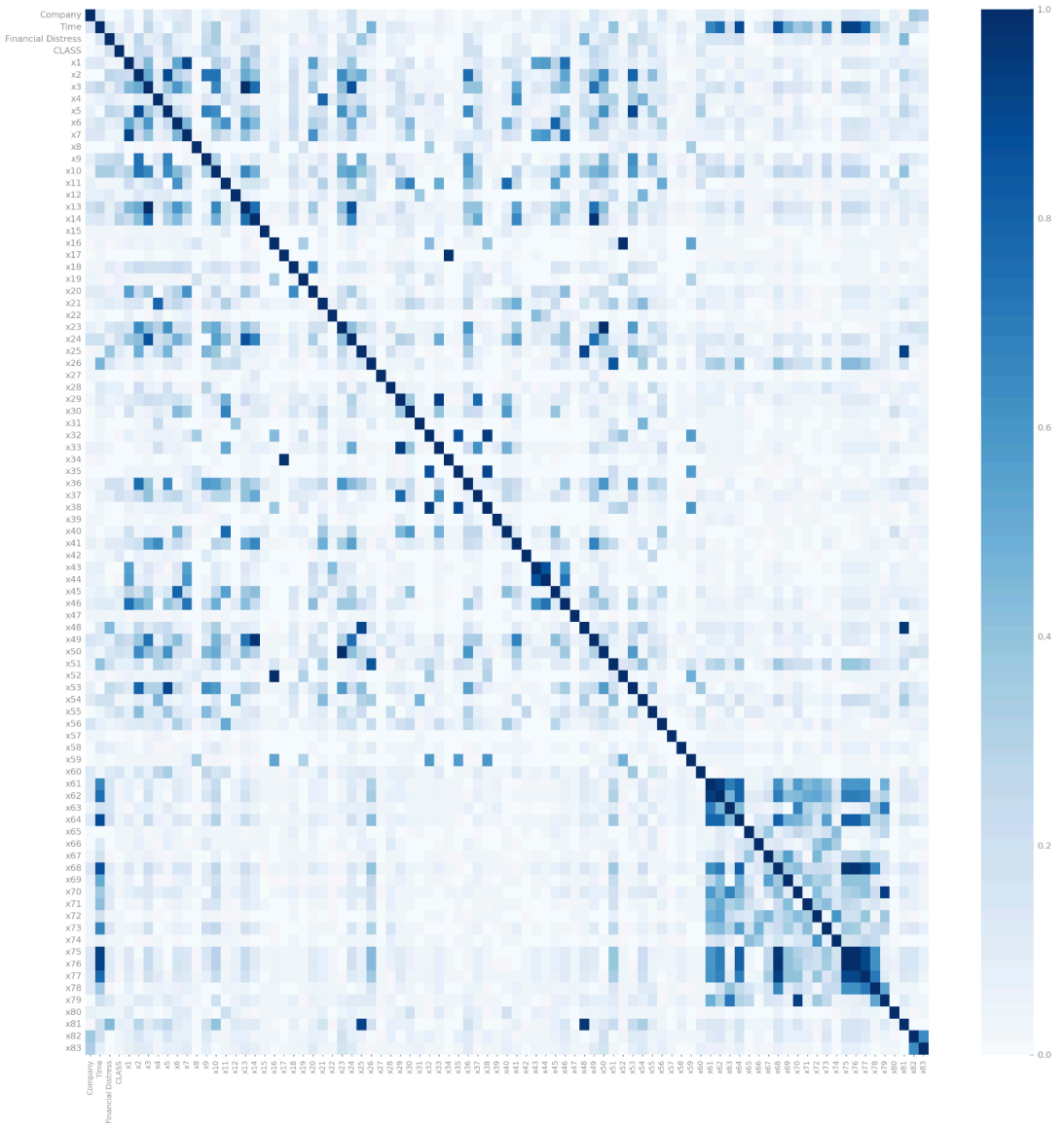



*Figure 17 Correlation analysis for dataset 2*

# 2 Data Preparation

## *Variables Encoding*

Dataset 1: AGE_GROUP, ARREST_BORO and PERP_RACE encoded to ordinal vars following Table 1. The target, JURISDICTION_CODE, encoded to binary var as suggested (NY(0)/nonNY(1)). LAW_CAT_CD and PERP_SEX encoded to binary vars. ARREST_DATE was split into ARREST_DAY, ARREST_MONTH, ARREST_QUARTER, ARREST_YEAR, ARREST_DAYOFWEEK, where ARREST_DAYOFWEEK and ARREST_QUARTER were encoded

as cyclic vars. PD_DESC and OFNS_DESC were tokenized and the top 20 words of each variable were dummified. LAW_CODE encoded to integer (example in Table 2).

Dataset 2: Variable encoding was not performed for this dataset because all variables from this dataset are already numeric.

| AGE_GROUP | | ARREST_BORO | | PERP_RACE | |
|---|---|---|---|---|---|
| <18 | 1 | M | 1 | BLACK | 1 |
| 18-24 | 2 | B | 2 | BLACK HISPANIC | 2 |
| 25-44 | 3 | Q | 3 | ASIAN / PACIFIC ISLANDER | 3 |
| 45-64 | 4 | K | 4 | AMERICAN INDIAN/ALASKAN NATIVE | 4 |
| 65+ | 5 | S | 5 | WHITE HISPANIC | 5 |
| | | | | WHITE | 6 |
| | | | | OTHER | 7 |

*Table 1 Ordinal variables encoding*

| LAW_CODE (original) | Prefix | Suffix | Prefix_encoded | Suffix_encoded | LAW_CODE (result) |
|---|---|---|---|---|---|
| PL 2200300 | PL | 2200300 | 0 | 1 | 1 |
| TAX1112223 | TAX | 1112223 | 1 | 1 | 10001 |
| TAX2223334 | TAX | 2223334 | 1 | 2 | 10002 |
| AVL1001001 | AVL | 1001001 | 2 | 1 | 20001 |

*Table 2 LAW_CODE encoding example*

### *Missing Value Imputation*

For both datasets, we decided to test three different approaches: Dropping MV (Row Removal), Mean & Most frequent Imputation and Median & Most frequent Imputation.

Dataset 1: below we can see the same results for all approaches. We decided to apply the approach of **dropping missing values** to the dataset.

Dataset 2: since the second dataset does not present any missing values, there was no need to handle missing values for this dataset. All approaches present the same results as expected.

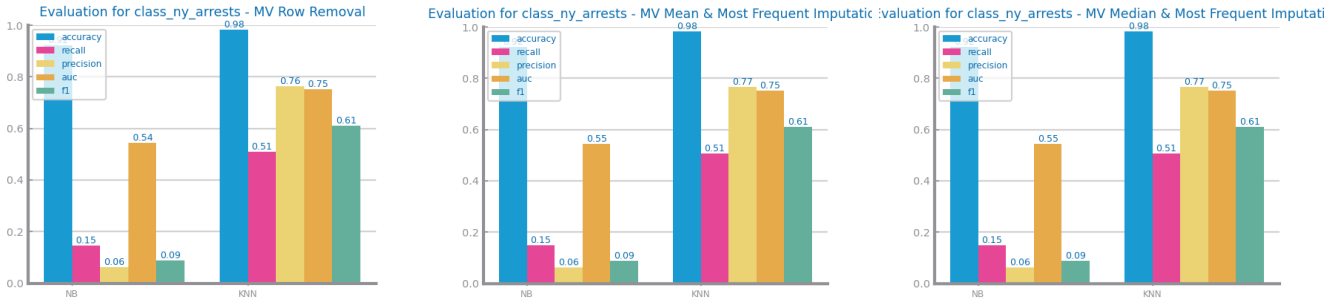



*Figure 18 Missing values imputation results with different approaches for dataset 1*

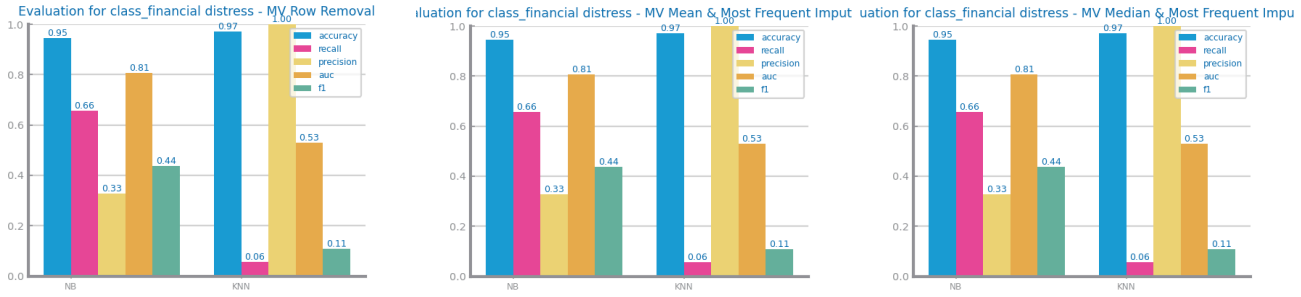



*Figure 19 Missing values imputation results with different approaches for dataset 2*

## *Outliers Treatment*

For both datasets, we decided to test three different approaches: Dropping outliers, Replacing outliers (with median) and Truncating outliers.

Dataset 1: better results for **replacing outliers** (with median), having higher overall recall between NB and KNN, so we decided to apply this approach to the **train** dataset.

Dataset 2: better results for both **replacing** (with median) and **truncating outliers** regarding recall. We decided to apply the **truncating outliers** approach to the **train** dataset since it improves precision instead.

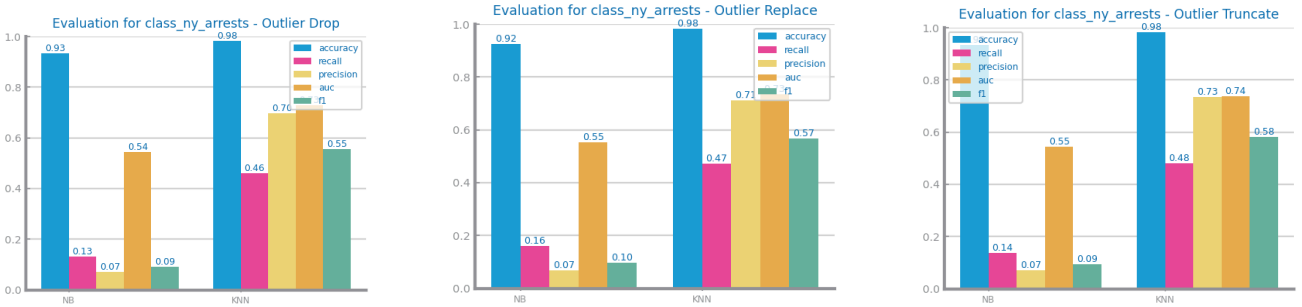



*Figure 20 Outliers imputation results with different approaches for dataset 1*

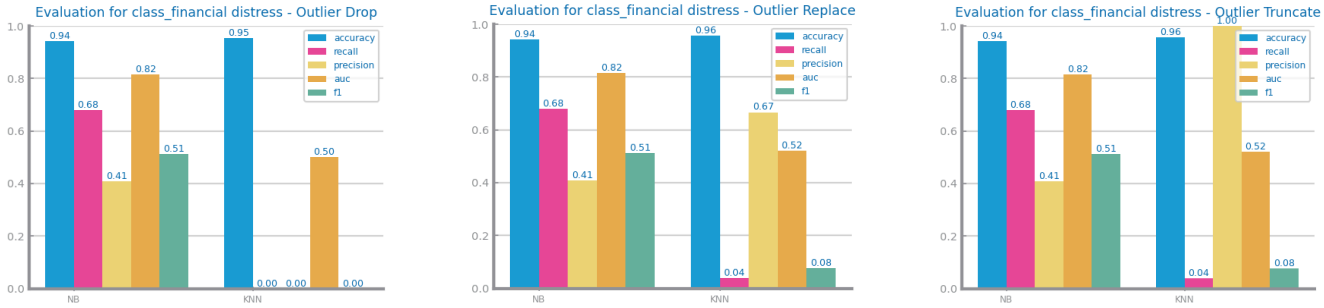



*Figure 21 Outliers imputation results with different approaches for dataset 2*

## *Scaling*

For both datasets, we decided to test two different approaches: Standard Scaling and MinMax Scaling.

Dataset 1: **we do not apply any scaling approach** for this dataset since it deteriorates the results for KNN.

Dataset 2: even though the recall remains the same, **we do not apply any scaling approach** for this dataset since it slightly deteriorates the results of accuracy and deteriorates the results of precision for KNN.

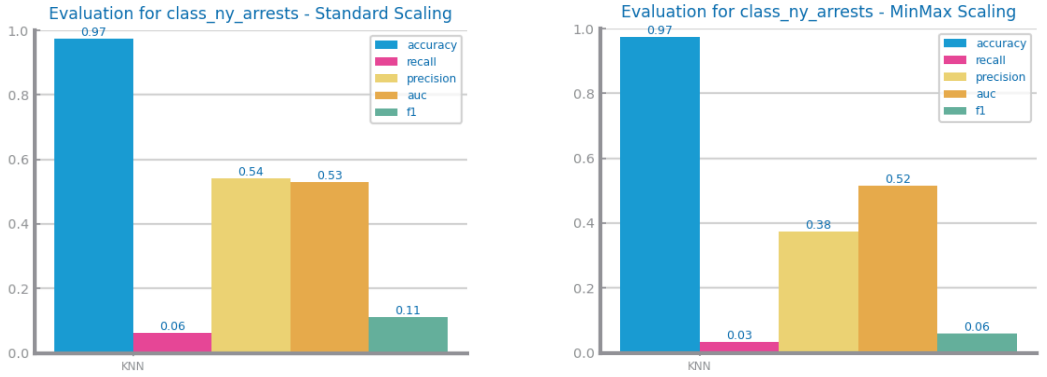



*Figure 22 Scaling results with different approaches for dataset 1*

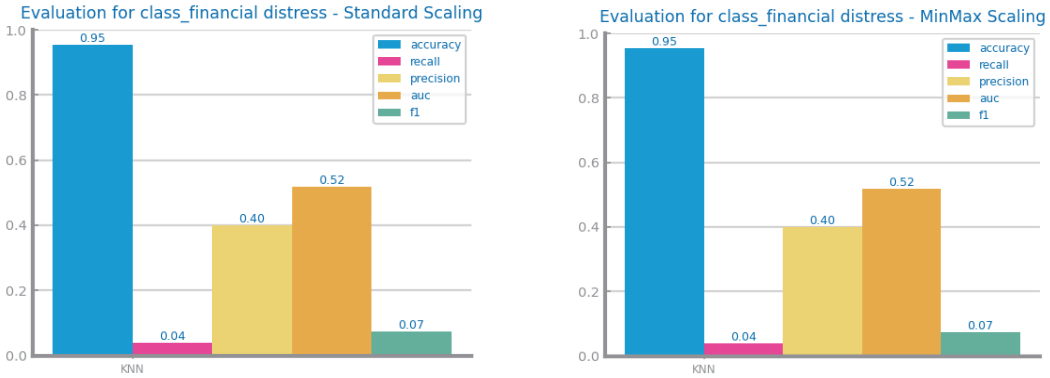



*Figure 23 Scaling results with different approaches for dataset 2*

## *Balancing*

For both datasets, we decided to test three different approaches: Undersampling, Oversampling and SMOTE.

Dataset 1: below we can see better results for **Undersampling**, having a higher overall recall for both NB and KNN, so we decided to apply this approach to the **train** dataset.

Dataset 2: below we can see better results for **Undersampling**, having a higher overall recall (especially for KNN), so we decided to apply this approach to the **train** dataset.

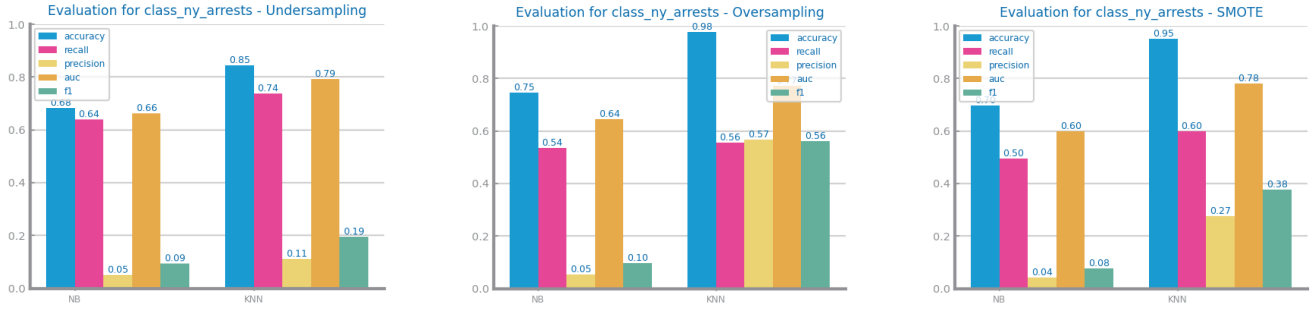



*Figure 24 Balancing results with different approaches for dataset 1*

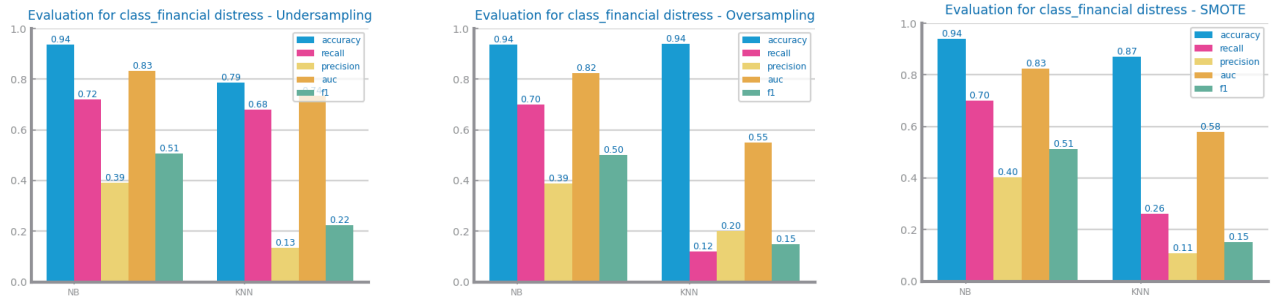



*Figure 25 Balancing results with different approaches for dataset 2*

## *Feature Selection*

Dataset 1: In the redundancy study, there is a slight improvement for NB for a **threshold of 0.5** without changes for KNN, so we apply it. In the variance study, we **decided not to filter out any variable based on variance**.

Dataset 2: In the redundancy study, the best results happen by **filtering redundant variables using a correlation threshold of 0.35 or 0.85**, so we apply **the 0.35 threshold**. In the variance study, we **decided not to filter out any variable based on variance**.

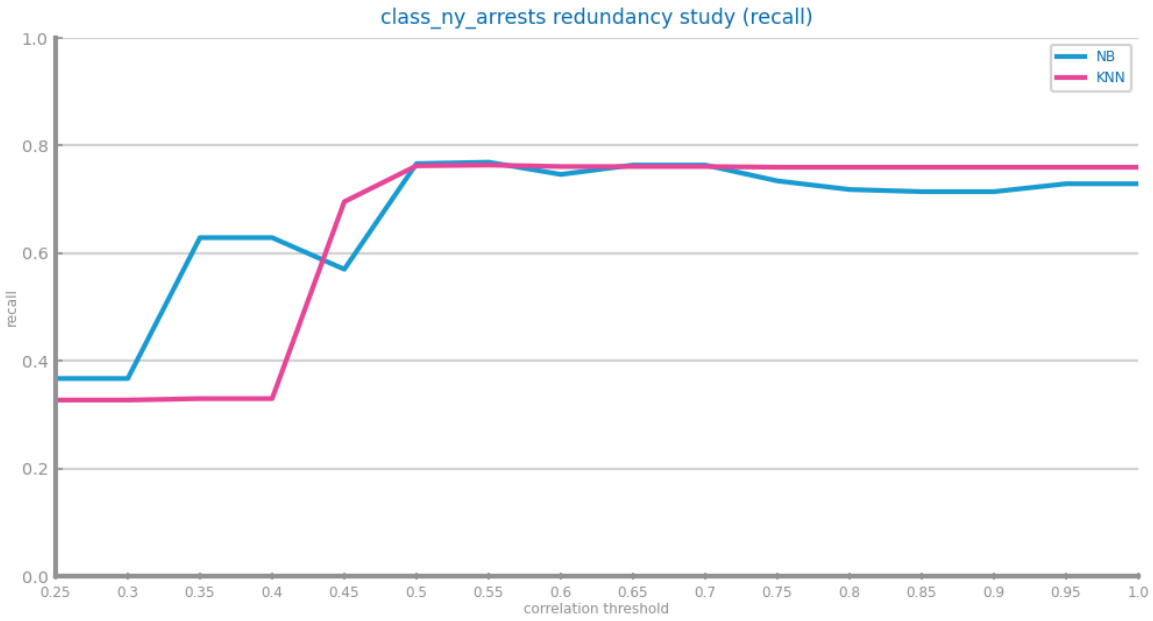



*Figure 26 Feature selection of redundant variables results with different parameters for dataset 1*

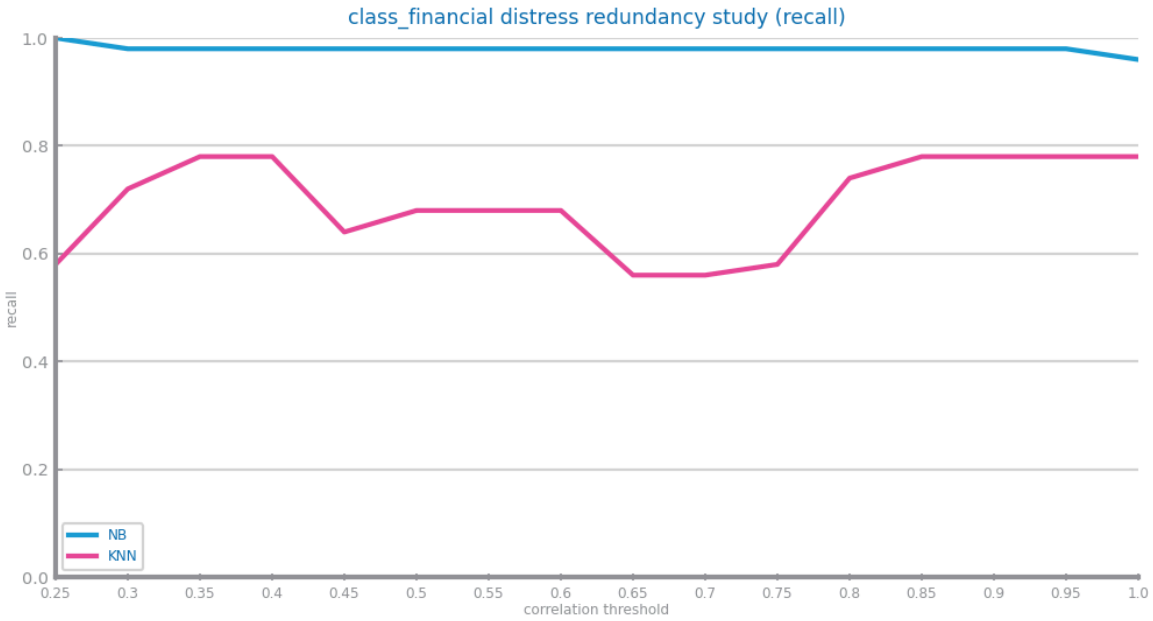



*Figure 27 Feature selection of redundant variables results with different parameters for dataset 2*

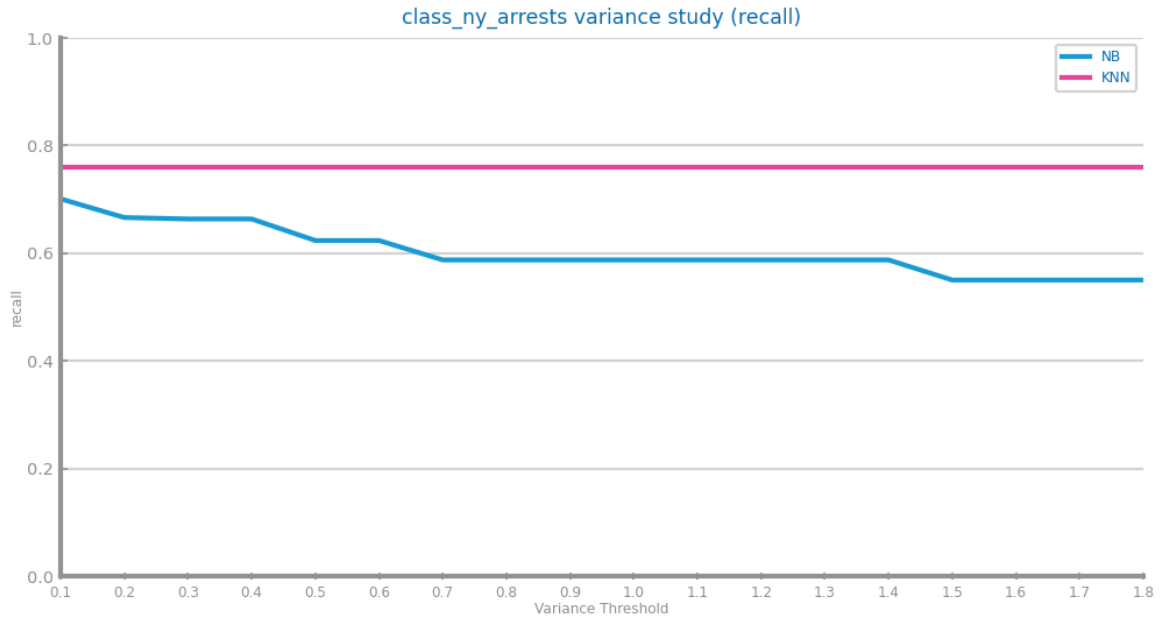



*Figure 28 Feature selection of relevant variables results with different parameters for dataset 1 (variance study)*

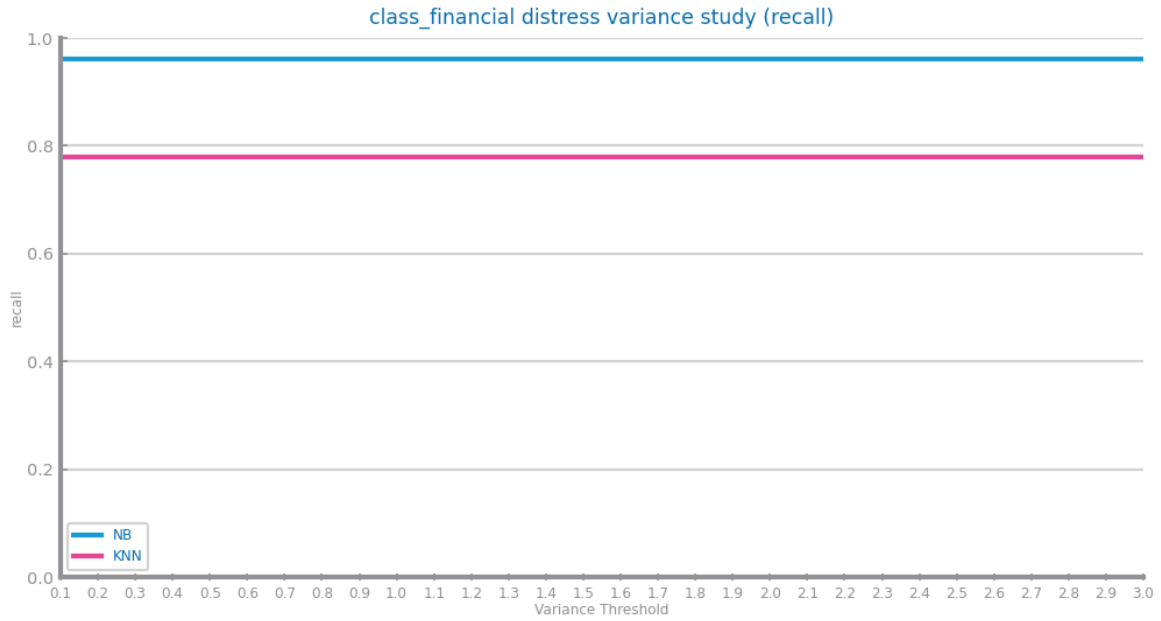



*Figure 29 Feature selection of relevant variables results with different parameters for dataset 2 (variance study)*

### *Additional Feature Generation*

All the feature generation we considered necessary was done in the variable encoding section.

## 3 Models' Evaluation

The metric chosen to use, for both datasets, was the **recall** as previously explained in the section **Data Distribution**. We trained each model 10 times, for both datasets, to get the confidence intervals that contextualize our results, which are shown in a chart in each model's section. Each run for dataset 1 used different samples of 100000 records and

different train-test-splits. Each run for dataset 2 used the whole dataset (due to the small size) but different train-test-splits. We will only show the best model for each model/dataset.

### *Naïve Bayes*

Multinomial alternative not used due to the presence of negative values in both datasets.

Dataset 1: best results for Gaussian with mean=0,69 and stdev=0,08.

Dataset 2: Best results for Gaussian with mean=0,79 and stdev=0,18. The high stdevs may be due to the high variation between gaussian and bernoulli choice between runs.

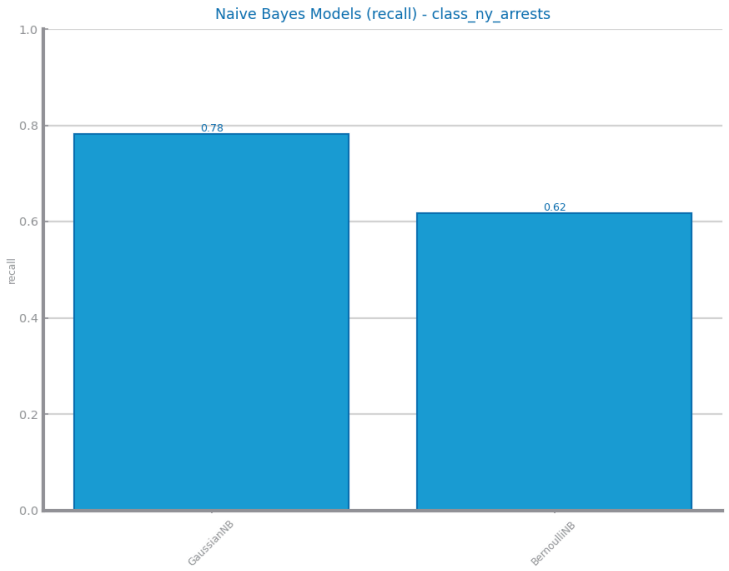



*Figure 30 Naïve Bayes alternatives comparison for dataset 1*

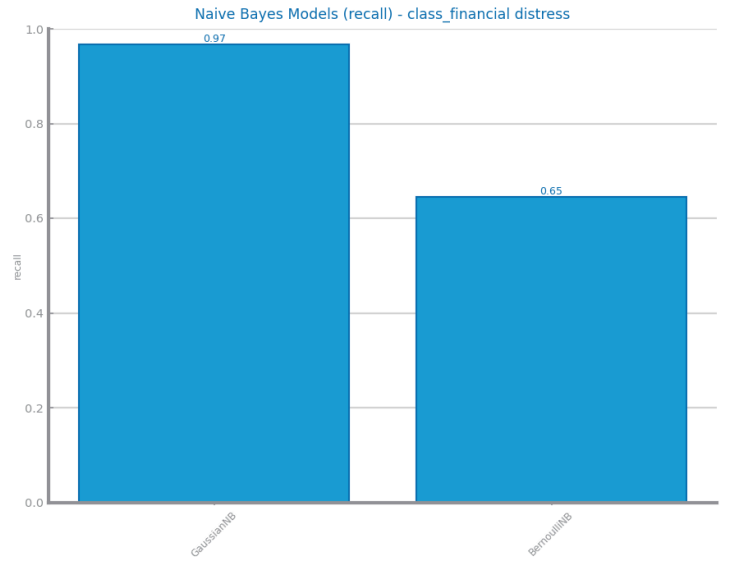



*Figure 31 Naïve Bayes alternative comparison for dataset 2*

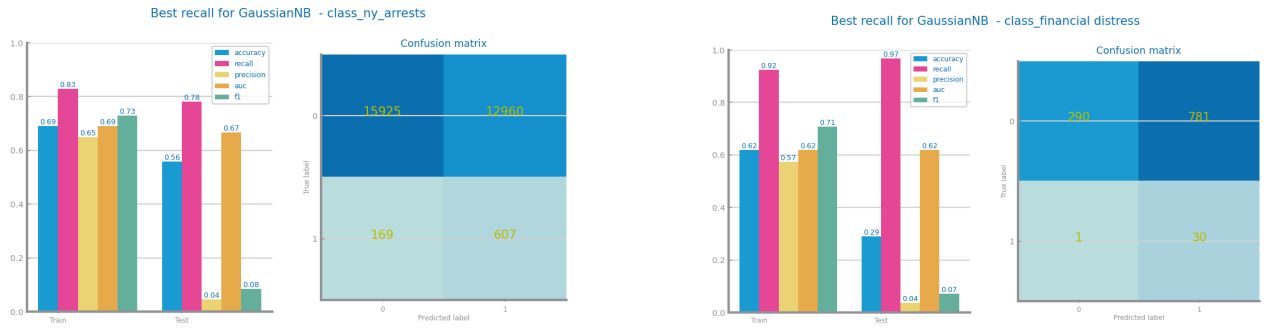



*Figure 32 Naïve Bayes best model results for dataset 1 (left) and dataset 2 (right)*

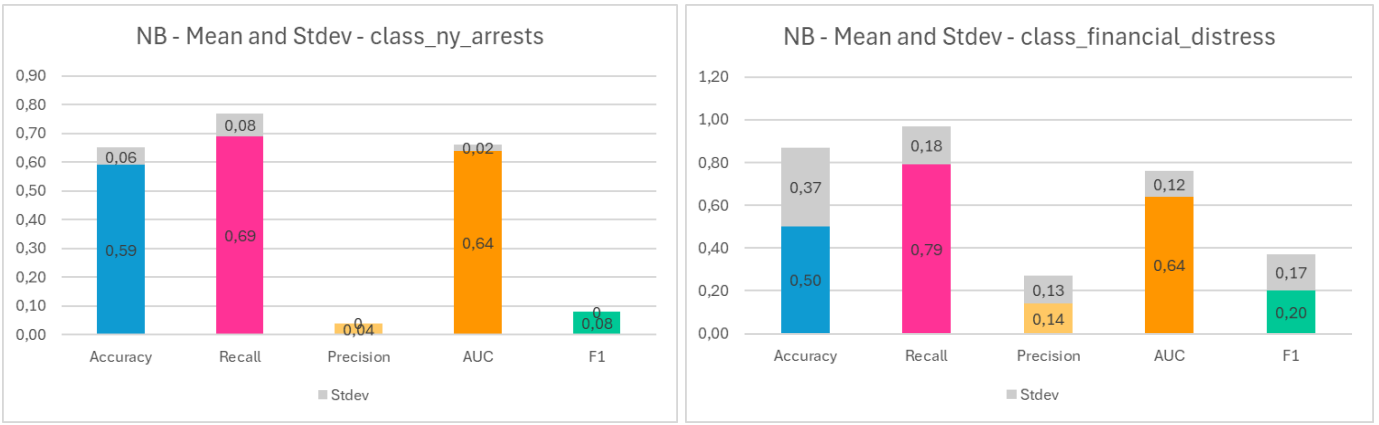



*Figure 33 Naïve Bayes confidence intervals for dataset 1 (left) and dataset 2 (right)*

## *KNN*

Dataset 1: Best results for manhattan and k=25, with mean=0,78 and stdev=0,01. This model does not show overfitting.

Dataset 2: Best results for manhattan and k=21, with mean=0,85 and stdev=0,07. This model does not show overfitting.

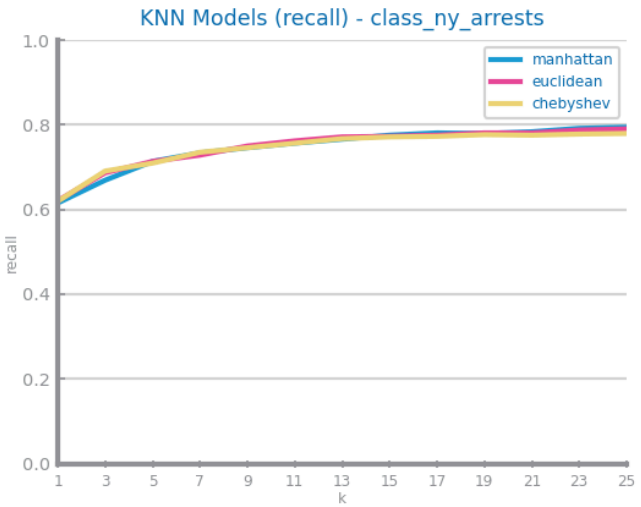



*Figure 34 KNN different parameterisations comparison for dataset 1*

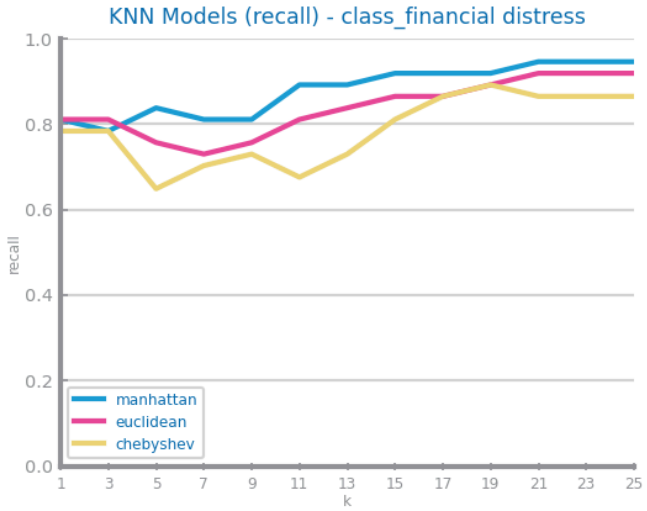



*Figure 35 KNN different parameterisations comparison for dataset 2*


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

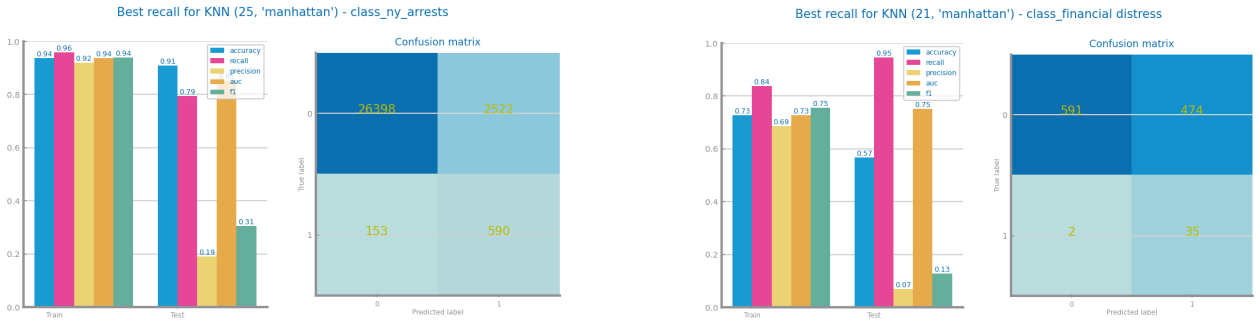



*Figure 37 KNN best model results for dataset 1 (left) and dataset 2 (right)*

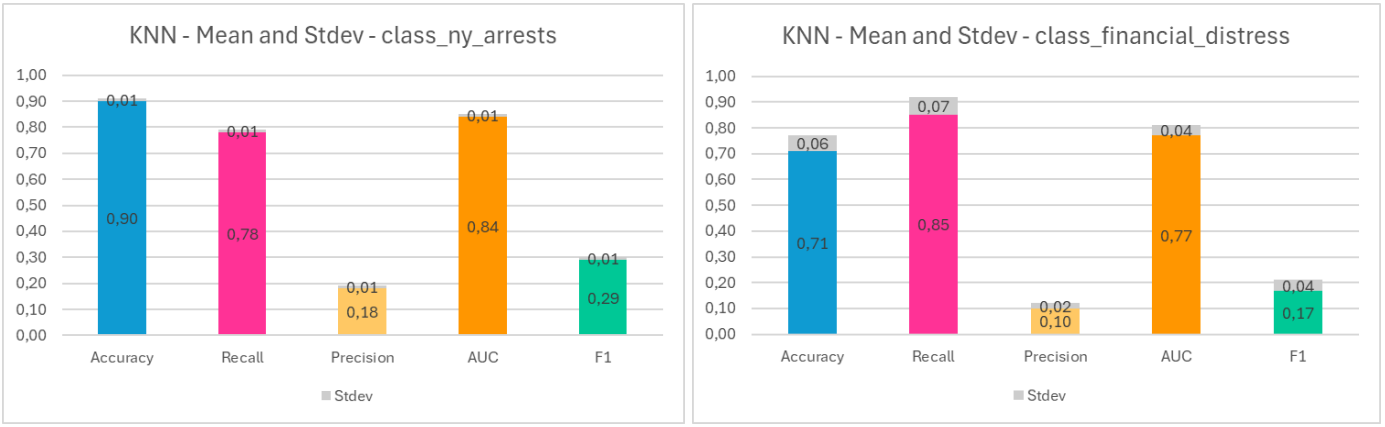



*Figure 38 KNN confidence intervals for dataset 1 (left) and dataset 2 (right)*

## *Decision Trees*

Dataset 1: Best results for entropy and d=2, with mean=0,70 and stdev=0,21. Even though there is a very small decrease after d=24, we would need another level of depth to be sure, so we assume this model does not show overfitting.

Dataset 2: Best results for entropy and d=2, with mean=0,90 and stdev=0,07. This model shows overfitting after max_depth=6 since the train performance is improving but the test is decreasing.

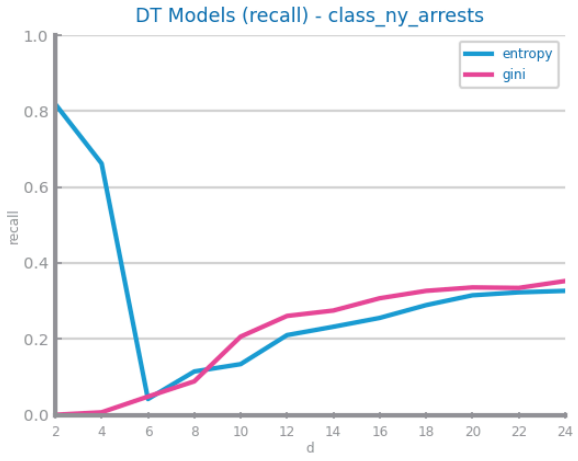



*Figure 39 Decision Trees different parameterisations comparison for dataset 1*

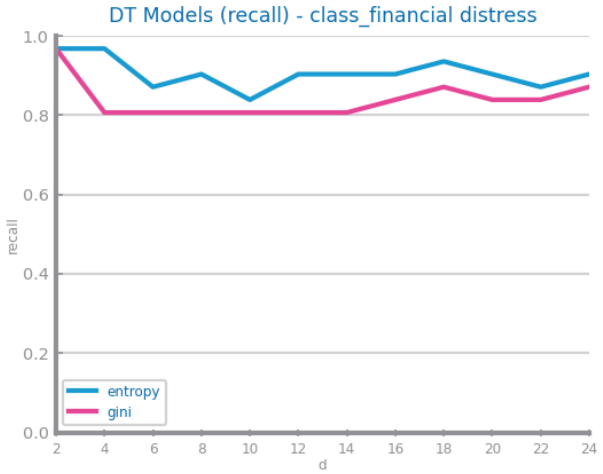



*Figure 40 Decision Trees different parameterisations comparison for dataset 2*

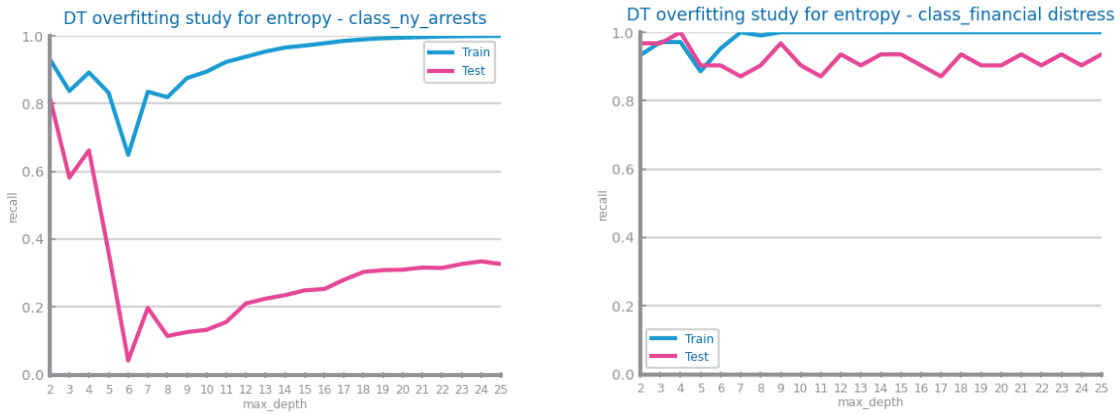



*Figure 41 Decision Trees overfitting analysis for dataset 1 (left) and dataset 2 (right)*

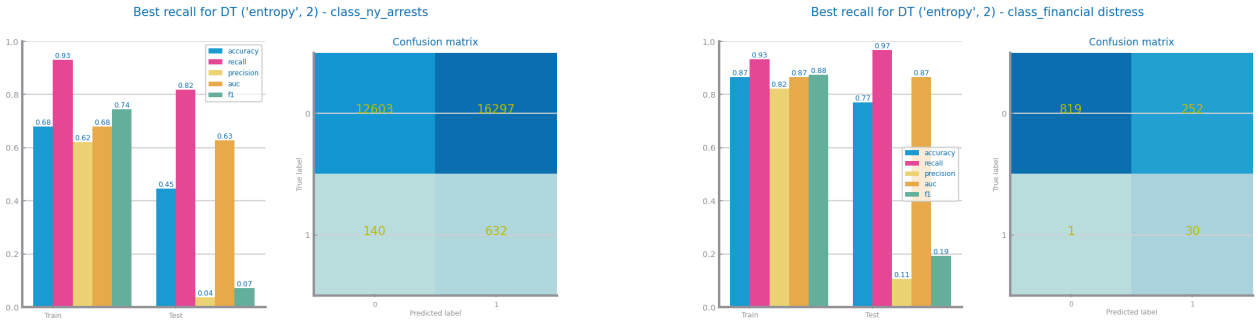



*Figure 42 Decision trees best model results for dataset 1 (left) and dataset 2 (right)*

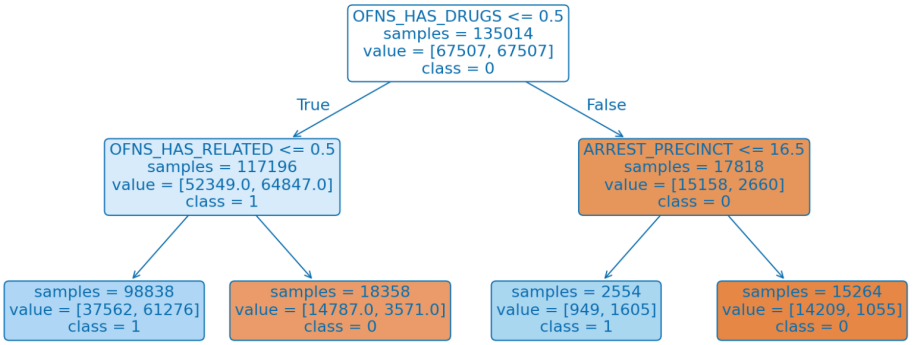



*Figure 43 Best tree for dataset 1*

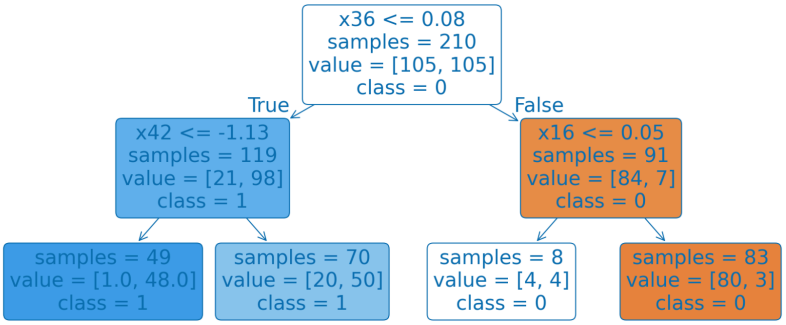



*Figure 44 Best trees for dataset 2*


*Figure 45 Decision Trees confidence intervals for dataset 1 (left) and dataset 2 (right)*

## *Random Forests*

Dataset 1: best results for d=2 and f=0.9, with mean=0,64 and stdev=0,11. This model does not show overfitting. The most important variables in the model are OFNS_HAS_DRUGS, OFNS_HAS_RELATED, ARREST_DAYOFWEEK_COS, ARREST_PRECINT and AGE_GROUP.

Dataset 2: best results for d=2 and f=0.7, with mean=0,90 and stdev=0,07. This model does not show overfitting. The most important variables in the model are x36, x12, x46, x8 and x10.

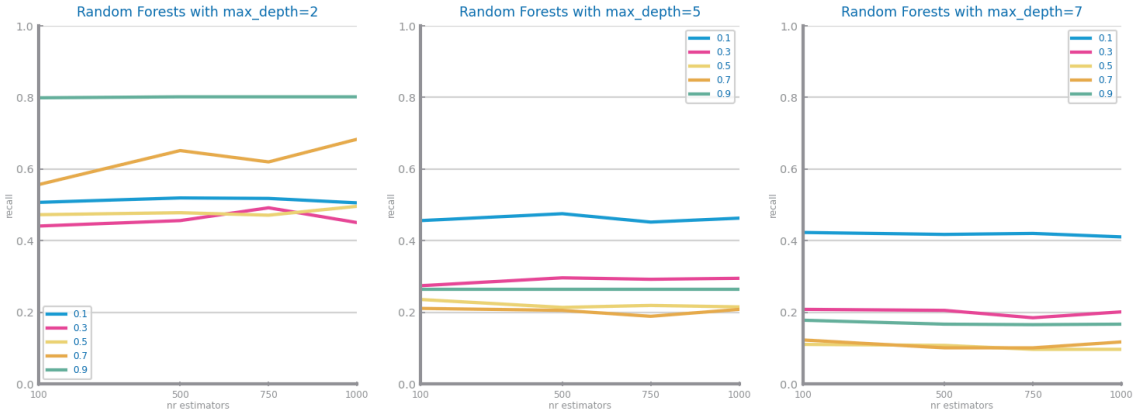



*Figure 46 Random Forests different parameterisations comparison for dataset 1*

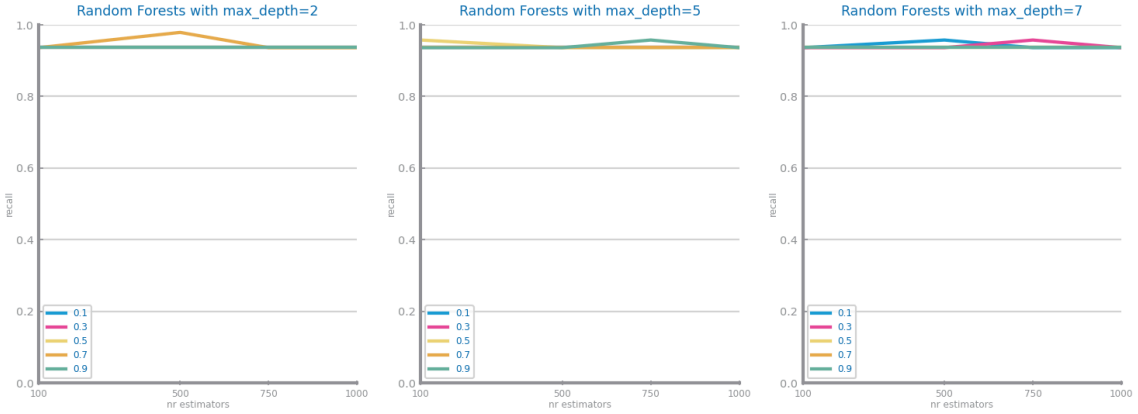



*Figure 47 Random Forests different parameterisations comparison for dataset 2*


*Figure 48 Random Forests overfitting analysis for dataset 1 (left) and dataset 2 (right)*

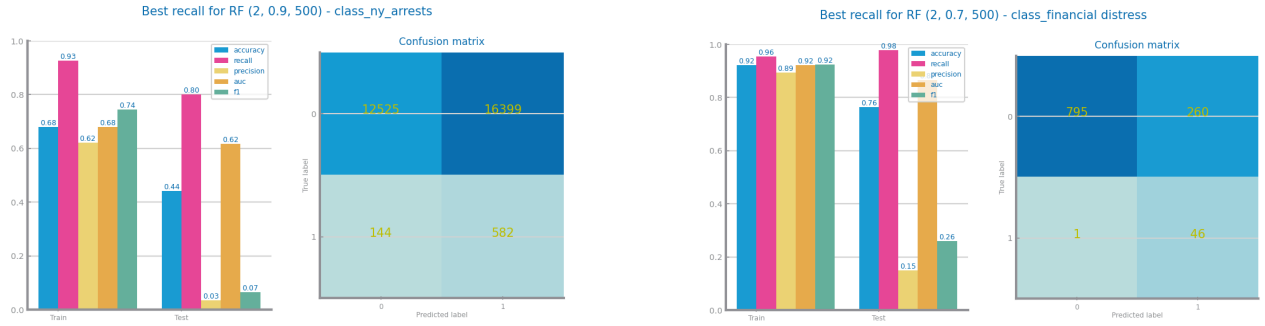



*Figure 49 Random Forests best model results for dataset 1 (left) and dataset 2 (right)*

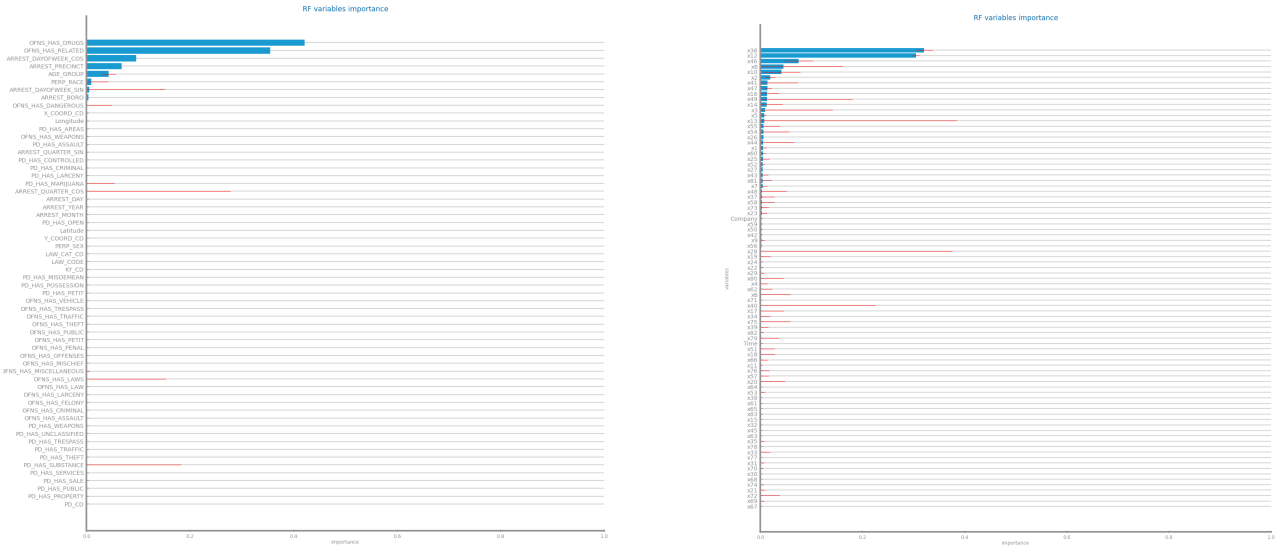



*Figure 50 Random Forests variables importance for dataset 1 (left) and dataset 2 (right)*

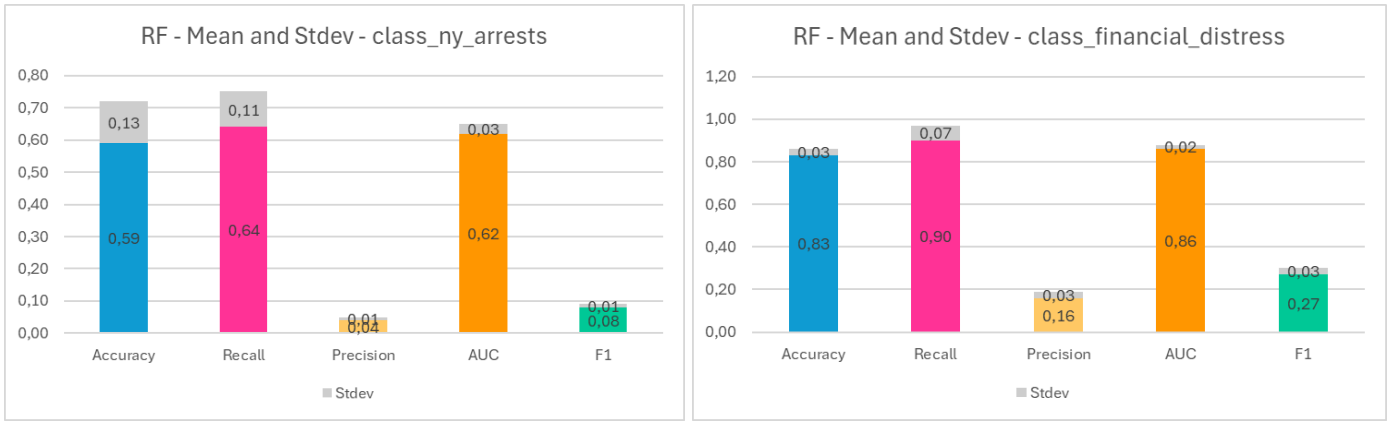



*Figure 51 Random Forests confidence intervals for dataset 1 (left) and dataset 2 (right)*

## *Gradient Boosting*

Dataset 1: best results for d=5 and f=0.5, with mean=0,51 and stdev=0,03. This model does not show overfitting. The most important variables in the model are ARREST_DAYOFWEEK_COS, AGE_GROUP, ARREST_DAYOFWEEK_SIN AND ARREST_QUARTER_SIN.

Dataset 2: best results for d=2 and f=0.9, with mean=0,91 and stdev=0,06. This model does not show overfitting. The most important variables in the model are x36, x12 and x10.

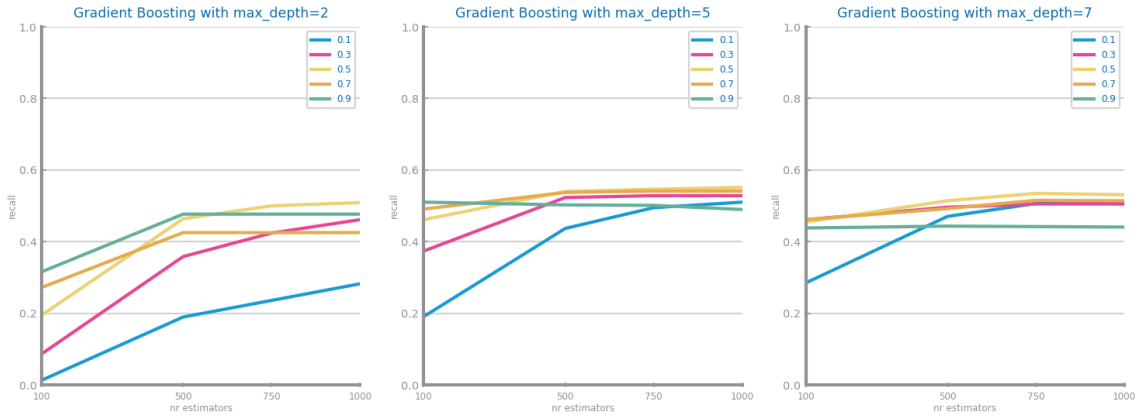



*Figure 52 Gradient boosting different parameterisations comparison for dataset 1*

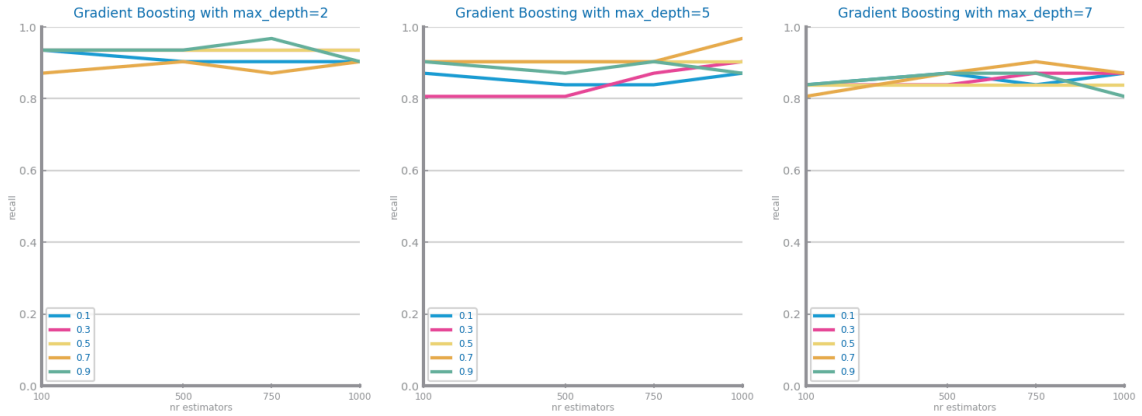



*Figure 53 Gradient boosting different parameterisations comparison for dataset 2*

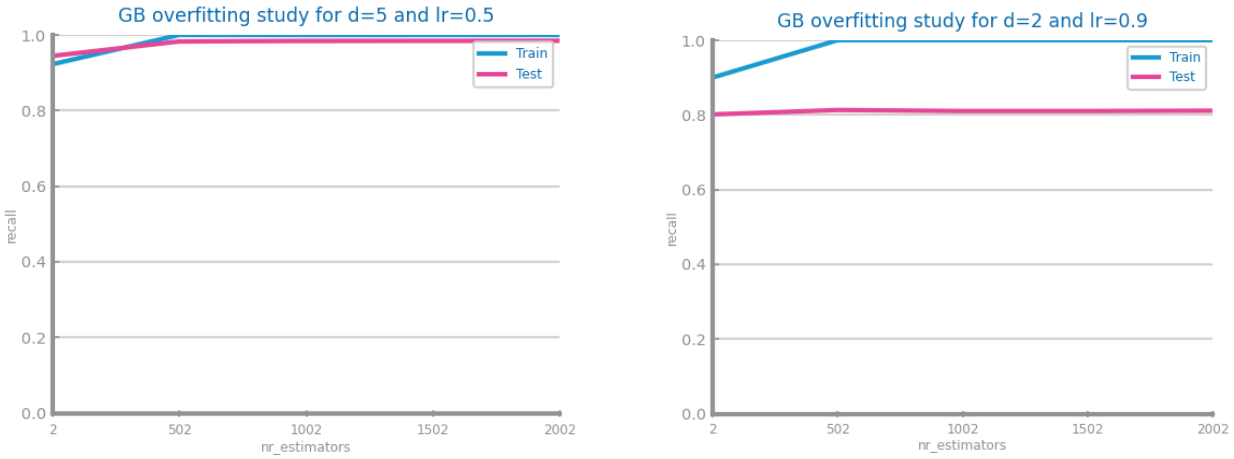



*Figure 54 Gradient boosting overfitting analysis for dataset 1 (left) and dataset 2 (right)*

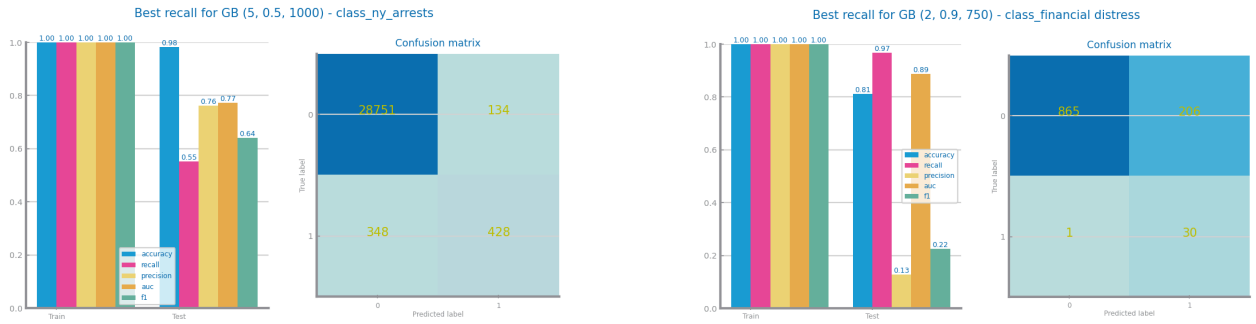



*Figure 55 Gradient boosting best model results for dataset 1 (left) and dataset 2 (right)*

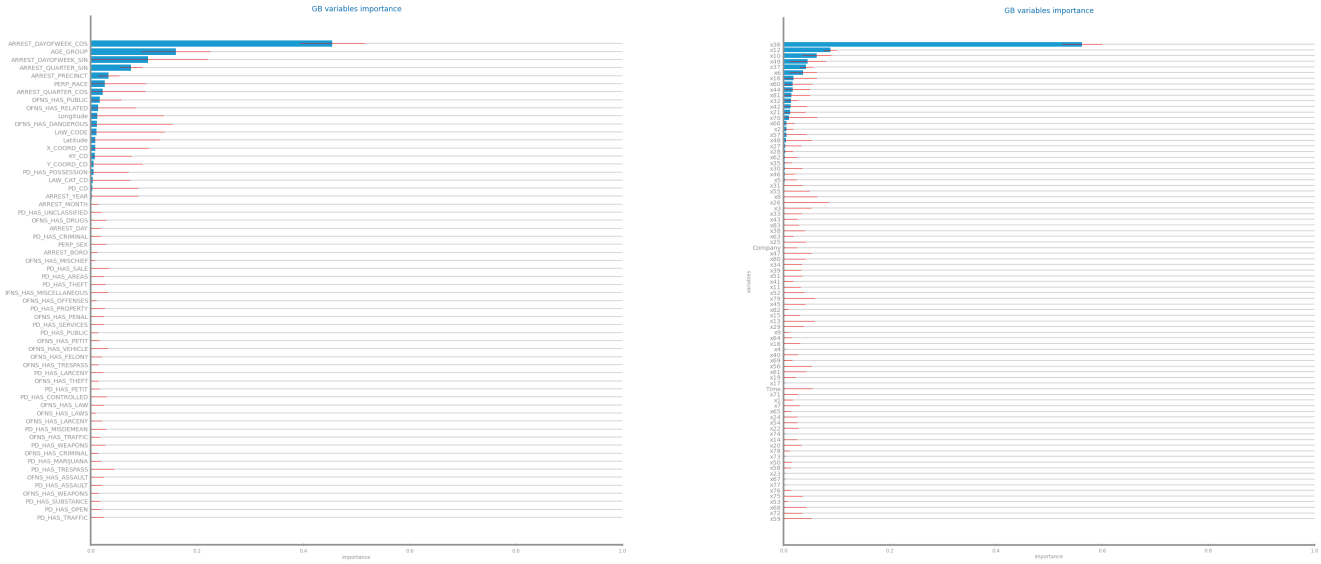



*Figure 56 Gradient boosting variables importance for dataset 1 (left) and dataset 2 (right)*


*Figure 57 Gradient boosting confidence intervals for dataset 1 (left) and dataset 2 (right)*

## *Multi-Layer Perceptrons*

For both datasets, even though the recall is shown as perfect, we concluded that the performance of MLP is very poor and that this model is not reliable by looking at the models' results and the parameterization study graphs.

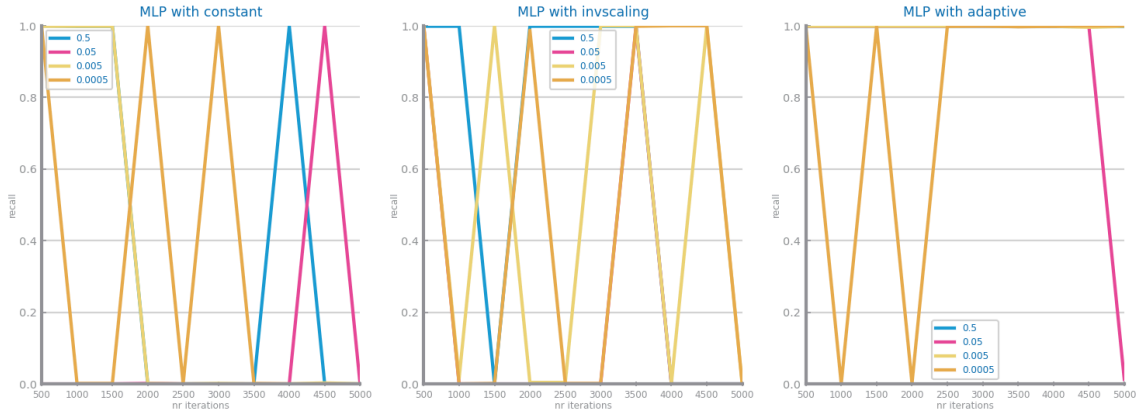



*Figure 58 MLP different parameterisations comparison for dataset 1*

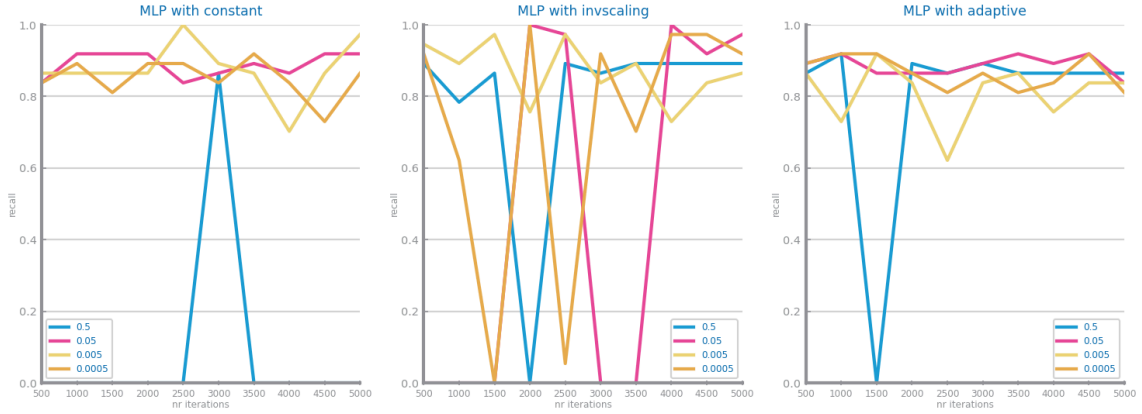



*Figure 59 MLP different parameterisations comparison for dataset 2*

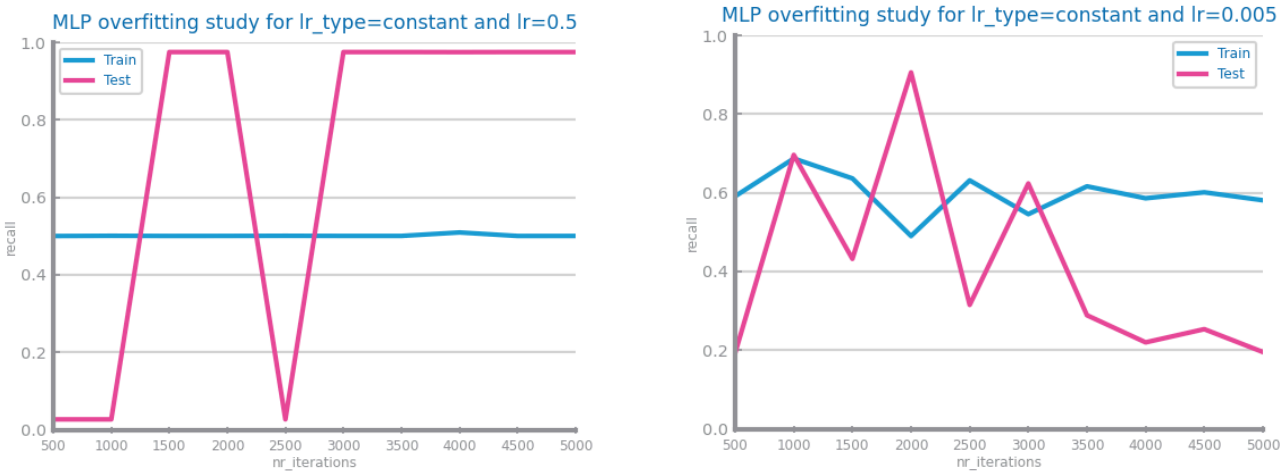



*Figure 60 MLP overfitting analysis for dataset 1 (left) and dataset 2 (right)*


*Figure 61 Loss curve analysis for dataset 1 (left) and dataset 2 (right)*

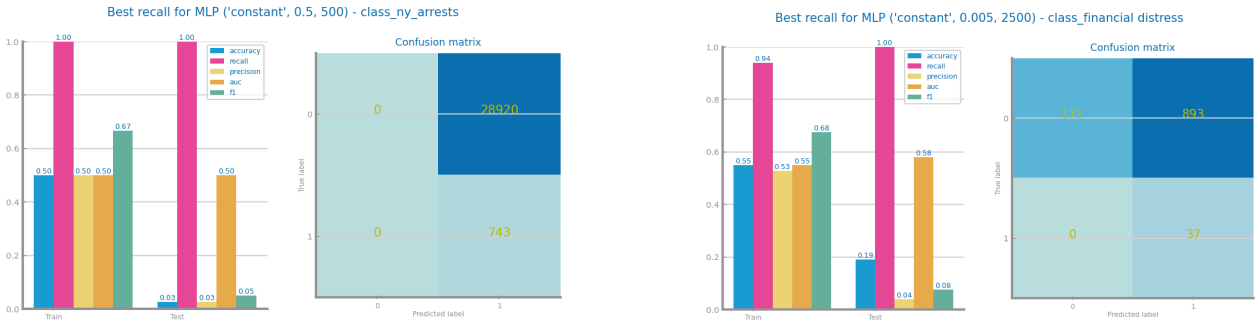



*Figure 62 MLP best model results for dataset 1 (left) and dataset 2 (right)*


*Figure 63 MLP confidence intervals for dataset 1 (left) and dataset 2 (right)*

# 4 Critical Analysis

As explained in the MLP section, these models presented very poor performance for both datasets. This can be due to the fact that we decided not to apply scaling in our data preparation phase. In a future iteration, scaling should be applied before training MLP models.

In dataset 1, the DT and RF models seem to show high importance of OFNS_HAS_DRUGS even though GB does not consider so. Additionally, from RF and GB, the variables AGE_GROUP, ARREST_PRECINT, PERP_RACE and the day of week and quarter cyclic variables also seem to be moderately important for the models.

In dataset 2, the DT, RF and GB models show that x36 is the most important variable, where we can also notice from RF and GB that x12 and x10 seem also moderately important and x16 may be relatively important.

The results between DT and RF are similar, which can be seen as expected if we take into account how the models work. This may also mean that we reached good DT models. Surprisingly, for dataset 1, the best DT model has higher recall than the best RF model, with higher mean and stdev values as well.

Ignoring MLP, dataset 1 shows overall good best models, being very similar among them (around 0,80 recall). One exception is the GB that has a recall=0,55 but, surprisingly, increases the accuracy to a good value of 0,97.

Ignoring MLP, dataset 2 shows overall good best models, being very similar among them (around 0.97 recall). One interesting aspect is that Naive Bayes, even though its best model is good, the stdev is very high (0,18).

Except for the GB of dataset 1, the precision is overall very low across all models. We believe that this may be a consequence of the imbalanced nature of the datasets, where the target contains a high number of negative values.

# Time Series Forecasting

## 5 Data Profiling

### *Data Dimensionality and Granularity*

Dataset 1: The chosen granularities were daily, monthly and yearly respectively, aggregated by sum. Shows fluctuations in trend and variability, with a noticeable decline over time.

Dataset 2: The chosen granularities were yearly, two years and five years respectively, aggregated by mean. Since the original data is by year it looks less noisy than the first dataset and also presents a clear upward trend.

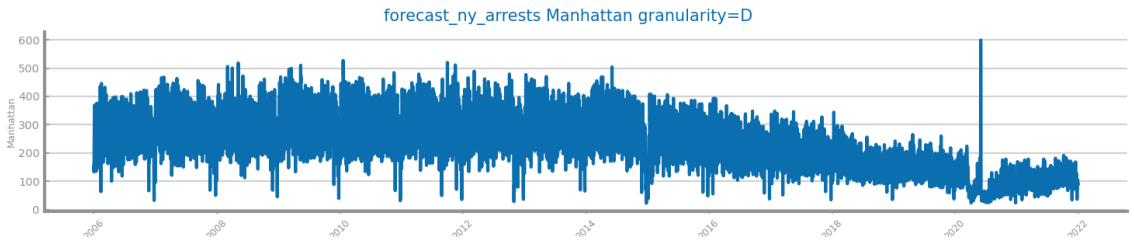



*Figure 64 Original time series 1 (the most atomic detail)*

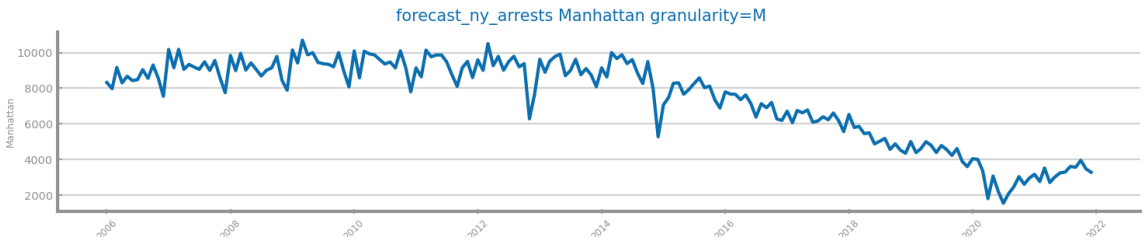



*Figure 65 Time series 1 at the second chosen granularity*

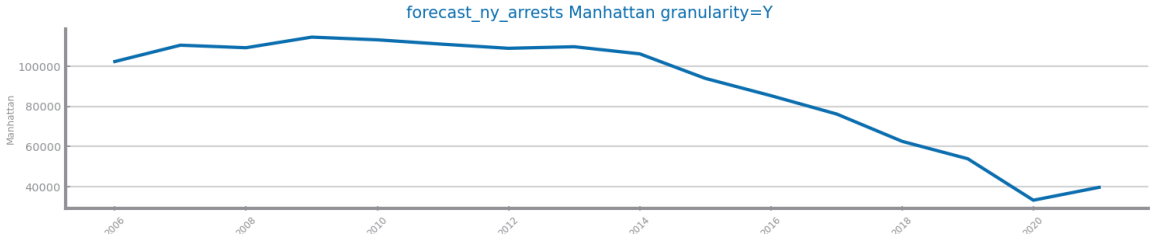



*Figure 66 Time series 1 at the third chosen granularity*

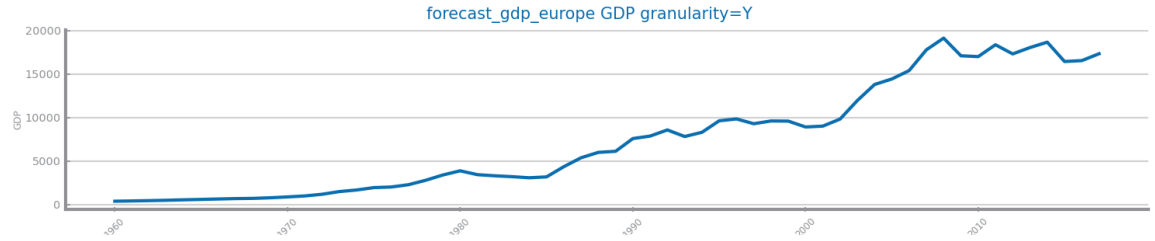



*Figure 67 Original time series 2 (the most atomic detail)*

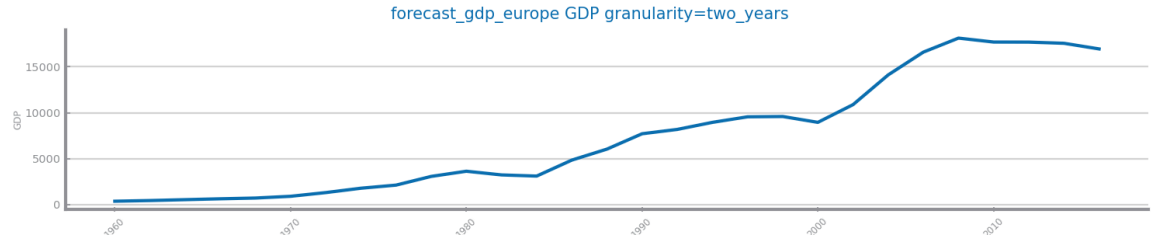



*Figure 68 Time series 2 at the second chosen granularity*

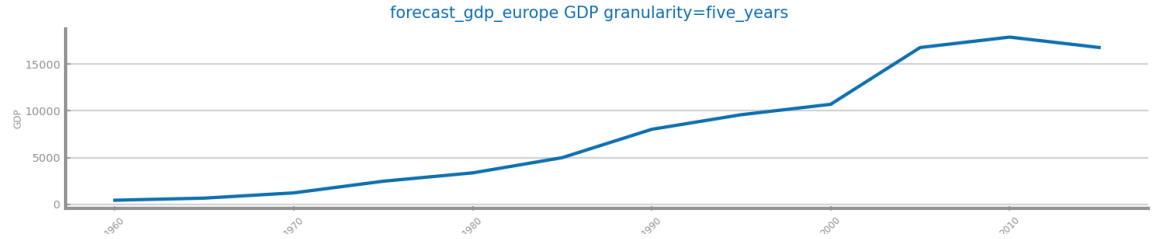
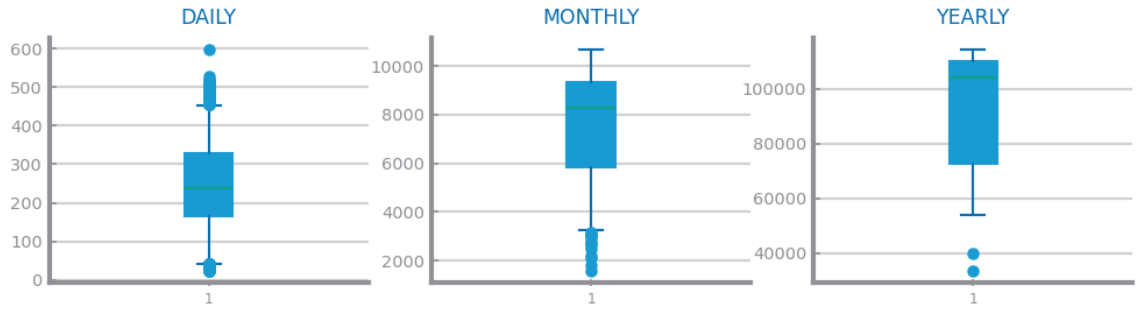



*Figure 69 Time series 2 at the third chosen granularity*

### *Data Distribution*

Dataset 1: The variable exhibits a positive-skewed distribution at the two higher granularities, becoming closer to normal at the daily level. Dispersion increases with coarser granularities, and histograms highlight uneven data behavior, with significant outliers in the yearly data.

Dataset 2: The variable exhibits a negative-skewed distribution at the yearly and five year granularities, becoming closer to normal at the two year level. Variability decreases with coarser granularities, reflecting smoother long-term trends.




| | |
|---|---|
| count | 5844.000000 |
| mean | 244.803730 |
| std | 103.735087 |
| min | 22.000000 |
| 25% | 166.000000 |
| 50% | 239.000000 |
| 75% | 330.000000 |
| max | 599.000000 |

Name: value, dtype: float64

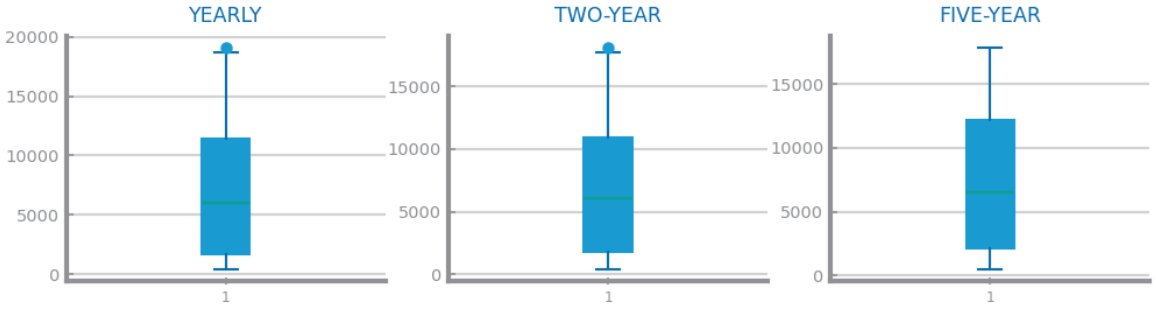
| | |
|---|---|
| count | 192.000000 |
| mean | 7451.213542 |
| std | 2341.787598 |
| min | 1553.000000 |
| 25% | 5848.250000 |
| 50% | 8302.000000 |
| 75% | 9340.000000 |
| max | 10699.000000 |

Name: value, dtype: float64

| | |
|---|---|
| count | 16.000000 |
| mean | 89414.562500 |
| std | 27763.472434 |
| min | 33255.000000 |
| 25% | 72799.250000 |
| 50% | 104272.000000 |
| 75% | 109932.250000 |
| max | 114527.000000 |

Name: value, dtype: float64

*Figure 70 Boxplots for time series 1 at different granularities*




| | |
|---|---|
| count | 58.000000 |
| mean | 7426.495552 |
| std | 6310.412525 |
| min | 359.029000 |
| 25% | 1720.540750 |
| 50% | 6044.174000 |
| 75% | 11420.337250 |
| max | 19137.007000 |

Name: value, dtype: float64

| | |
|---|---|
| count | 29.000000 |
| mean | 7426.495552 |
| std | 6347.216085 |
| min | 374.958000 |
| 25% | 1789.014500 |
| 50% | 6044.174000 |
| 75% | 10882.922000 |
| max | 18119.774000 |

Name: value, dtype: float64

| | |
|---|---|
| count | 12.000000 |
| mean | 7738.176733 |
| std | 6598.851289 |
| min | 433.734800 |
| 25% | 2154.457850 |
| 50% | 6504.120100 |
| 75% | 12214.310000 |
| max | 17884.646600 |

Name: value, dtype: float64

*Figure 71 Boxplots for time series 2 at different granularities*

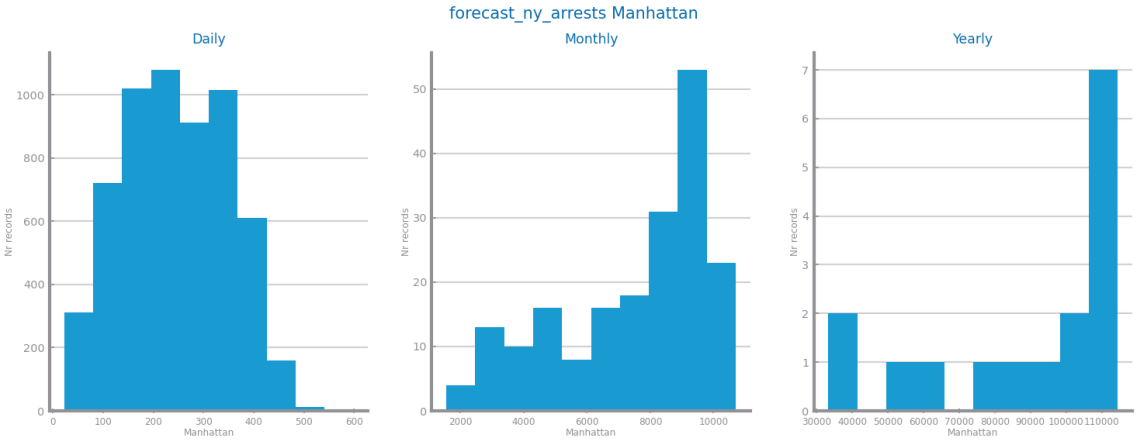



*Figure 72 Histograms for time series 1 at different granularities*

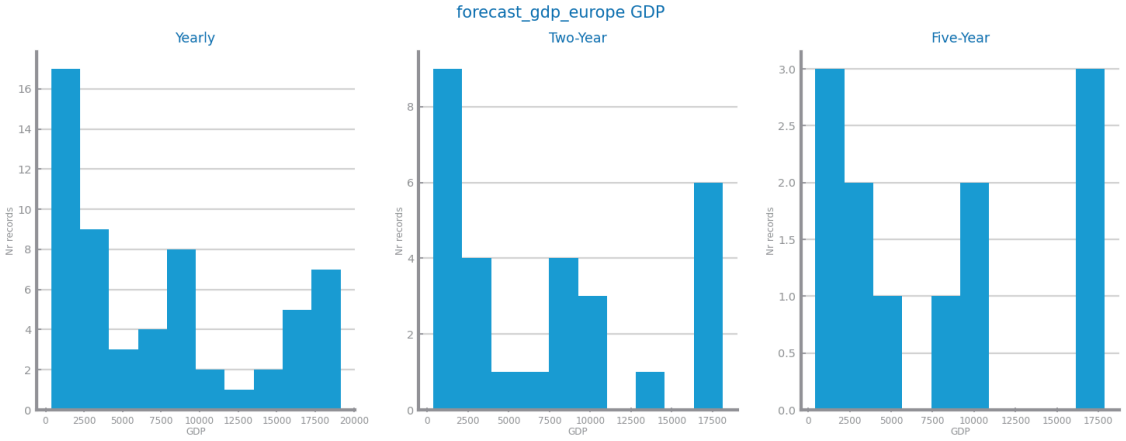



*Figure 73 Histograms for time series 2 at different granularities*

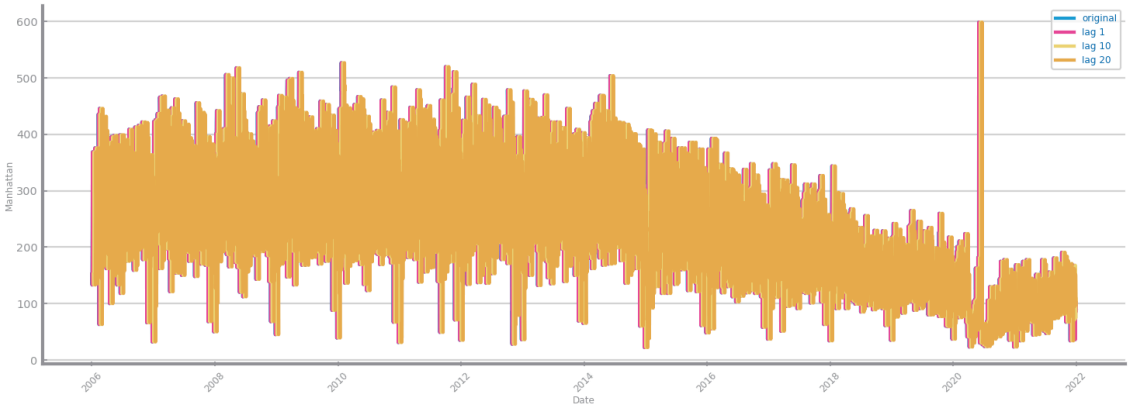



*Figure 74 Autocorrelation lag-plots for original time series 1*

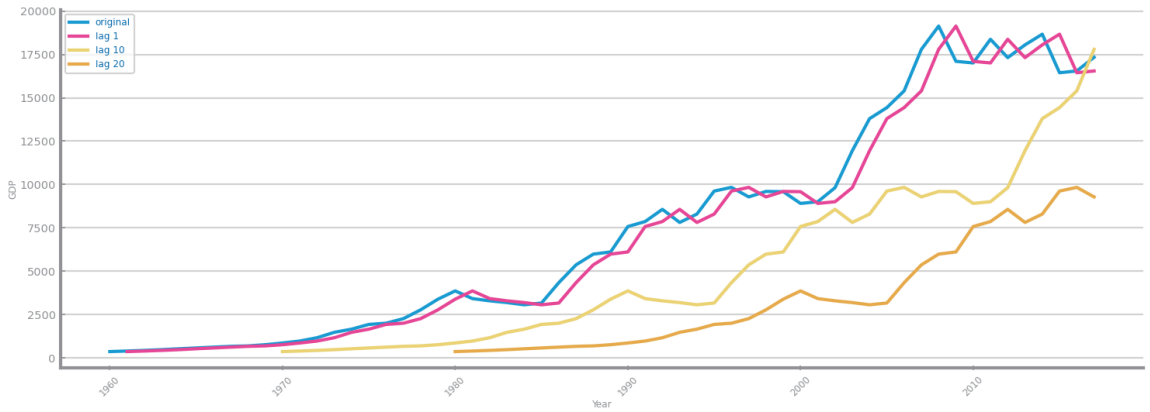



*Figure 75 Autocorrelation lag-plots for original time series 2*

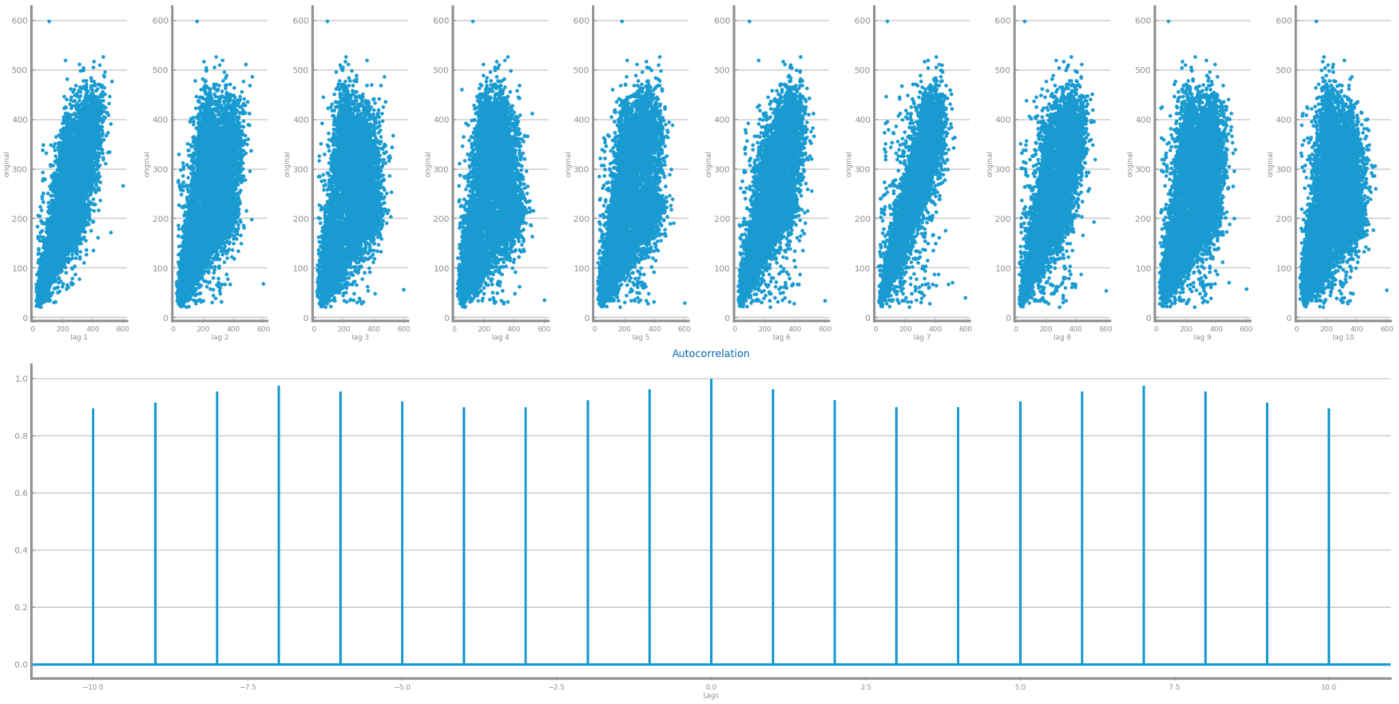



*Figure 76 Autocorrelation correlogram for original time series 1*

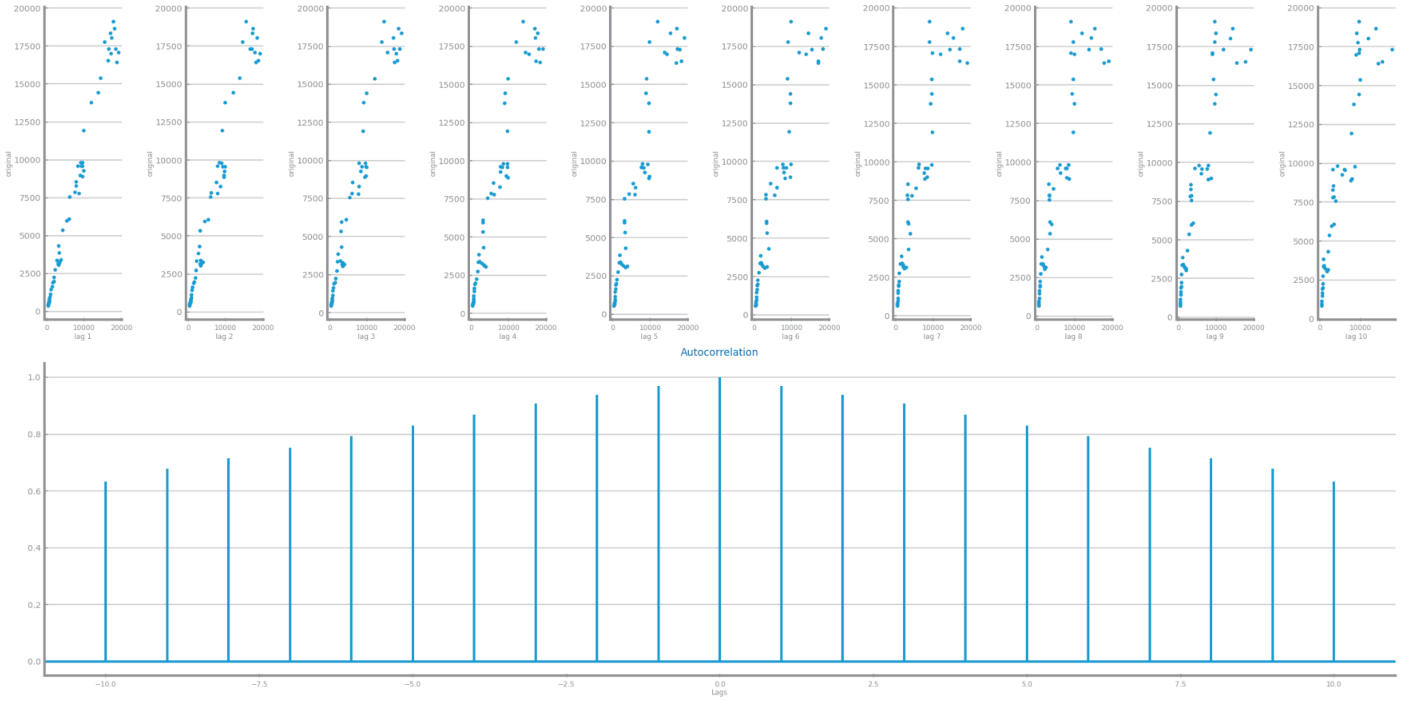



*Figure 77 Autocorrelation correlogram for original time series 2*

## *Data Stationarity*

Dataset 1: The Dickey-Fuller test returned a p-value of 0.338, indicating non-stationarity that is confirmed by the mean and standard deviation plots due to the shifting patterns.

Dataset 2: The Dickey-Fuller test returned a p-value of 0.869 that also confirms non-stationarity.

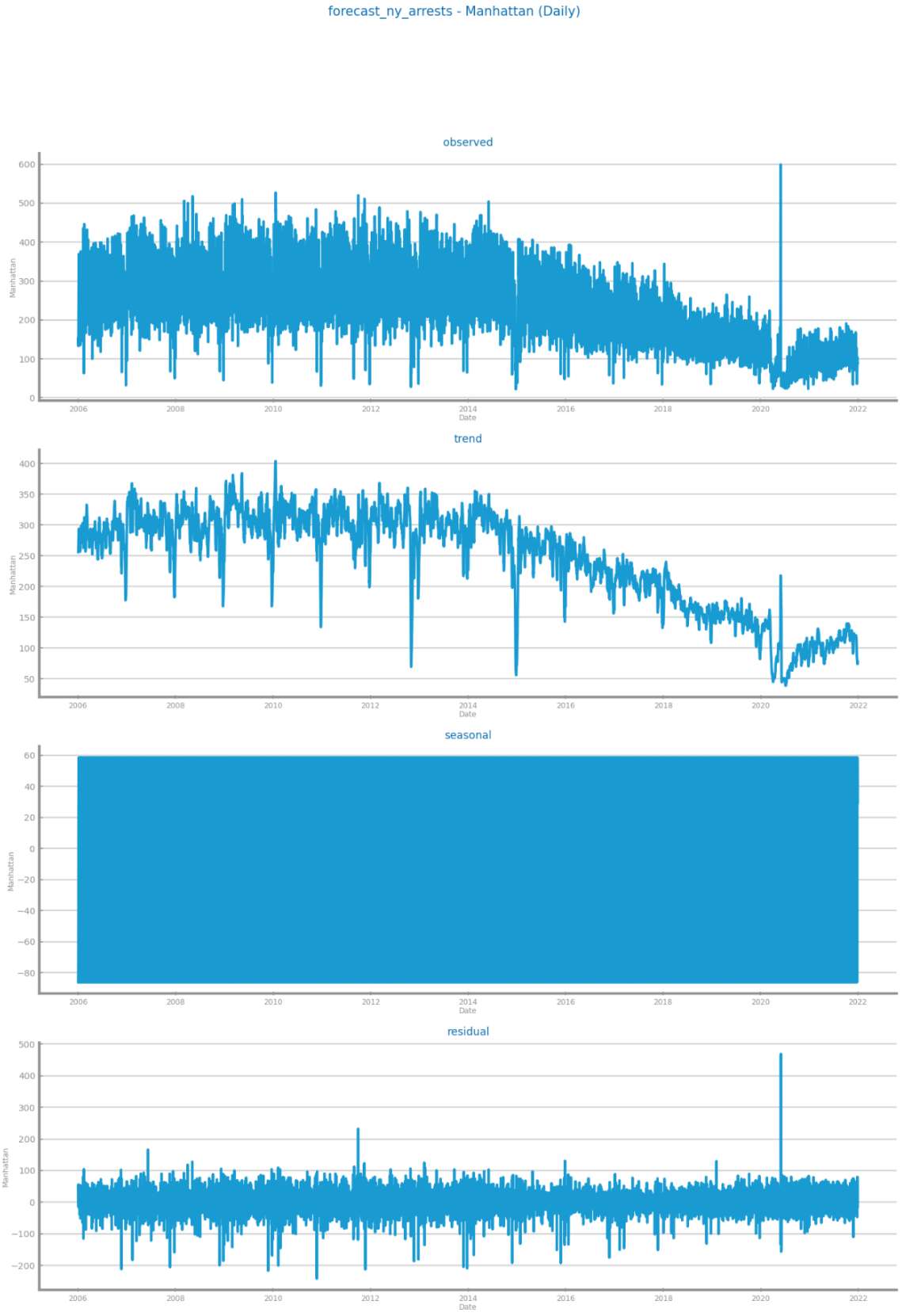



*Figure 78 Components study for time series 1*

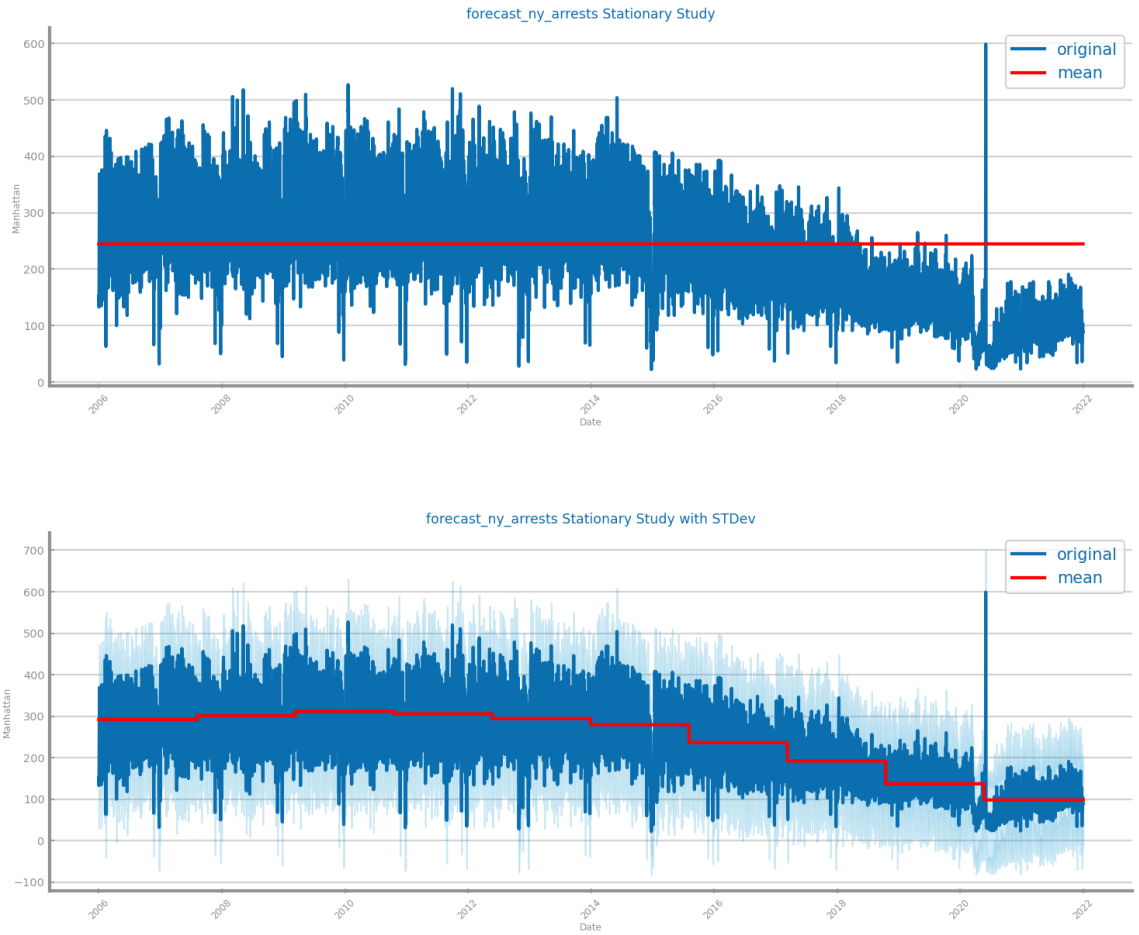



*Figure 79 Stationarity study for time series 1*

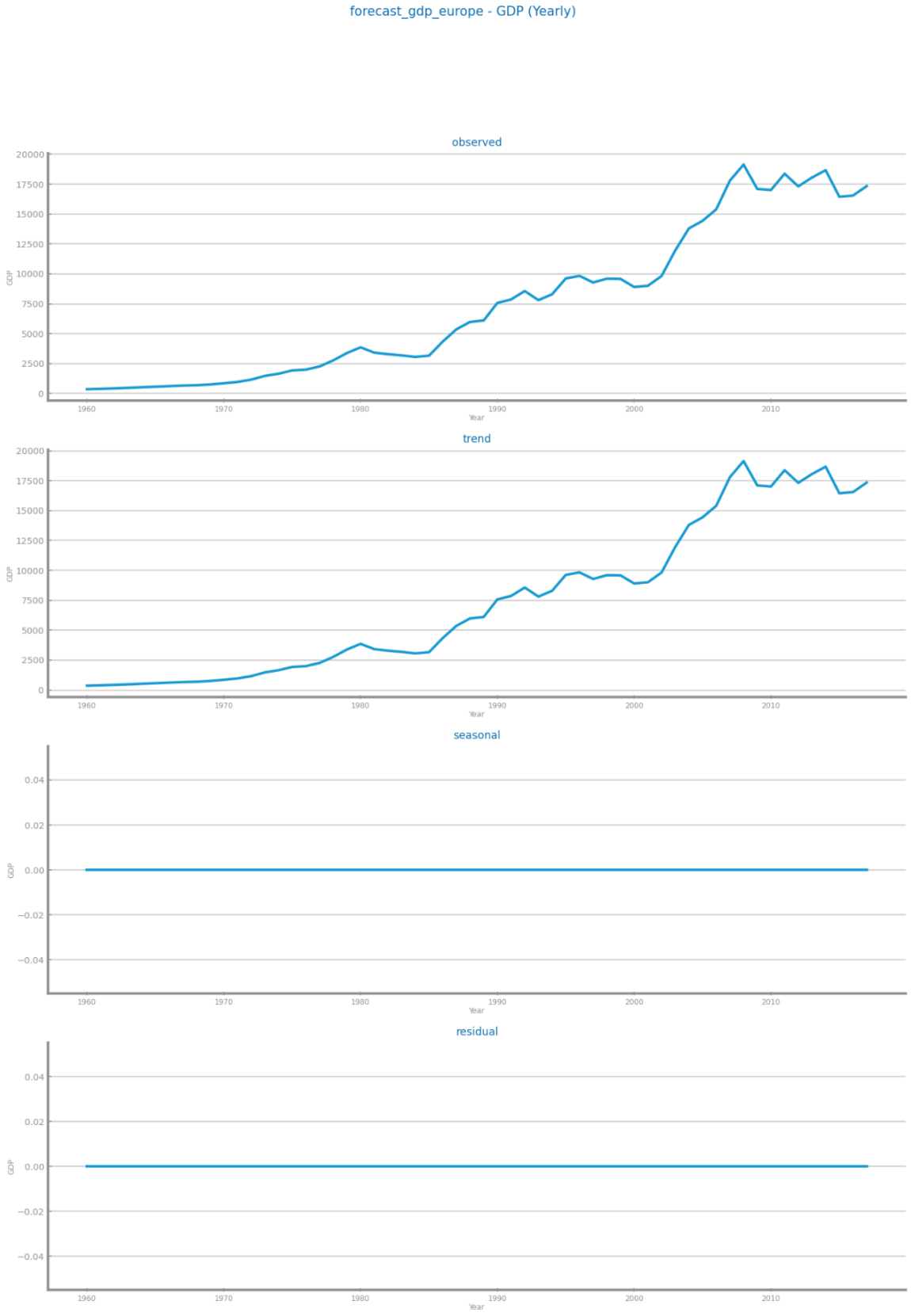



*Figure 80 Components study for time series 2*

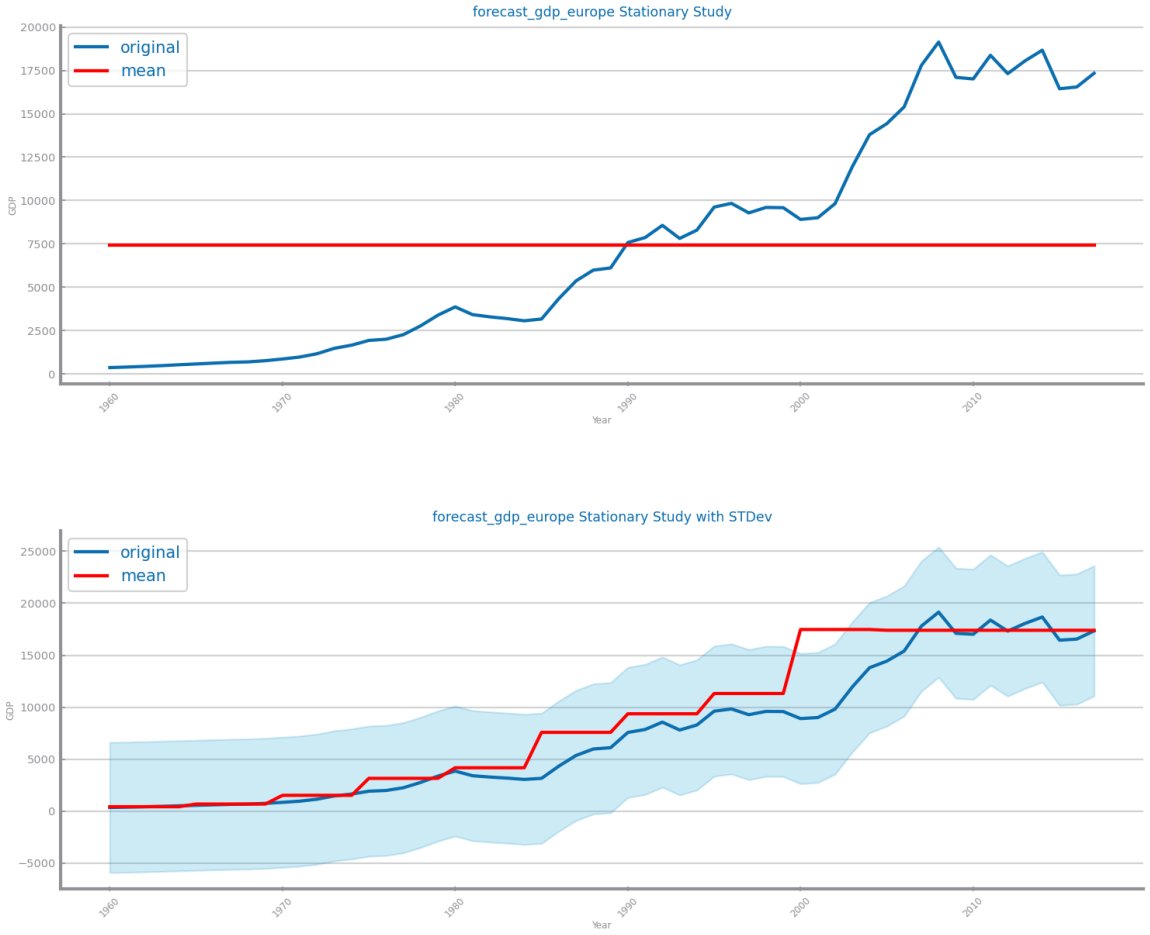



*Figure 81 Stationarity study for time series 2*

# 6 Data Transformation

## *Aggregation*

Dataset 1: We chose to aggregate to a **monthly granularity** due to improved performance and a cleaner trend with less noise while preserving variability.

Dataset 2: We chose the **original** annual aggregation since it looks the best in our eyes, by preserving the GDP trend and not showing presence of noise.

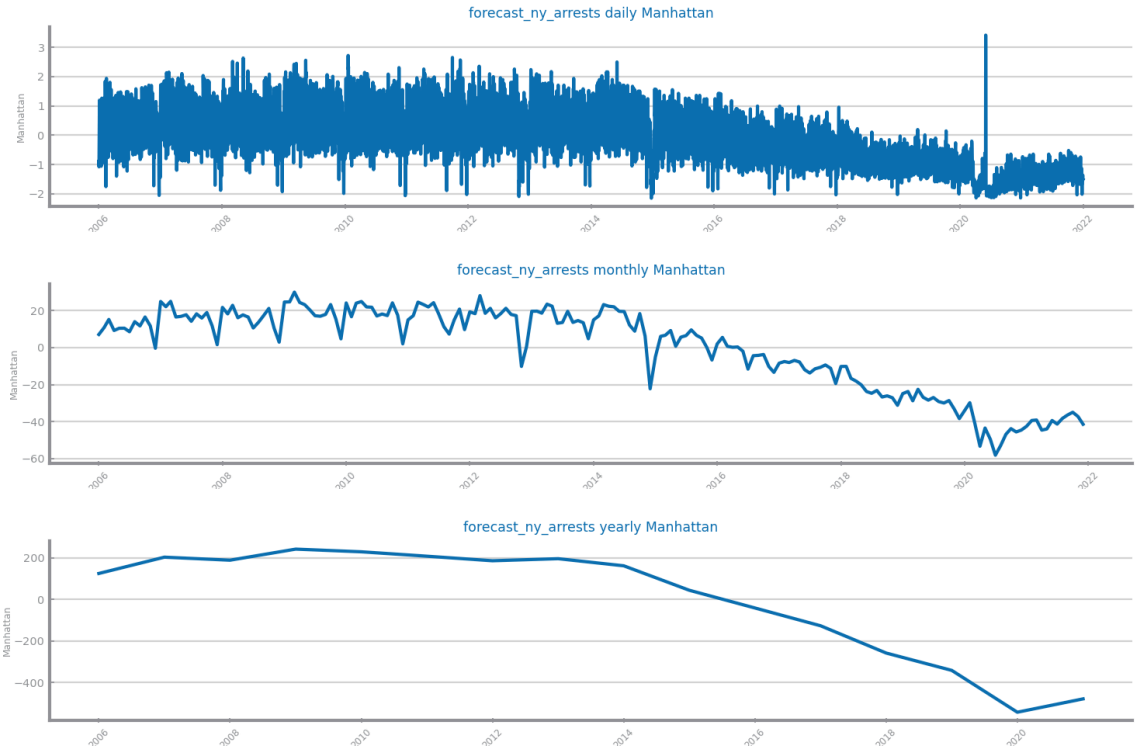
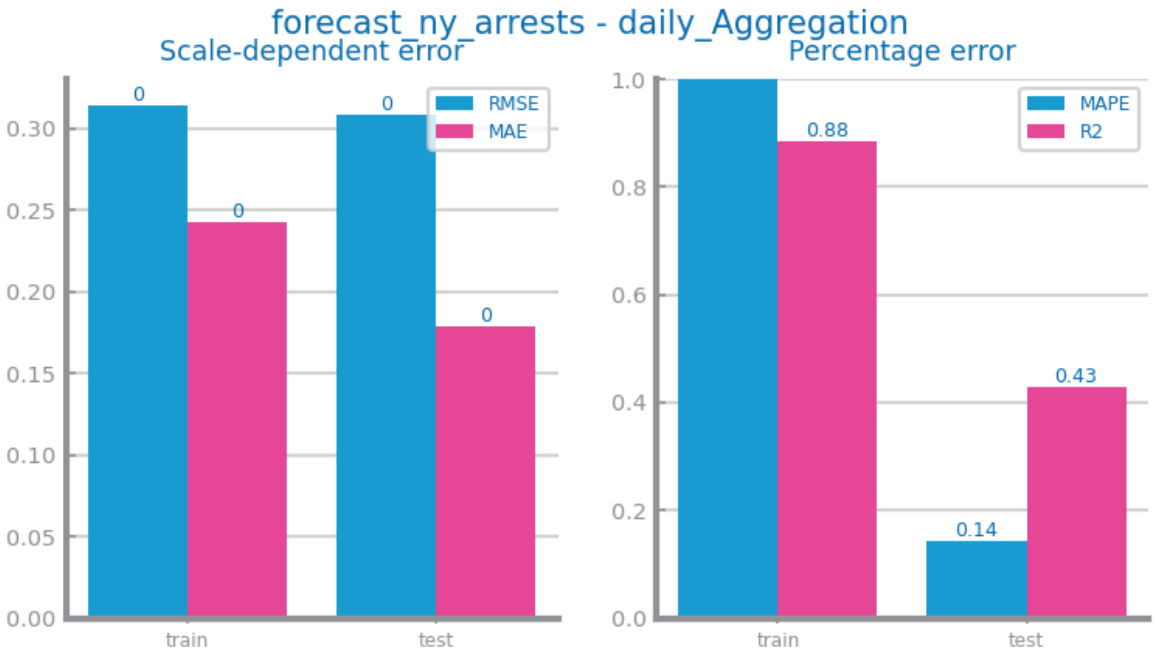
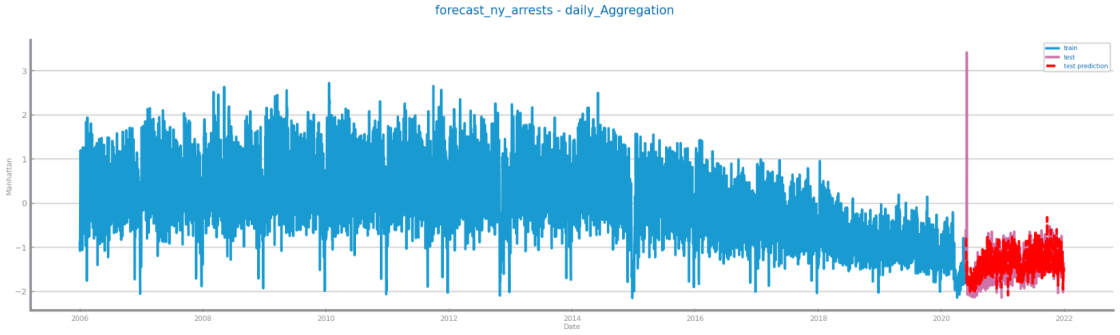
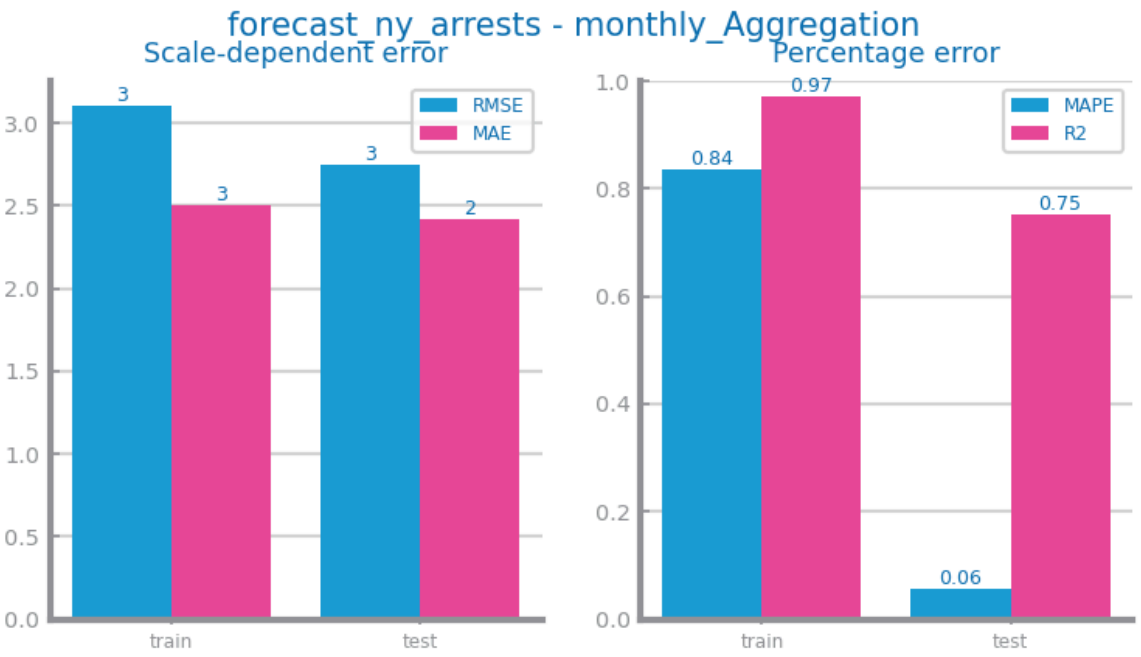
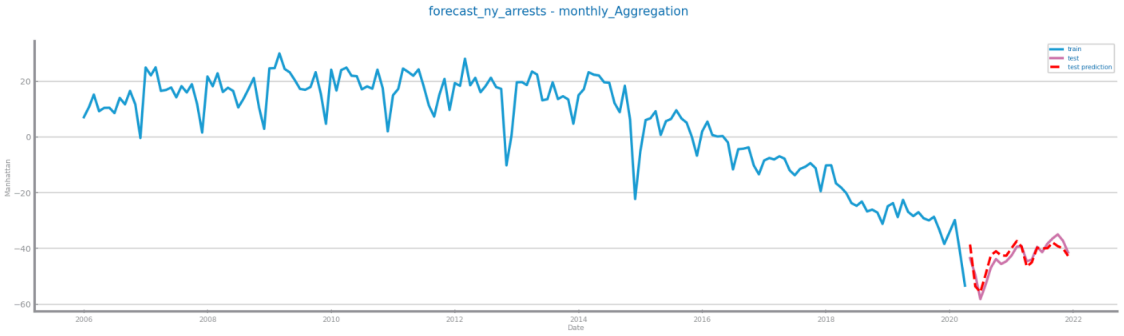



*Figure 82 Forecasting plots after different aggregations on time series 1*




forecast_ny_arrests - monthly_Aggregation
Scale-dependent error
Percentage error
RMSE
MAE
MAPE
R2
3
3
3
2
0.84
0.97
0.06
0.75
train
test
train
test


forecast_ny_arrests - monthly_Aggregation
train
test
test prediction
Manhattan
Date

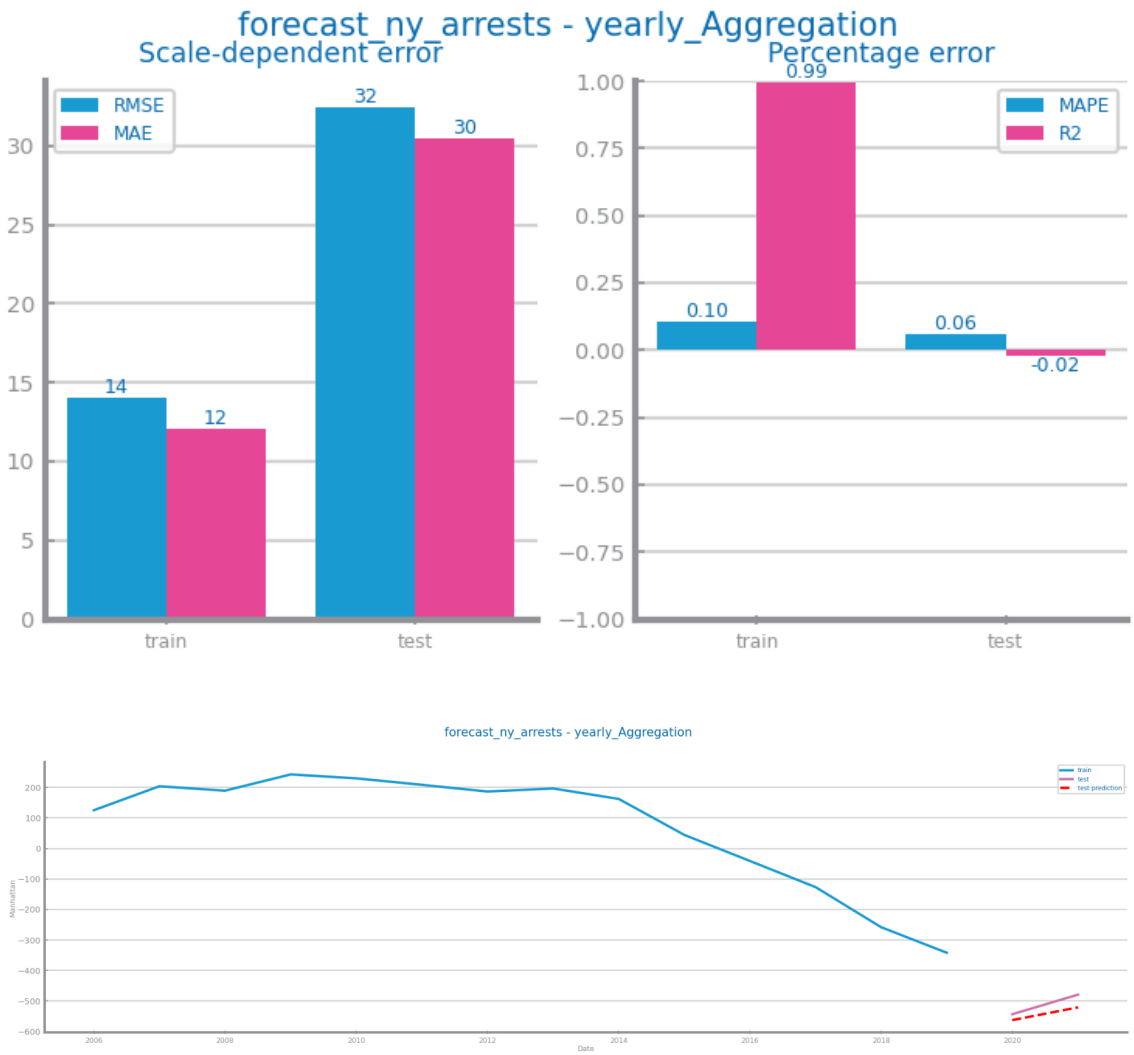



*Figure 83 Forecasting results after different aggregations on time series 1*

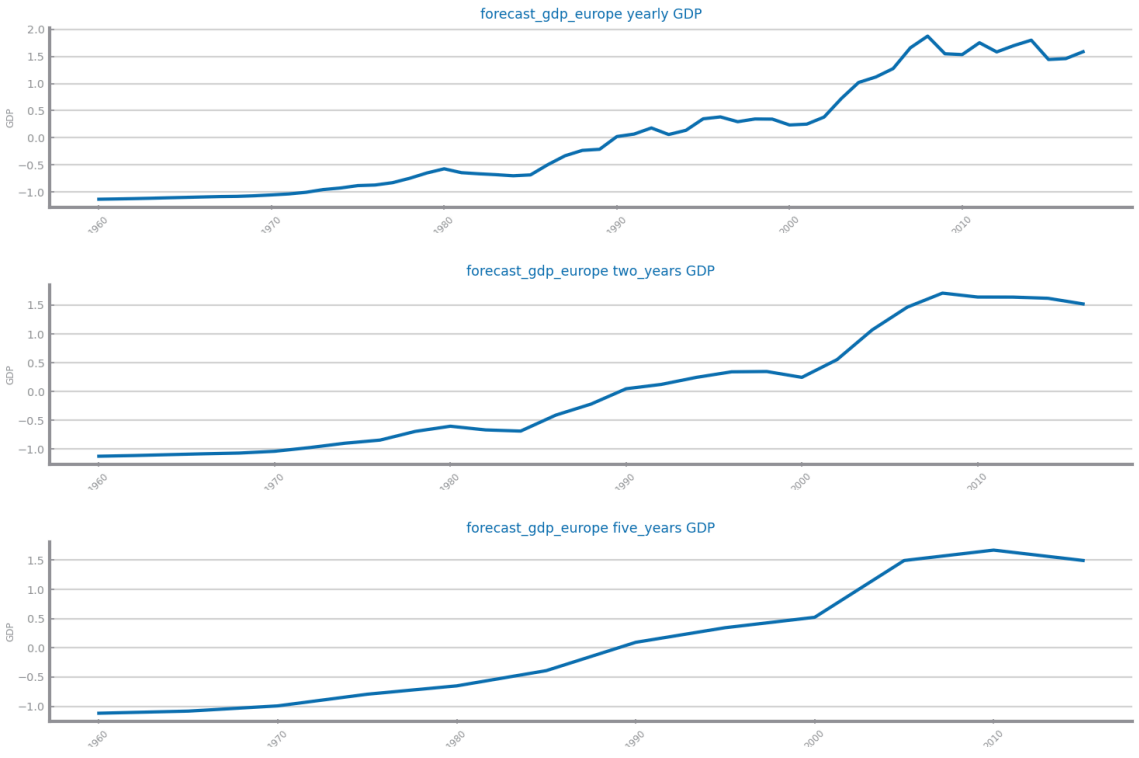


*Figure 84 Forecasting plots after different aggregations on time series 2*

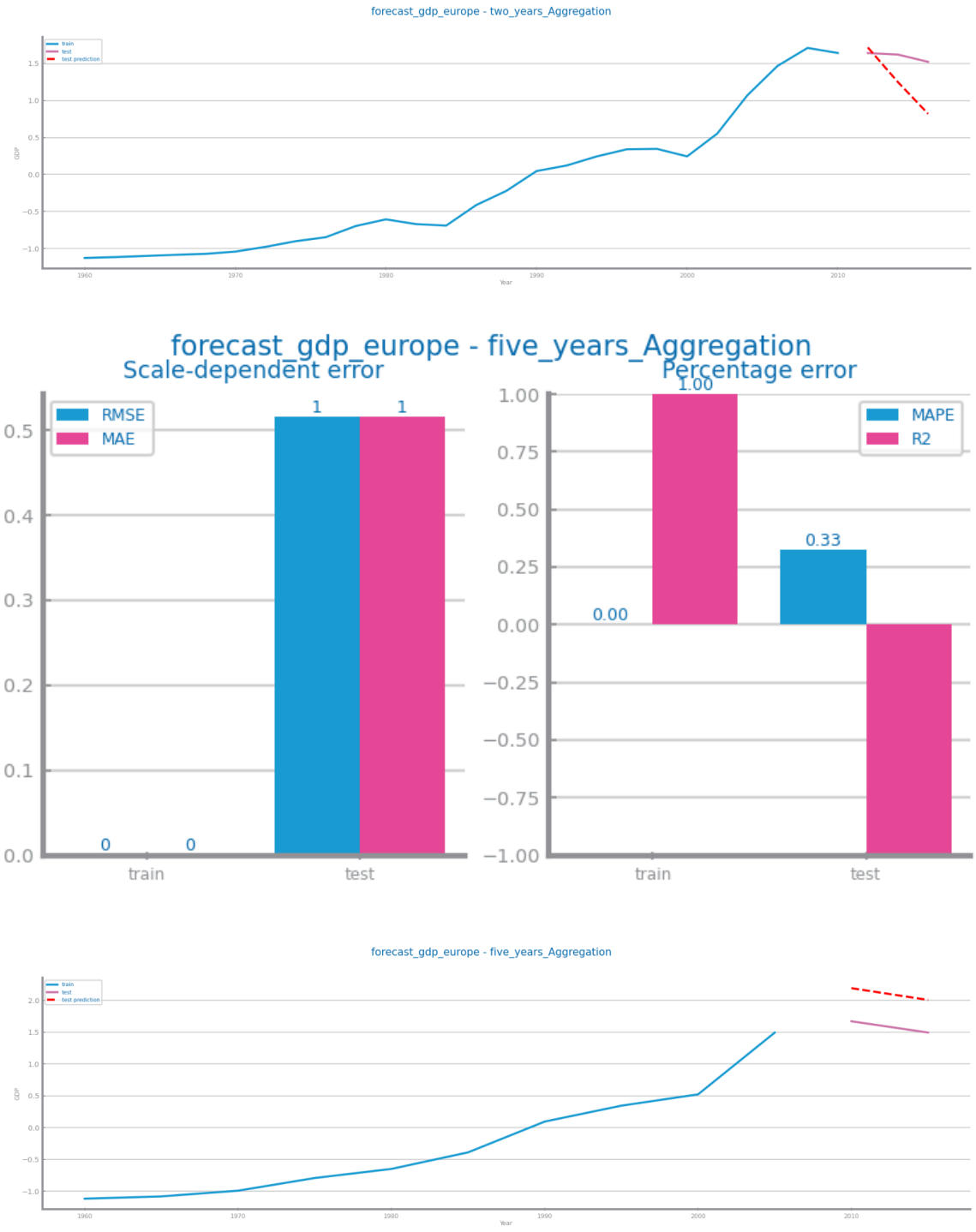



*Figure 85 Forecasting results after different aggregations on time series 2*

### *Smoothing*

Dataset 1: We applied smoothing for the window size of 10, 25, 50, 75 and 100 but **decided to stick with the original** in the end.

Dataset 2: We applied smoothing for the window size of 5, 10,20, 35 and 50 but **decided to stick with the original** in the end.

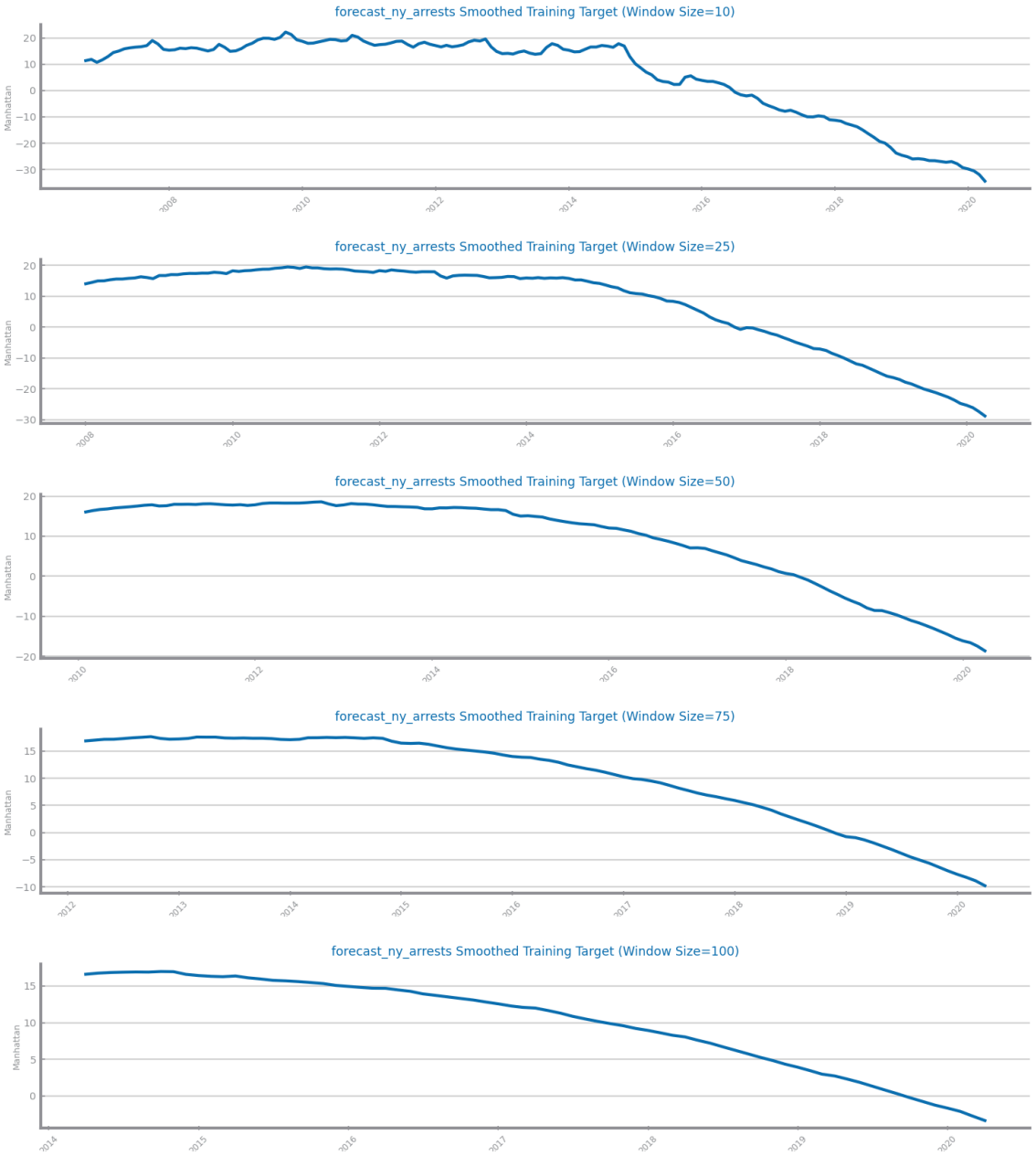
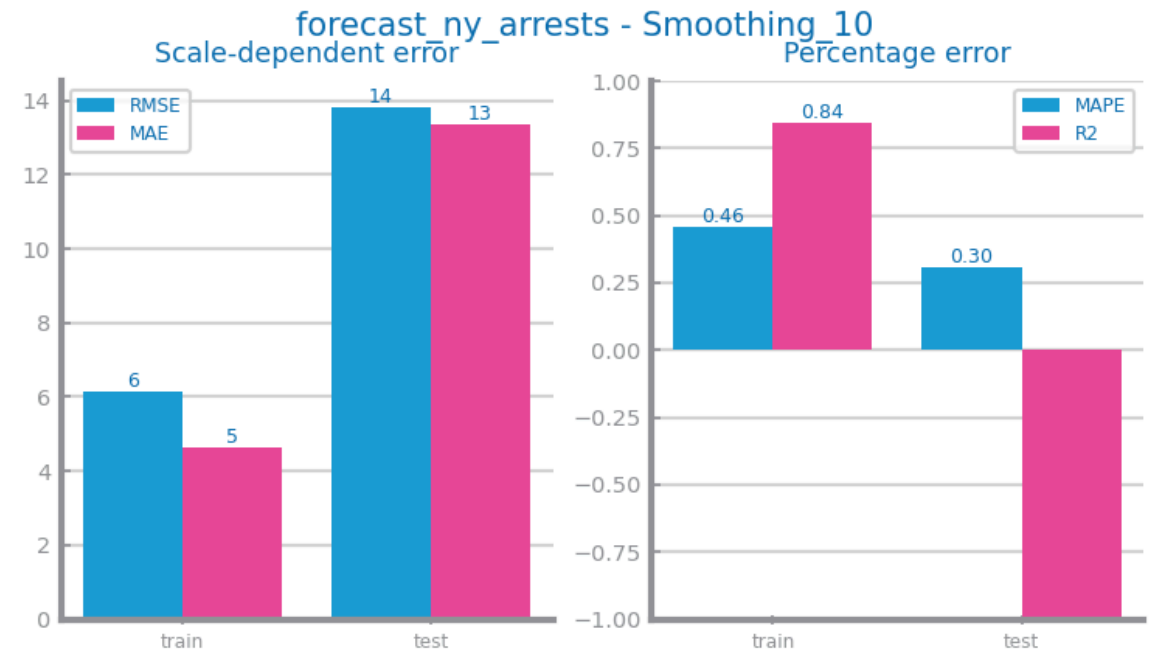
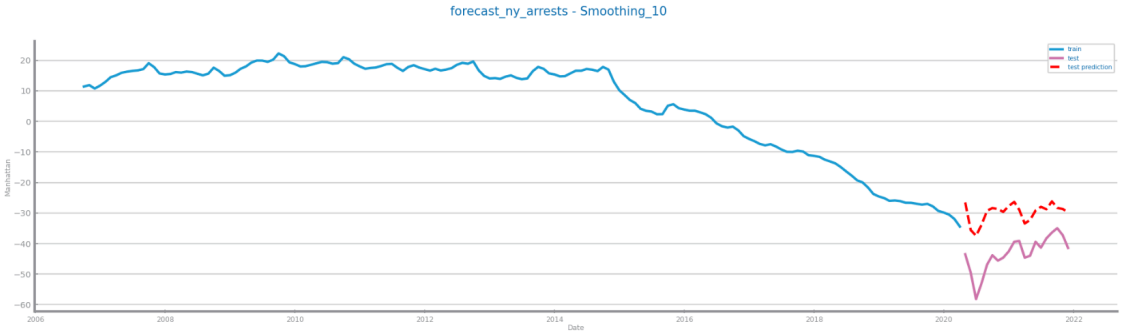
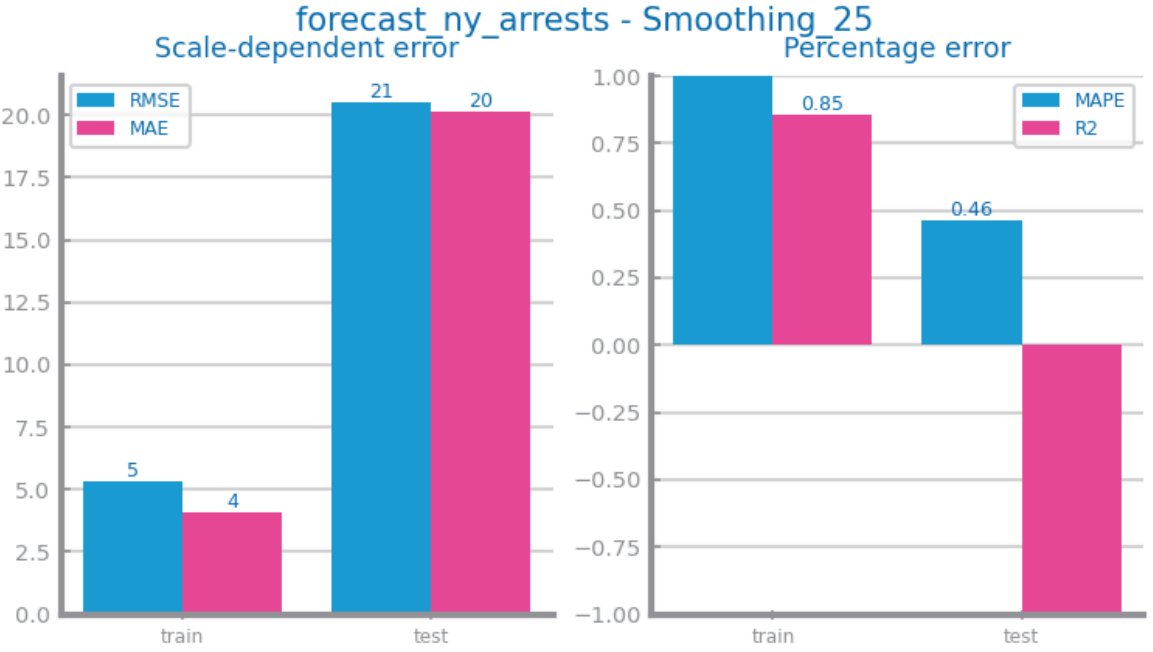
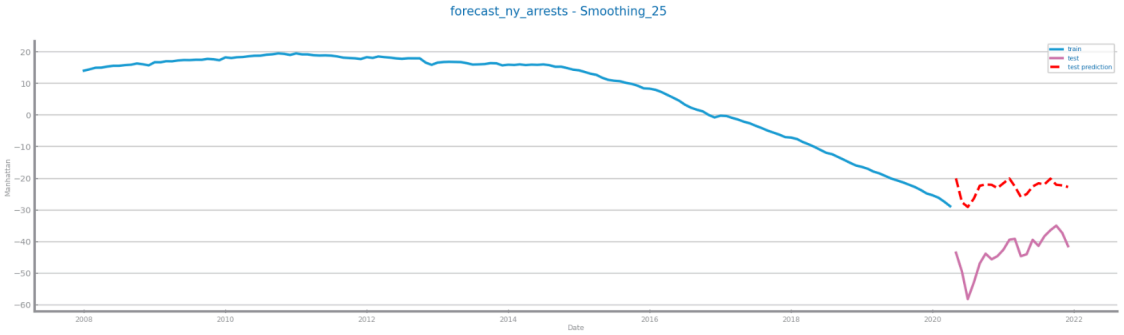


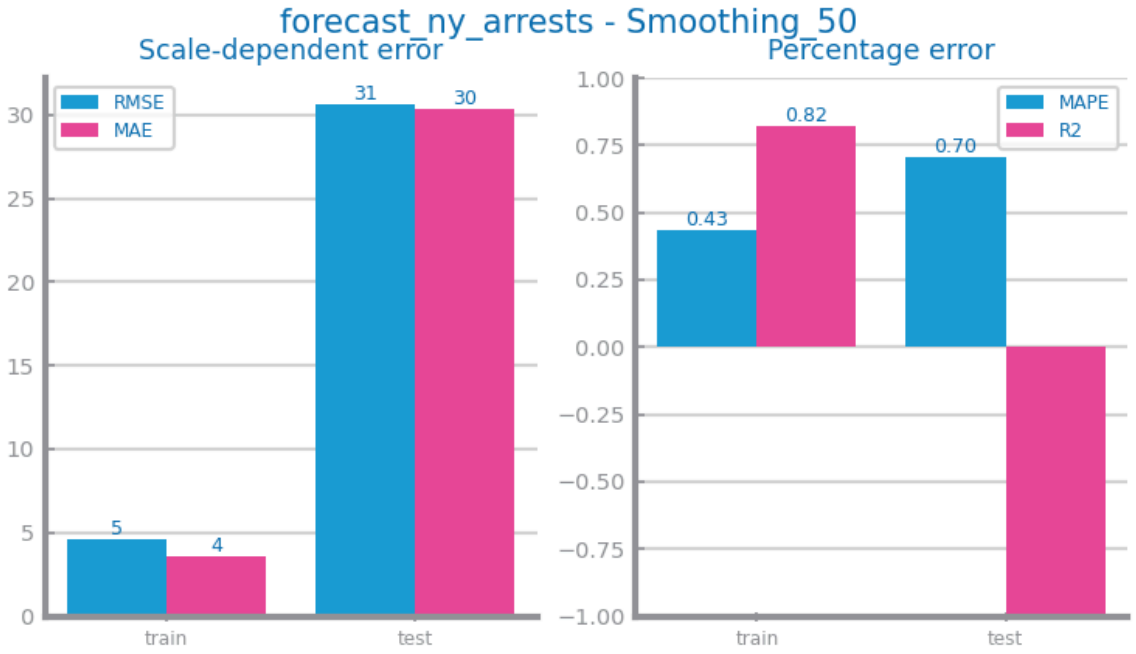

*Figure 86 Forecasting plots after different smoothing parameterisations on time series 1*

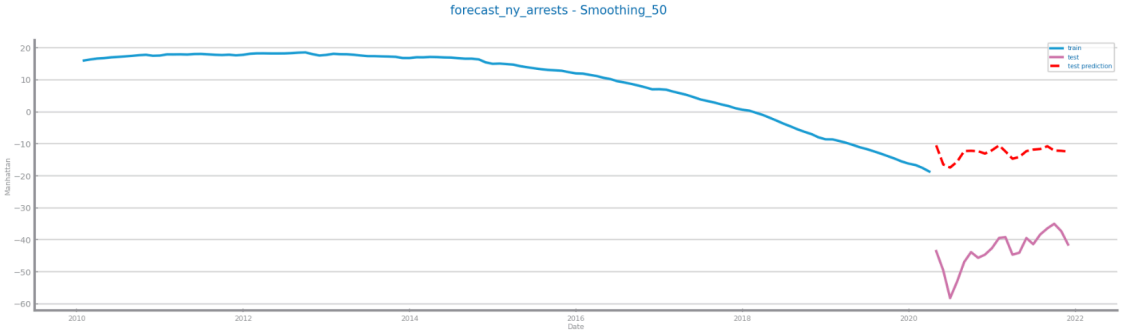

forecast_ny_arrests - Smoothing_10
Scale-dependent error
Percentage error
RMSE
MAE
MAPE
R2
6
5
14
13
0.46
0.84
0.30
train
test

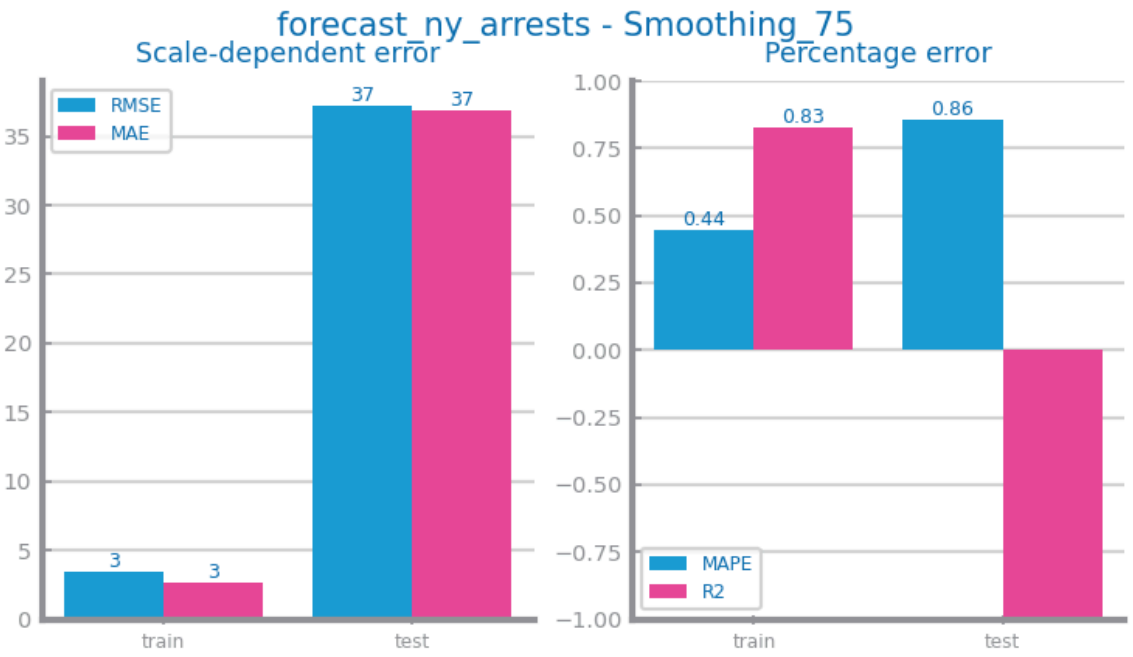

forecast_ny_arrests - Smoothing_10
train
test
test prediction
Manhattan
Date

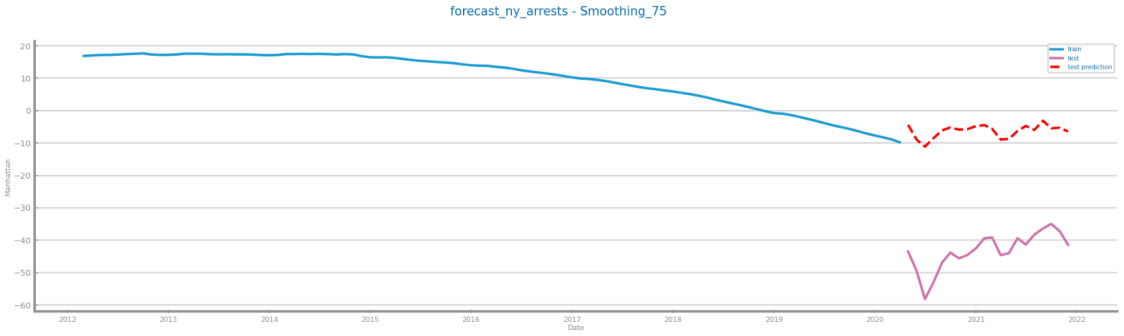

forecast_ny_arrests - Smoothing_25
Scale-dependent error
Percentage error
RMSE
MAE
MAPE
R2
5
4
21
20
0.85
0.46
train
test


forecast_ny_arrests - Smoothing_25
train
test
test prediction
Manhattan
Date


forecast_ny_arrests - Smoothing_50
Scale-dependent error
RMSE
MAE
5
4
31
30
train
test
Percentage error
MAPE
R2
0.43
0.82
0.70
train
test


forecast_ny_arrests - Smoothing_50
train
test
test prediction
Manhattan
Date


forecast_ny_arrests - Smoothing_75
Scale-dependent error
Percentage error
RMSE
MAE
MAPE
R2
3
3
37
37
0.44
0.83
0.86
train
test
train
test


forecast_ny_arrests - Smoothing_75
train
test
test prediction
Manhattan
Date

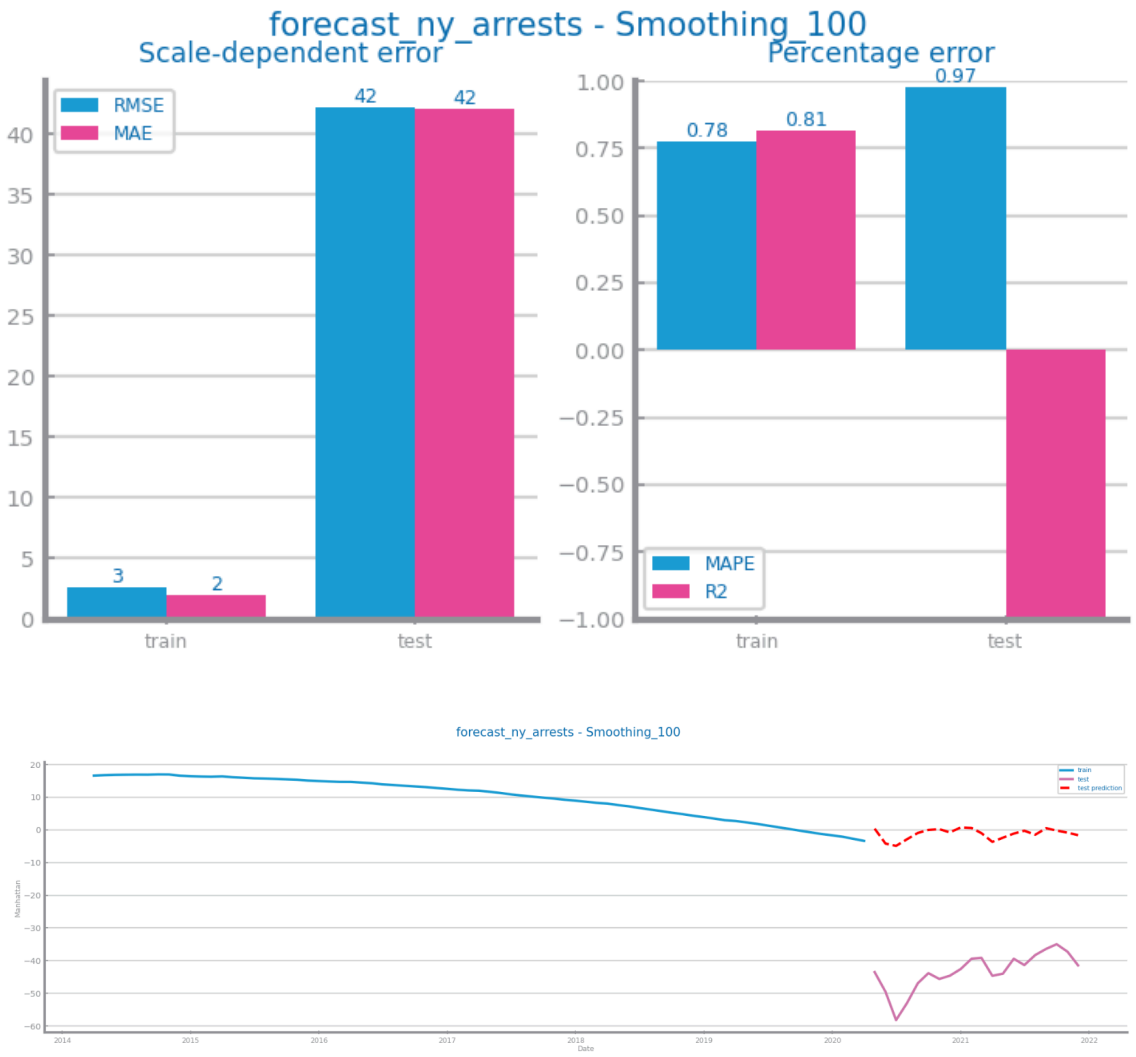
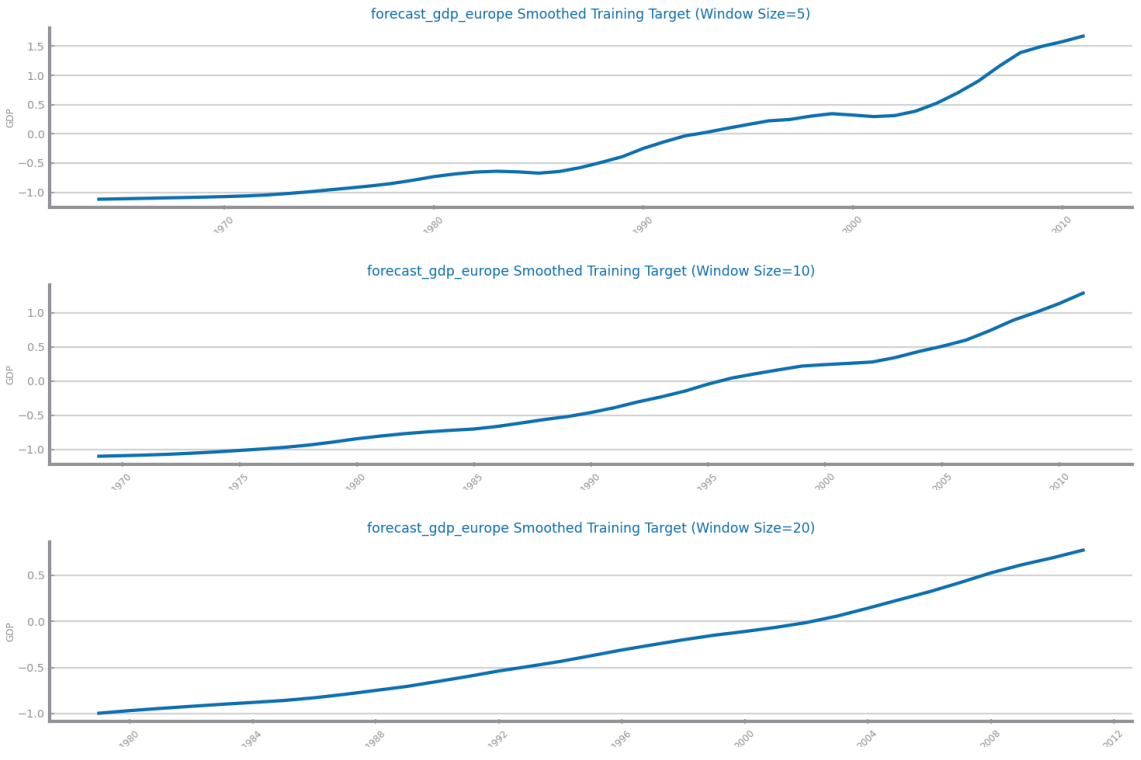



*Figure 87 Forecasting results after different smoothing parameterisations on time series 1*



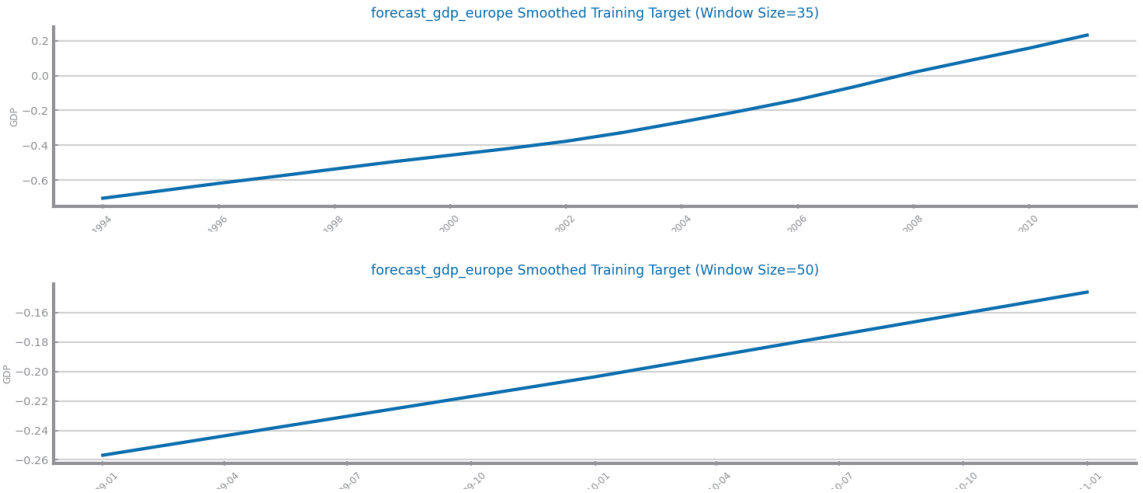
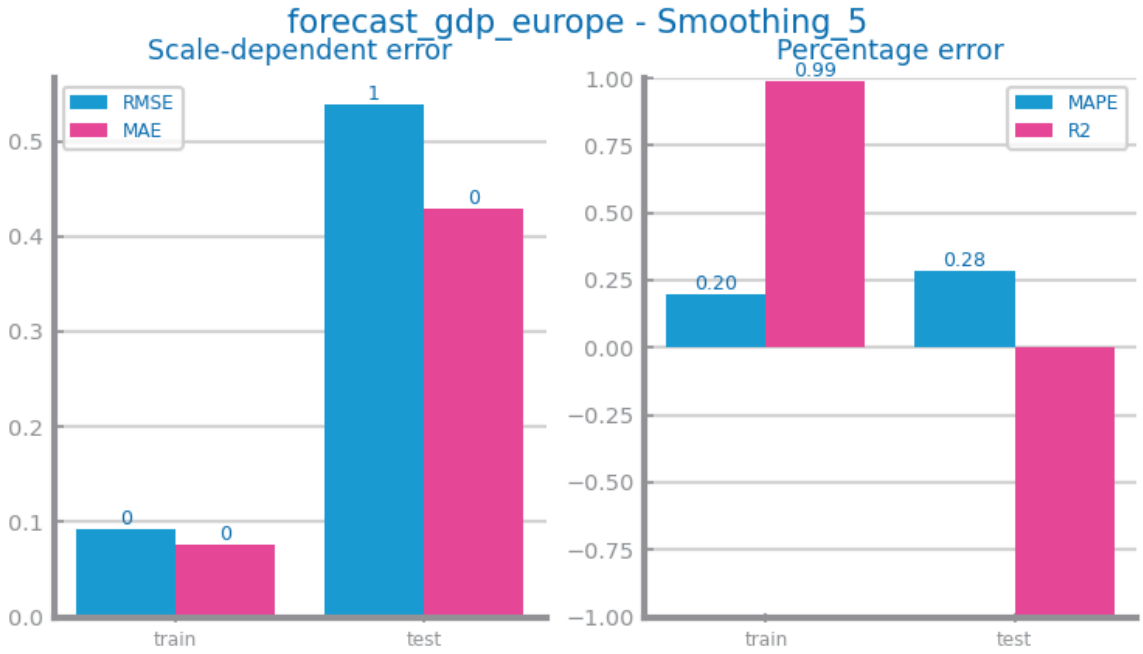
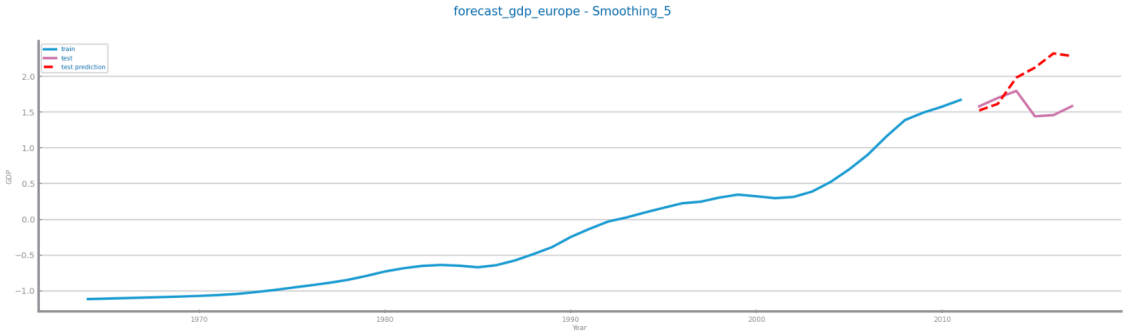
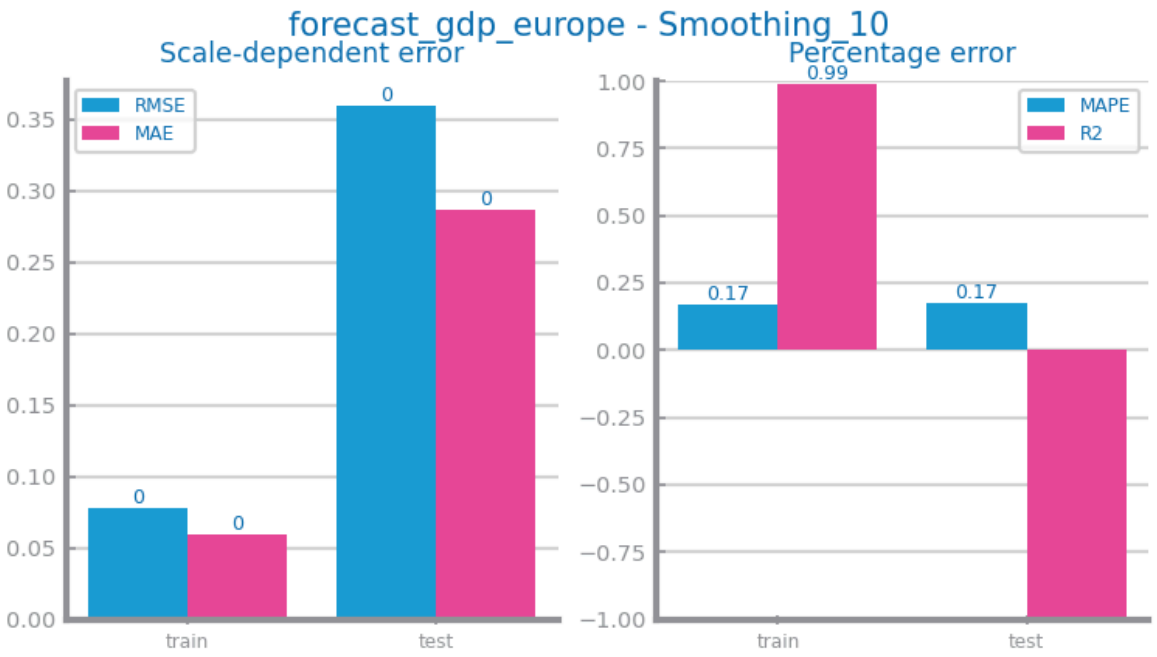
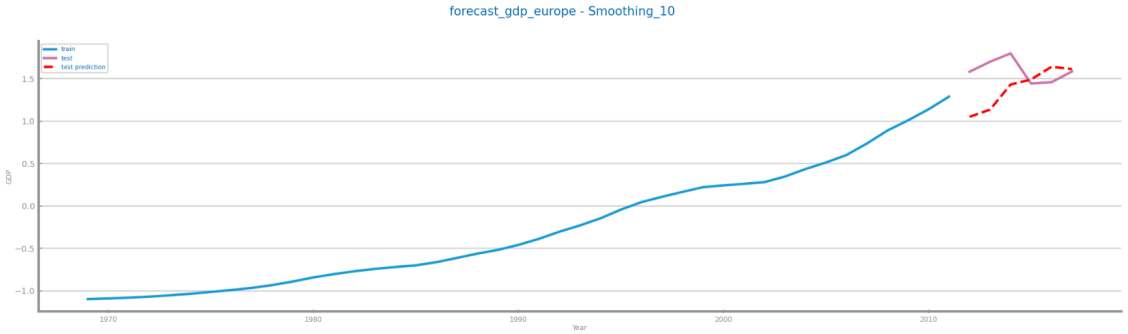


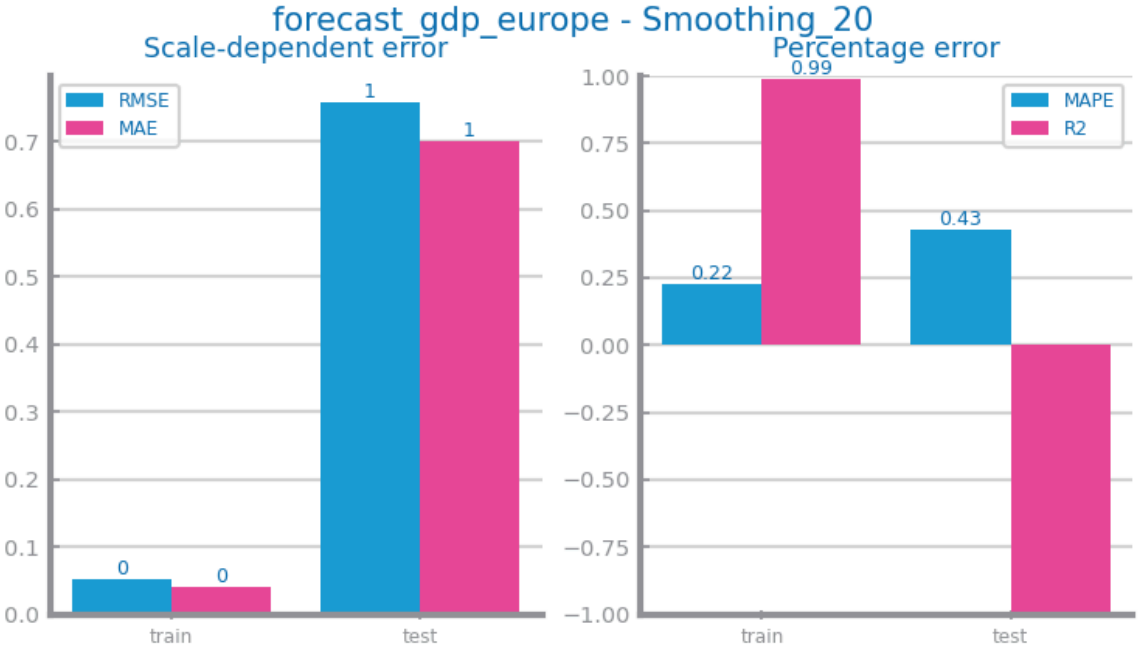

*Figure 88 Forecasting plots after different smoothing parameterisations on time series 2*

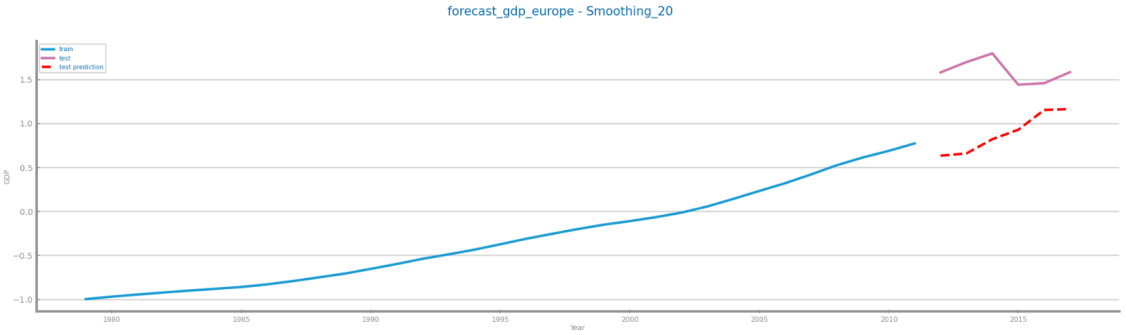


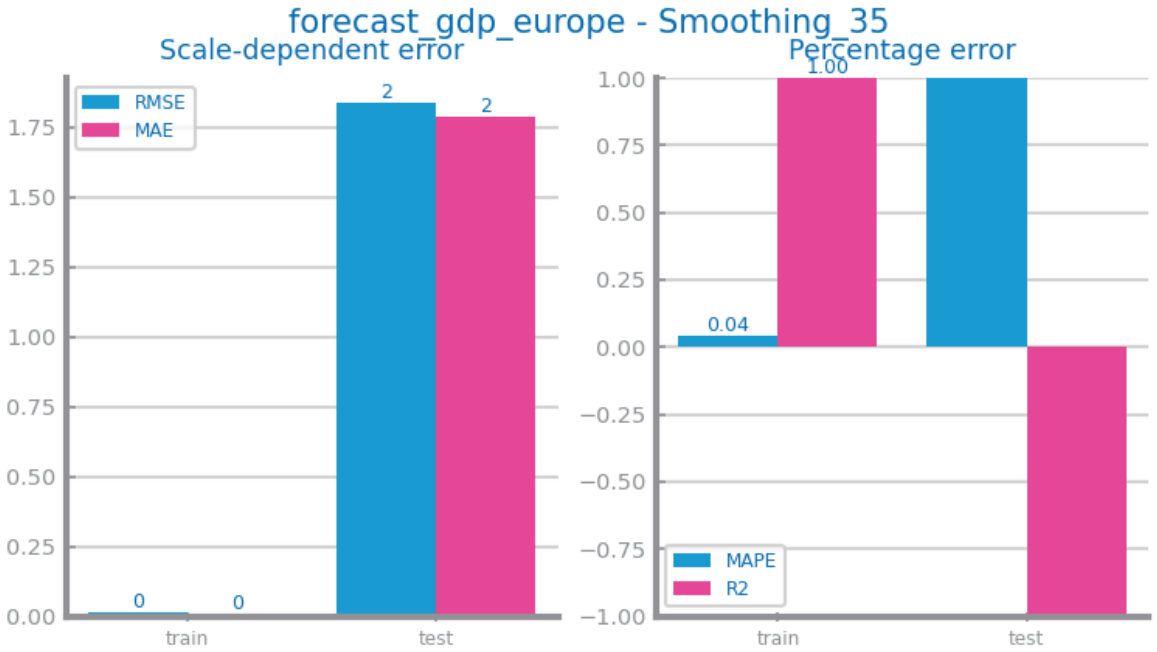



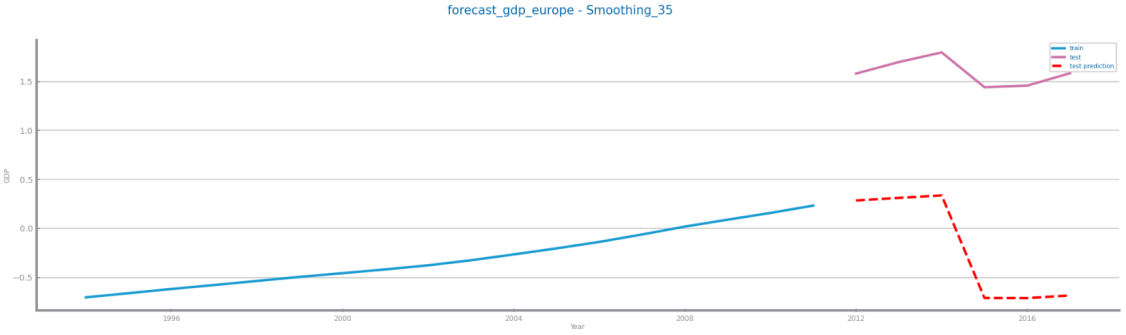

forecast_gdp_europe - Smoothing_10
Scale-dependent error
Percentage error
RMSE
MAE
MAPE
R2
0
0
0
0
0.17
0.99
0.17
train
test
train
test


forecast_gdp_europe - Smoothing_10
train
test
test prediction
GDP
Year


forecast_gdp_europe - Smoothing_20
Scale-dependent error
Percentage error
RMSE
MAE
MAPE
R2
0
0
1
1
0.22
0.99
0.43
train
test
train
test


forecast_gdp_europe - Smoothing_20
train
test
test prediction
GDP
Year


forecast_gdp_europe - Smoothing_35
Scale-dependent error
RMSE
MAE
0
0
2
2
train
test
Percentage error
1.00
0.04
MAPE
R2
train
test


forecast_gdp_europe - Smoothing_35
train
test
test prediction
GDP
Year

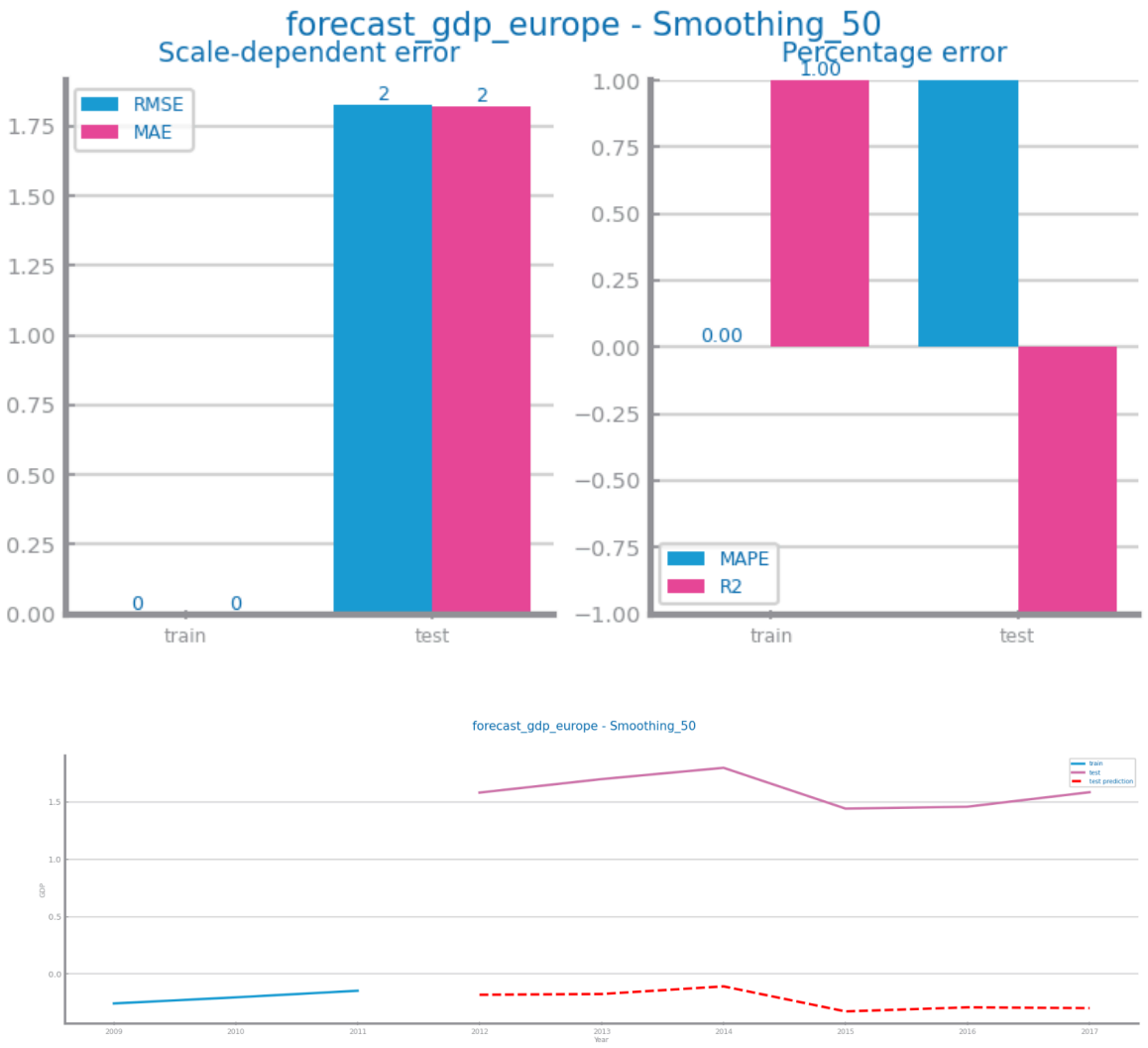


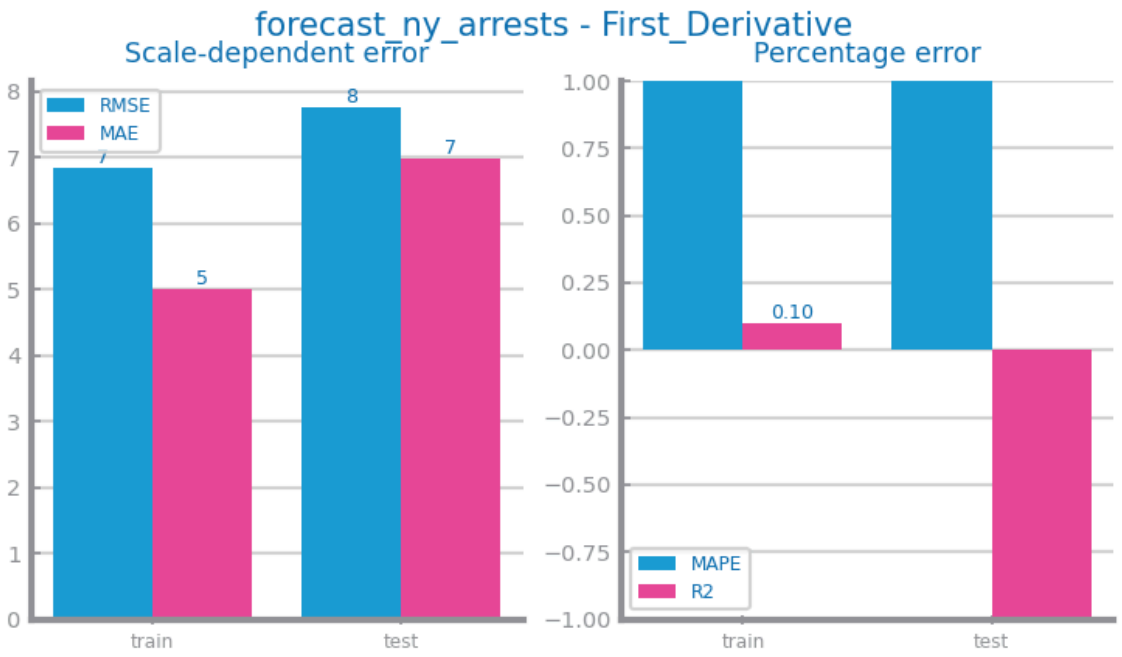

*Figure 89 Forecasting results after different smoothing parameterisations on time series 2*

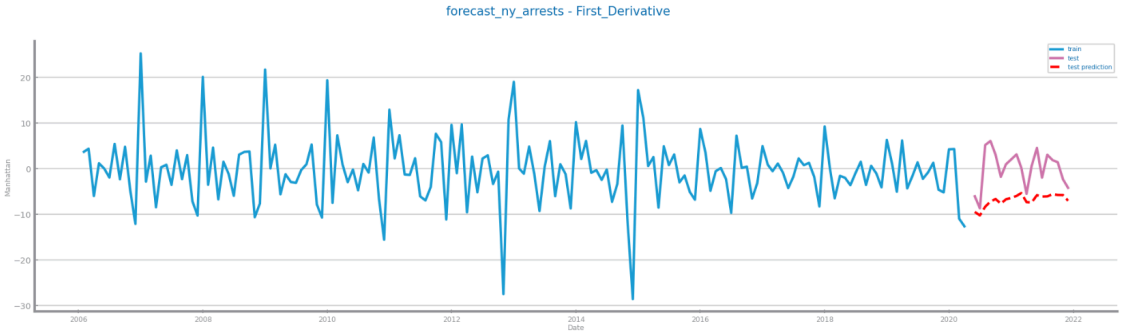
### *Differentiation*

Dataset 1: Differentiation was **not applied**, as the original data yielded better performance in evaluation.

Dataset 2: The **second derivative** was selected, as it improved evaluation results compared to the first derivative and the original. It also improves the model performance in the Model Evaluation part.


*Figure 90 Forecasting plots after first and second differentiation of time series 1*


forecast_ny_arrests - First_Derivative
Scale-dependent error
Percentage error
RMSE
MAE
MAPE
R2
7
5
8
7
0.10
train
test
train
test


forecast_ny_arrests - First_Derivative
train
test
test prediction
Manhattan
Date

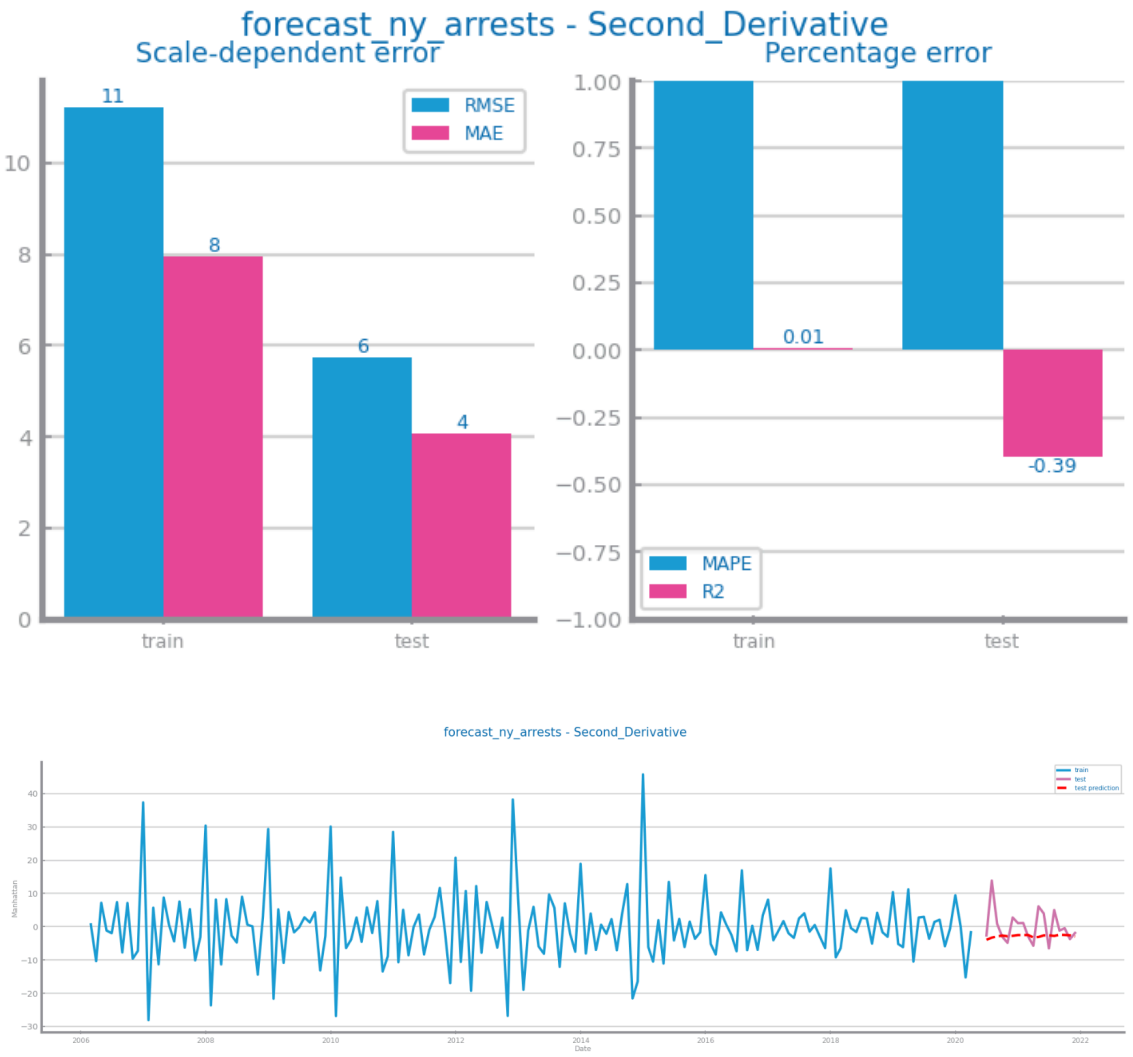



*Figure 91 Forecasting results after first and second differentiation of time series 1*

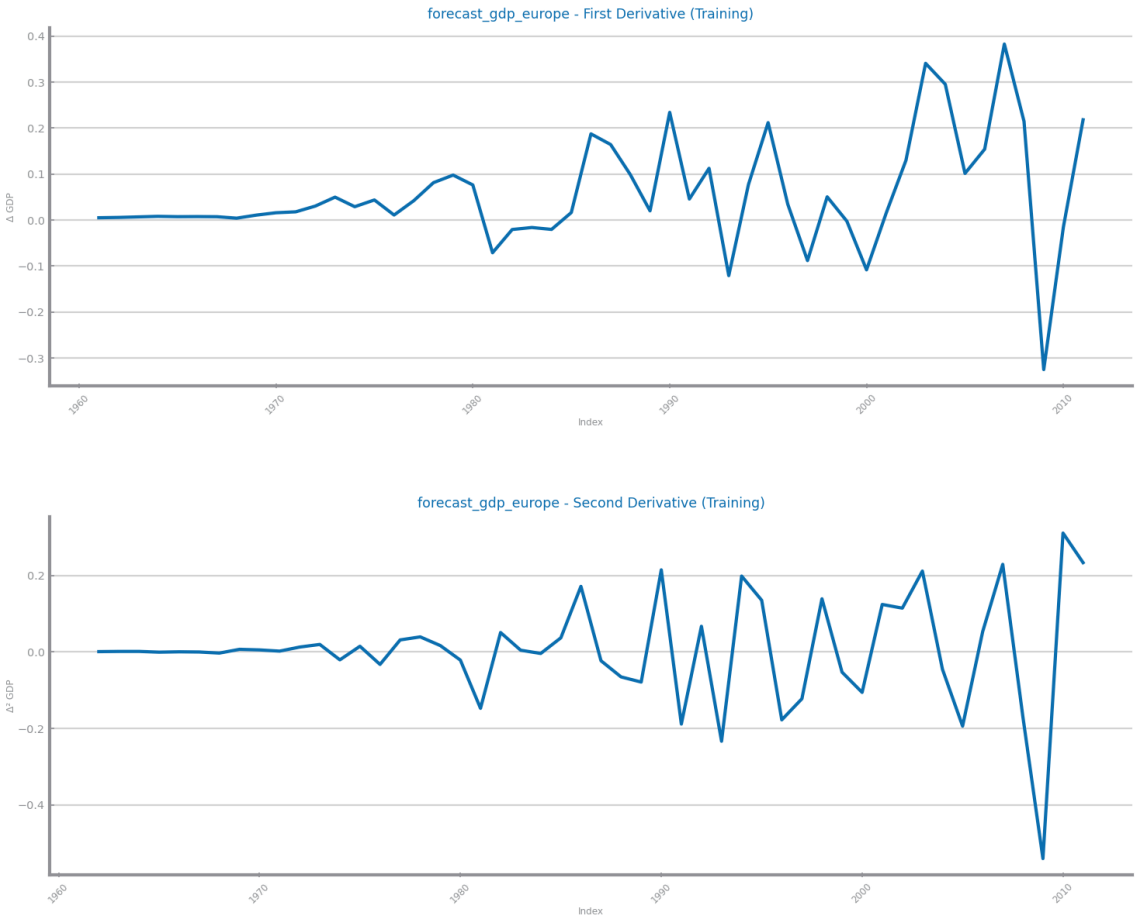
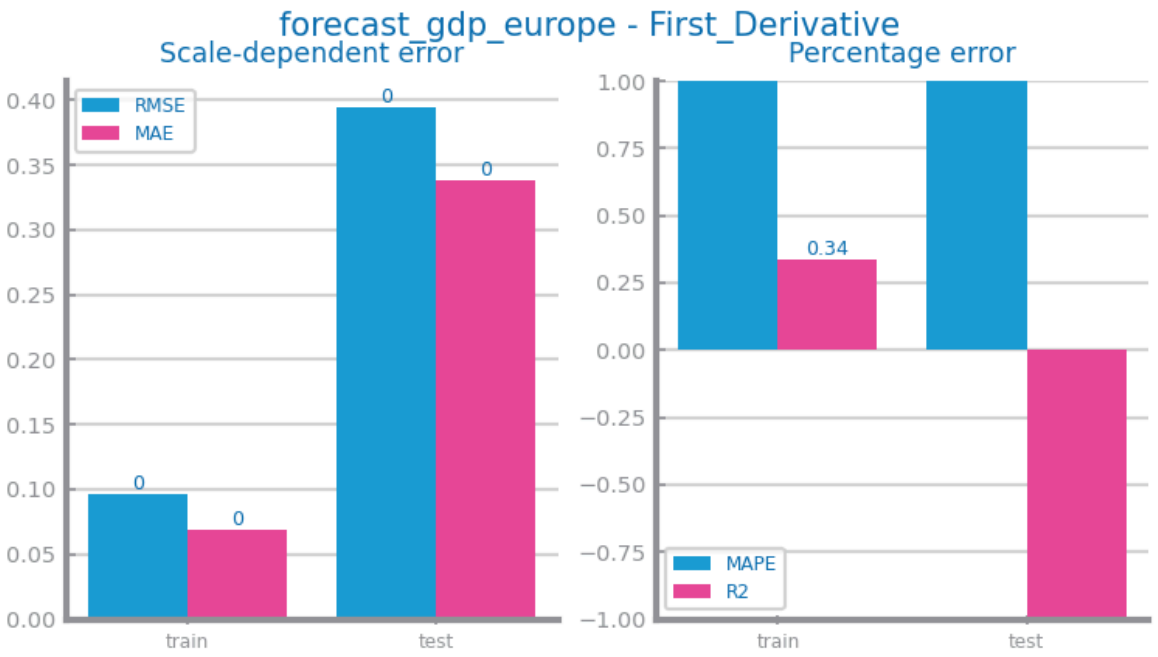
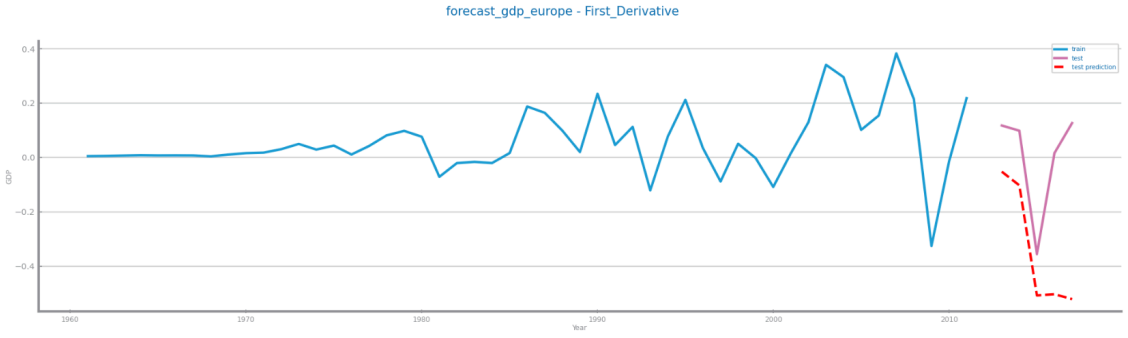



*Figure 92 Forecasting plots after first and second differentiation of time series 2*


forecast_gdp_europe - First_Derivative
Scale-dependent error
Percentage error
RMSE
MAE
MAPE
R2
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0
0
0
0
0.34
train
test
train
test


forecast_gdp_europe - First_Derivative
GDP
Year
1960
1970
1980
1990
2000
2010

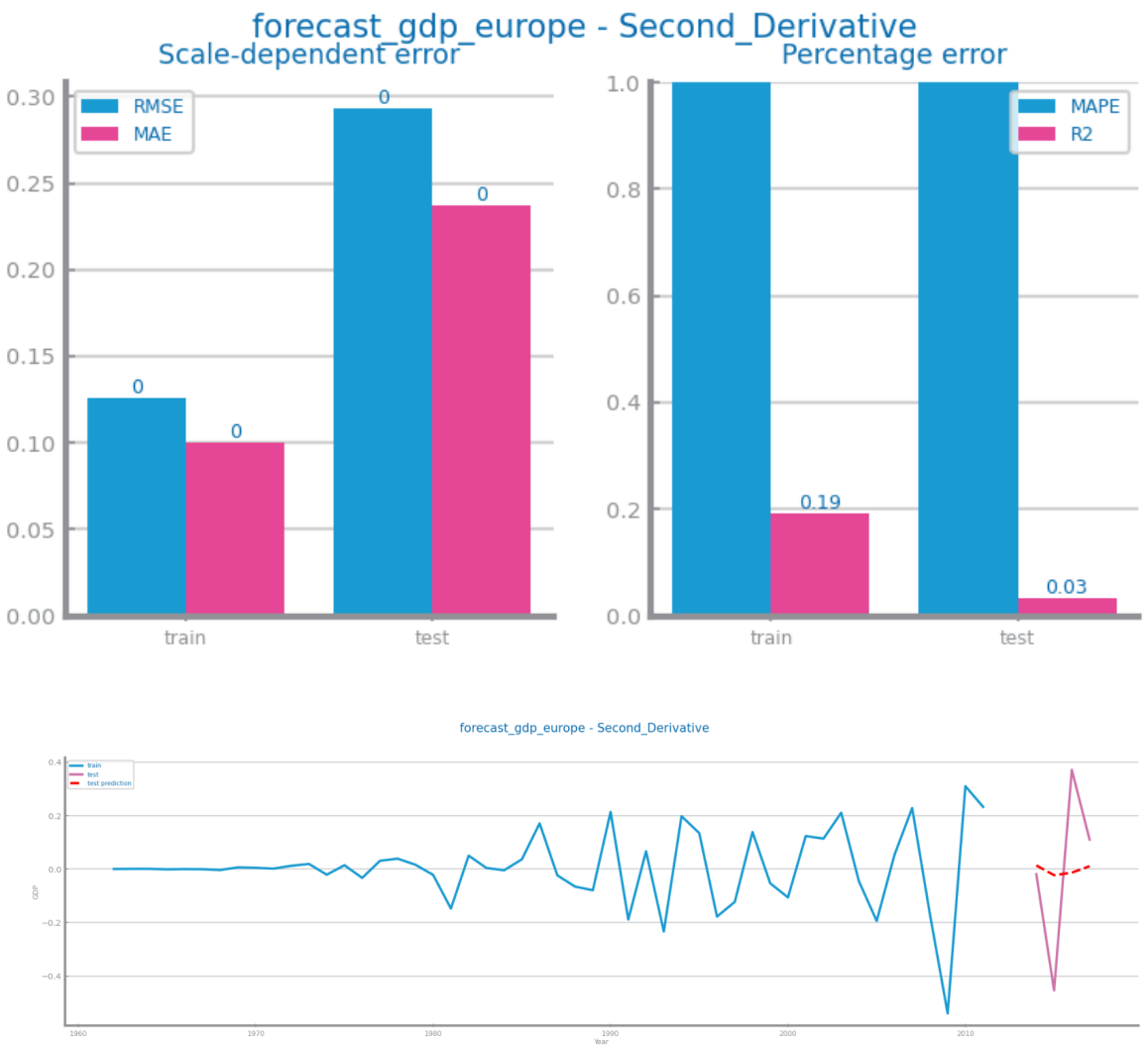



*Figure 93 Forecasting results after first and second differentiation of time series 2*

## *Other transformations (optional)*

Mean imputation was applied to handle missing values in both datasets, ensuring data consistency for model training. Scaling was also performed on both datasets to normalize the data even if it did not lead to any changes in the evaluation results of the evaluation.

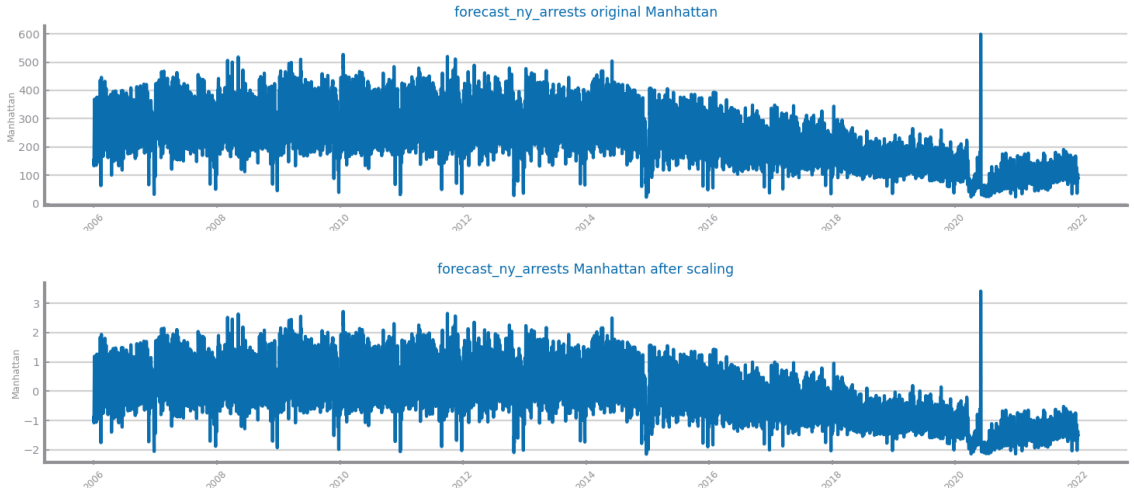
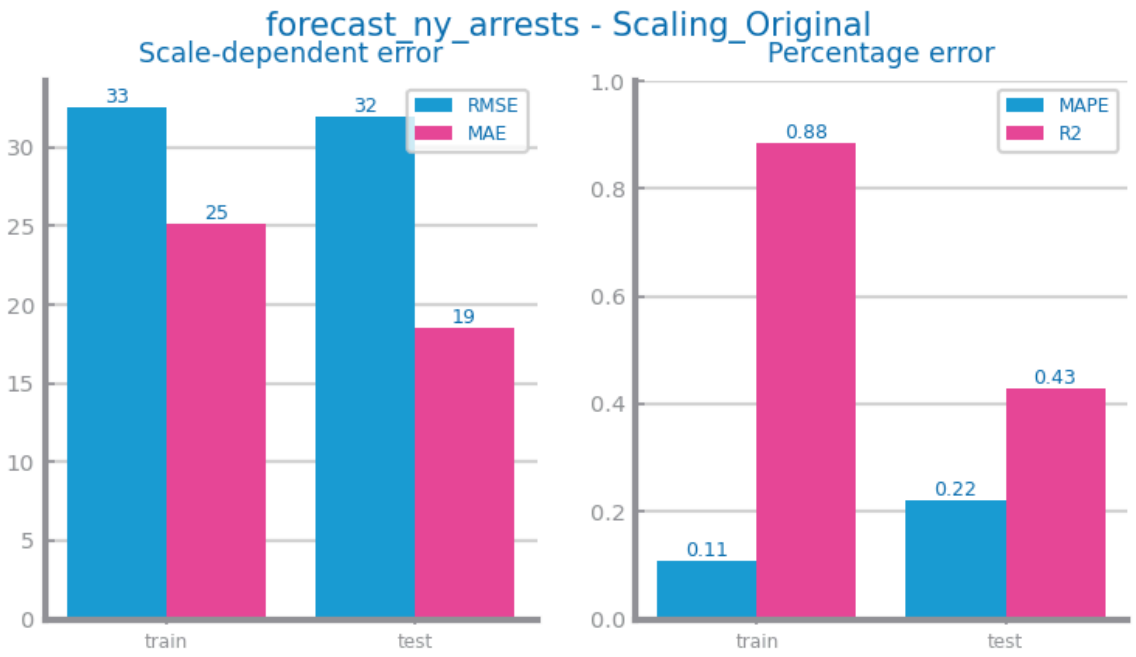
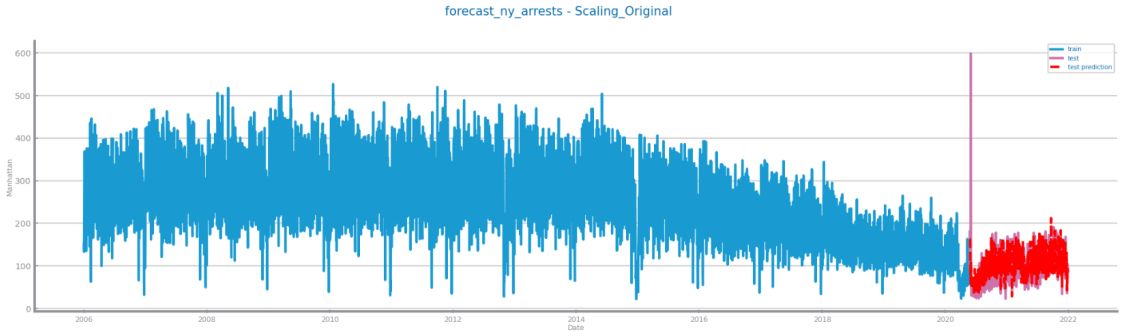



*Figure 94 Forecasting plots after applying other transformations over time series 1*


forecast_ny_arrests - Scaling_Original
Scale-dependent error
Percentage error
RMSE
MAE
MAPE
R2
33
25
32
19
0.11
0.88
0.22
0.43
train
test
0
5
10
15
20
25
30
0.0
0.2
0.4
0.6
0.8
1.0


forecast_ny_arrests - Scaling_Original
train
test
test prediction
Manhattan
Date
2006
2008
2010
2012
2014
2016
2018
2020
2022

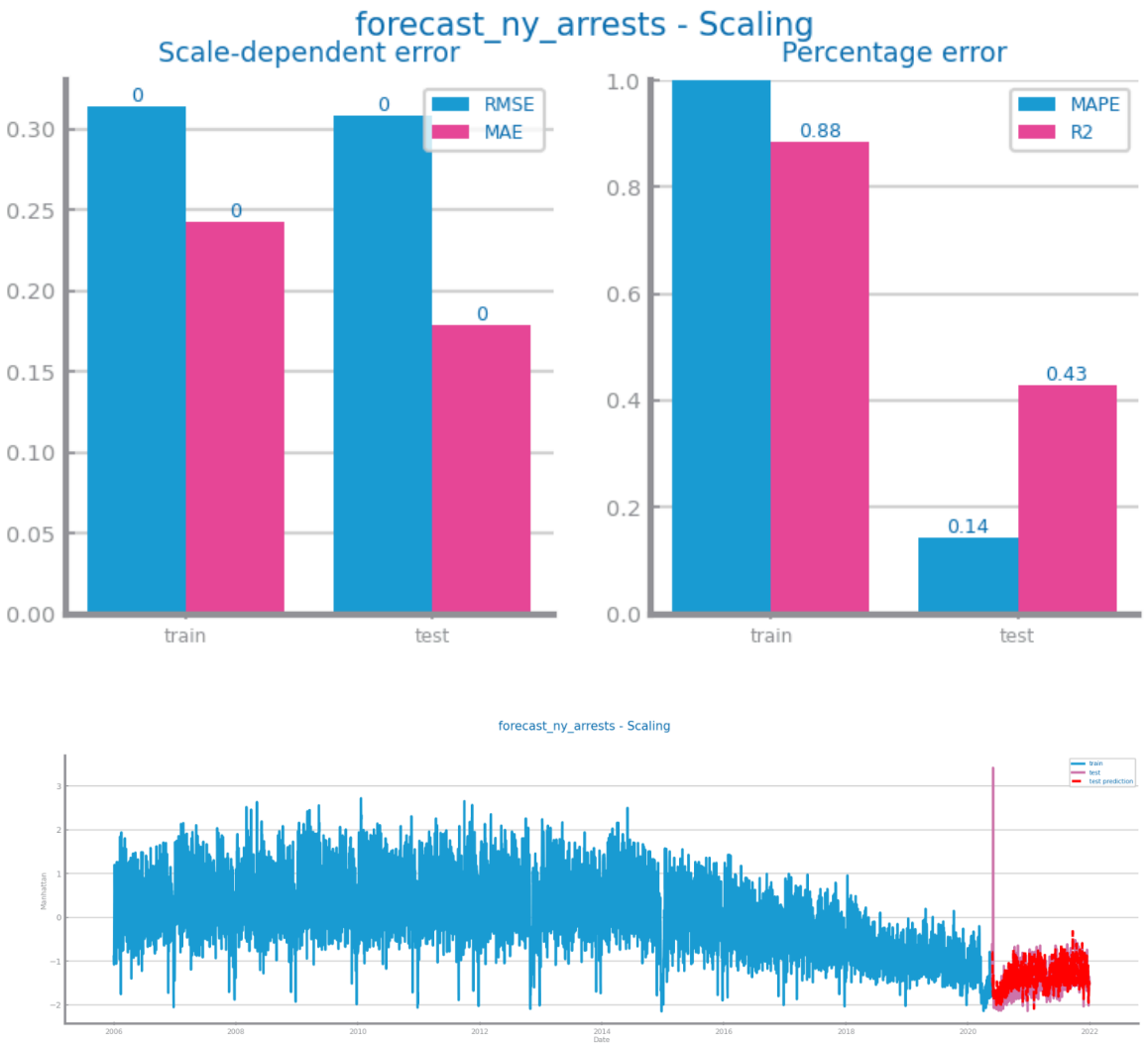



*Figure 95 Forecasting results after applying other transformations over time series 1*

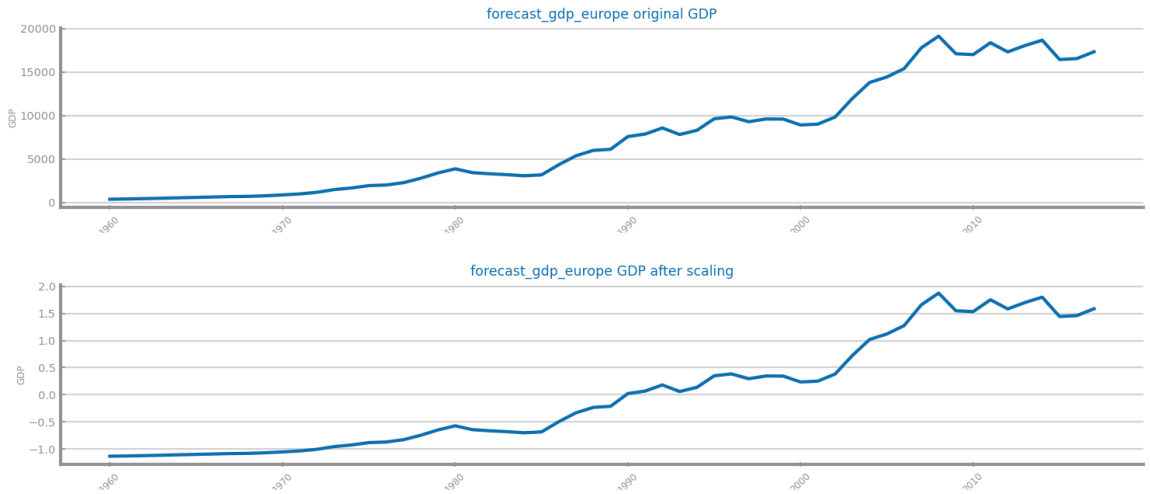
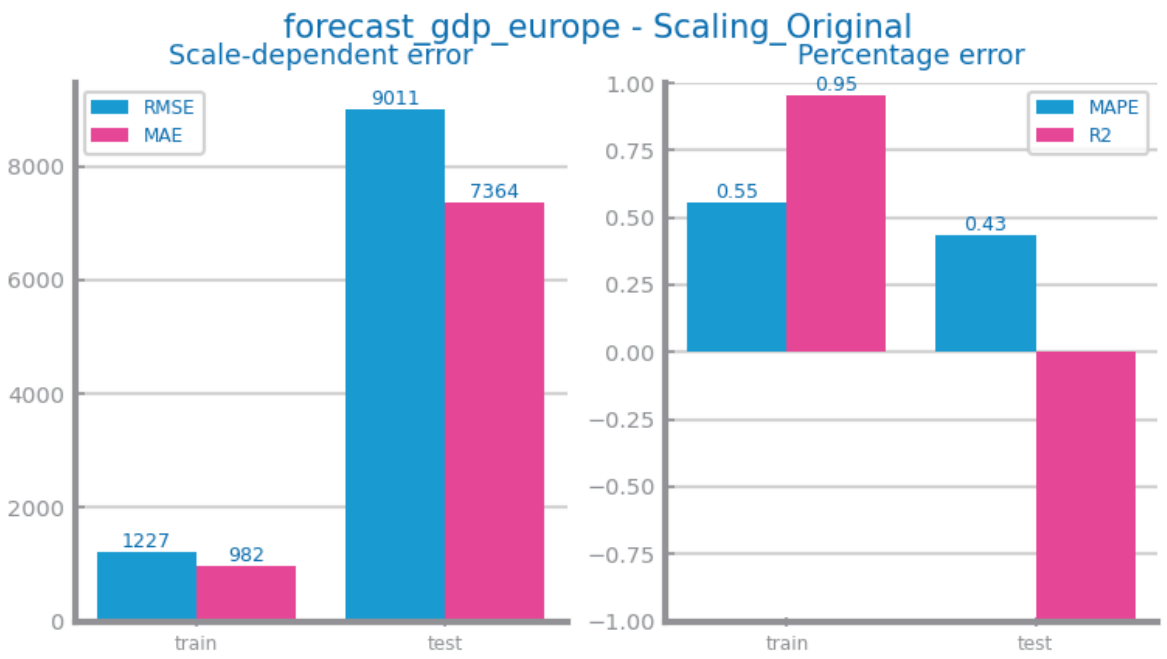



*Figure 96 Forecasting plots after applying other transformations over time series 2*


forecast_gdp_europe - Scaling_Original
Scale-dependent error
Percentage error
RMSE
MAE
MAPE
R2
9011
7364
1227
982
0.95
0.55
0.43
8000
6000
4000
2000
0
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
train
test
train
test


forecast_gdp_europe - Scaling_Original
GDP
Year
train
test
test prediction
20000
17500
15000
12500
10000
7500
5000
2500
0
1960
1970
1980
1990
2000
2010

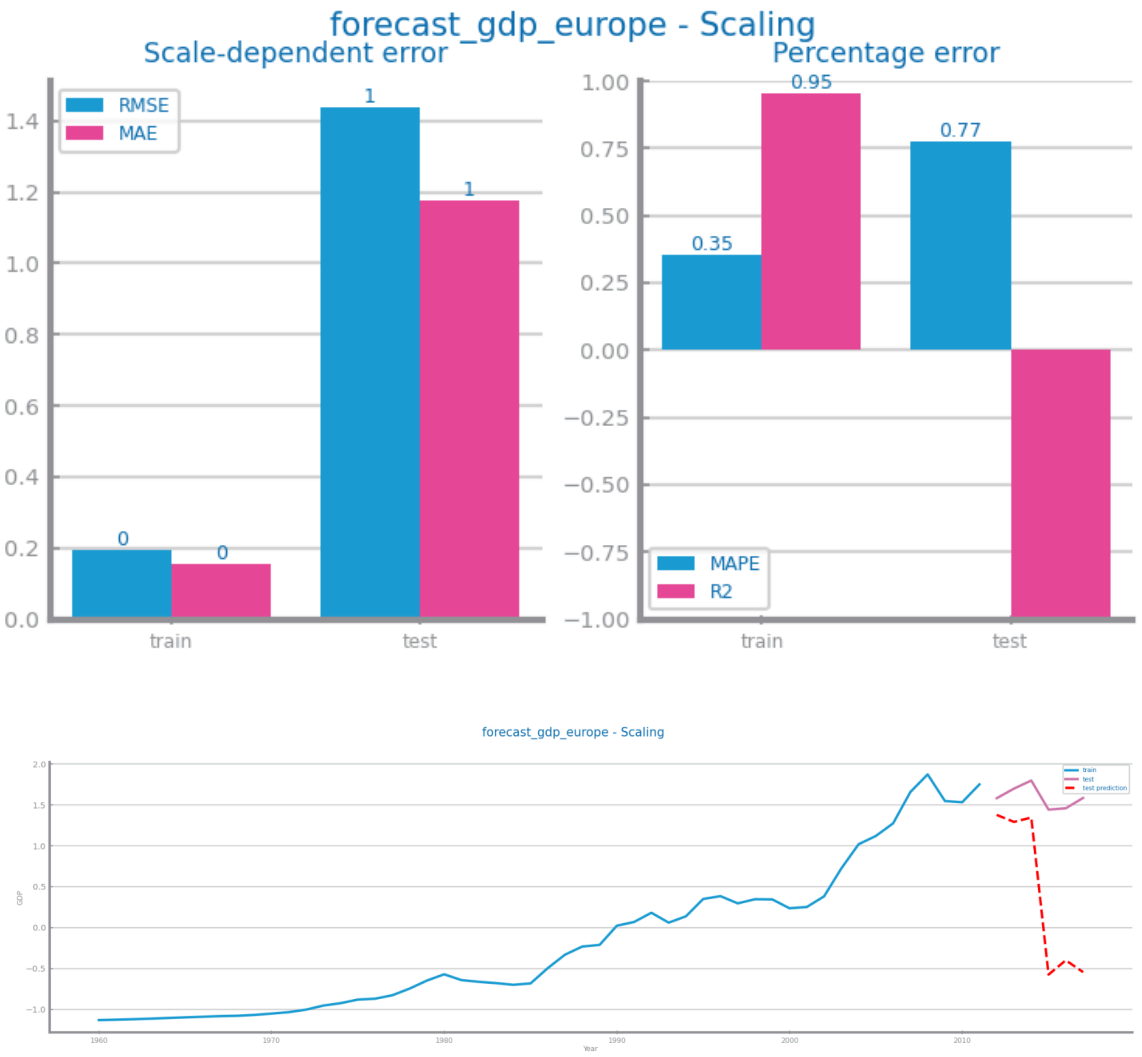



*Figure 97 Forecasting results after applying other transformations over time series 2*

# 7 Models' Evaluation

Dataset 1: Mean imputation and scaling were applied. Monthly aggregation was used, while no smoothing or differentiation was applied. The chosen measure to maximize is R2.

Dataset 2: Mean imputation and scaling were applied. Second differentiation was used, while no aggregation or smoothing was applied. The chosen measure to maximize is R2.

## *Simple Average Model*

The simple average model performed poorly on both datasets due to its simplicity. This bad performance is more notable on the first dataset since its trend is changing to the end of the train dataset.


*Figure 98 Forecasting plots obtained with Simple Average model over time series 1*

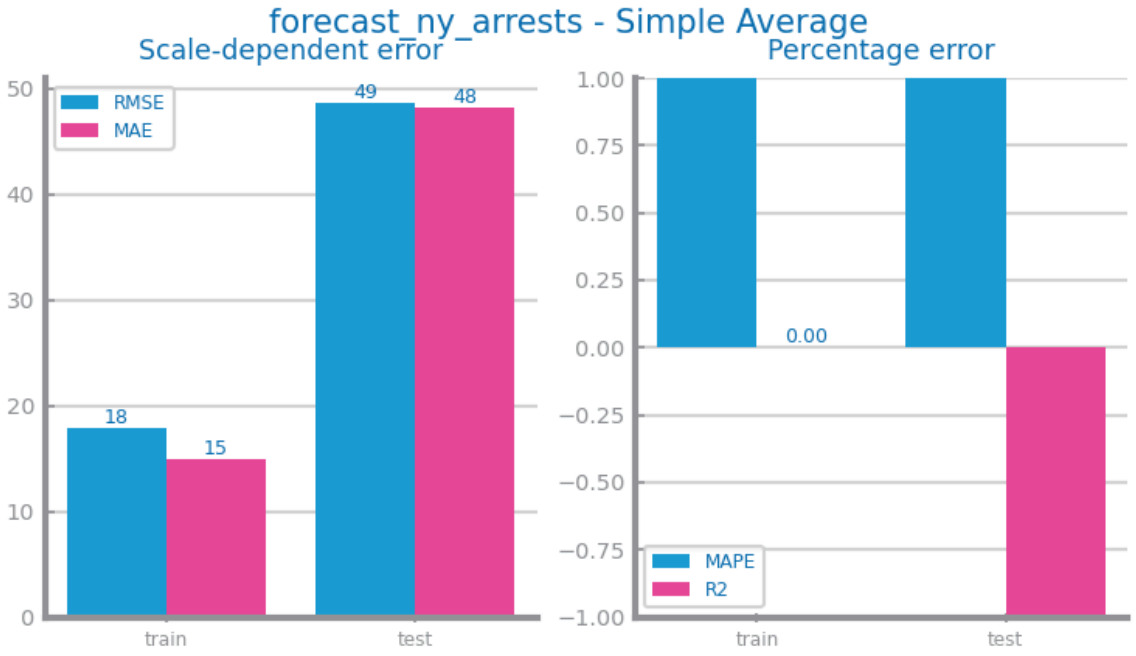



*Figure 99 Forecasting results obtained with Simple Average model over time series 1*

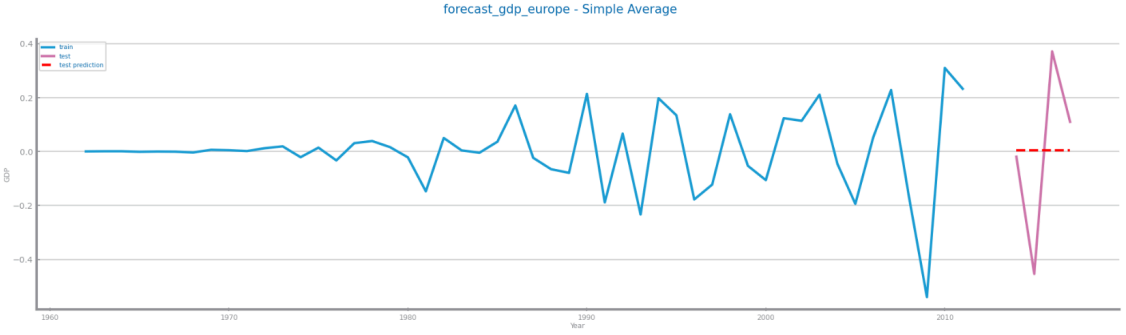



*Figure 100 Forecasting plots obtained with Simple Average model over time series 2*

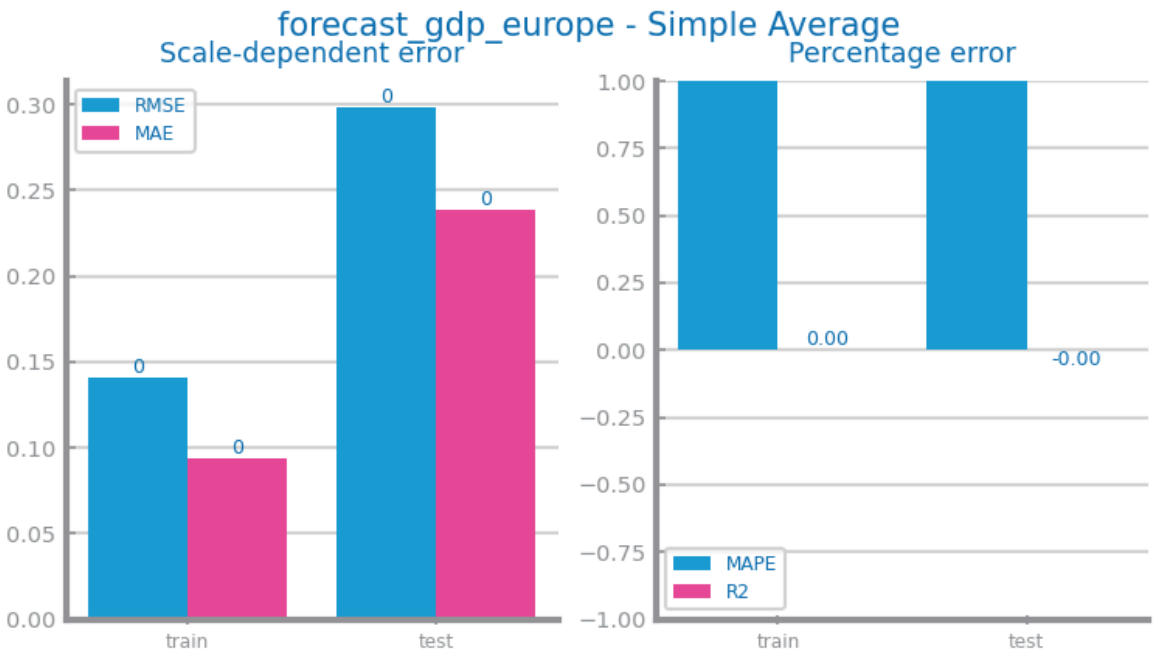



*Figure 101 Forecasting results obtained with Simple Average model over time series 2*

### *Persistence Model*

Dataset 1: Both persistence models performed poorly, with the optimist model showing the best results compared to the realist model. Despite the poor performance, the optimist model provided relatively better predictions.

Dataset 2: Overall performance was worse than on the first dataset, where both models struggled significantly. Surprisingly the realistic model achieved better performance compared to the optimistic model.


*Figure 102 Forecasting plots obtained with Persistence model (long term) over time series 1*


*Figure 103 Forecasting plots obtained with Persistence model (next point) over time series 1*


*Figure 104 Forecasting results obtained with Persistence model in both situations over time series 1*

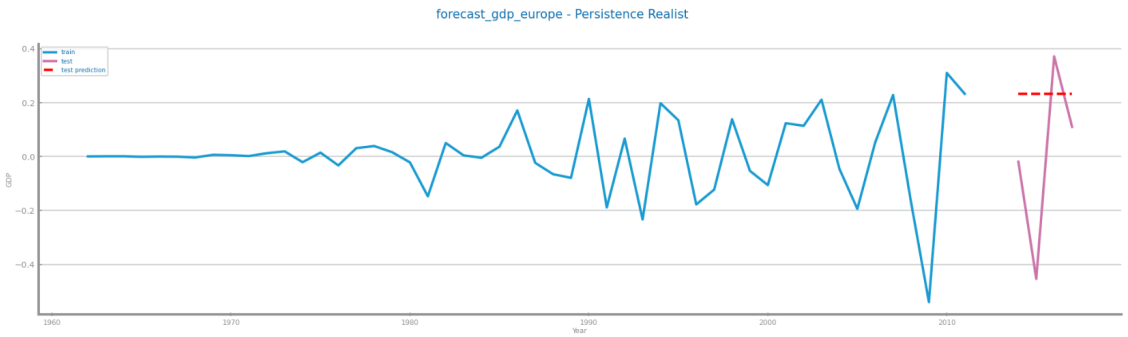



*Figure 105 Forecasting plots obtained with Persistence model (long term) over time series 2*

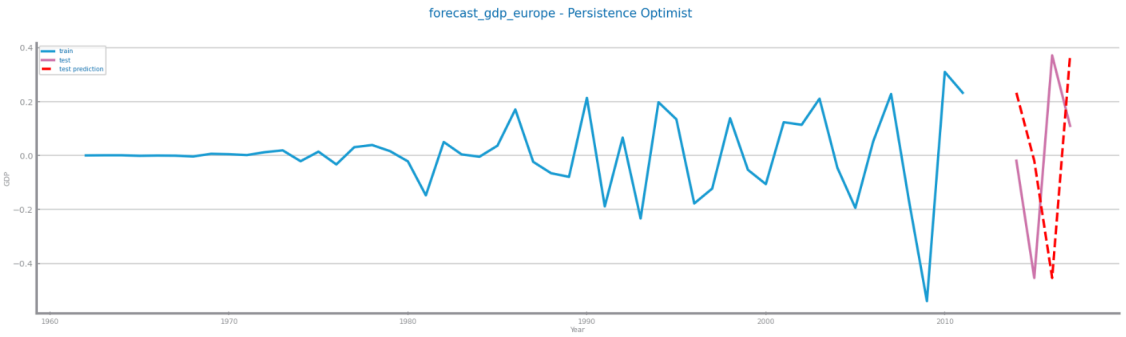



*Figure 106 Forecasting plots obtained with Persistence model (next point) over time series 2*


*Figure 107 Forecasting results obtained with Persistence model in both situation over time series 2*

### *Rolling Mean Model*

Dataset 1: The study of parameters showed that larger window sizes worsened performance, with the best results obtained using a window size of 3. However, smaller window sizes caused instability in the forecasts likely for not being able to capture trends, with this the results were poor overall, possibly due to overfitting.

Dataset 2: Parameter study was inconclusive, likely due to the small test size, but a smaller window size of 3 was assumed to be optimal. Similar to the first dataset, results were poor, indicating consistent limitations of the method.

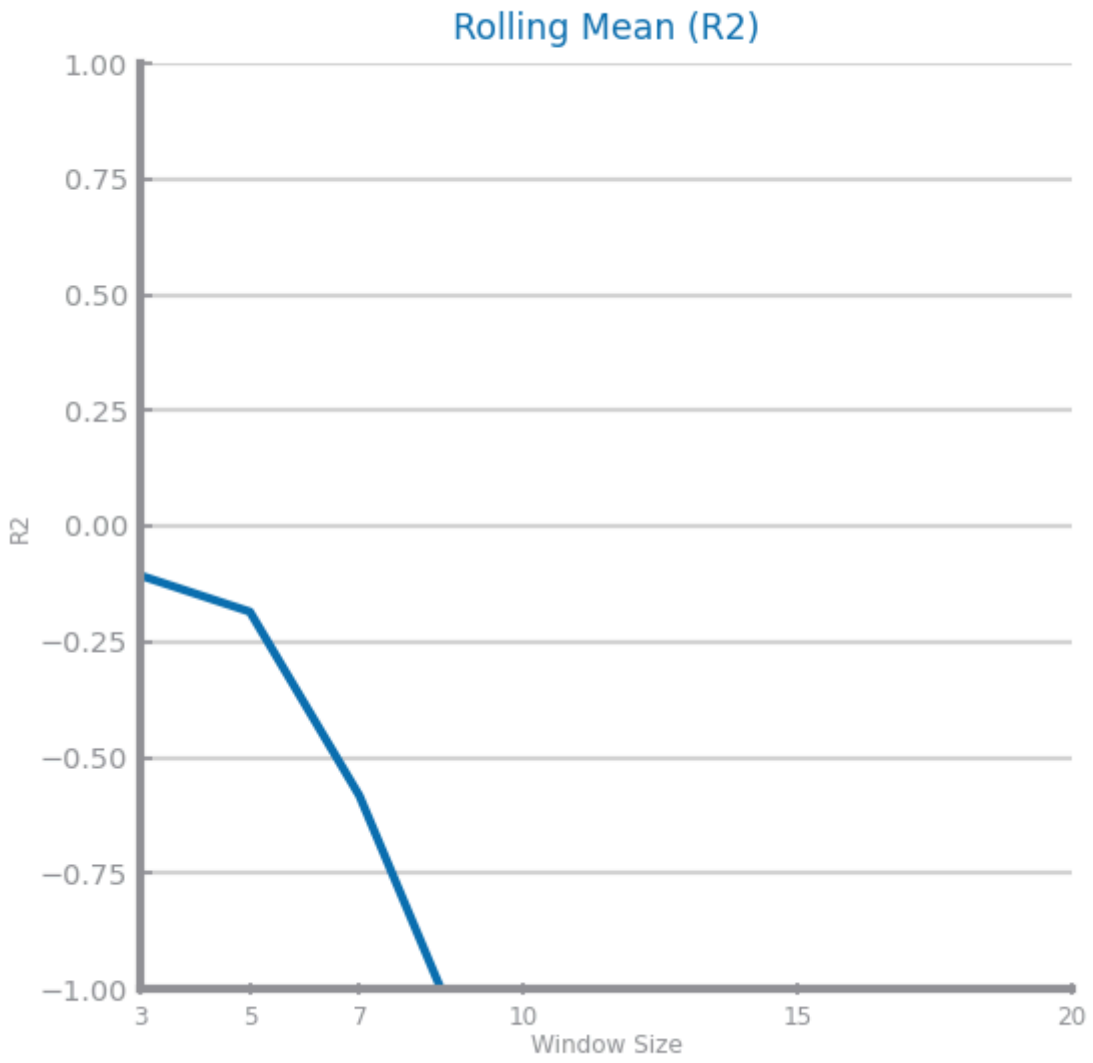



*Figure 108 Forecasting study over different parameterisations of the Rolling Mean algorithm over time series 1*


*Figure 109 Forecasting plots obtained with the best parameterisation of Rolling Mean algorithm, over time series 1*

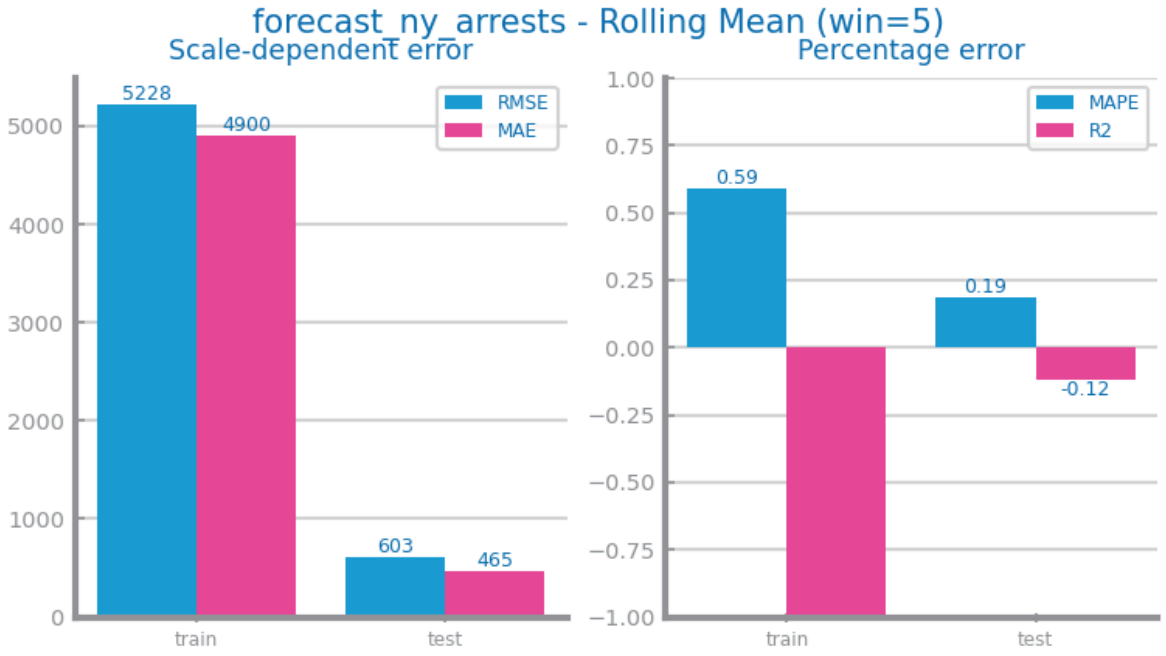



*Figure 110 Forecasting results obtained with the best parameterisation of Rolling Mean algorithm, over time series 1*

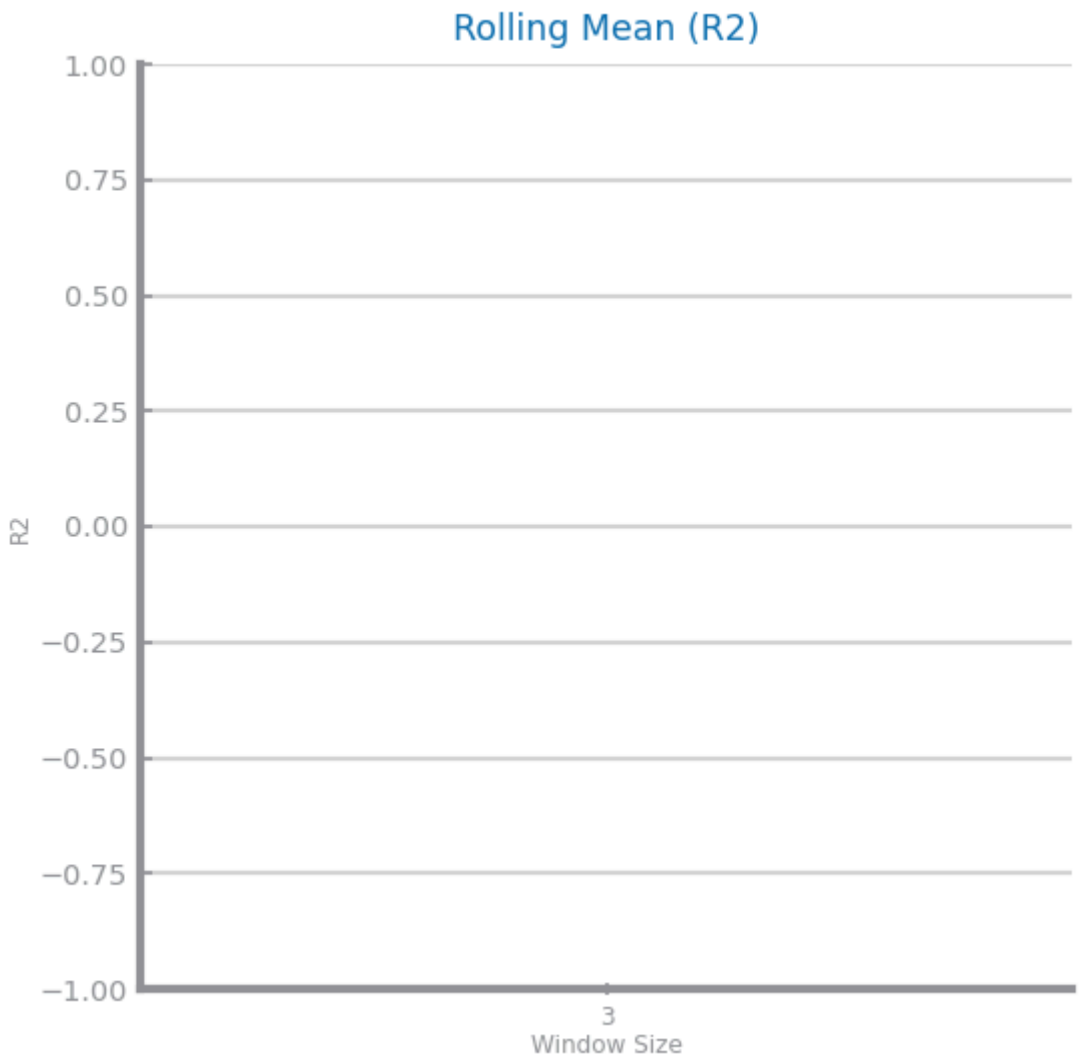



*Figure 111 Forecasting study over different parameterisations of the Rolling Mean algorithm over time series 2*

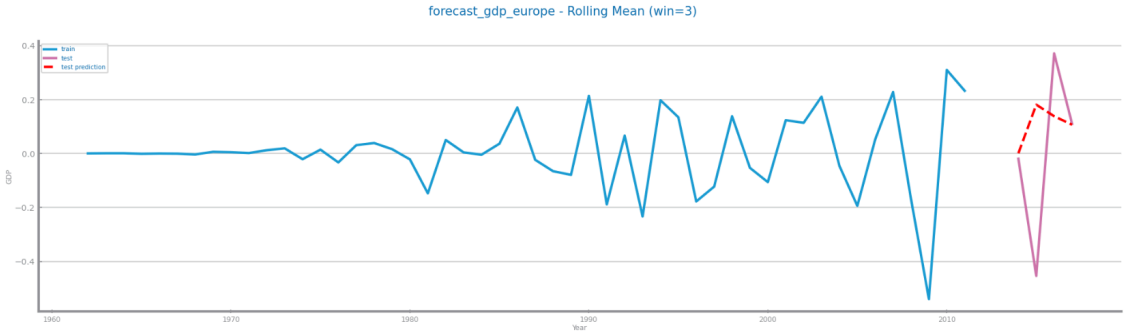



*Figure 112 Forecasting plots obtained with the best parameterisation of Rolling Mean algorithm, over time series 2*

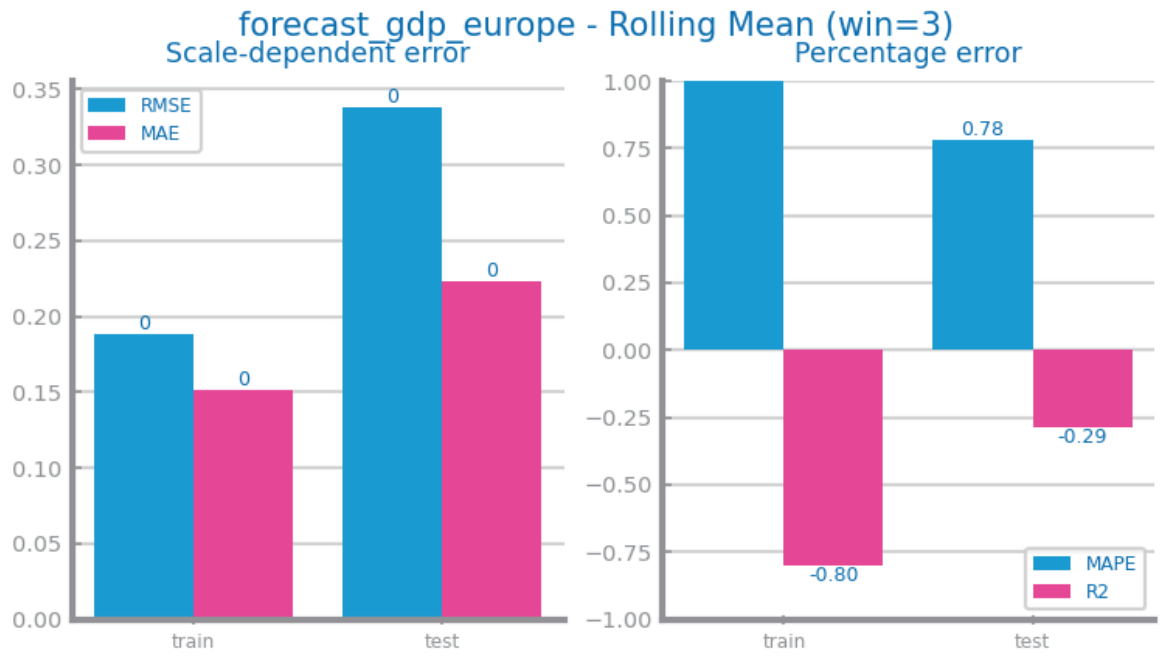



*Figure 113 Forecasting results obtained with the best parameterisation of Rolling Mean algorithm, over time series 2*

### *Exponential Smoothing*

Dataset 1: The optimal alpha value was 0.4, with R2 decreasing for values below and above, likely because smaller alphas smoothed too much and larger ones overreacted to noise. We suppose the results were poor due to the model's inability to capture complex patterns.

Dataset 2: An alpha of 0.1 was optimal, with higher values reducing R2 likely by overemphasizing short-term changes. Test results were similar to Dataset 1, with poor performance linked to the model's simplicity and dataset variability.

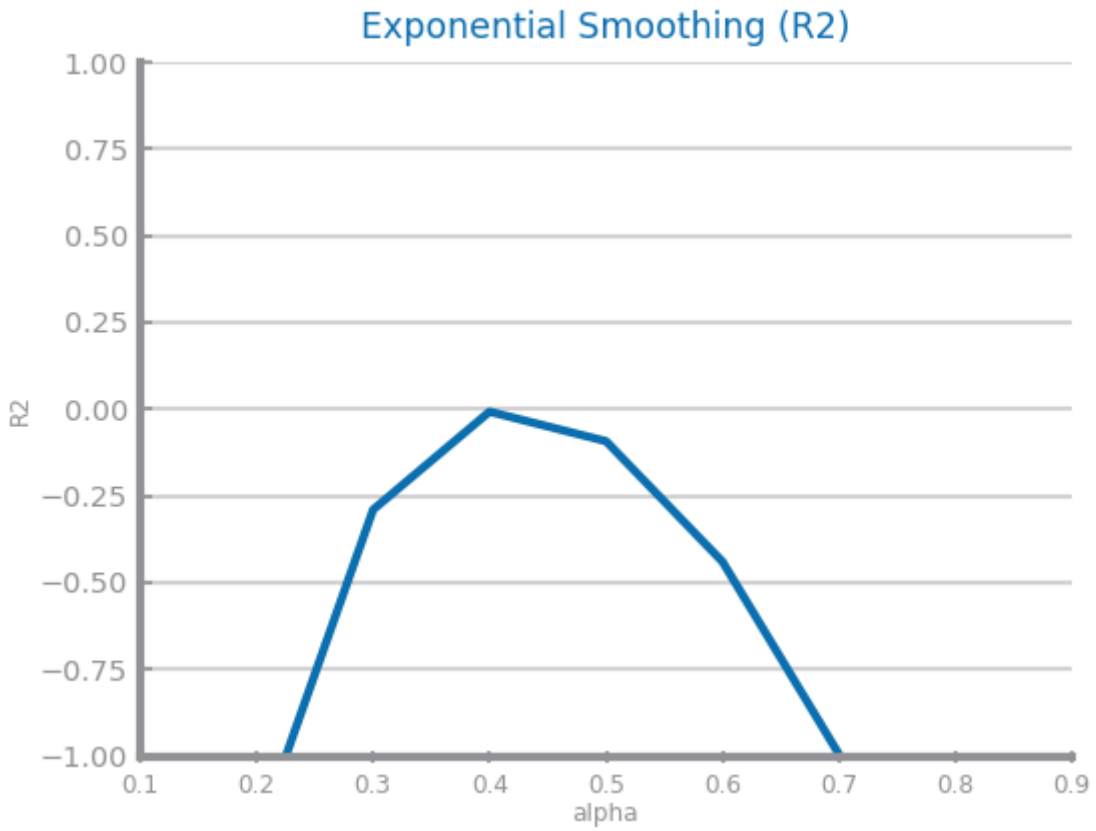



*Figure 114 Forecasting study over different parameterisations of the Exponential Smoothing algorithm over time series 1*


*Figure 115 Forecasting plots obtained with the best parameterisation of Exponential Smoothing algorithm, over time series 1*

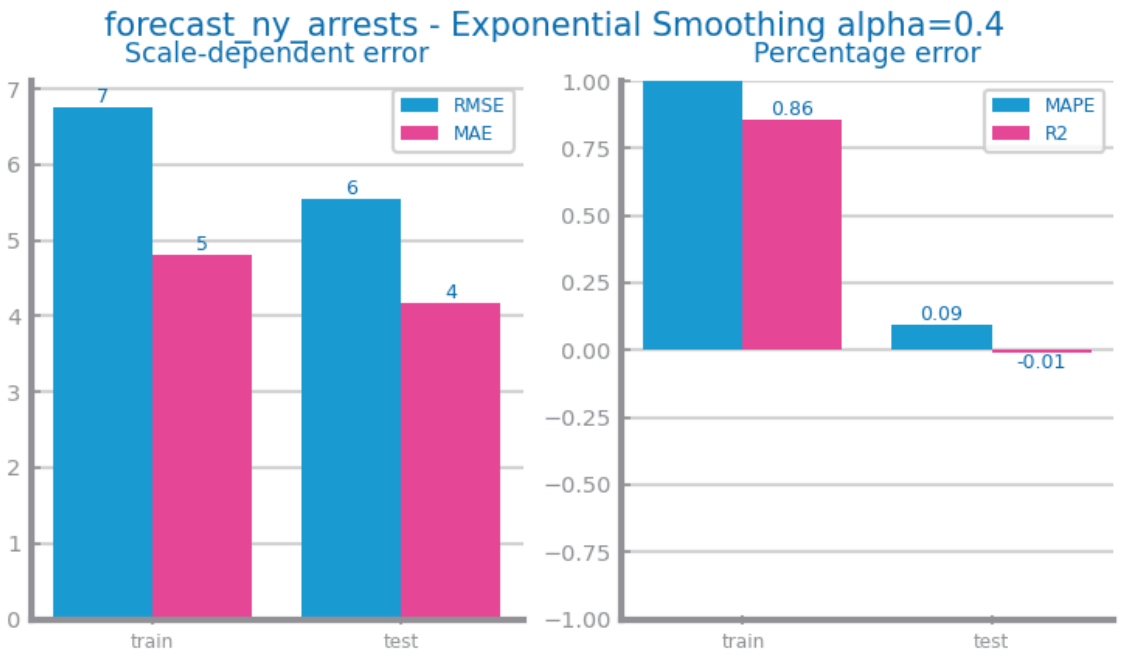



*Figure 116 Forecasting results obtained with the best parameterisation of Exponential Smoothing algorithm, over time series 1*

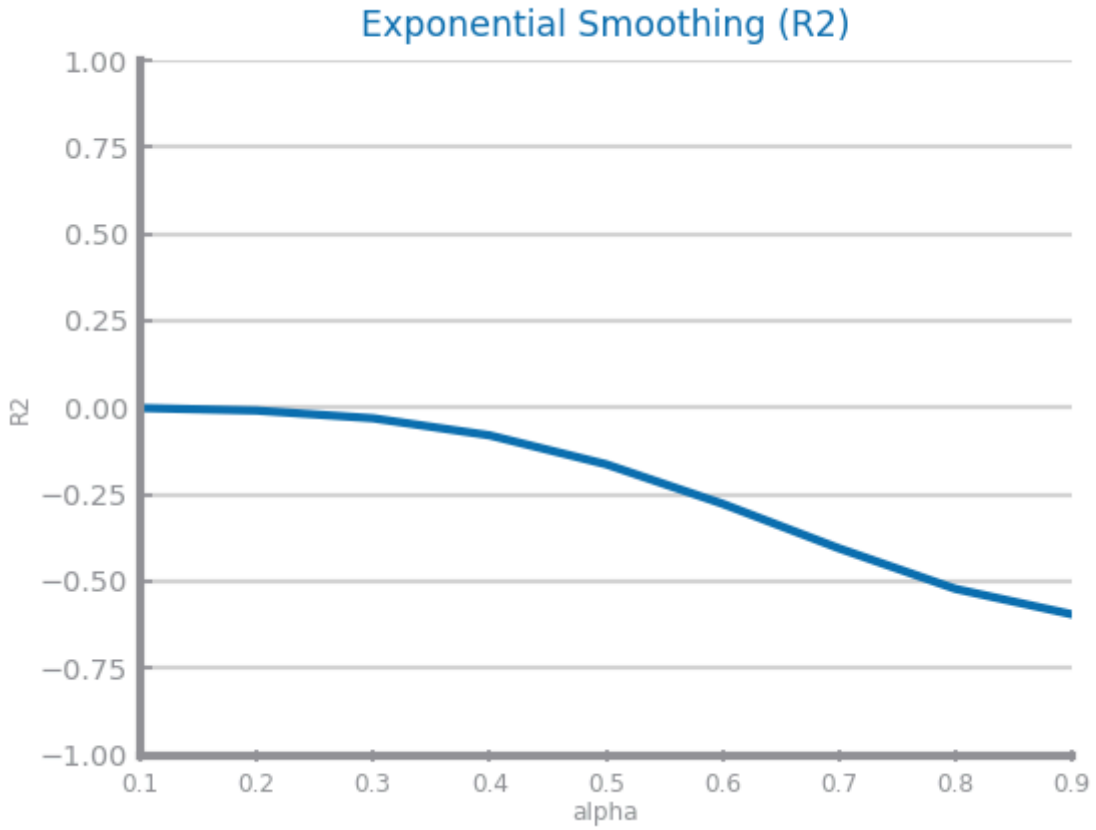



*Figure 117 Forecasting study over different parameterisations of the Exponential Smoothing algorithm over time series 2*

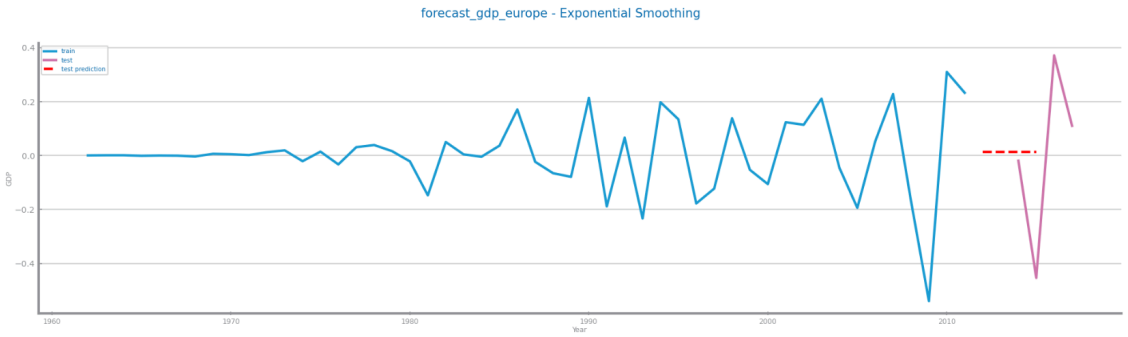



*Figure 118 Forecasting plots obtained with the best parameterisation of Exponential Smoothing, over time series 2*

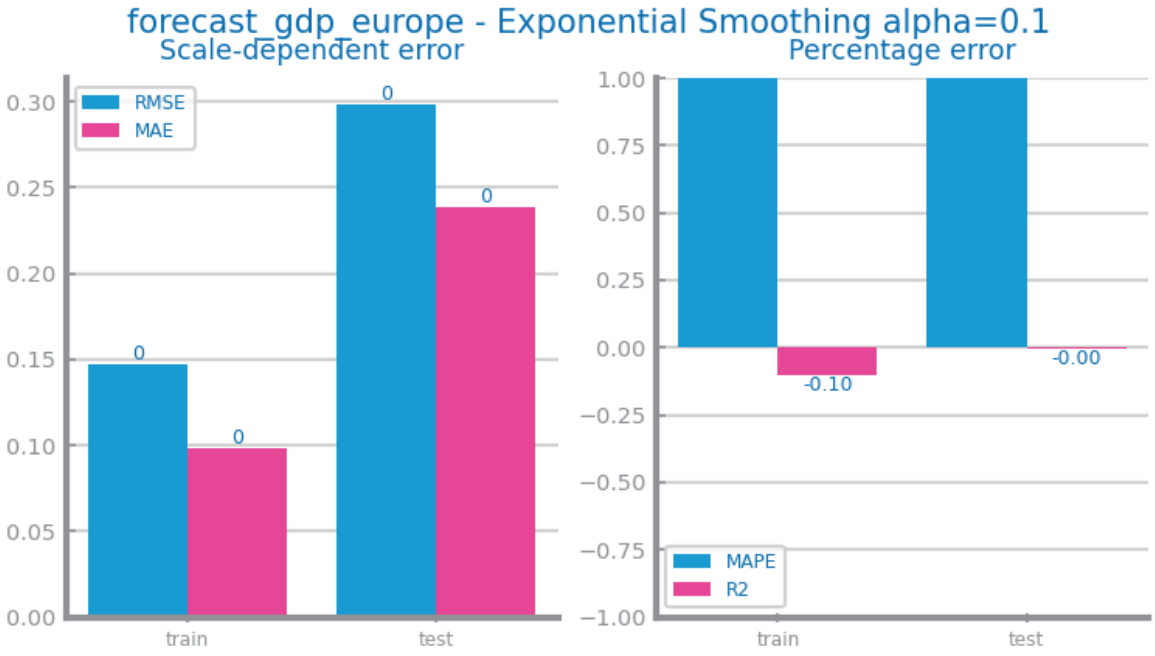



*Figure 119 Forecasting results obtained with the best parameterisation of Exponential Smoothing algorithm, over time series 2*

## *Linear Regression*

Dataset 1: Linear regression performed very well, achieving high R2 values for both train and test sets. It is the first model to provide good results, indicating that the dataset's patterns are well-suited to a linear approach.

Dataset 2: The results were mediocre, with test R2 values still below expectations, linear regression remains one of the best-performing models so far, showing some capacity to handle the dataset's trends.

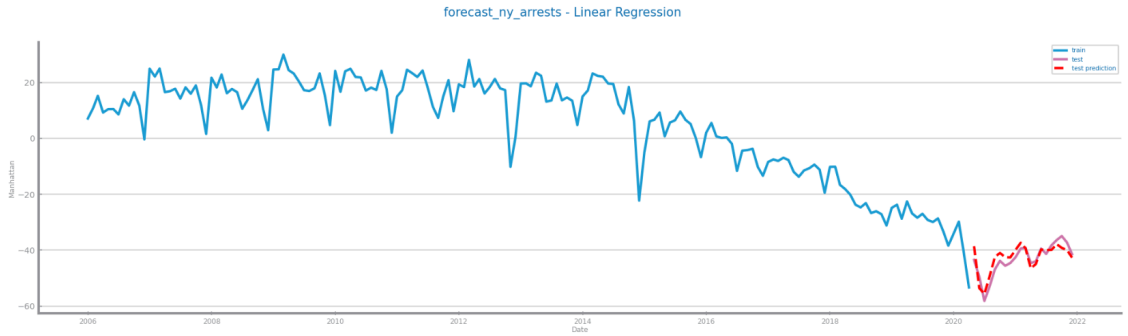



*Figure 120 Forecasting plots obtained with Linear Regression model over time series 1*

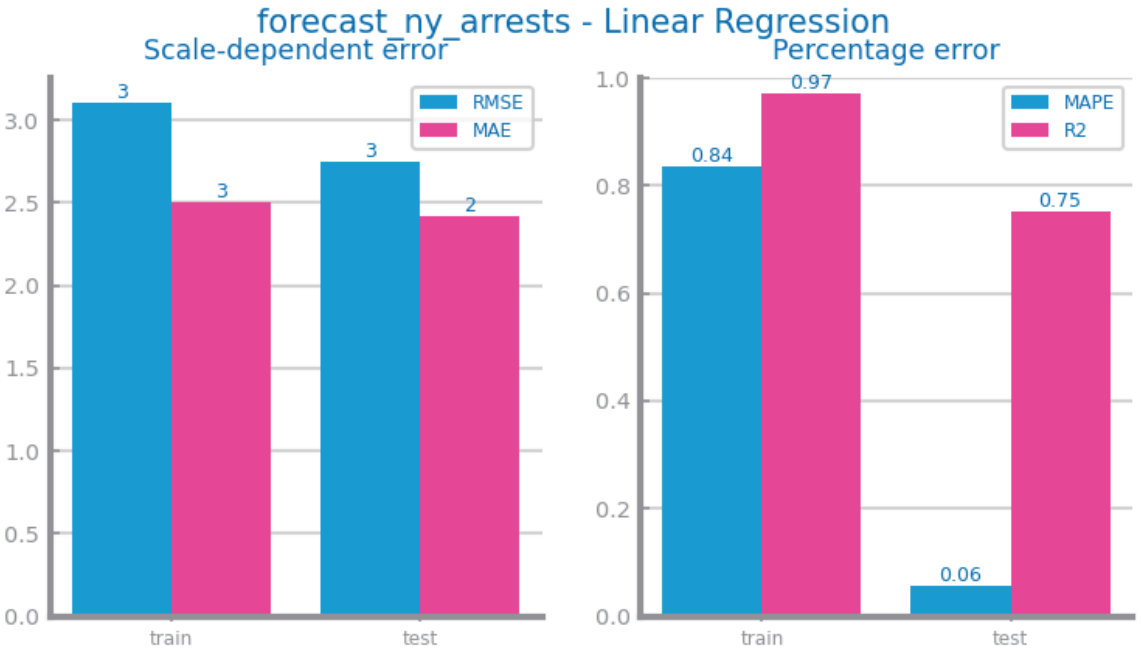



*Figure 121 Forecasting results obtained with Linear Regression model over time series 1*

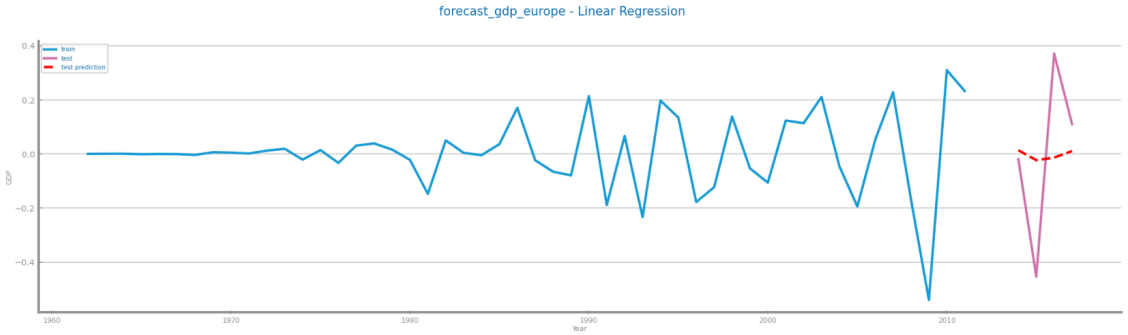



*Figure 122 Forecasting plots obtained with Linear Regression model over time series 2*

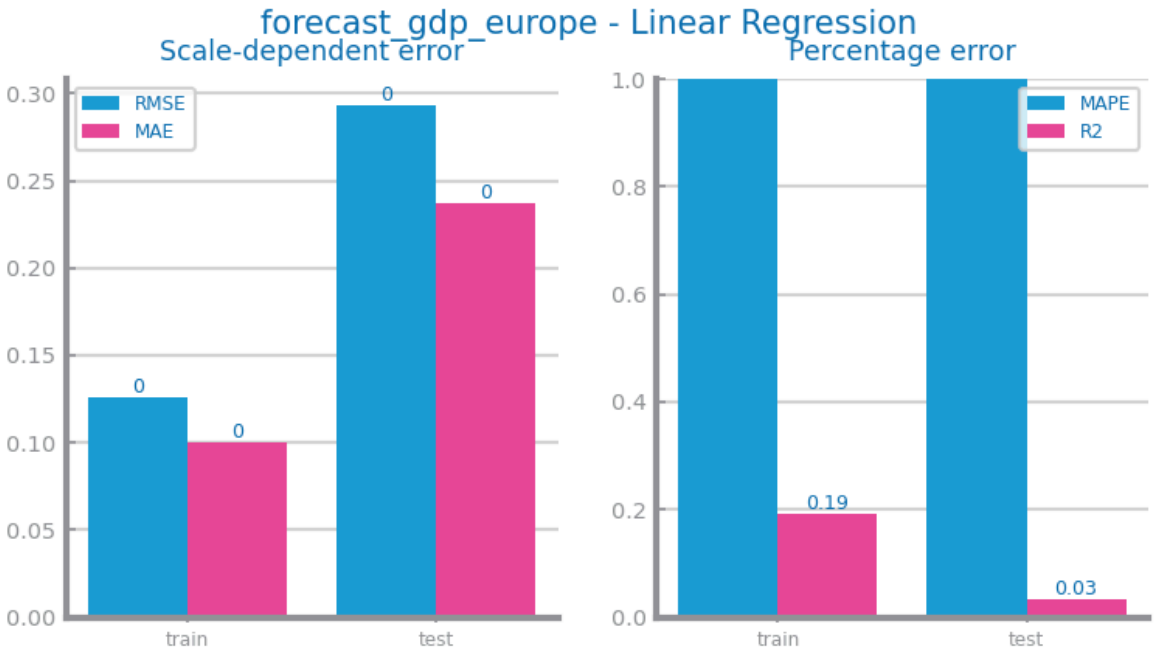


*Figure 123 Forecasting results obtained with Linear Regression model over time series 2*

### *ARIMA Model*

Dataset 1: ARIMA performed best with d=0, highlighting challenges in modeling non-stationary data without differentiation. The model overfitted, showing strong training performance but poor generalization on the test set.

Dataset 2: Univariate differentiation (with second derivative, d=2) improved generalization, achieving strong test performance. Multivariate differentiation (with first derivative, d=1) struggled to match this accuracy, suggesting univariate approaches were more effective for this dataset.

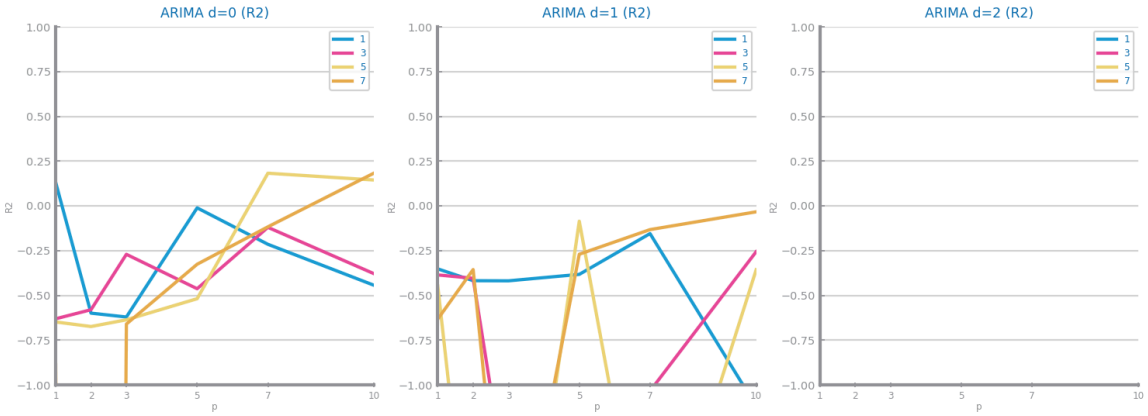



*Figure 124 Forecasting study over different parameterisations of the ARIMA algorithm over time series 1, only with the target variable*

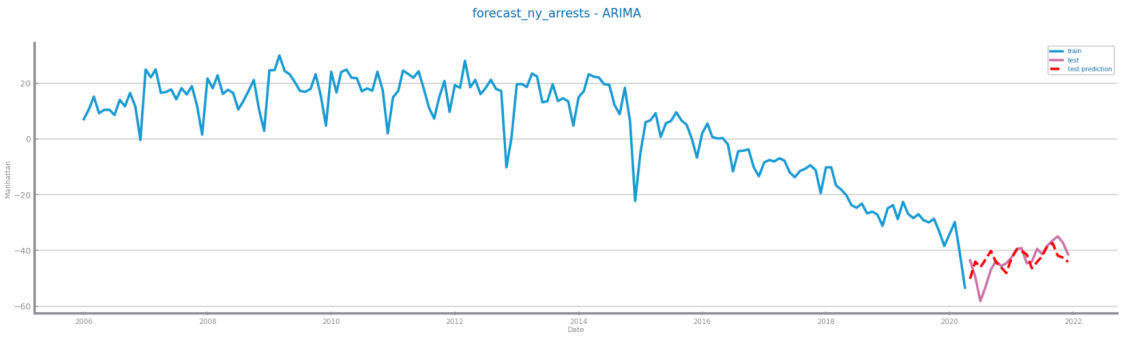



*Figure 125 Forecasting plots obtained with the best parameterisation of ARIMA algorithm, over time series 1, only with the target variable*

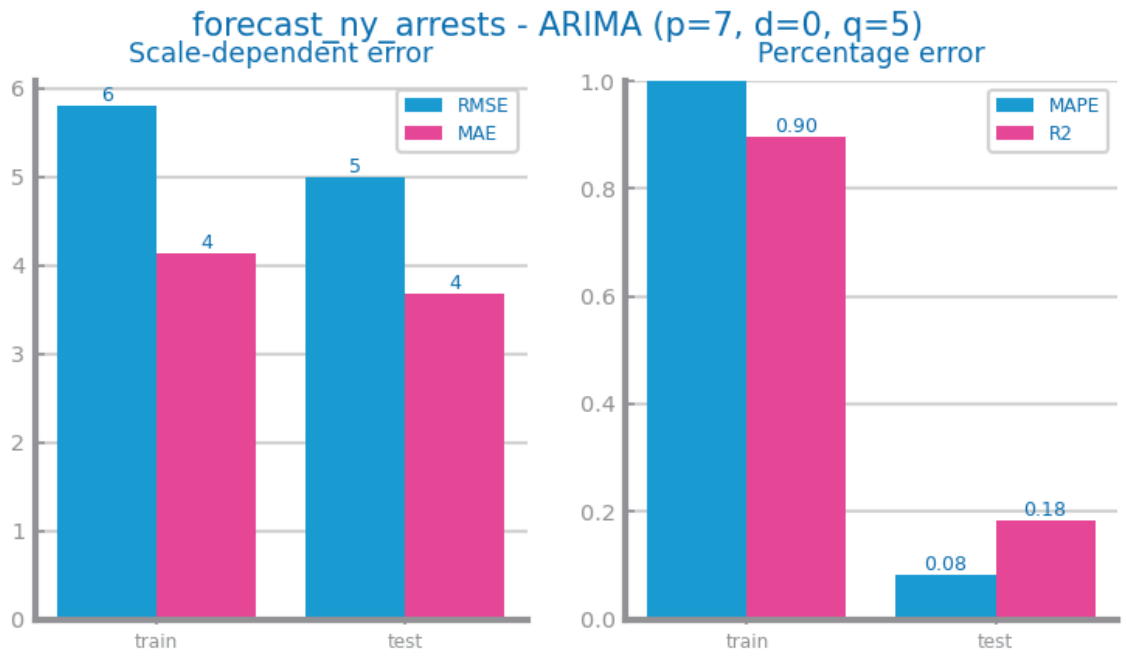



*Figure 126 Forecasting results obtained with the best parameterisation of ARIMA algorithm, over time series 1, only with the target variable*

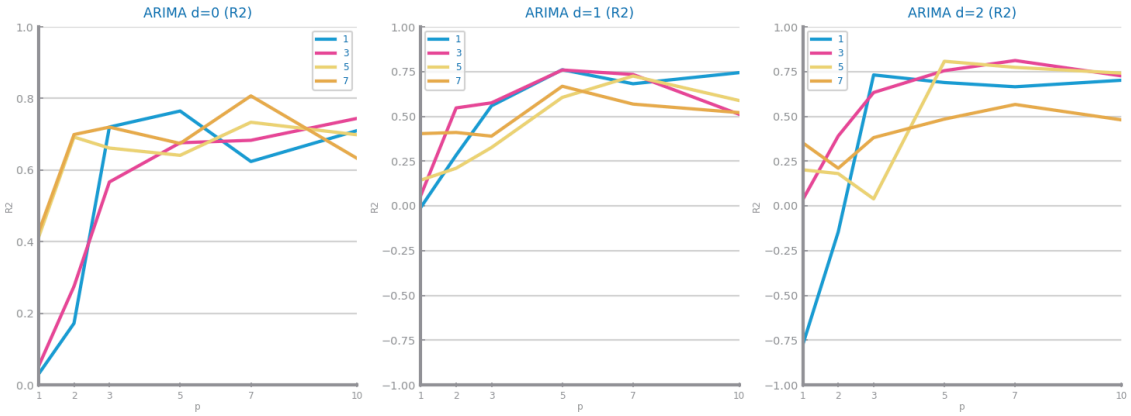



*Figure 127 Forecasting study over different parameterisations of the ARIMA algorithm over time series 2, only with the target variable*

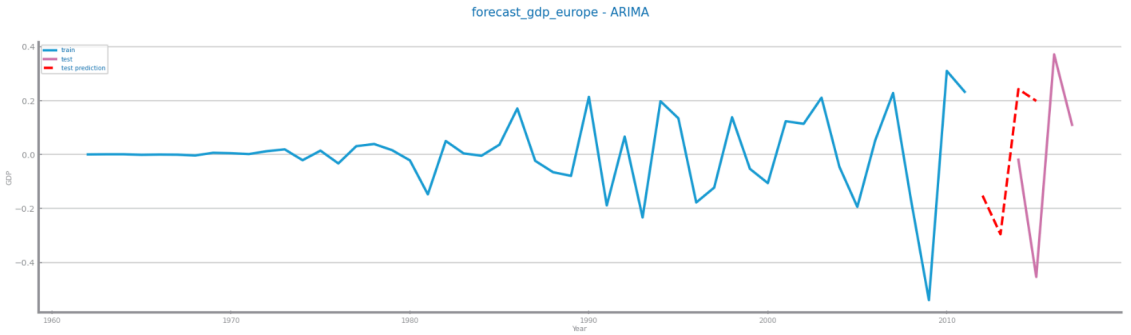



*Figure 128 Forecasting plots obtained with the best parameterisation of ARIMA algorithm, over time series 2, only with the target variable*

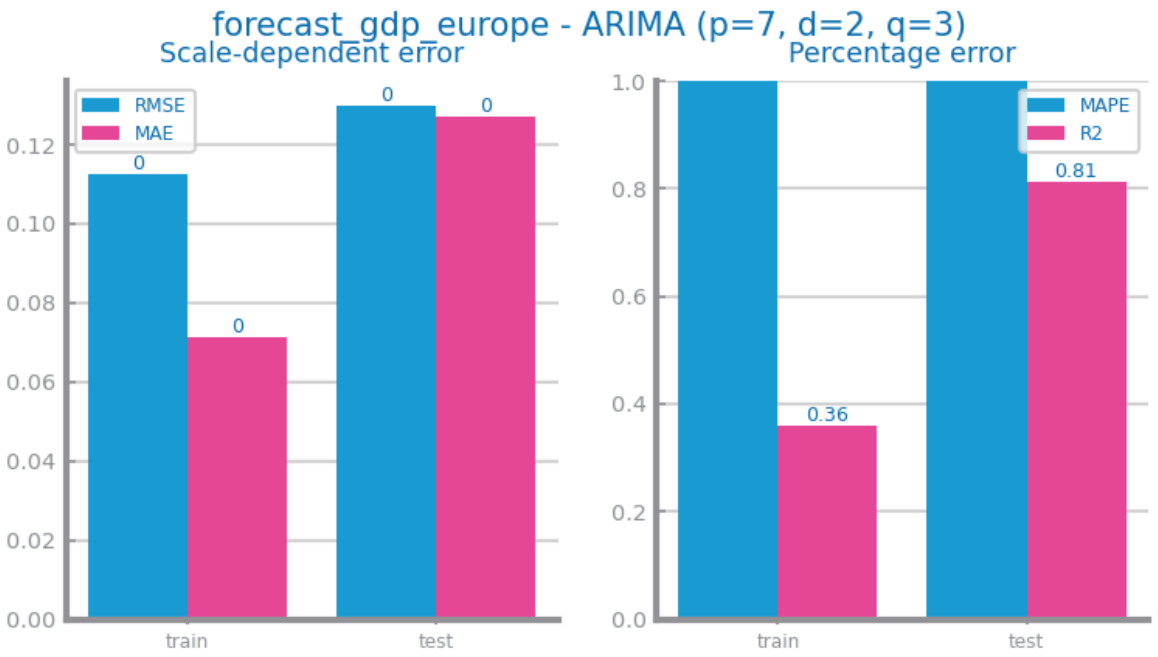



*Figure 129 Forecasting results obtained with the best parameterisation of ARIMA algorithm, over time series 2, only with the target variable*

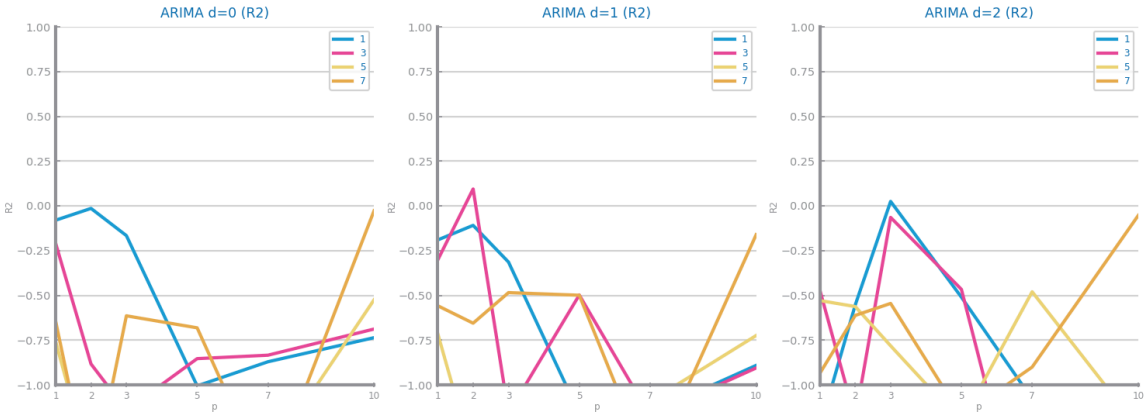



*Figure 130 Forecasting study over different parameterisations of the ARIMA algorithm over time series 2, with multiple variables*

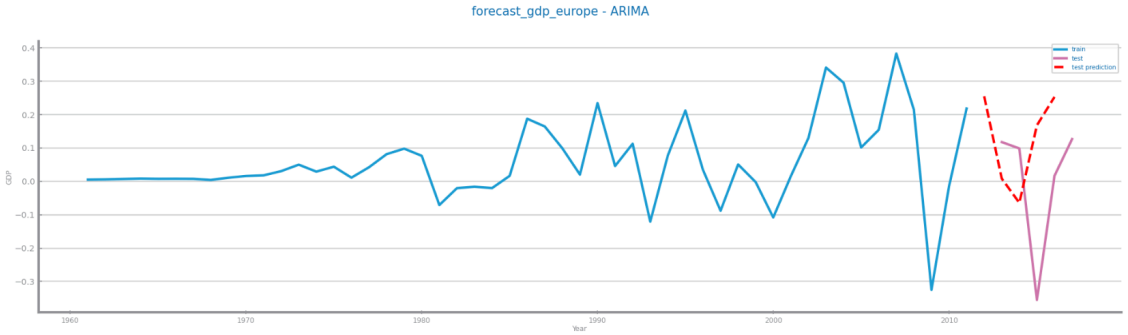



*Figure 131 Forecasting plots obtained with the best parameterisation of ARIMA algorithm, over time series 2, with multiple variables*

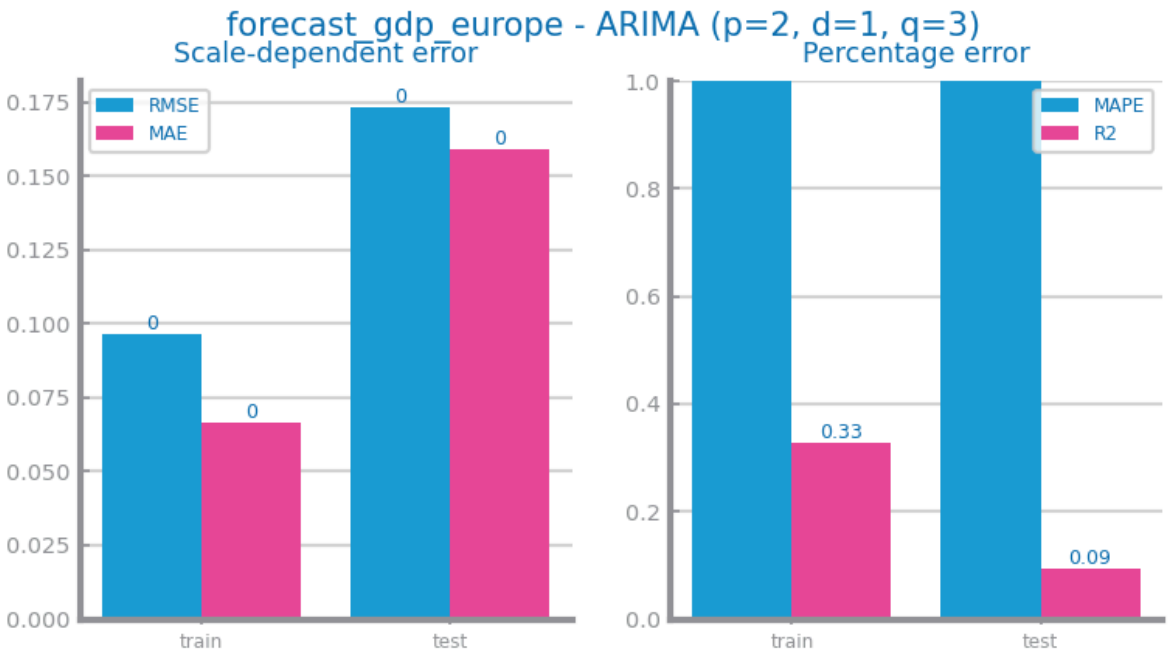



*Figure 132 Forecasting results obtained with the best parameterisation of ARIMA algorithm, over time series 2, with multiple variables*

### *LSTMs Model*

Dataset 1: The parameter study shows that LSTM performed best with length=2, hidden=100, and epochs=900. The model demonstrated strong training performance but struggled with generalization, highlighting overfitting issues when no differentiation was applied to univariate data.

Dataset 2: Univariate differentiation (with second derivative, length=2, hidden=100, epochs=3000) improved generalization with balanced results. Multivariate differentiation (with first derivative, length=2, hidden=25, epochs=0) struggled, univariate approaches were more effective.

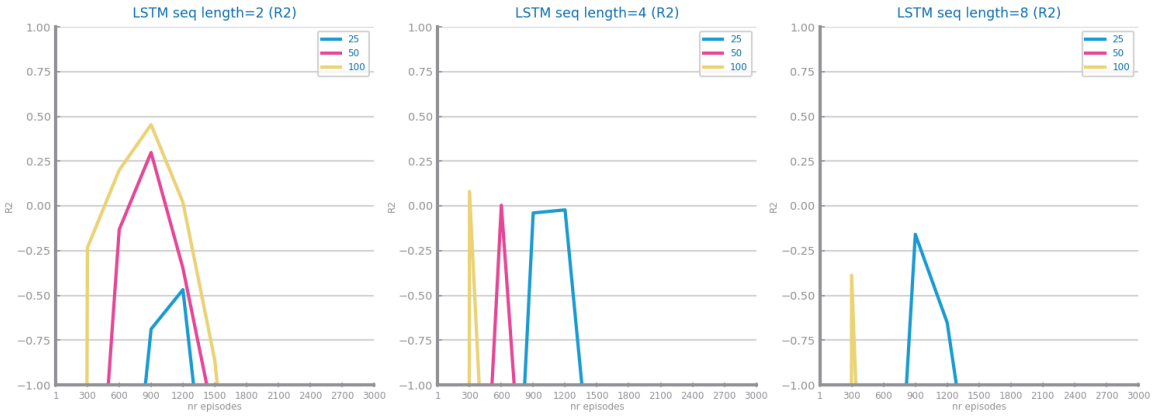



*Figure 133 Forecasting study over different parameterisations of LSTMs over time series 1, only with the target variable*


*Figure 134 Forecasting plots obtained with the best parameterisation of LSTMs over time series 1, only with the target variable*

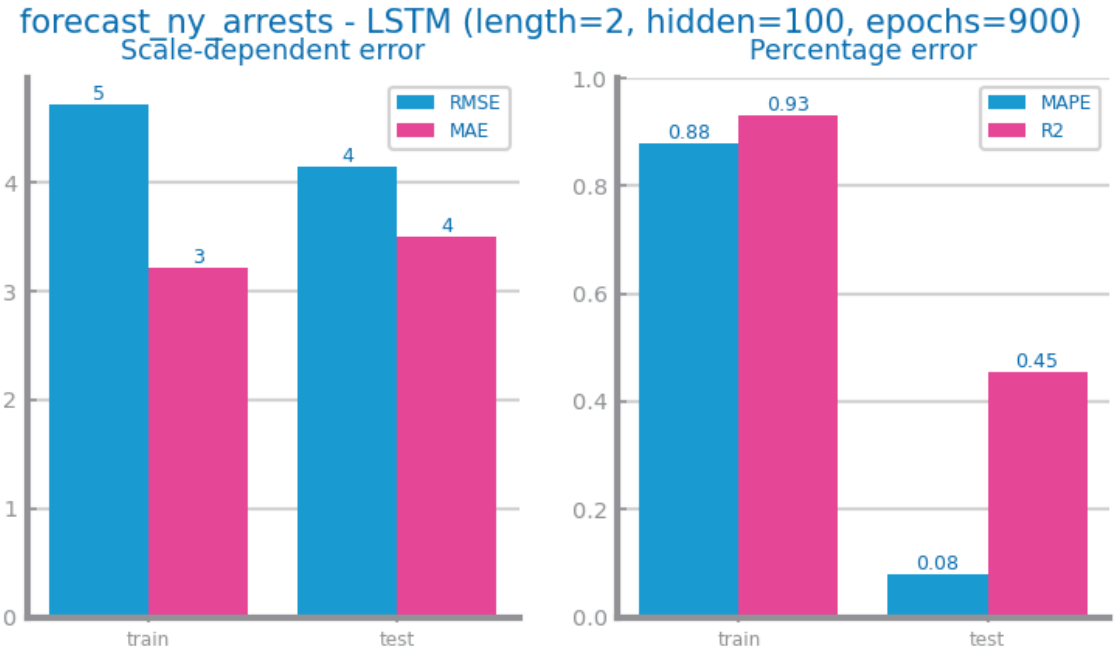



*Figure 135 Forecasting results obtained with the best parameterisation of LSTMs over time series 1, only with the target variable*

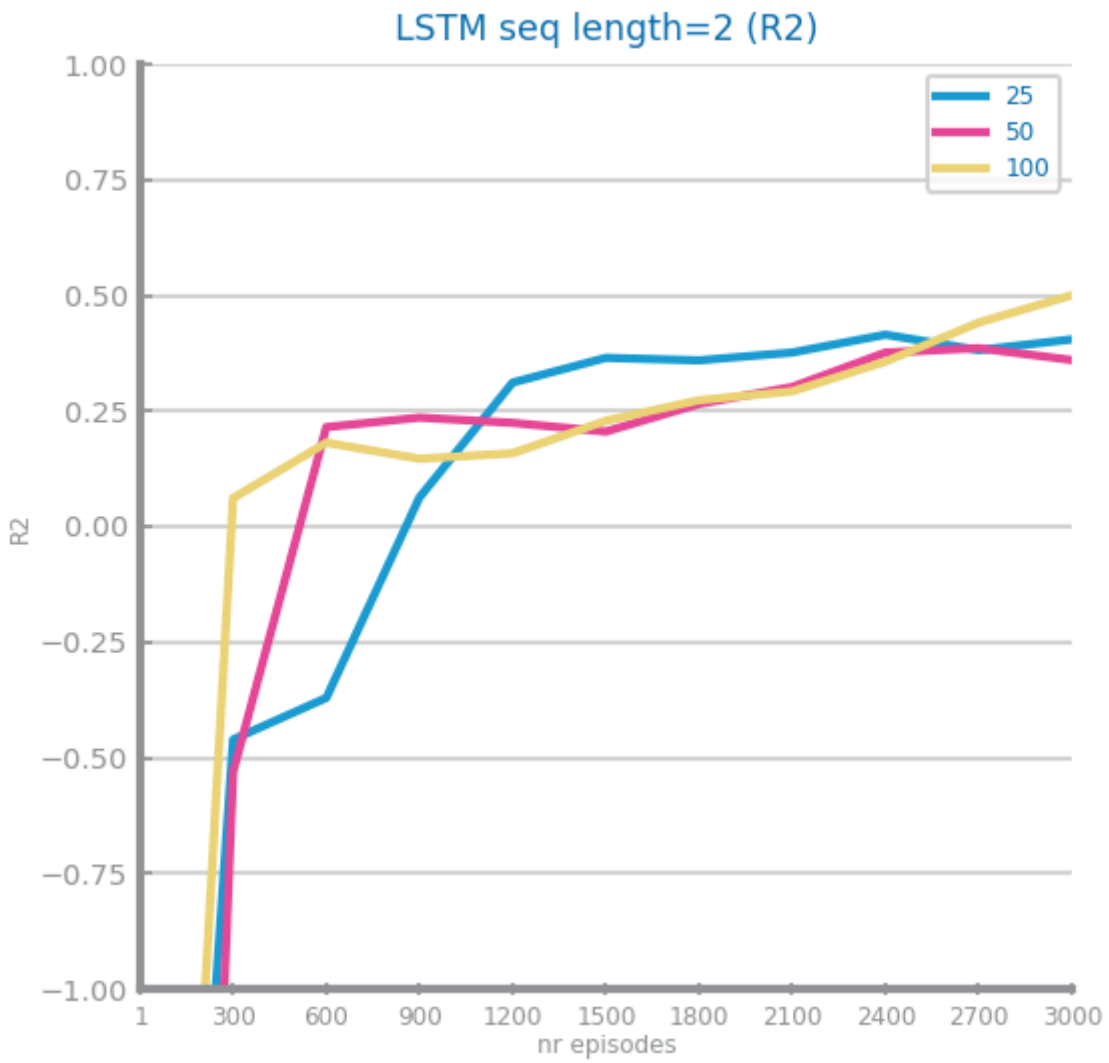



*Figure 136 Forecasting study over different parameterisations of the LSTMs over time series 2, only with the target variable*

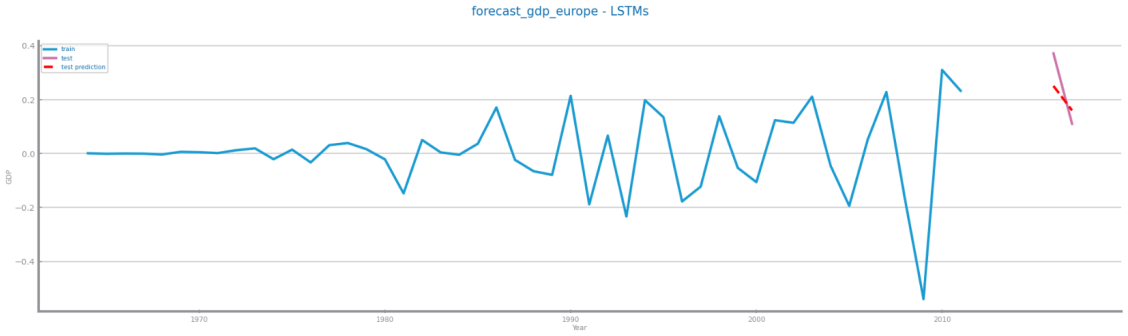



*Figure 137 Forecasting plots obtained with the best parameterisation of LSTMs over time series 2, only with the target variable*

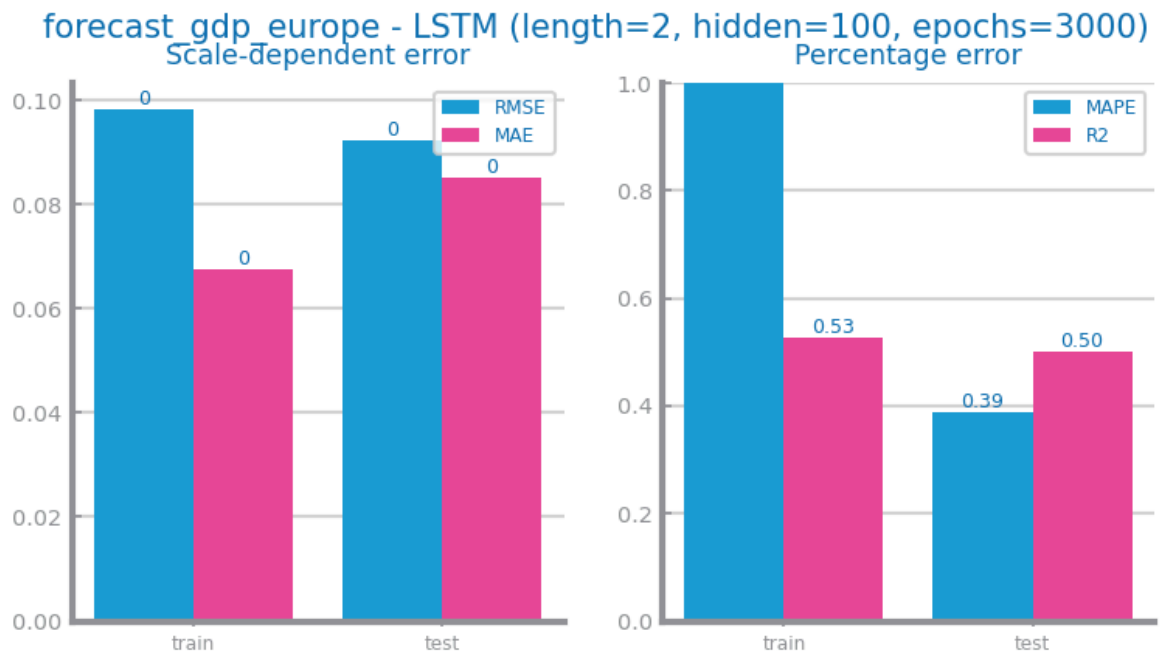



*Figure 138 Forecasting results obtained with the best parameterisation of LSTMs over time series 2, only with the target variable*

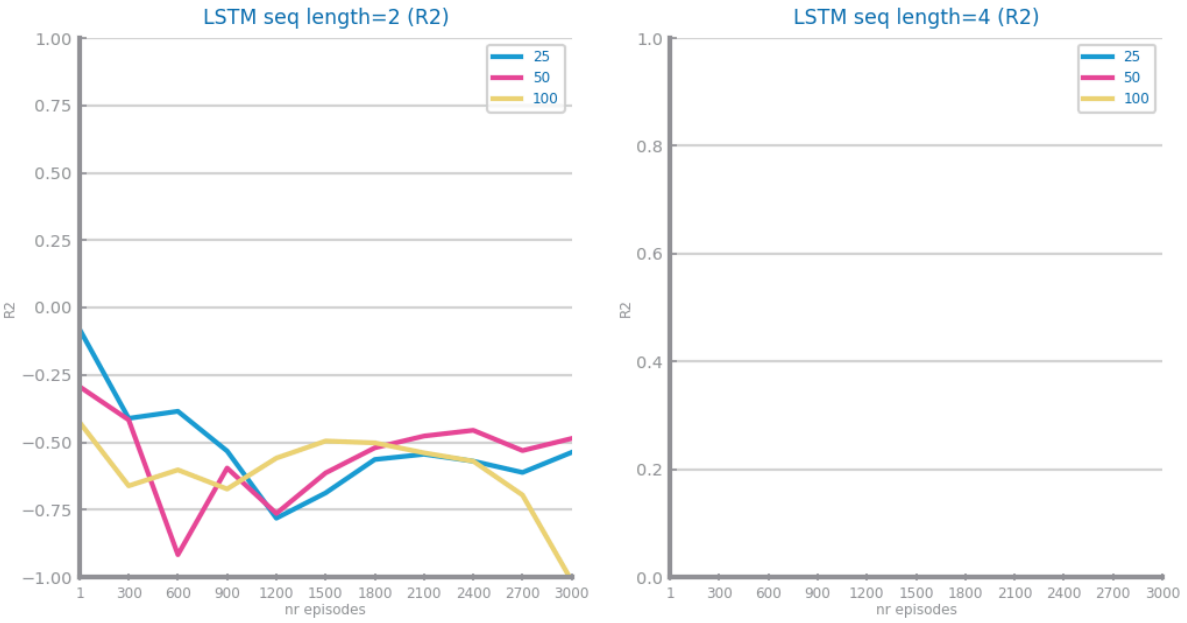



*Figure 139 Forecasting study over different parameterisations of the LSTMs over time series 2, with multiple variables*

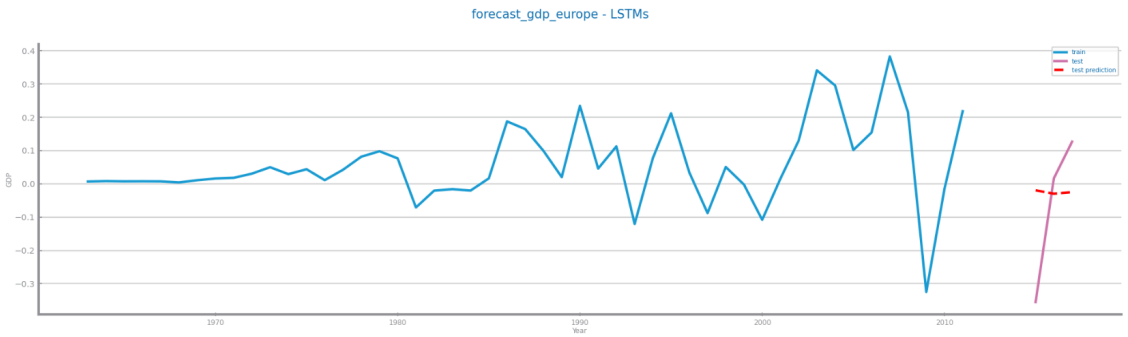


*Figure 140 Forecasting plots obtained with the best parameterisation of LSTMs over time series 2, with multiple variables*

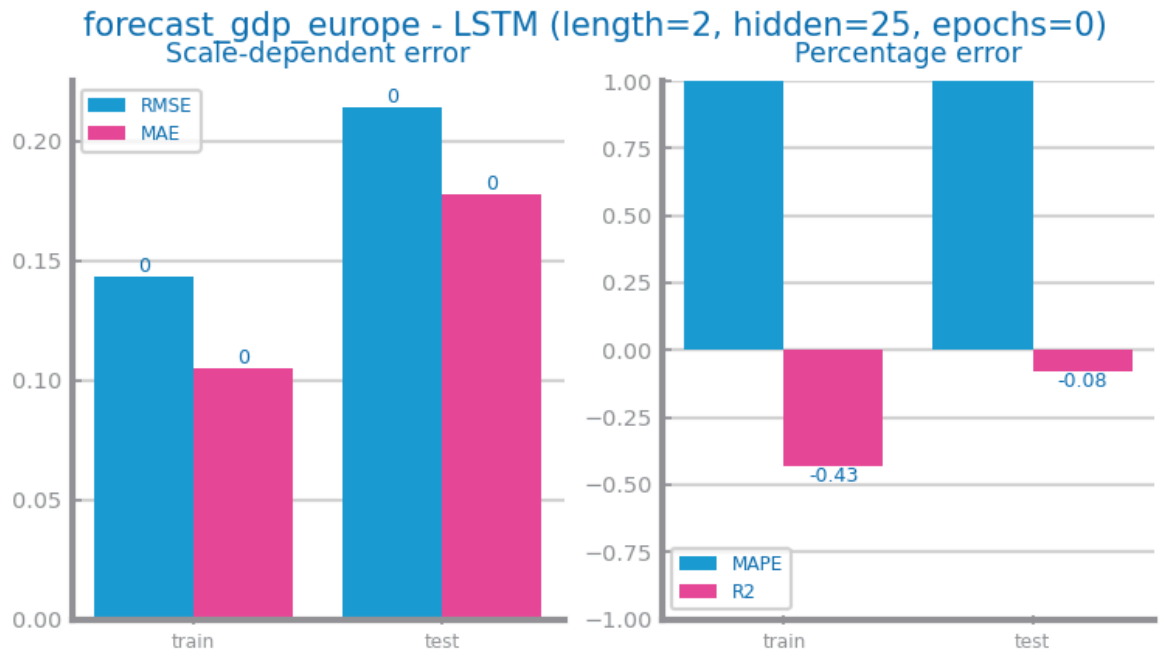



*Figure 141 Forecasting results obtained with the best parameterisation of LSTMs over time series 2, with multiple variables*

# 8 Critical Analysis

Dataset 1: The applied transformations, particularly the aggregation to monthly data, reduced noise and improved model performance. However, most models performed poorly on average, with the exception of Linear Regression, Persistence Optimistic, ARIMA, and LSTM, which performed above the rest.

Among these, Linear Regression emerged as the most reliable, as the other models exhibited clear overfitting and insufficient R2 values on the test set.

Differentiation studies with LSTM and ARIMA (univariate and multivariate) failed to yield better results, so the original data was retained in both.

Dataset 2: Transformations, particularly univariate differentiation, significantly improved ARIMA and LSTM performance. Most models performed poorly possibly due to the datasets size (the dataset is really small and with the split we get small training and especially small test sets). ARIMA was the only one that performed well and LSTM was another who outperformed others on univariate differentiated data.

ARIMA demonstrated strong test performance, though its training performance was much lower, possibly due to the small dataset size. LSTM, with comparable train and test performance, showed limited utility as R2 values hovered around 0.50, suggesting it may be guessing.

For LSTM and ARIMA, univariate differentiation (second derivative) yielded the best performance boost, while multivariate differentiation (first derivative) was slightly better for evaluation but still underperformed.