TÉCNICO LISBOA

DS 2022

# Data Science Project

| | | |
|---|---|---|
| **Team nr:** 21 | **Student1 :** Basanta Poudel | **IST nr:** 80894 |
| | **Student 2 :** Maribel Jaramillo | **IST nr:** 105560 |
| | **Student 3 :** Niels Vullings | **IST nr:** 104778 |
| | **Student 4:** Miguel Cruz | **IST nr**: 92527 |

## CLASSIFICATION

## 1 DATA PROFILING

Dataset1 must analyze if a patient was readmitted (NO, >30, <30) in some hospital.

Dataset2 must analyze drought (1 and 0 values) using weather & soil data.

### *Data Dimensionality*

For both datasets, there are plenty records for analysis. There are many missing values for three variables in dataset 1 and, there are no missing values in dataset 2

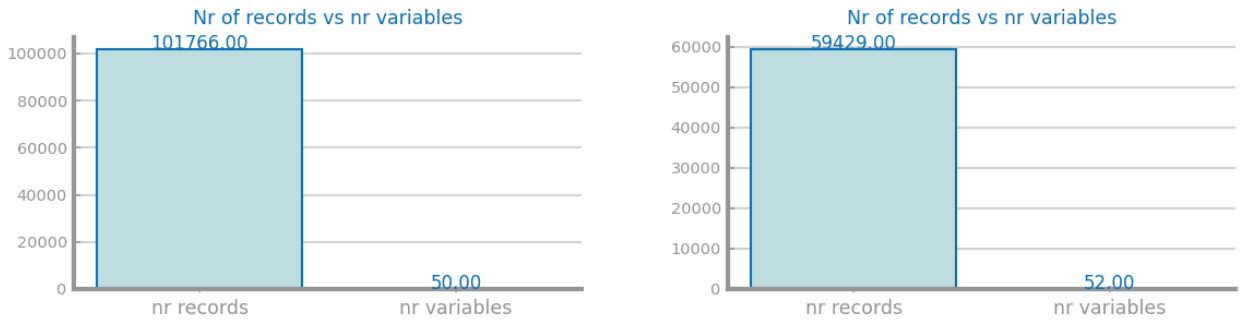



*Figure 1 Nr Records x Nr variables for dataset 1 (left) and dataset 2 (right)*

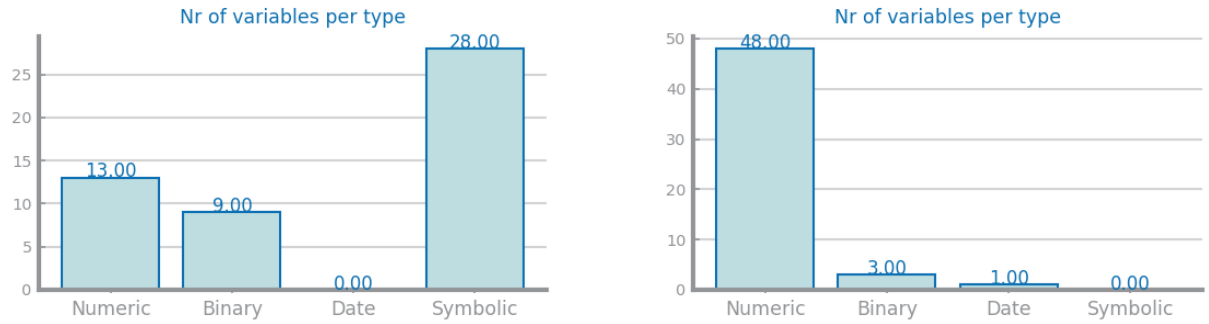



*Figure 2 Nr variables per type for dataset 1 (left) and dataset 2 (right)*

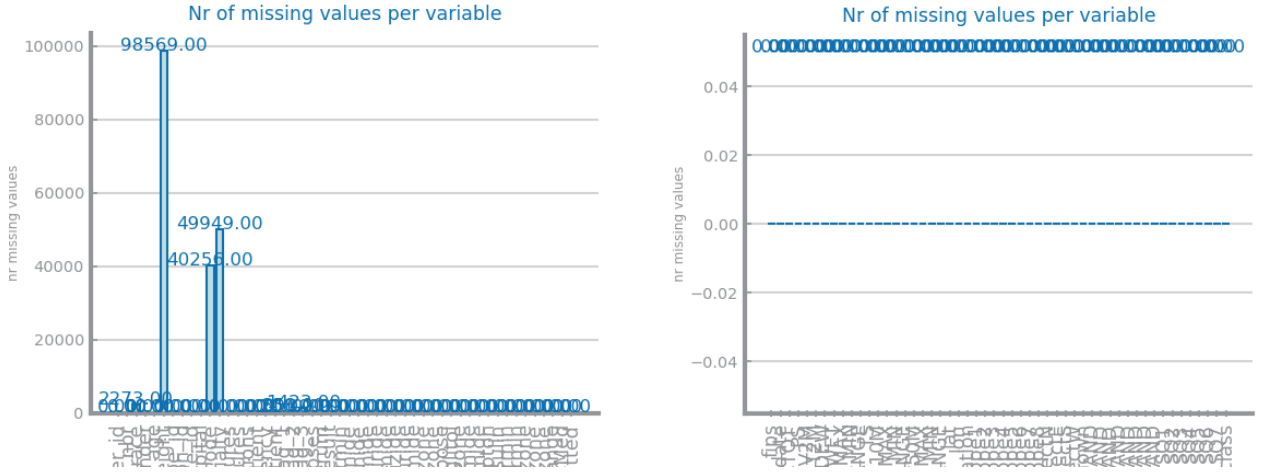



*Figure 3 Nr missing values for dataset 1 (left) and dataset 2 (right)*

### *Data Distribution*

For dataset 1, most of our variables are symbolic and numerical and values have exponential distributions very concentrated in low values, some variables have Normal distribution such as number_patient. The dataset seems to have few outliers. Dataset 2, most of our variables are numerical with a normal distribution, and the dataset has outliers.

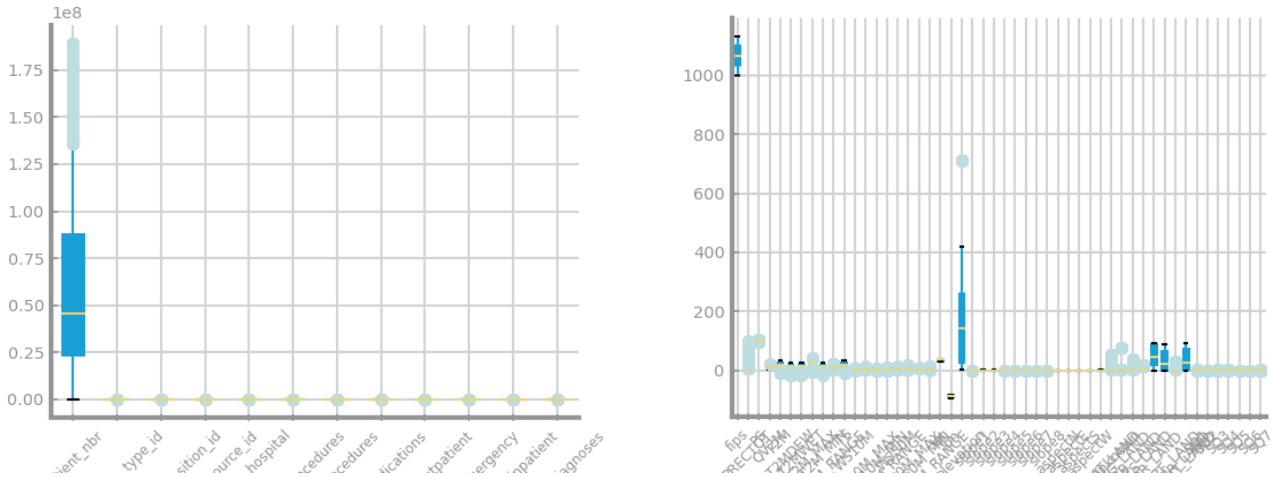



*Figure 4 Global boxplots dataset 1 (left) and dataset 2 (right)*

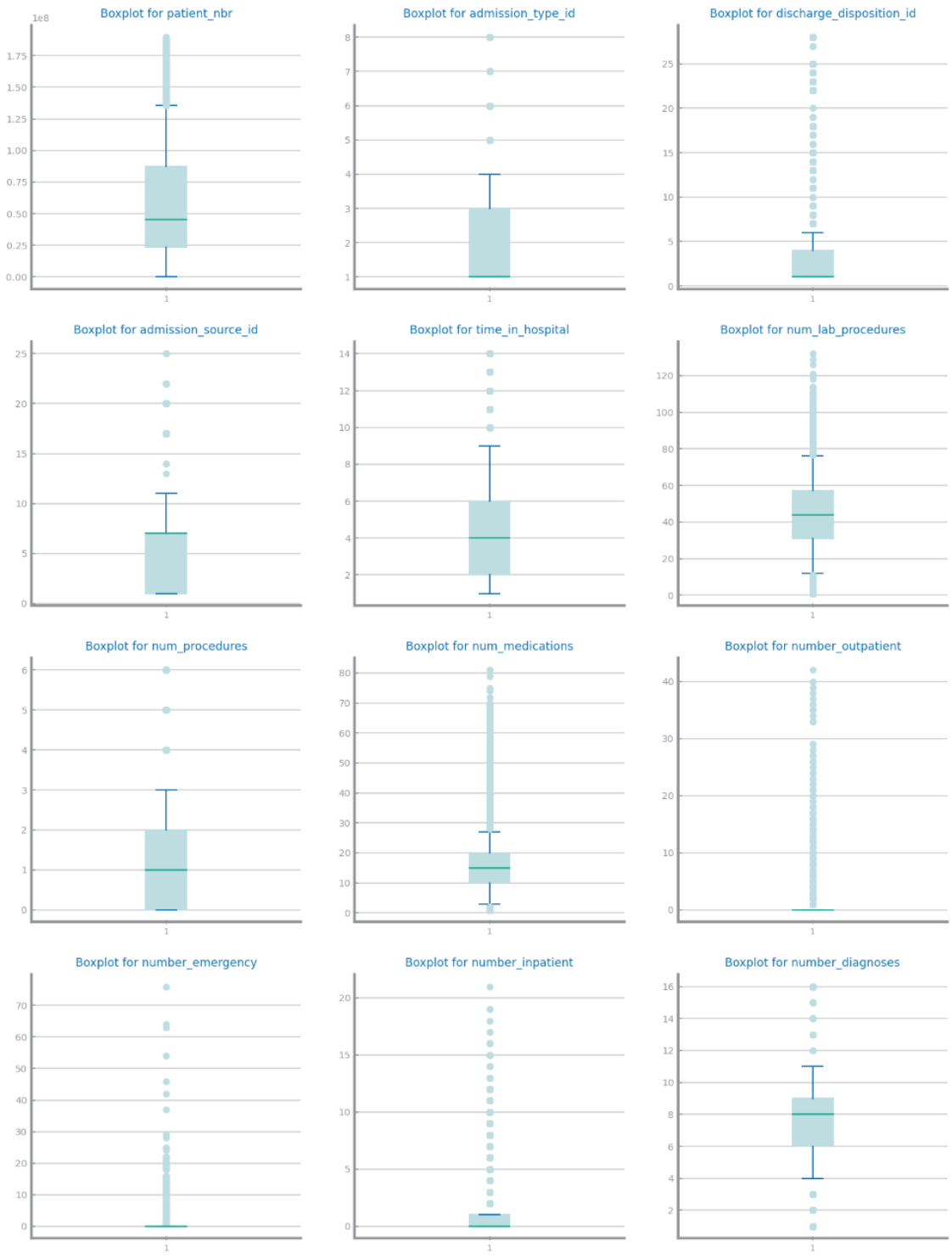



*Figure 5 Single variable boxplots for dataset 1*

*Figure 6 Single variable boxplots for dataset 2*

*Figure 7 Symbolic Histograms for dataset 1*


*Figure 8 Numeric Histograms for dataset 1*

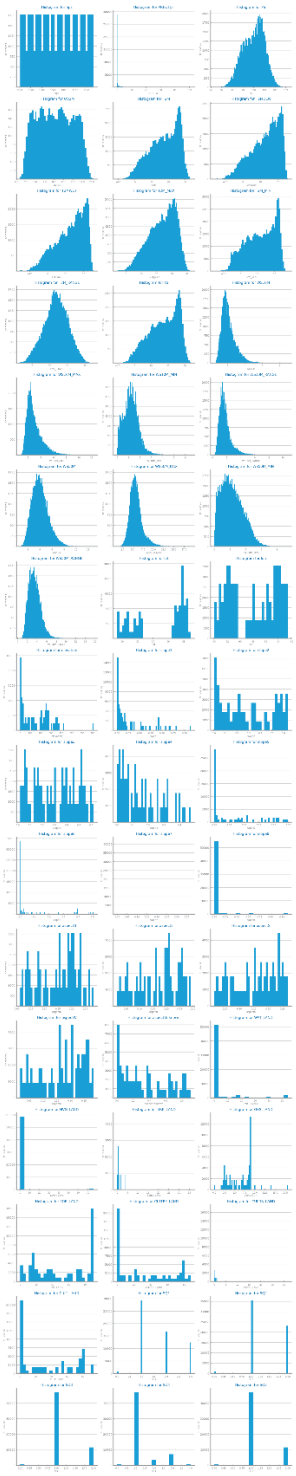


*Figure 9 Histograms for dataset 2*

### *Data Granularity*

**Dataset1**: Symbolic variables are best represented with 10 bins. This is not true for the ''diag'' variables, due to their many number of categories.

**Dataset2**: Variables 'QV2M', 'T2-', 'TS' and 'WS' show high levels of granularity.

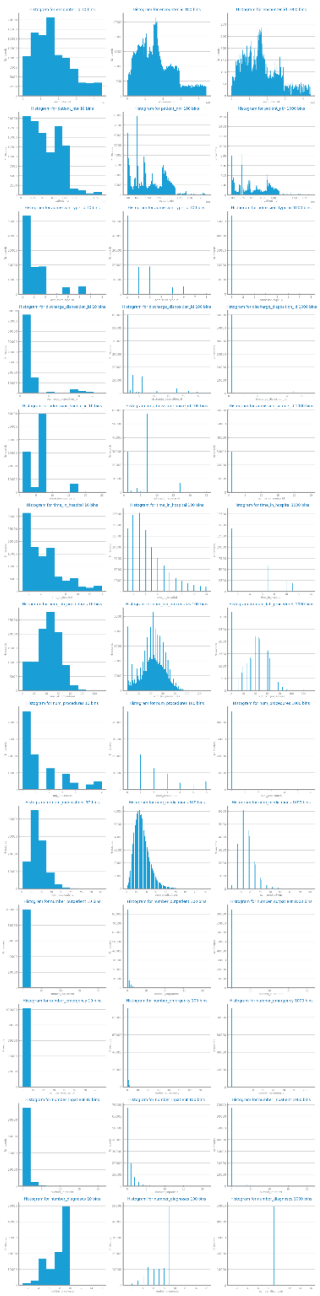


*Figure 10 Numeric Variables - Granularity analysis for dataset 1*

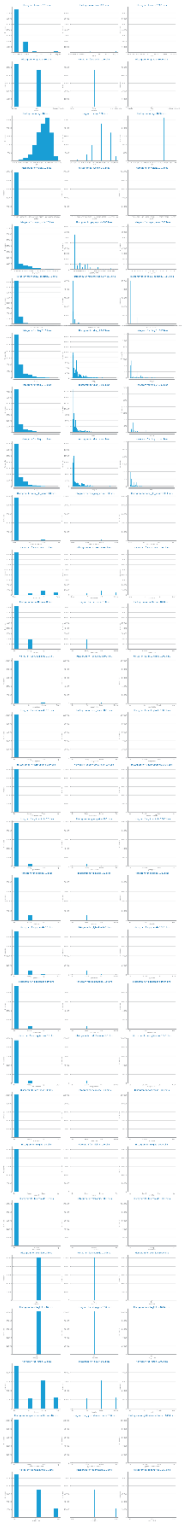


*Figure 11 Symbolic Variables - Granularity for dataset 1*

*Figure 12 Numeric Variables - Granularity analysis for dataset 2*

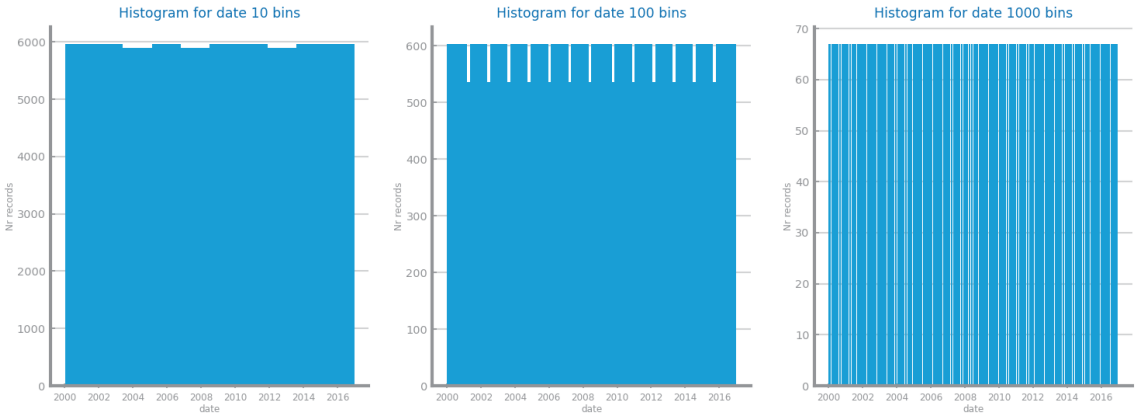



*Figure 13 Date - Granularity analysis for dataset 2*

### *Data Sparsity*

**Dataset1-** The correlation between variables are not high specially with the classification variable.

**Dataset2-** There are some correlated variables but the classification variable is not highly correlated with any other variable.

*Figure 14 Numeric Variables: Sparsity analysis for dataset 1*

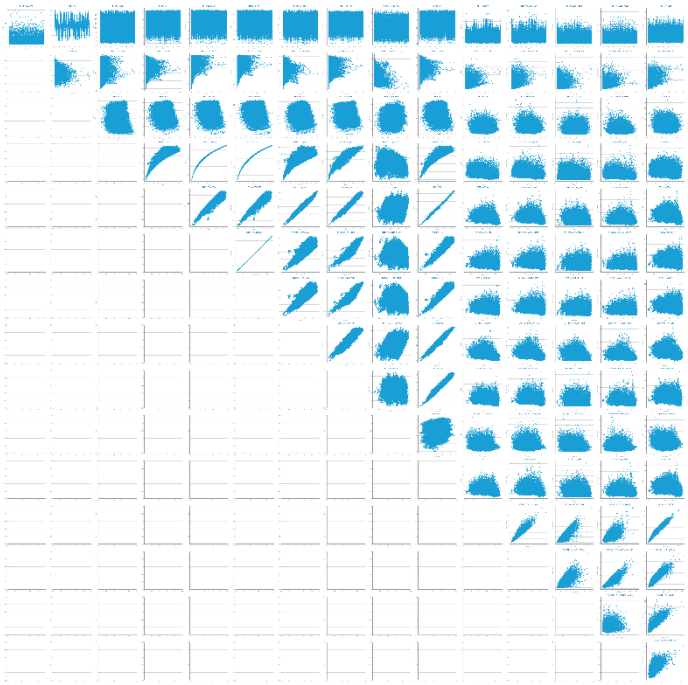


*Figure 15 Numeric Variable: Sparsity analysis for dataset 2*


*Figure 16 Correlation analysis for dataset 1*

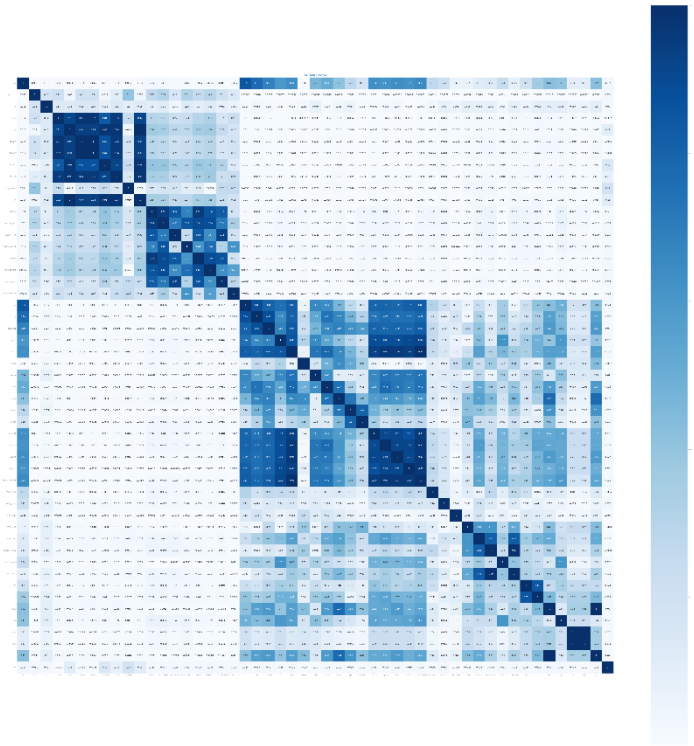


*Figure 17 Correlation analysis for dataset 2*

# 2 DATA PREPARATION

## *Variables Encoding*

**Dataset1:** We convert categorical variables to numbers that the model can understand and extract valuable information.

**Dataset2-** Date Variable is encoded to numeric by applying the formula which generates unique number for each date - 10000*data['date'][n].year + 100*data['date'][n].month + data['date'][n].day

| Transformation method | Variables |
|---|---|
| Random code assignment | ['race', 'payer_code', 'medical_specialty'] |
| Binary code [0,1] | ['gender',] [diabetes_med and change] |
| Encoding according to ordinal value | ['age', 'max_glu_serum', 'A1Cresult', 'metformin', 'repaglinide', 'nateglinide', 'chlorpropamide', 'glimepiride', 'glipizide', 'glyburide', 'pioglitazone', 'rosiglitazone', 'acarbose', 'miglitol', 'tolazamide', 'insulin', 'readmitted'] |
| Encoding according to dataset description ICD9 | ['diag_1', 'diag_2', 'diag_3'] |

*Figure 18 Variable Encoding - dataset 1*

### *Missing Value Imputation*

**Dataset1-** Variable weight contains approximately 98% of the missing values so we dropped, and we apply **dropna** to the dataset to clean up.

**Dataset2-** Not applicable - No missing values.


*Figure 19 Missing Value Imputation results - Mean approach dataset 1 - knn (left) and nb (right)*

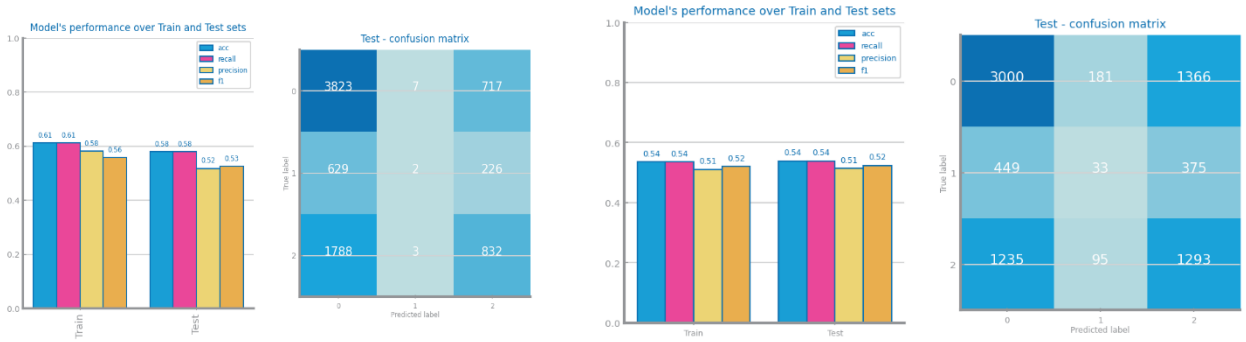



*Figure 20 Missing Value Imputation results - DropNa approach dataset 1 - knn (left) and nb (right)*

### *Outliers Treatment*

**Dataset1-** Outliers columns discovered - [‘num_lab_procedures ']. Outliers Treatment gives better results. **Replace Outliers** has the best results.

**Dataset2-** Outliers columns discovered - ['CULTIR_LAND', 'URB_LAND']. Outliers Treatment gives better results. **Drop Outliers** has the best results.

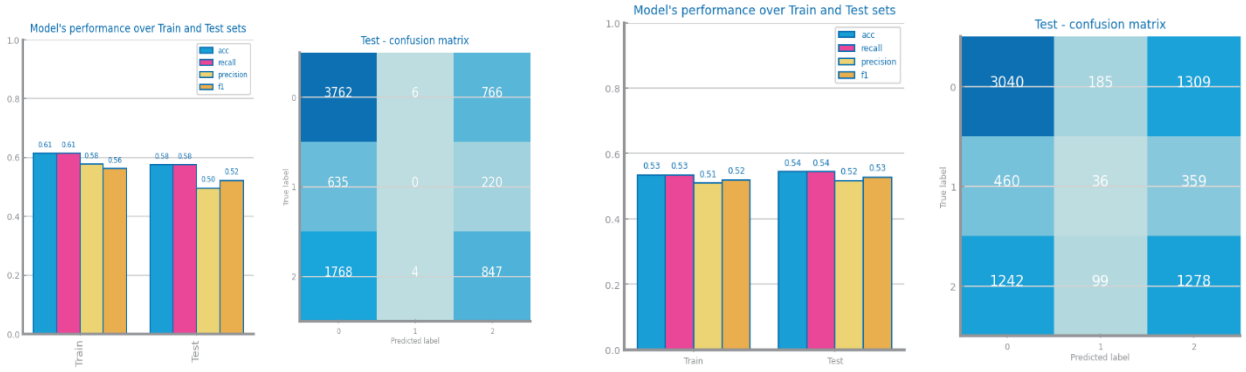



*Figure 21 Outliers imputation results - Drop outliers treatment dataset 1 - knn (left) and nb (right)*

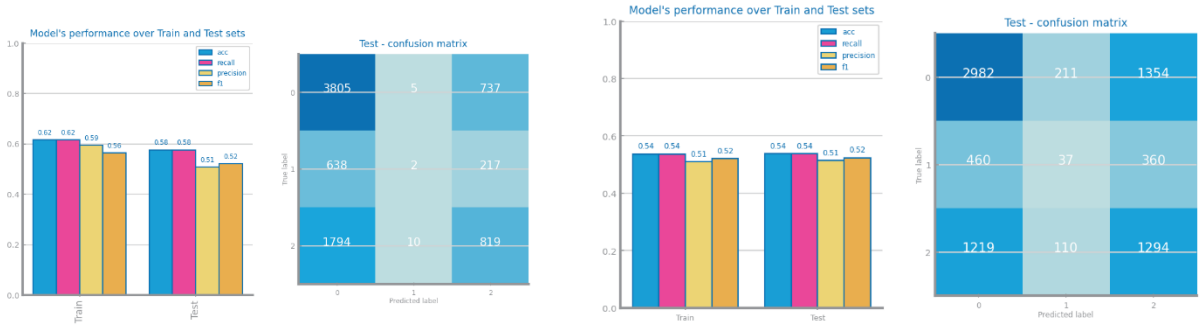



*Figure 22 Outliers imputation results - Replace outliers treatment dataset 1 - knn (left) and nb (right)*

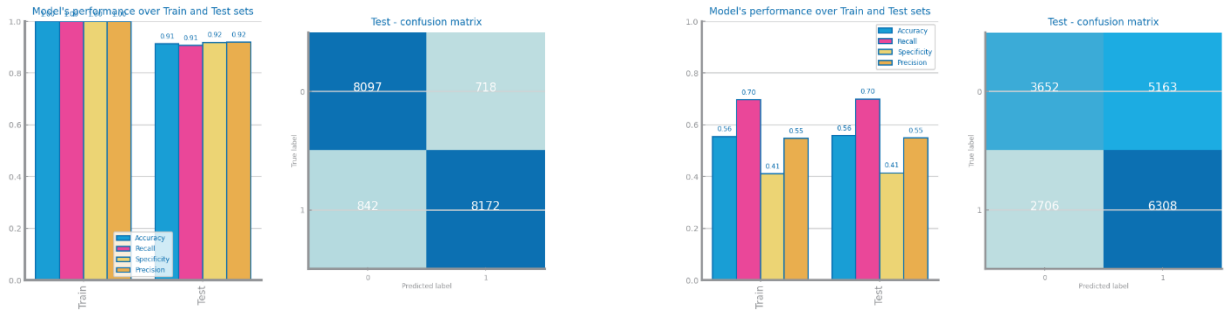



*Figure 23 Outliers imputation results – No outliers treatment dataset 2 - knn (left) and nb (right)*

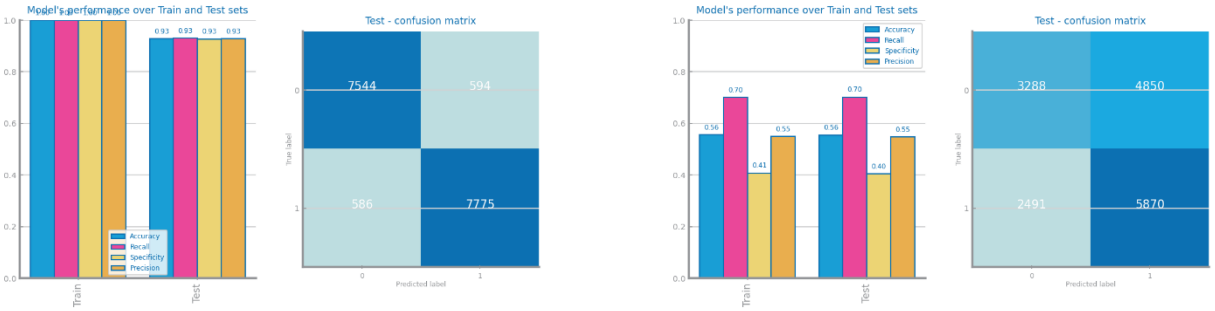



*Figure 24 Outliers imputation results - Drop outliers treatment dataset 2 - knn (left) and nb (right)*

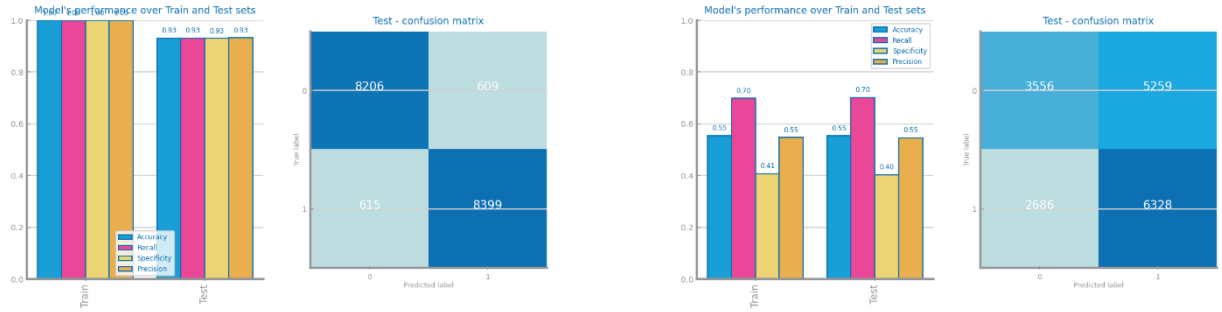



*Figure 25 Outliers imputation results - Truncate outliers treatment dataset 2 - knn (left) and nb (right)*

## *Scaling*

**Dataset1-** Variables have the same ranges due to the coding performed, which does not help with any scaling.
**Dataset2-** None of the strategies improved the evaluation score. (Selected - output of the drop outliers approach)

*Results for minmax scaling dataset 1: KNN (left), NB (right)*

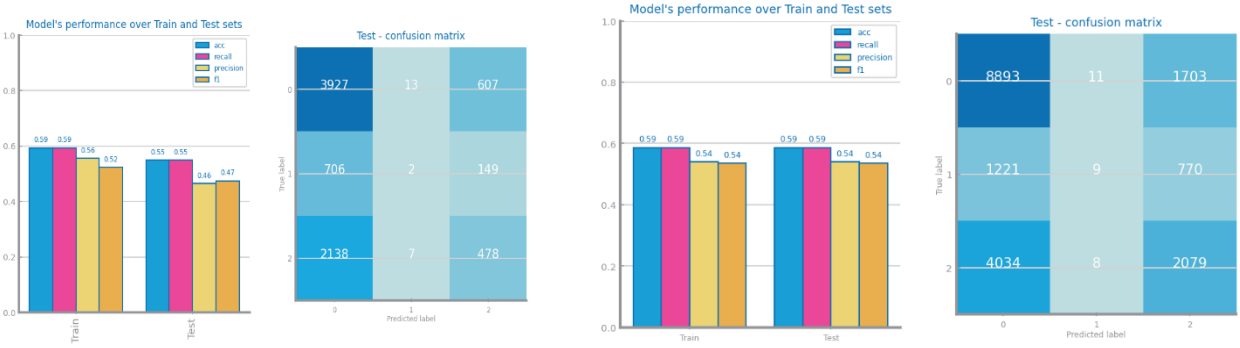



*Figure 26 Scaling results - MinMax approach dataset 1 - knn (left) and nb (right)*

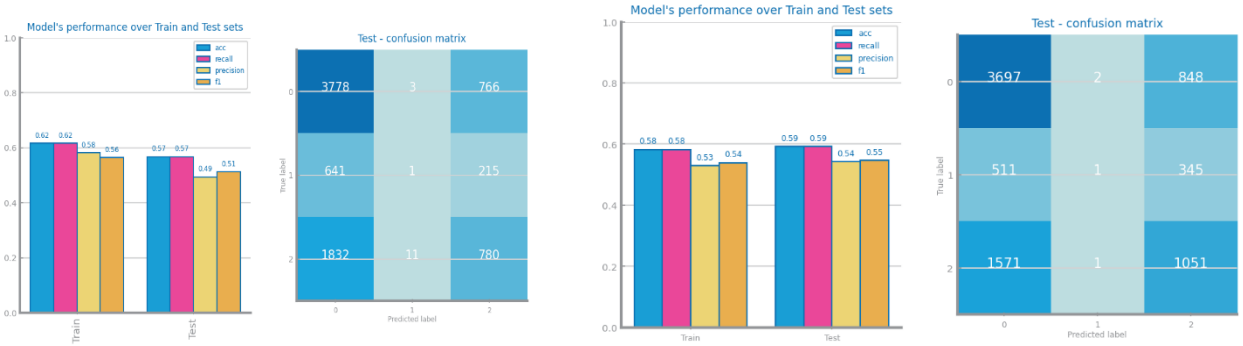



*Figure 27 Scaling results - No Scaling (previous preparation results) dataset 1 - knn (left) and nb (right)*

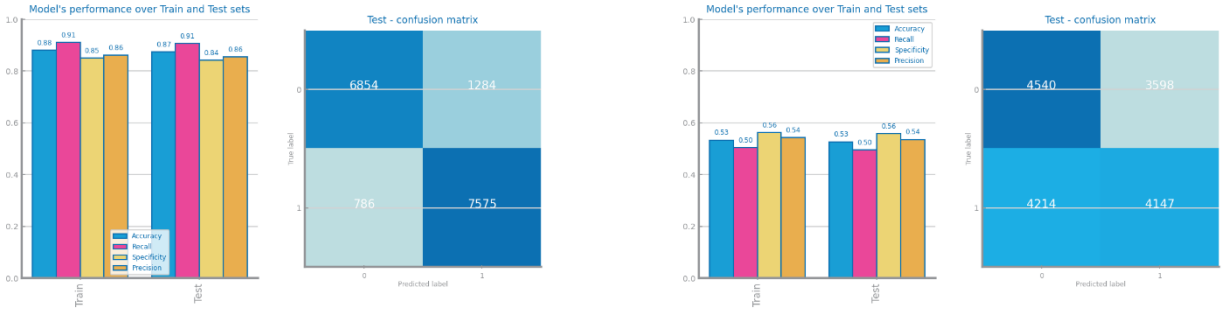



*Figure 28 Scaling results - Minmax approach for dataset 2 - knn (left) and nb (right)*

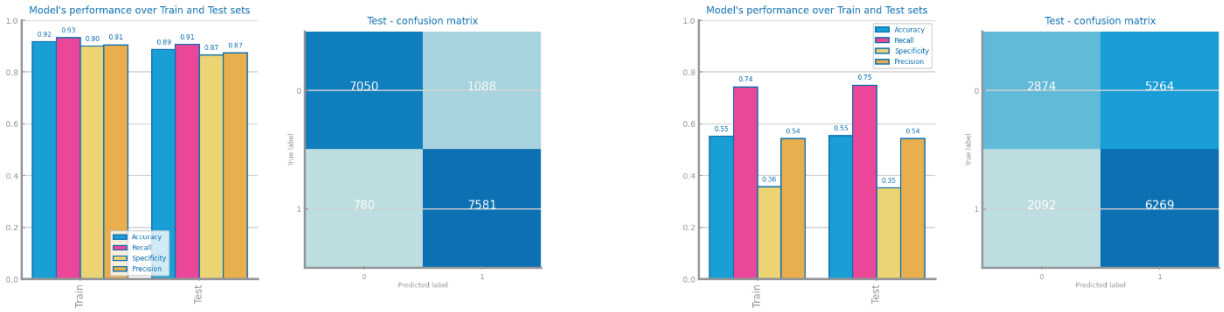



*Figure 29 Scaling results – Z-Score approach for dataset 2 - knn (left) and nb (right)*

## *Feature Selection*

**Dataset1-** Variables with the same values are eliminated by t**hreshold 0.2**, also payer_code and patient_nbr.

**Dataset2-** From NB results, feature selection with the **threshold 0.7** gives better recall.

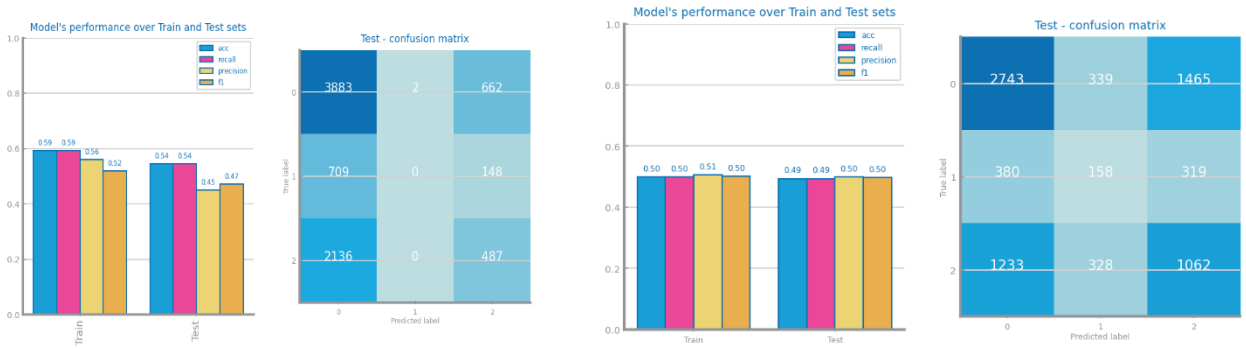



*Figure 30 Feature selection of redundant variables results – threshold:0.9 for dataset 1 - knn (left) and nb (right)*

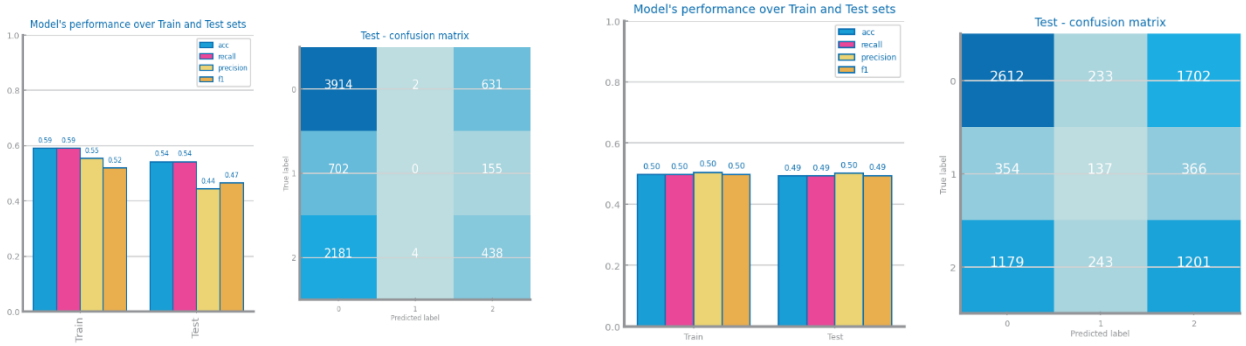



*Figure 31 Feature selection of redundant variables results – threshold:0.5 for dataset 1 - knn (left) and nb (right)*

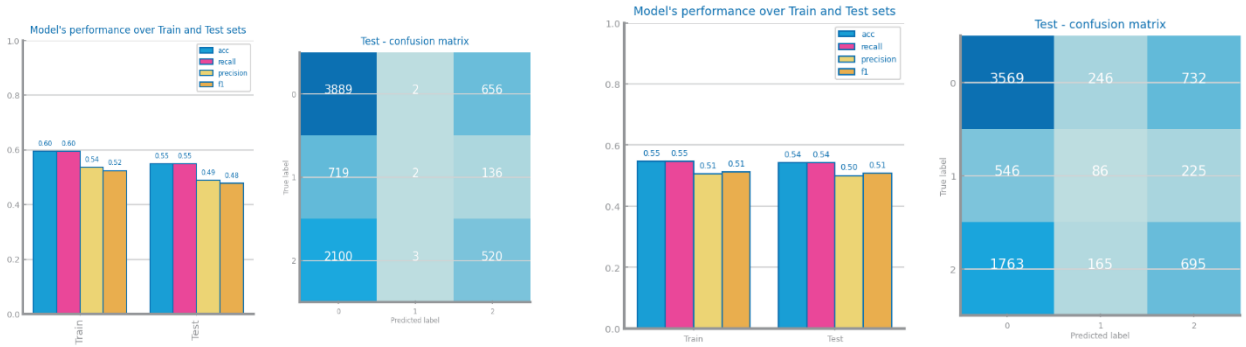



*Figure 32 Feature selection of redundant variables results – threshold:0.2 for dataset 1 - knn (left) and nb (right)*

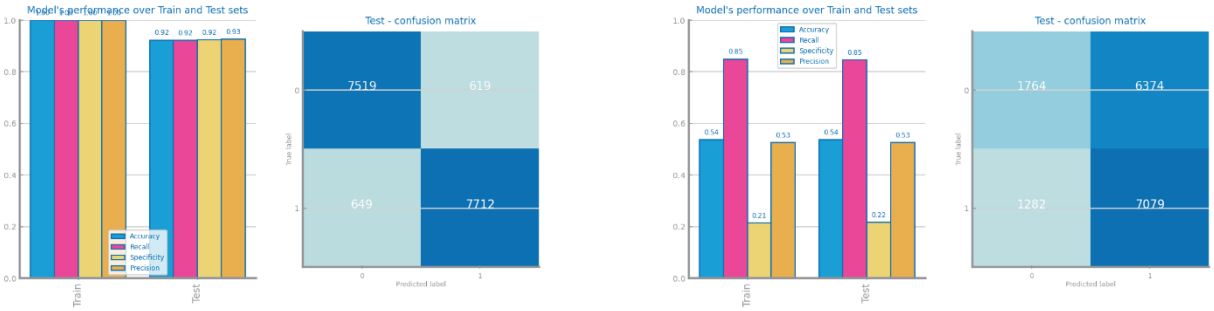



*Figure 33 Feature selection of redundant variables results – threshold:0.7 for dataset 2 - knn (left) and nb (right)*

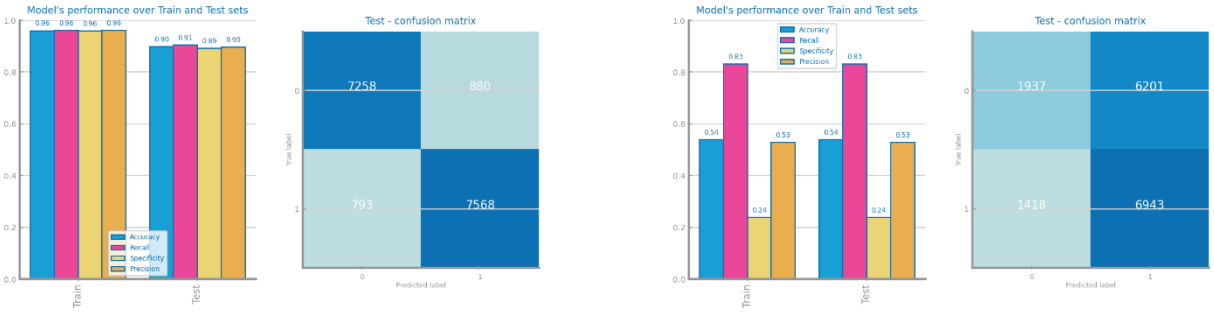



*Figure 34 Feature selection of redundant variables results – threshold:0.8 for dataset 2 - knn (left) and nb (right)*

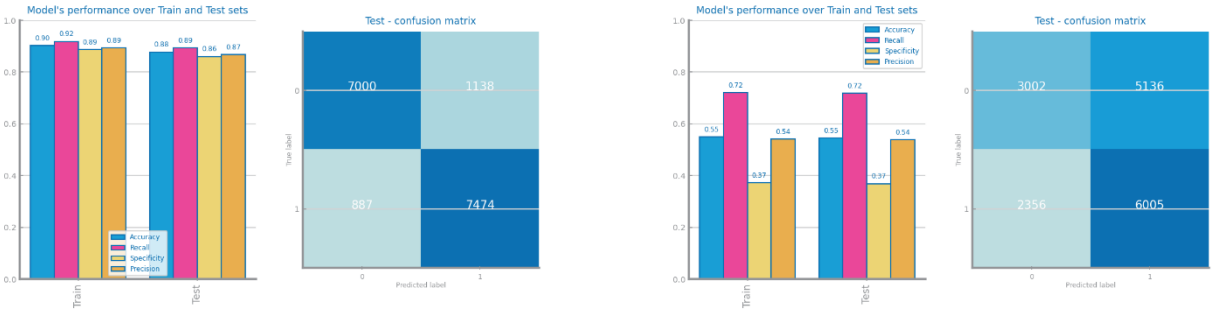



*Figure 35 Feature selection of redundant variables results – threshold:0.2 for dataset 2 - knn (left) and nb (right)*

## *Balancing*

**Dataset1–** We selected oversample because we obtained a higher precision and accuracy.

**Dataset2** - None of the strategies improved the evaluation score. (output of the feature selection with threshold 0.7 approach)

**Dataset1 Results**

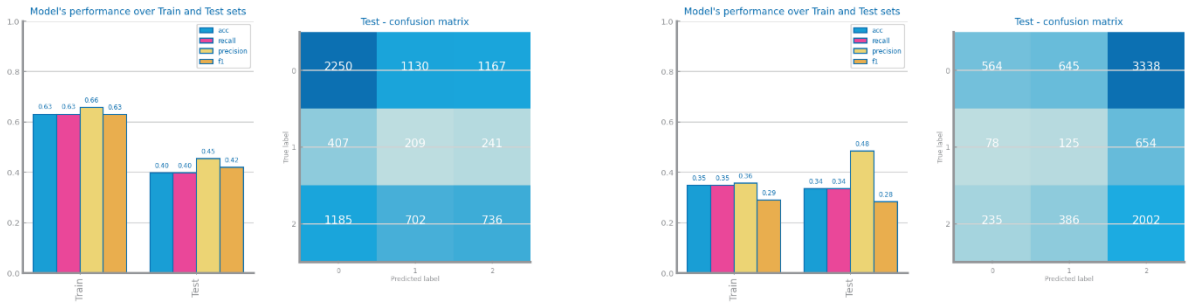



*Figure 36 Balancing results – Undersampling approach for dataset 1 - knn (left) and nb (right)*

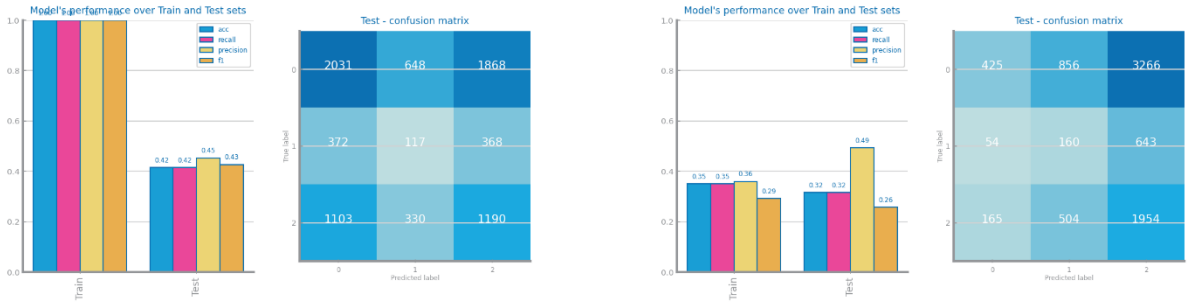



*Figure 37 Balancing results – Smote approach for dataset 1 - knn (left) and nb (right)*

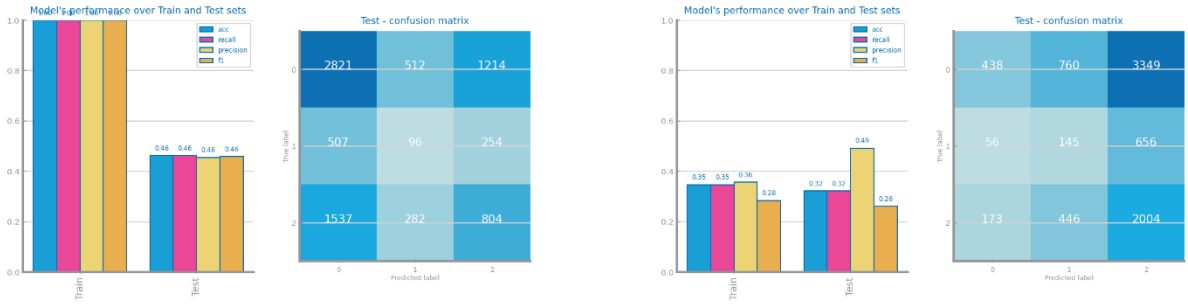



*Figure 38 Balancing results – Oversampling approach for dataset 1 - knn (left) and nb (right)*

**Dataset2 Results**

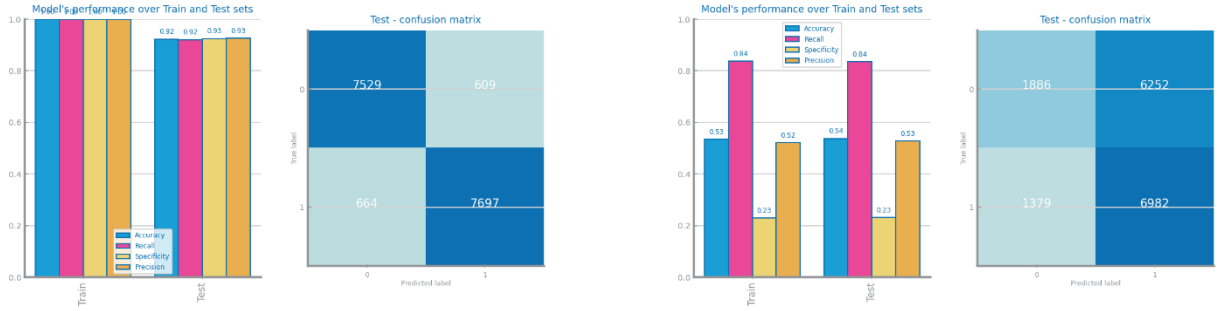



*Figure 39 Balancing results – Undersampling approach for dataset 2 - knn (left) and nb (right)*

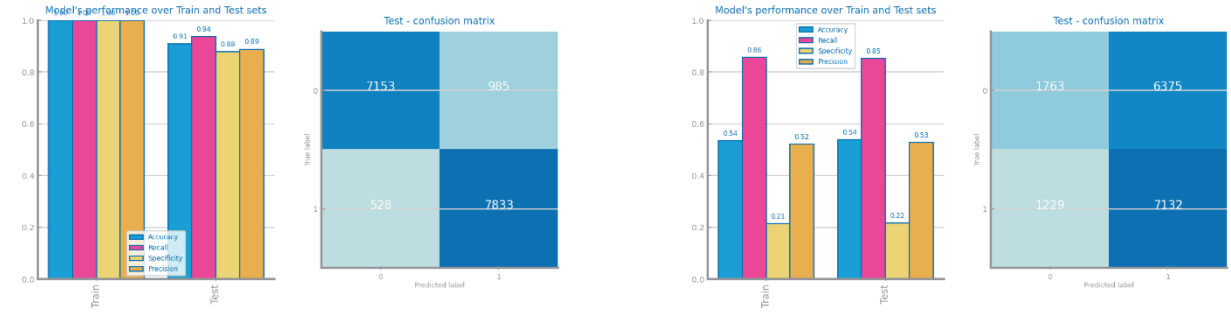



*Figure 40 Balancing results – Oversampling approach for dataset 2 - knn (left) and nb (right)*

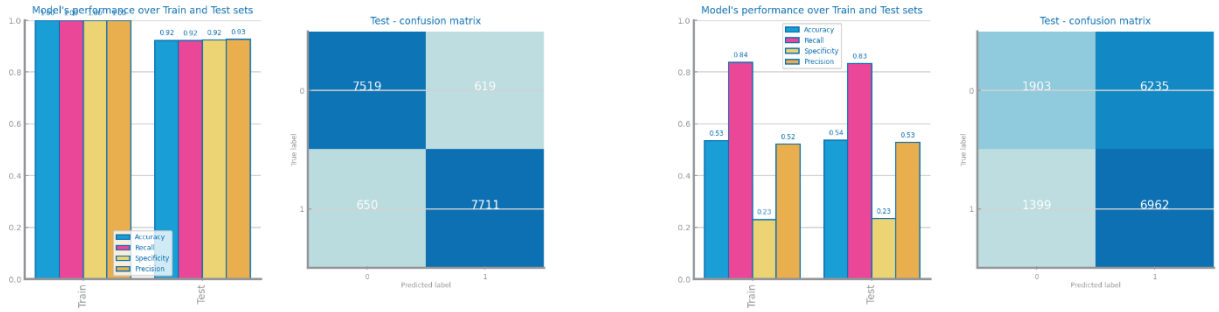



*Figure 41 Balancing results – Smote approach for dataset 2 - knn (left) and nb (right)*

# 3 MODELS' EVALUATION

For the evaluation we identified Negative=0, Positive<30 and Intermediate>30 for dataset1 and, Negative = 1, Positive = 0 for the dataset2, where we gave more importance to ACC; Recall, Accuracy and F1 results.

### *Naïve Bayes*

**Dataset1-** Accuracy is very low in the train and test dataset around 30% and it only hits that 30% of the times, the dataset is balanced (Oversample) we could say that the probability is equal to choose any random output.

**Dataset2** – We couldn't use Multinomial NB as there are negative values. Recall value is higher than the accuracy values.

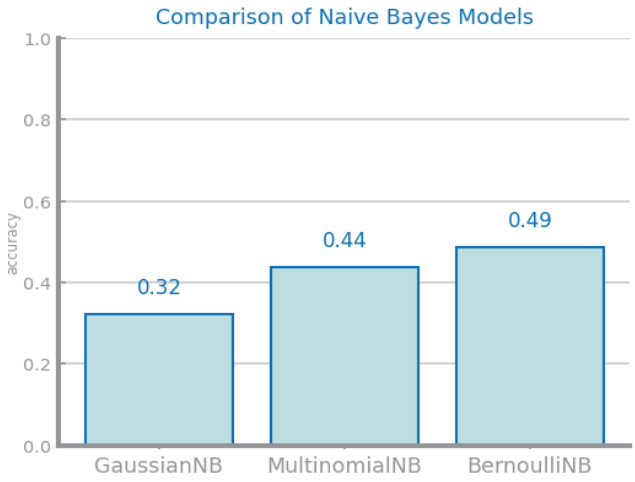



*Figure 42 Naïve Bayes alternatives comparison for dataset 1*

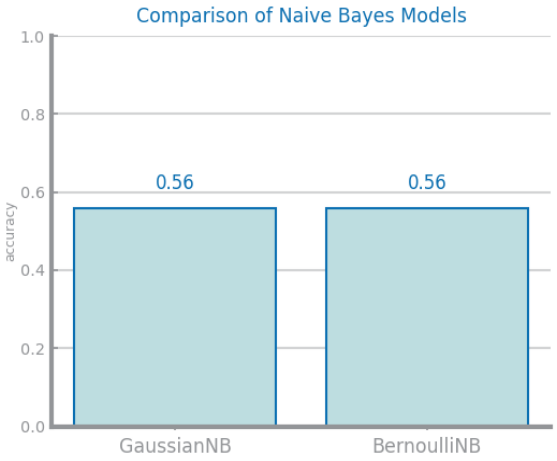



*Figure 43 Naïve Bayes alternative comparison for dataset 2*

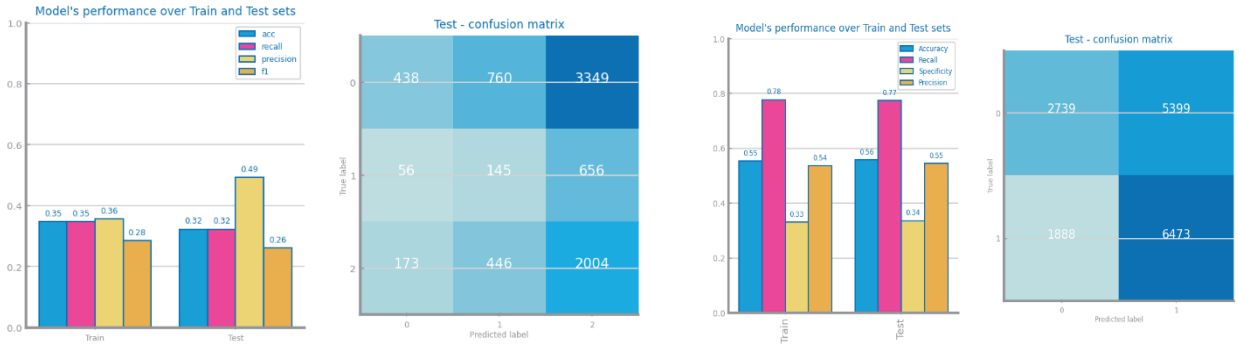



*Figure 44 Naïve Bayes best model results for dataset 1 (left) and dataset 2 (right)*

## *KNN*

**Dataset1-** Best result for us was with a neighbor near equal to 1 and Manhattan metric, precision and accuracy of 100% on the train dataset, we can say that our dataset is Overfitting because we did not have the same result for the Test dataset where we obtained a precision around 46%

**Dataset2** – In general knn has good results in terms of accuracy, recall and precision. The best result was with n_neighbours=3 and distance=manhattan

**Dataset1 Results:**

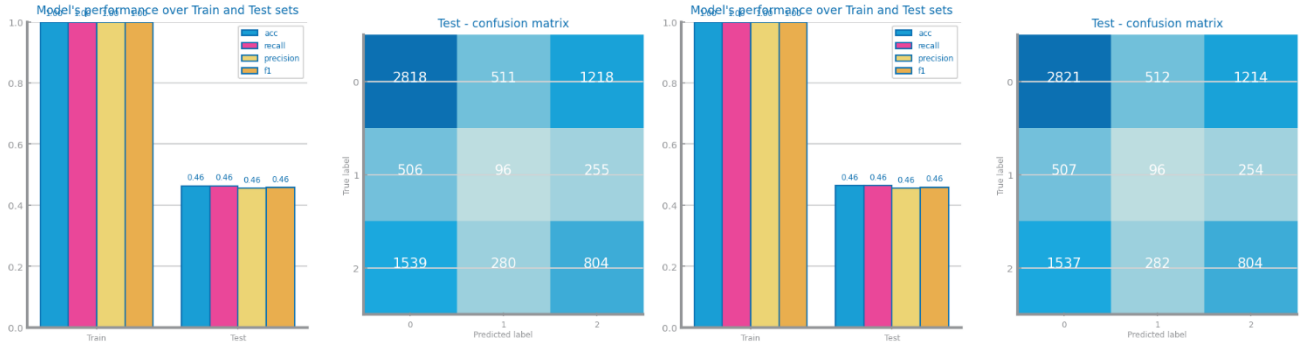



*Figure 45 KNN – parameter: distance_metric approaches comparison for dataset 1 - euclidean (left), manhattan(right)*

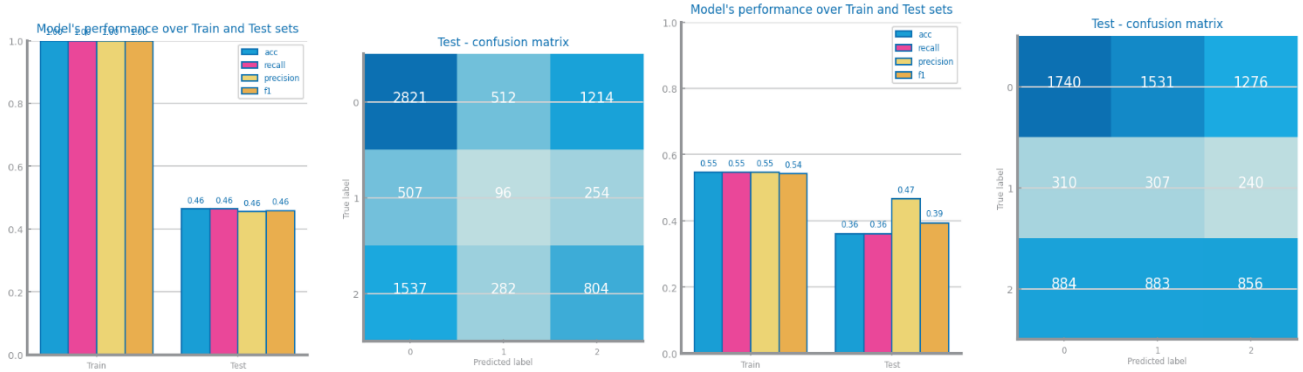



*Figure 46 KNN – parameter: n_neighbours approaches comparison for dataset 1 - n_neighbours =1 (left), n_neighbours =20 (right)*

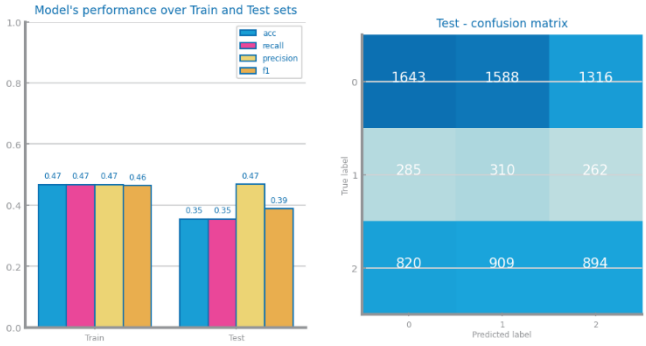



*Figure 47 KNN – parameter: n_neighbours approaches comparison for dataset 1 - n_neighbours = 50*

**Dataset2 Results:**

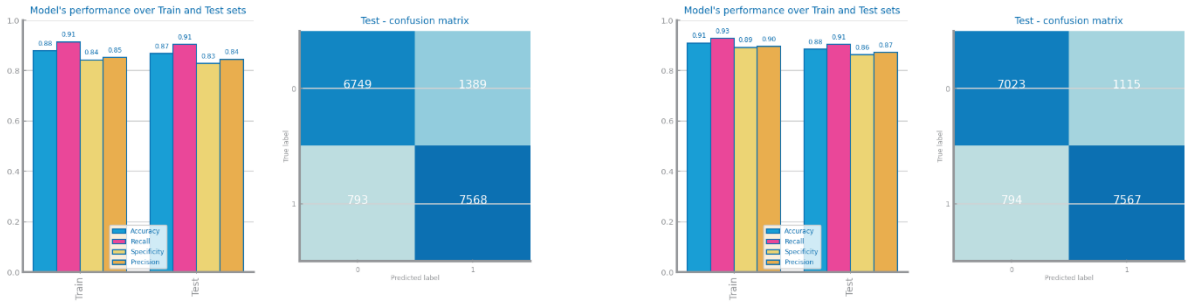



*Figure 48 KNN – parameter: distance_metric approaches comparison for dataset 2 - euclidean (left), manhattan(right)*

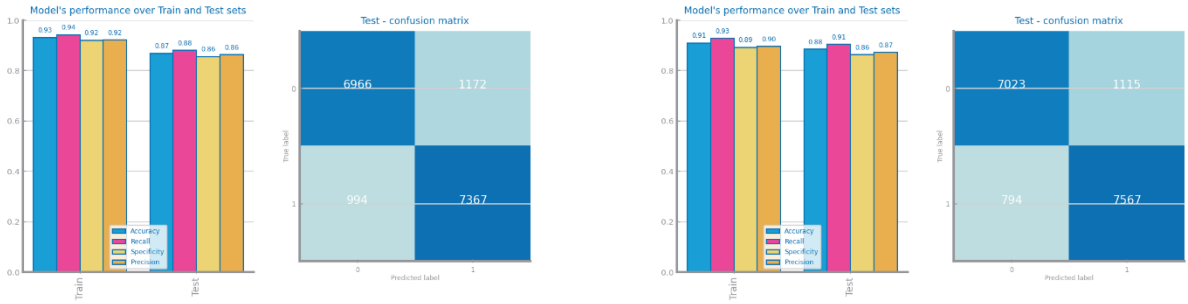



*Figure 49 KNN – parameter: n_neighbours approaches comparison for dataset 1 - **n_neighbours =3** (left), n_neighbours =9 (right)*

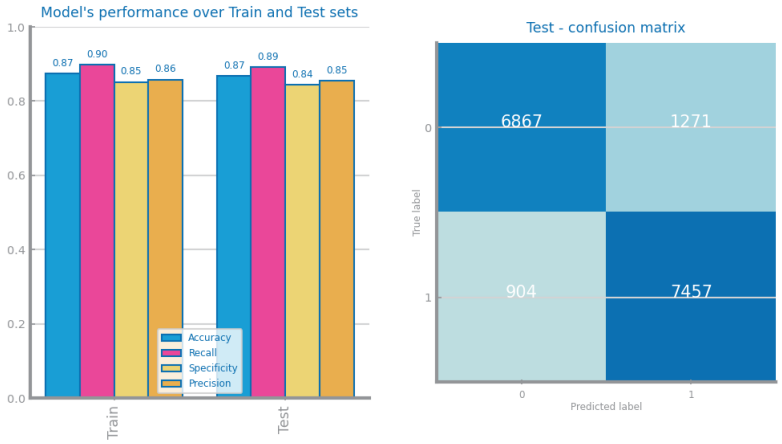



*Figure 50 KNN – parameter: n_neighbours approaches comparison for dataset 1 - n_neighbours = 50*

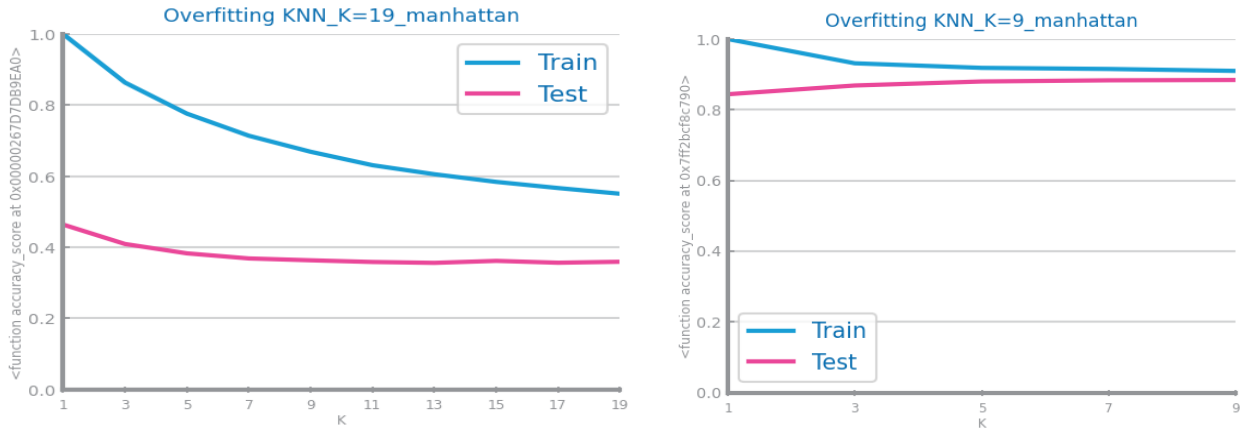



*Figure 51 KNN overfitting analysis for dataset 1 (left) and dataset 2 (right)*

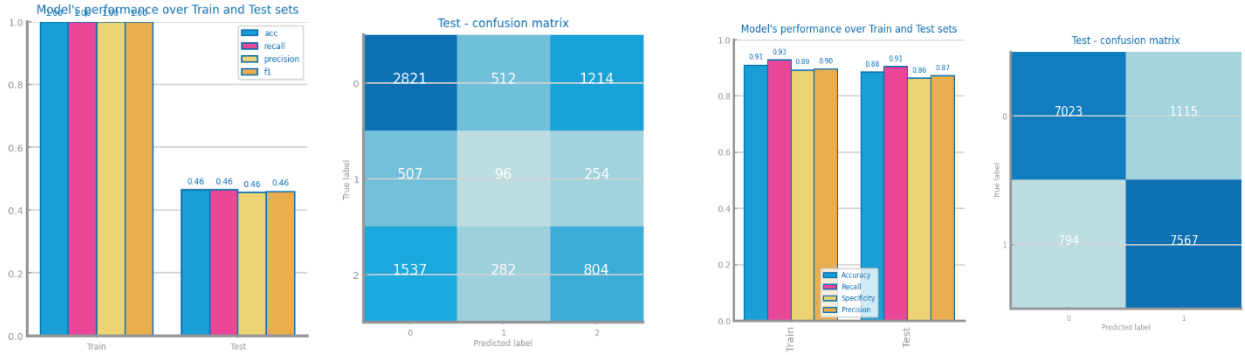



*Figure 52 KNN best model results for dataset 1 (left) and dataset 2 (right)*

## *Decision Trees*

Dataset 1: **Best: gini criteria, depth=5 and min_impurity_decrease=0.01000 ==> accuracy=0.56646**. Our classifier does not handle the minority and mainly intermediate class, because of the result presented in the confusion matrix

Dataset 2: **Best: entropy criteria, depth=20 and min_impurity_decrease=0.00050 ==> accuracy=0.88024**

For overfitting, both datasets show an increasing performance on the train dataset. Dataset 1: the model seems to be overfitting quite quickly. For dataset 2, the performance of test and train is nearly equal, which indicates that the model is not overfitting.

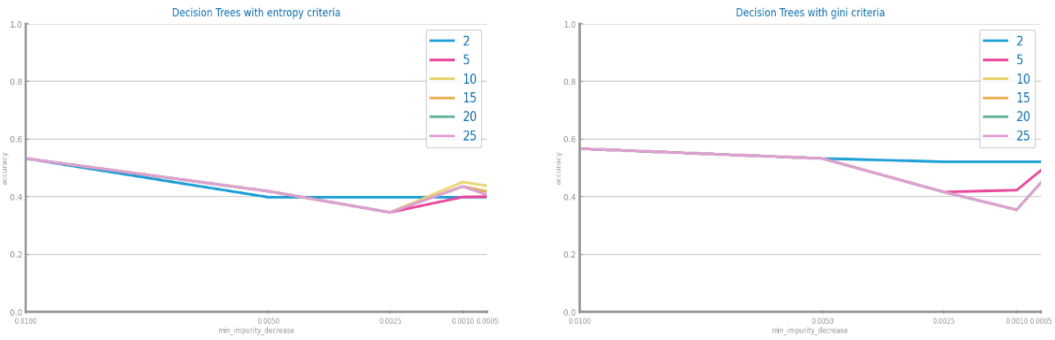



*Figure 53 Decision Trees different parameterizations comparison for dataset 1*

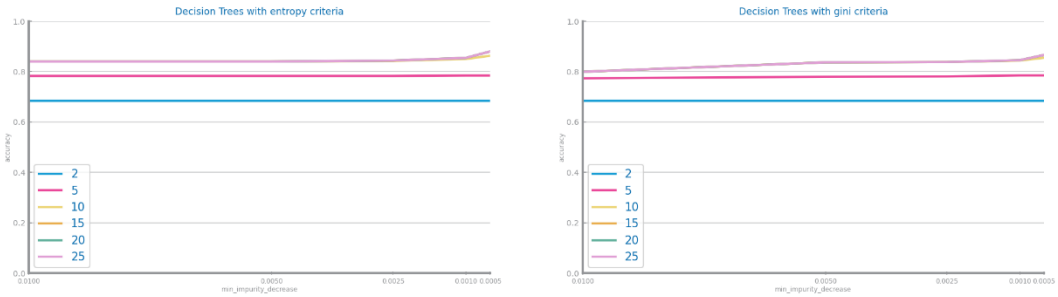



*Figure 54 Decision Trees different parameterizations comparison for dataset 2*


*Figure 55 Decision Trees overfitting analysis for dataset 1 (left) and dataset 2 (right)*

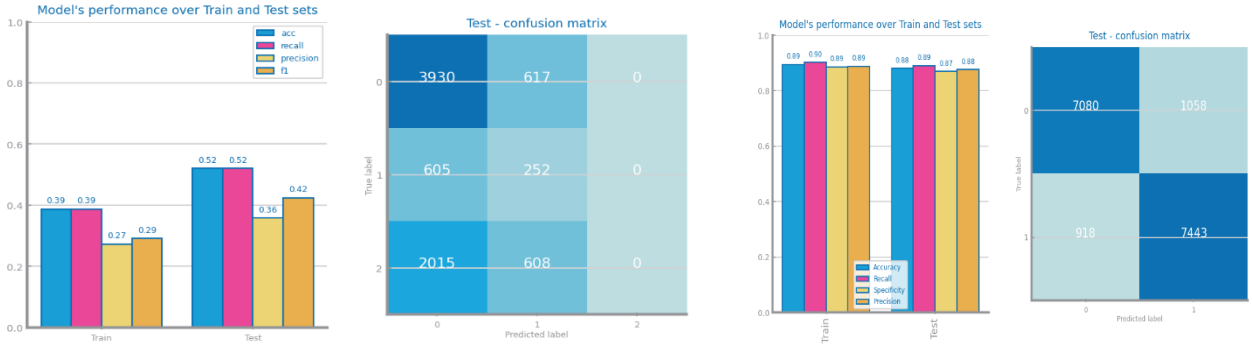



*Figure 56 Decision trees best model results for dataset 1 (left) and dataset 2 (right)*

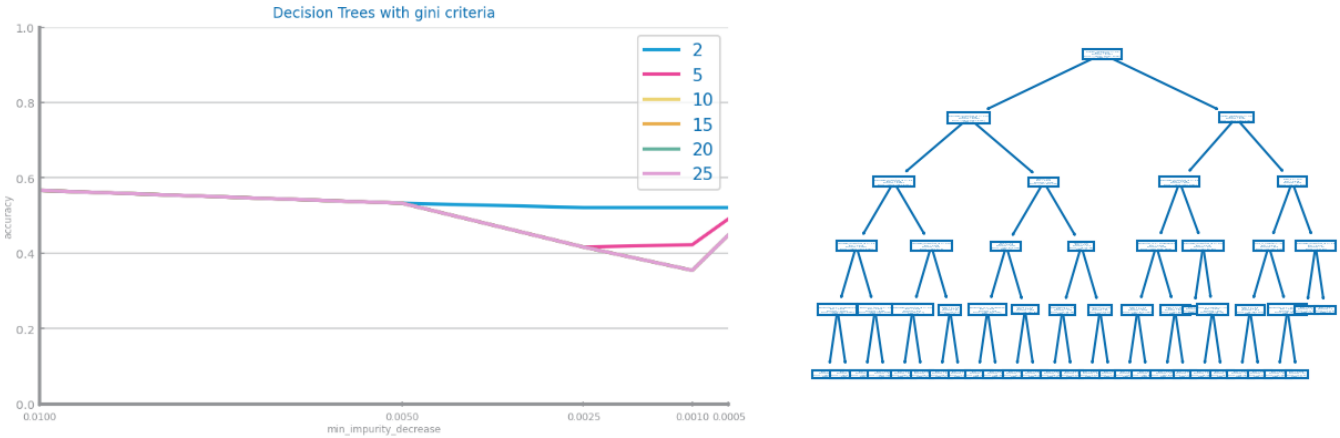
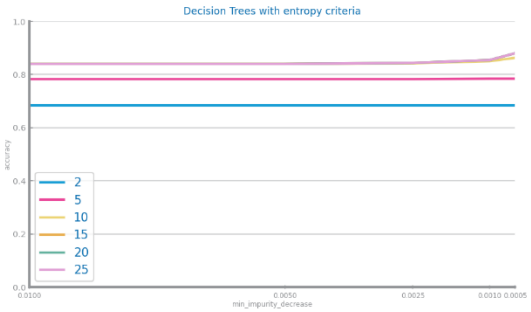



*Figure 57 Best tree for dataset 1*



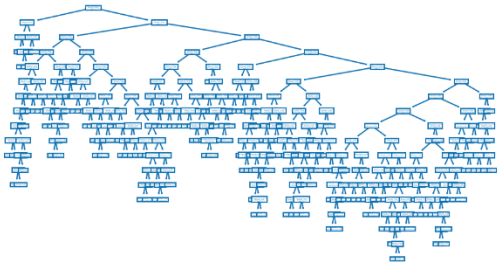



*Figure 58 Best trees for dataset 2*

## *Random Forests*

**Dataset1: Best results with depth=25, 0.30 features and 400 estimators, with accuracy=0.59**

**Dataset2: Best results with depth=25, 0.70 features and 400 estimators, with accuracy=0.88**

Both overfitting graphs show a gap between the train and test performance. This may indicate that there is an overfitting issue. The gap is however not increasing or decreasing. Especially for dataset 1 overfitting may be an issue, as the train performance is close to 1.

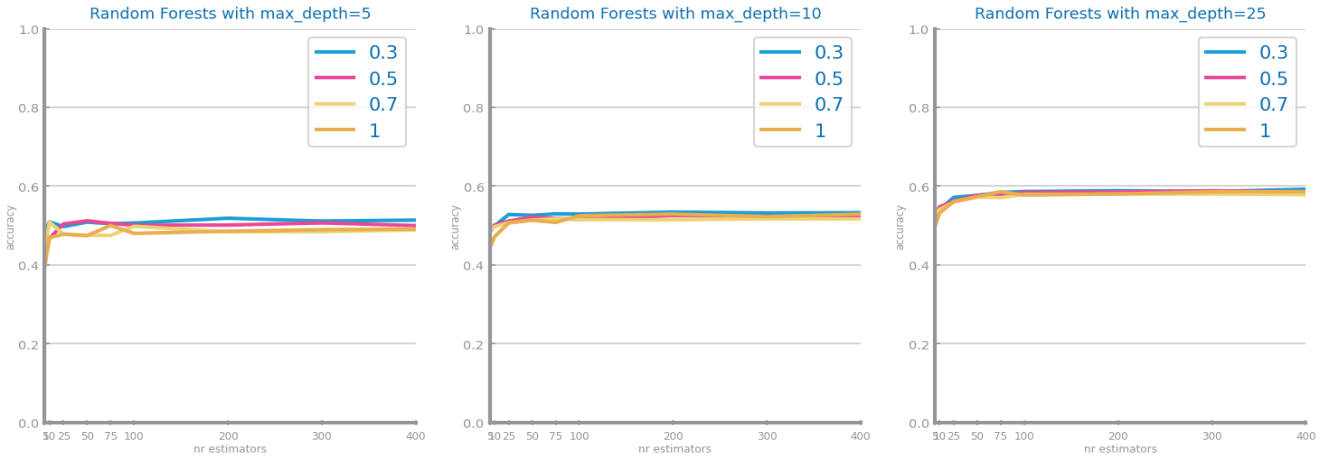



*Figure 59 Random Forests different parameterizations comparison for dataset 1*

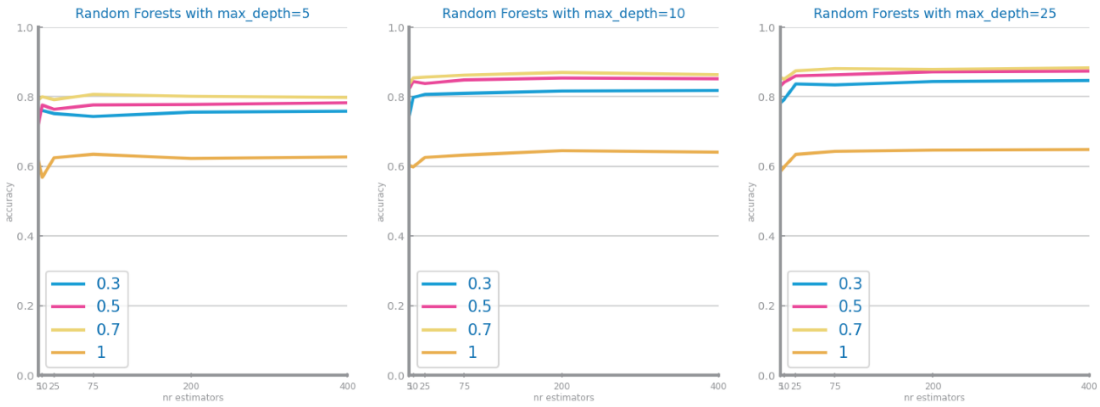



*Figure 60 Random Forests different parameterizations comparison for dataset 2*

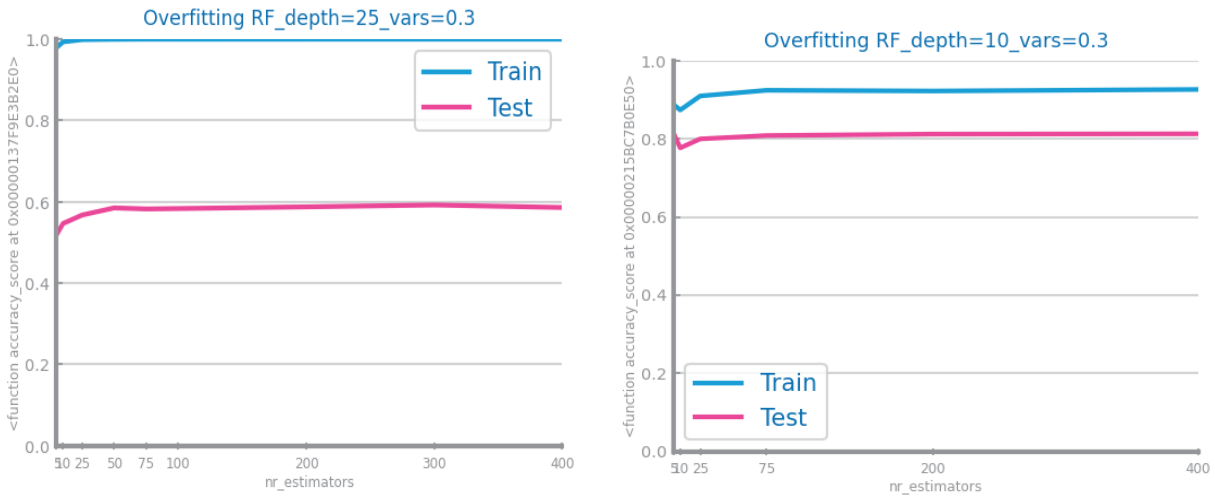



*Figure 61 Random Forests overfitting analysis for dataset 1 (left) and dataset 2 (right)*

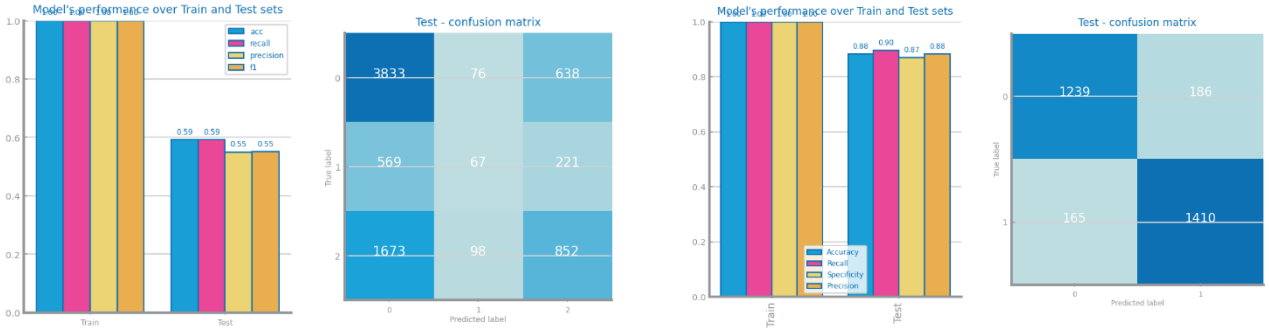



*Figure 62 Random Forests best model results for dataset 1 (left) and dataset 2 (right)*

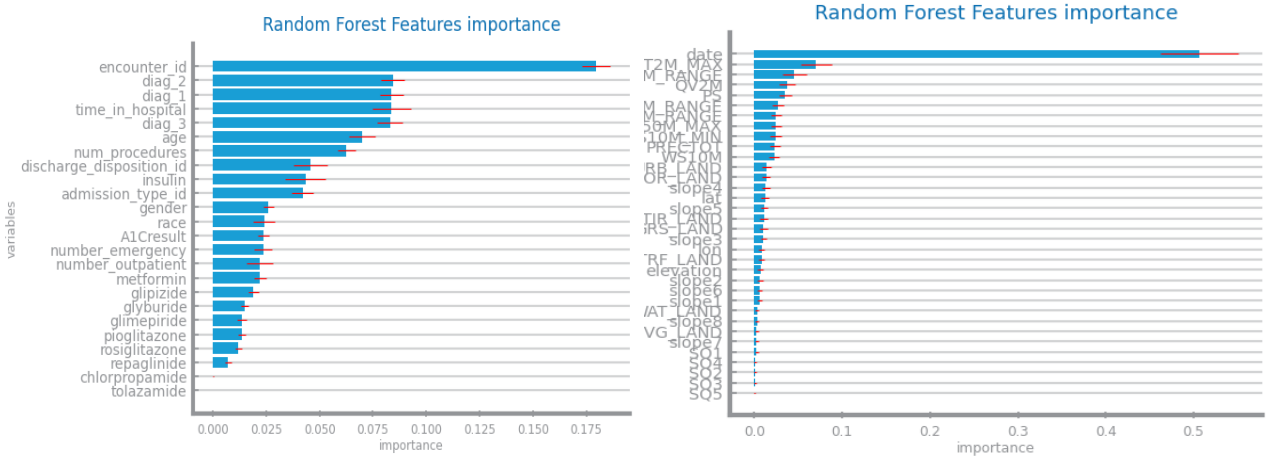



*Figure 63 Random Forests variables importance for dataset 1 (left) and dataset 2 (right)*

## *Gradient Boosting*

Dataset1**-** This classifier gives us a good result in the analysis with respect to accuracy and precision for the training dataset and not so bad in the test dataset **accuracy is 0.58, the learning rate is 0.50 with a depth of 25.**

Dataset2: The accuracy for dataset 2 is quite good because the RMSE and MAE present results close to 0 with an R-Squared close to 100%. **Best results achieved with 0.1 learning rate, depth=25 and nr estimators=100 ==> measure=0.11**

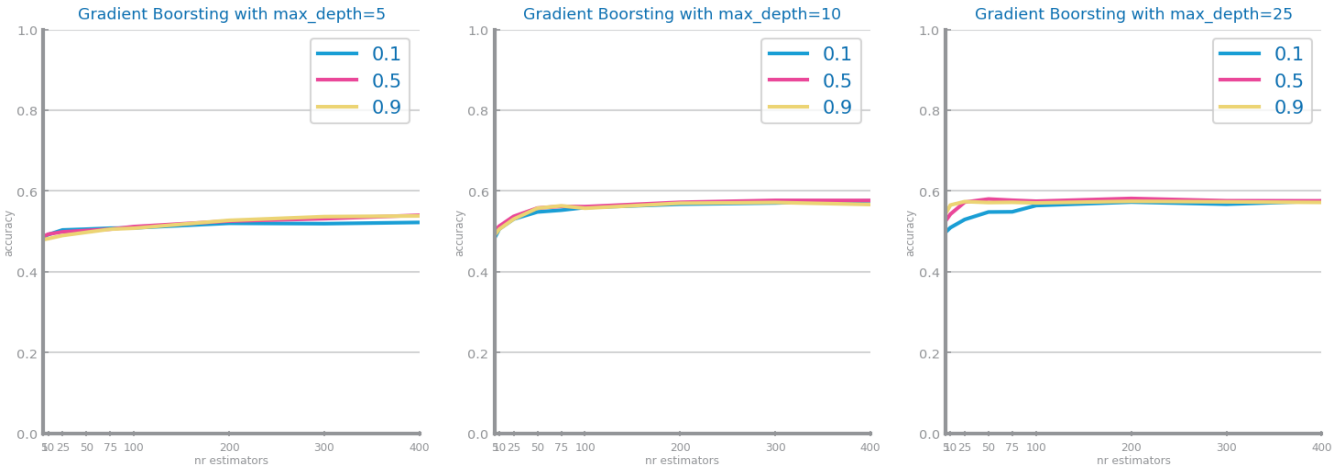



*Figure 64 Gradient boosting different parameterizations comparison for dataset 1*

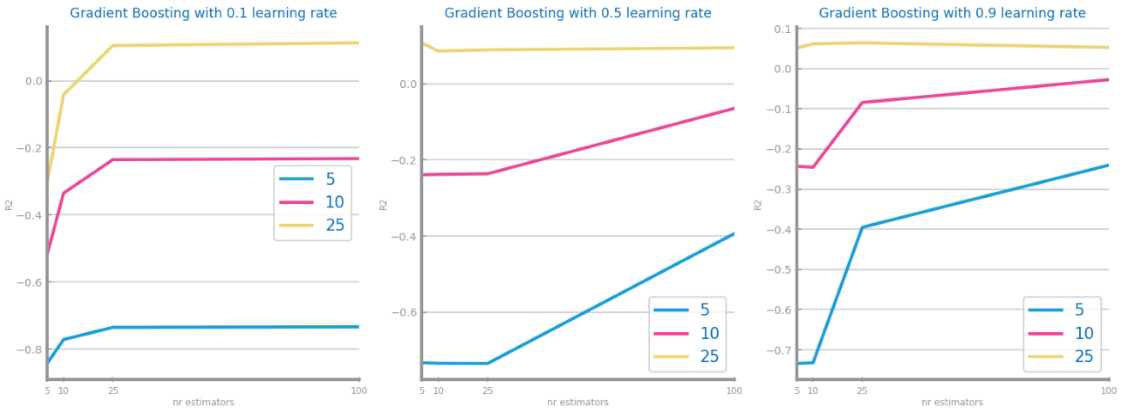



*Figure 65 Gradient boosting different parameterizations comparison for dataset 2*

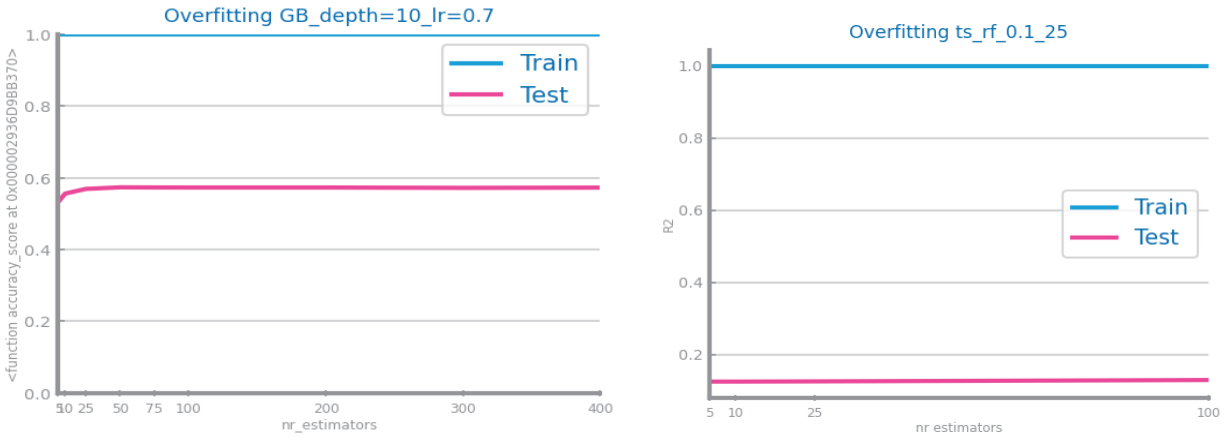



*Figure 66 Gradient boosting overfitting analysis for dataset 1 (left) and dataset 2 (right)*

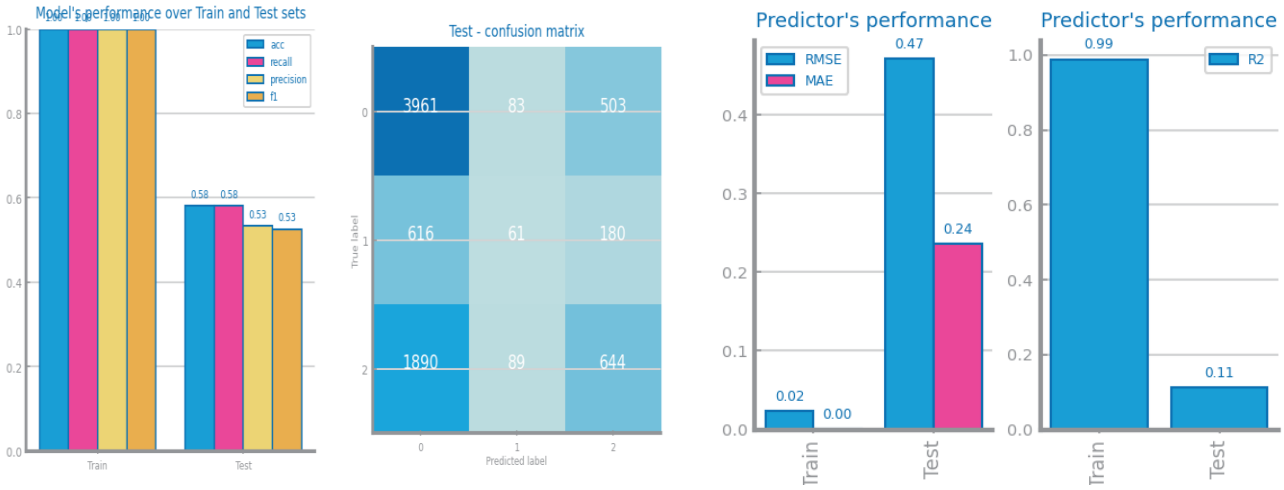



*Figure 67 Gradient boosting best model results for dataset 1 (left) and dataset 2 (right)*

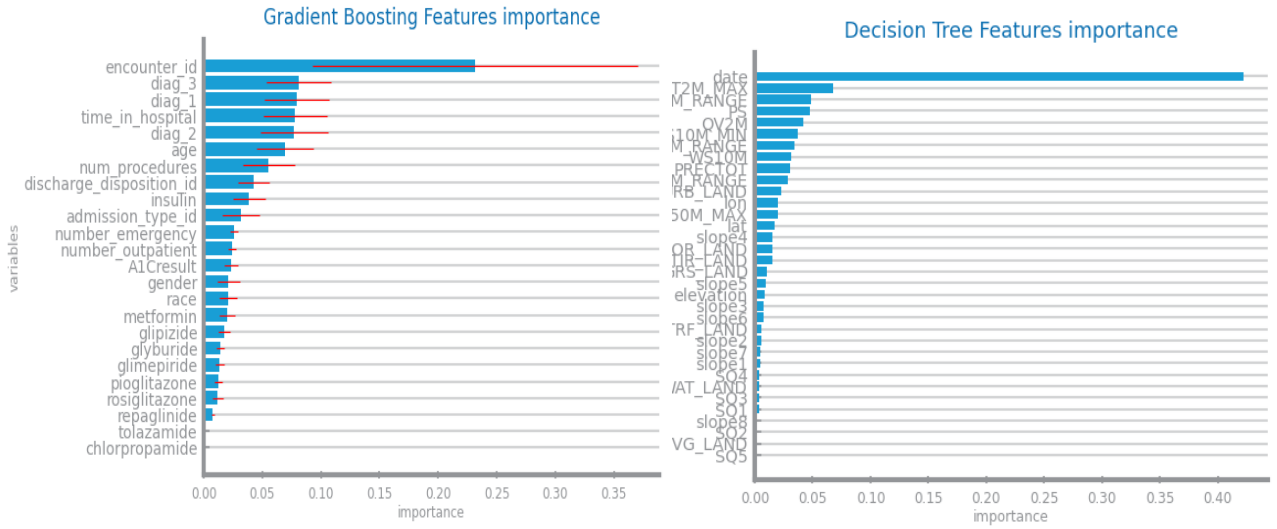



*Figure 68 Gradient boosting variables importance for dataset 1 (left) and dataset 2 (right)*

## *Multi-Layer Perceptrons*

**Dataset1-** We can observe for the MAE and RMSE are higher, and we can improve our model by reducing the error. Best results with **lr_type=constant, learning rate=0.1 and 100 max iter, with accuracy=0.5664631867447365**

**Dataset2:** Best results with **lr_type=constant, learning rate=0.1 and 1000 max iter ==> measure=0.00**

The MLP for both datsets do not learn any logic. The measures are as good as random guessing. All the parameters have been tuned for testing: hidden layers, optimizers (sgd and adam), act. function, without any result.

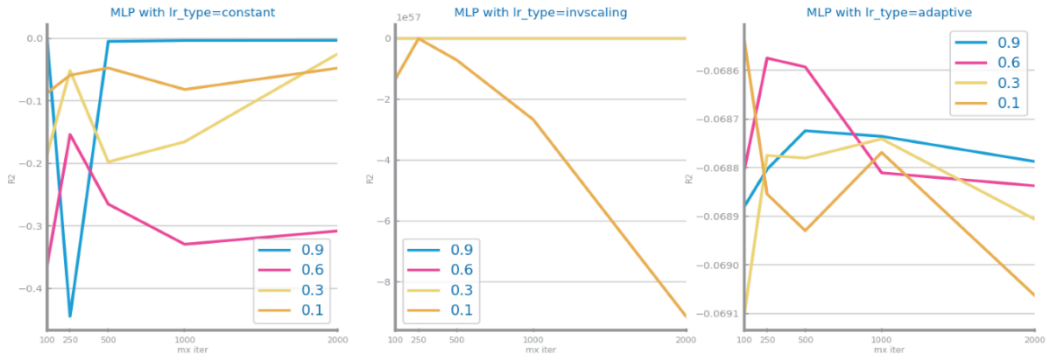



*Figure 69 MLP different parameterizations comparison for dataset 1*

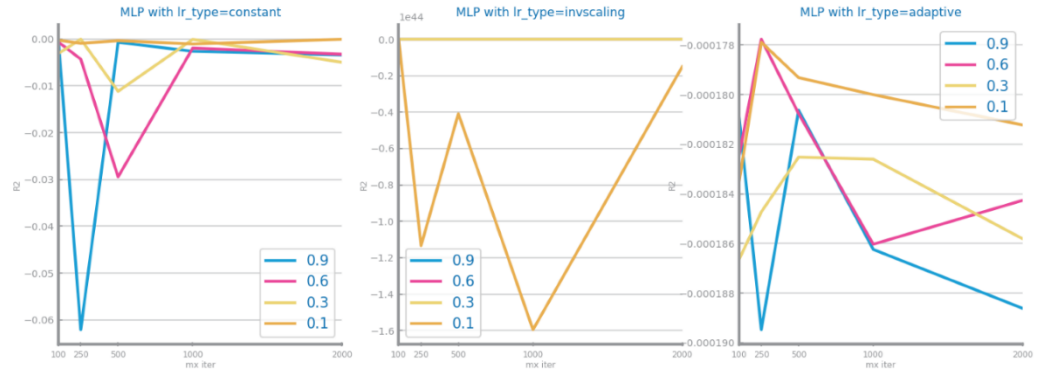



*Figure 70 MLP different parameterizations comparison for dataset 2*

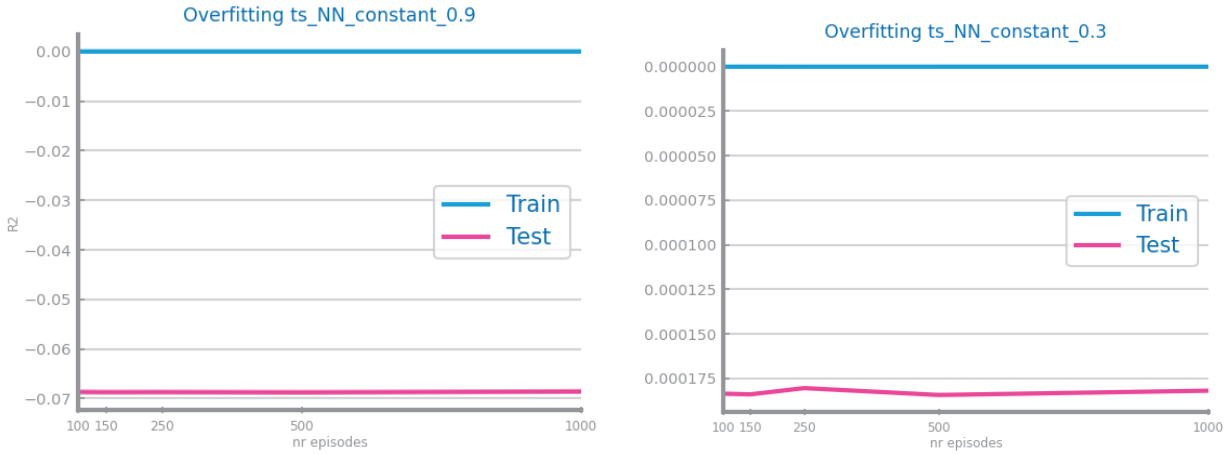



*Figure 71 MLP overfitting analysis for dataset 1 (left) and dataset 2 (right)*

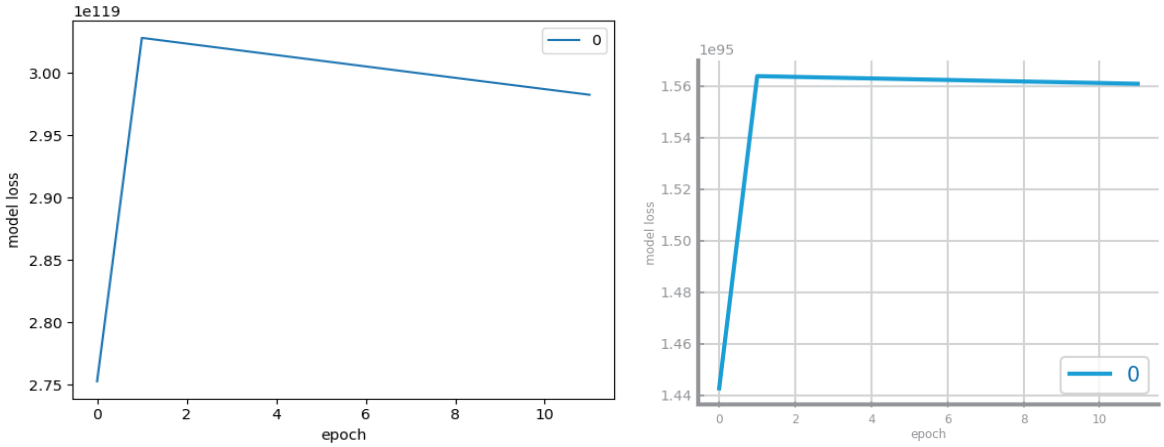



*Figure 72 Loss curves analysis for dataset 1 (left) and dataset 2 (right)*

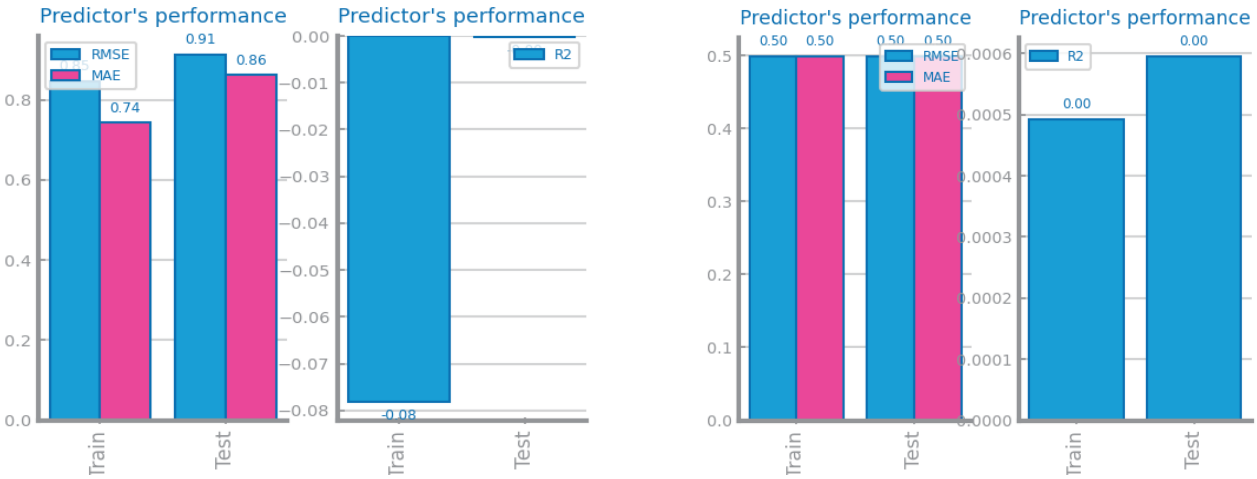



*Figure 73 MLP best model results for dataset 1 (left) and dataset 2 (right)*

# 4 CRITICAL ANALYSIS

For dataset 1 since most of our variables were categorical we decided to do the coding step using OneHotEncoder for all categorical variables, the binary variables have simple values like 0 and 1, for the case of diagnostic variables we decided to use the International Classification of Diseases, according to the Center for Disease Prevention and Control. After performing the variable encoding these variables present discrete values. For the missing value imputation, we deeply analyzed and ran different results to obtain the method that best fits our analysis, we decided to remove some columns, eliminate the missing values and we replaced some outliers in the column num_lab_procedures.

For dataset 2, Variables 'QV2M', 'T2-', 'TS' and 'WS' show high levels of granularity. Problems occurring can be a high storage space, and worse results. Storage space is not an issue for now, since the programming software handles the data fine. We could conclude that balancing was not needed for the analysis which can be reflected by the almost equal class distribution during the data profiling.

For both datasets, the random forest and decision tree give decent classification results. As expected, the random forest measure is as least as good as decision tree, because it is an aggregation of decision trees. The MLP and gradient boost results are bad. For the MLP to have this bad performance, is probably due to the data being not suited for learning, or parameterization of the MLP. The gradient boosting seems to be overfitting very quickly. The performance of the gradient boosting algorithm should be improved by reducing the overfitting. This may be a hard thing to do, since it can also be a result of the data being noisy.

# 5 DATA PROFILING

## *Data Granularity*

**TS1- T**he most atomic granularity considered is **hourly** with consistent interval at 08, 12, 18 and 22. To get the granularity of an hour we need to interpolate the measurements during the intervals.

**TS1-** The data present a daily granularity with no repeated records almost every day, so we considered **daily** to be the most atomic.

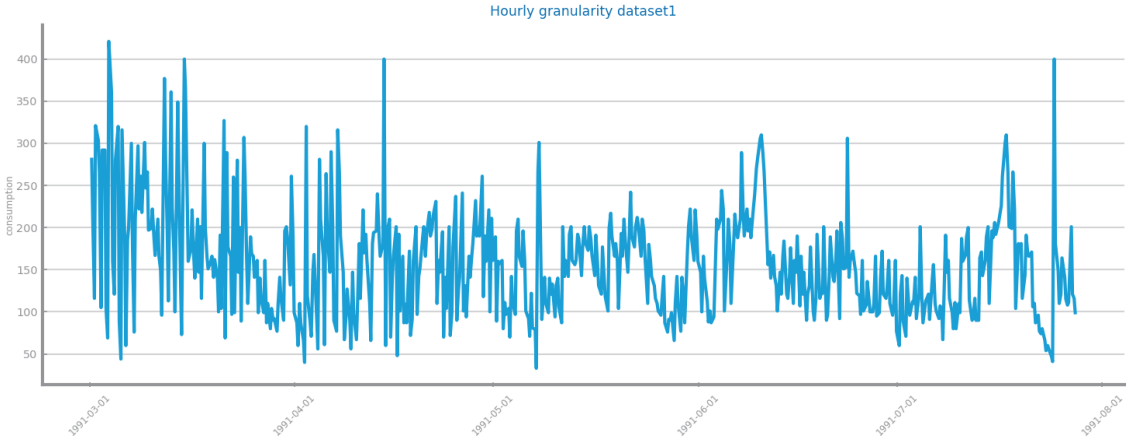



*Figure 74 Time series 1 at the most granular detail - hourly*

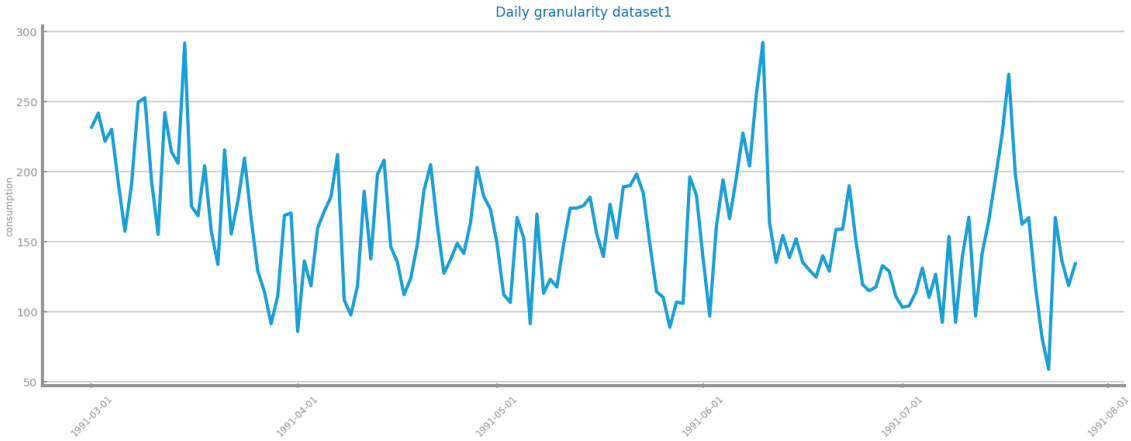



*Figure 75 Time series 1 at the second chosen granularity - daily*

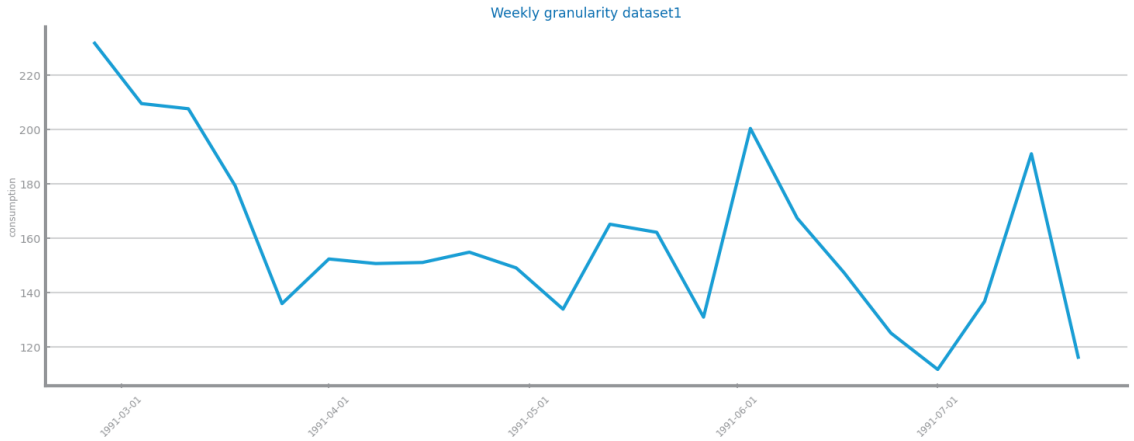



*Figure 76 Time series 1 at the third chosen granularity - weekly*

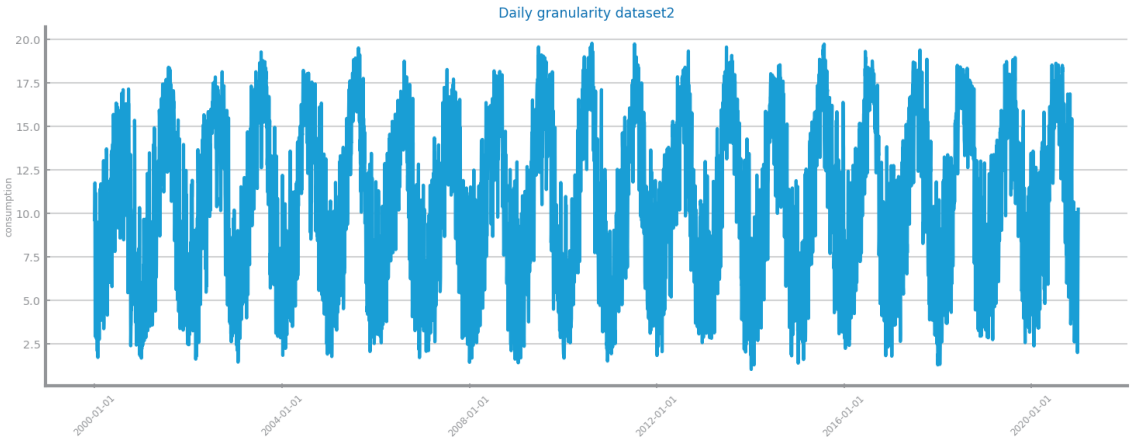



*Figure 77 Time series 2 at the most granular detail - daily*

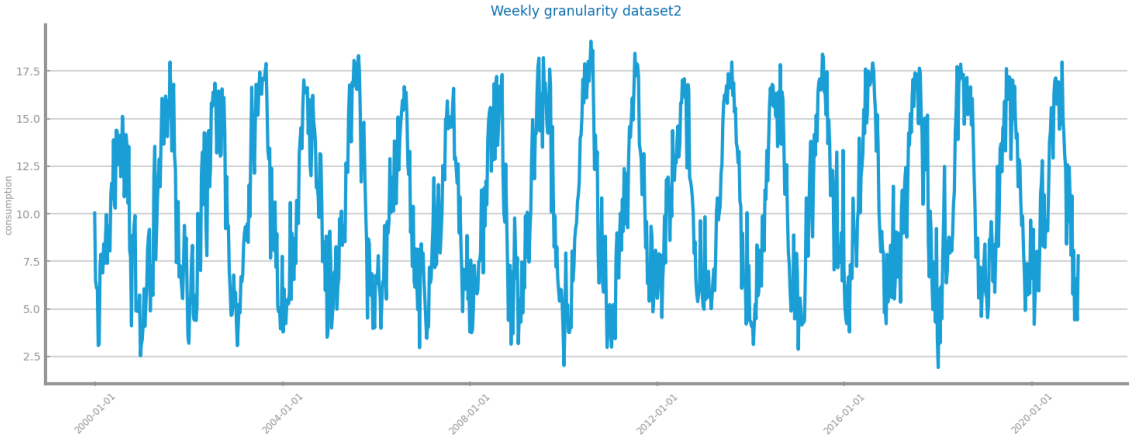



*Figure 78 Time series 2 at the second chosen granularity - weekly*

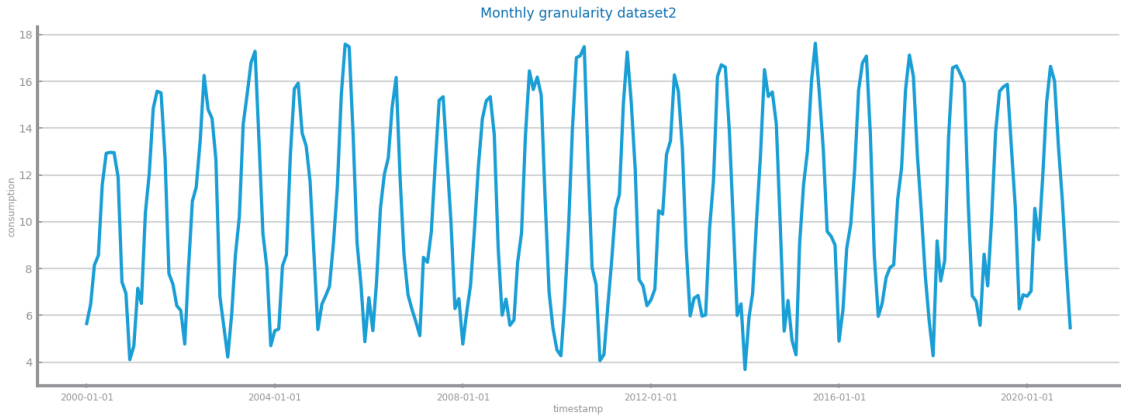



*Figure 79 Time series 2 at the third chosen granularity - monthly*

### *Data Distribution and Stationarity*

**TS1-** There are only 4 data that are greater than or equal to 400, so we can consider that there are no outliers.

**TS2-** Stationarity has the property that the mean and target variable do not change over time.

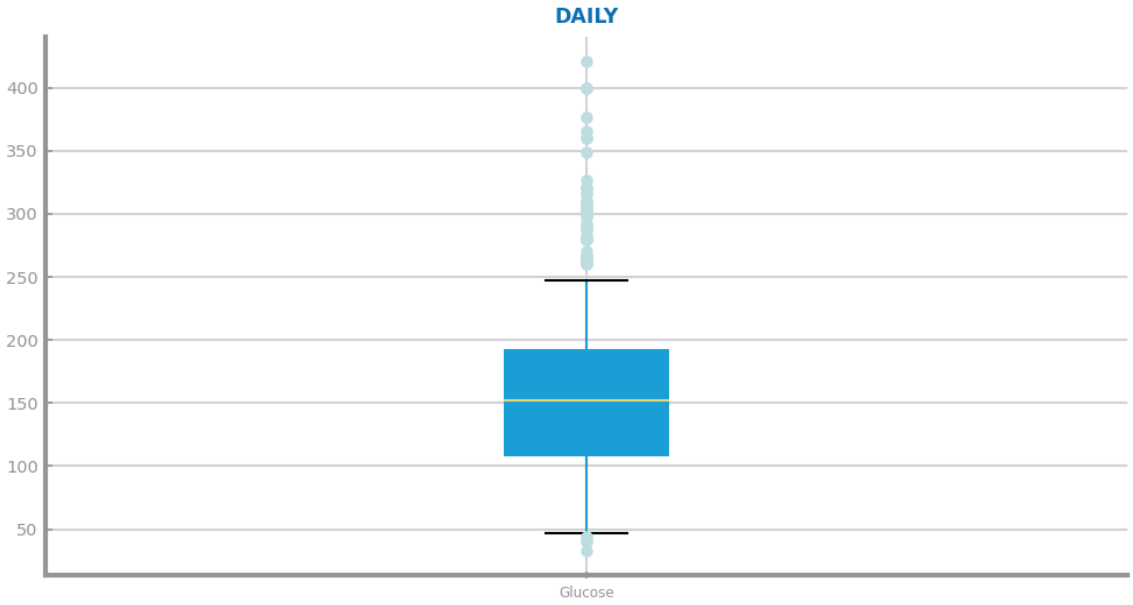



*Figure 80 Boxplot(s) for time series 1*

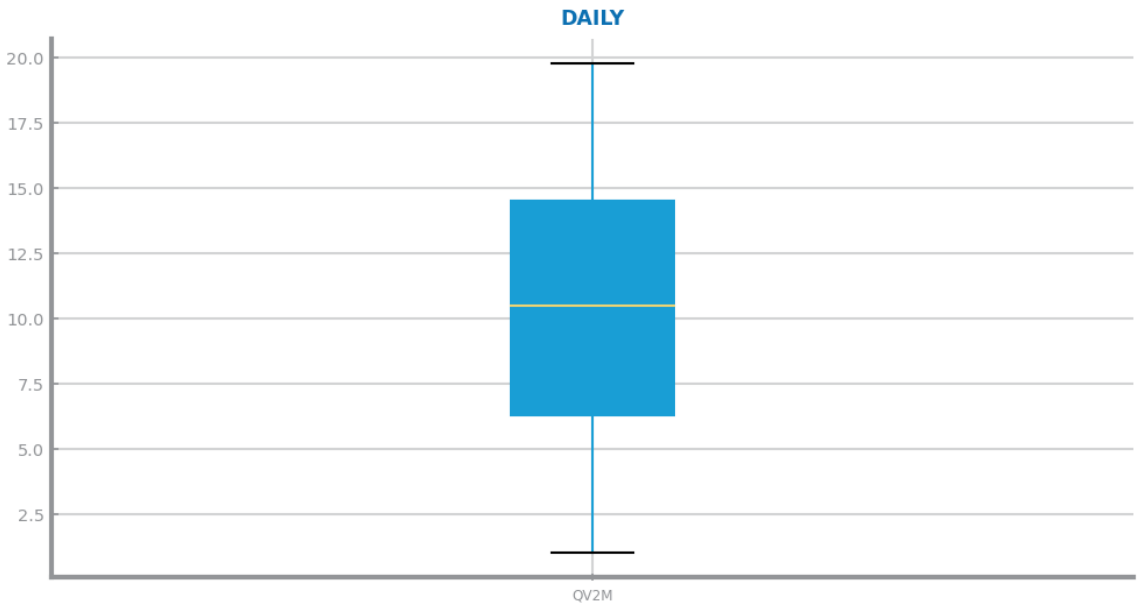



*Figure 81 Boxplot(s) for time series 2*

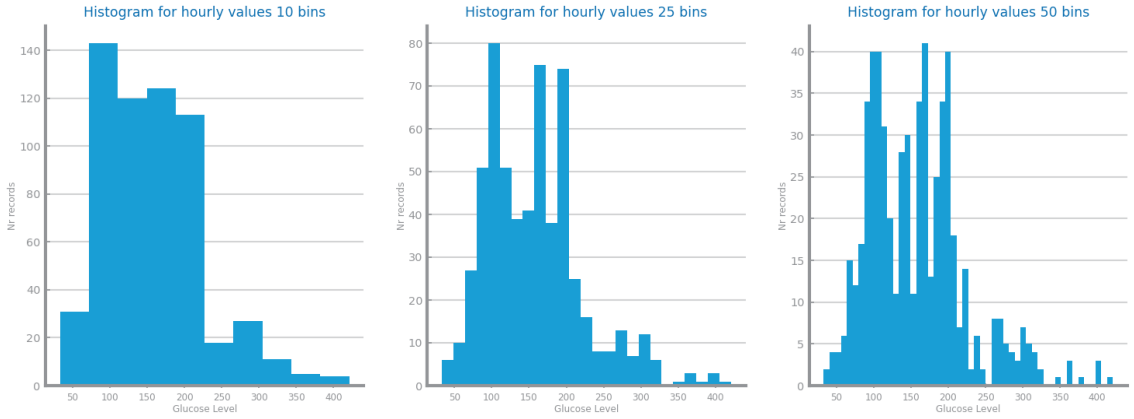



*Figure 82 Histogram(s) for time series 1 - hourly*

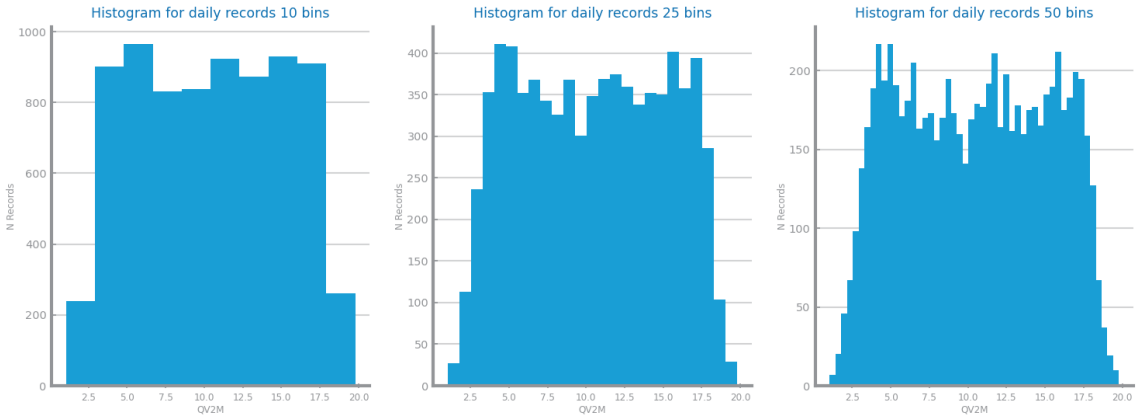



*Figure 83 Histogram(s) for time series 2 – daily records*

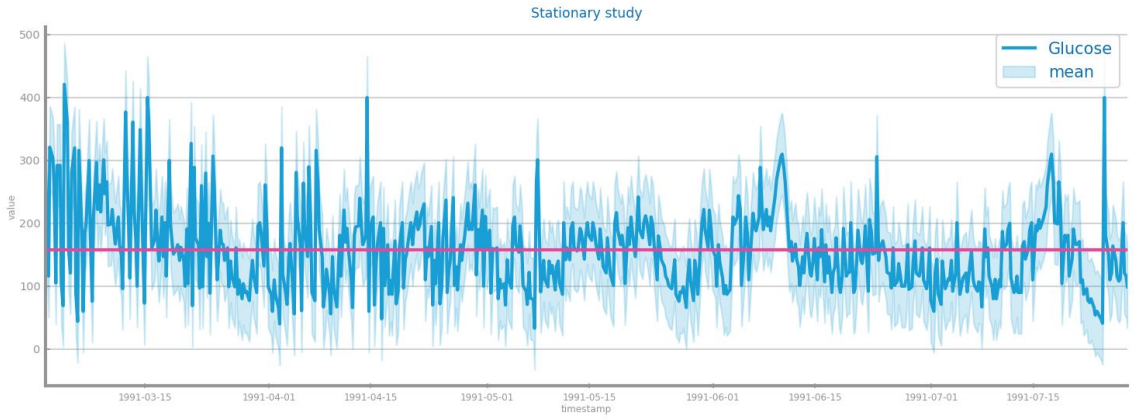



*Figure 84 Stationarity study for time series 1*

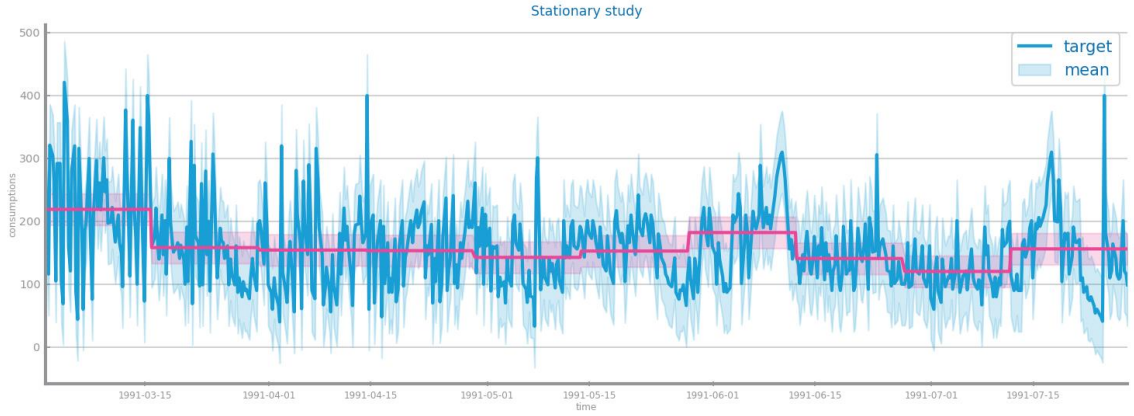



*Figure 85 Segmented Stationarity study for time series 1*

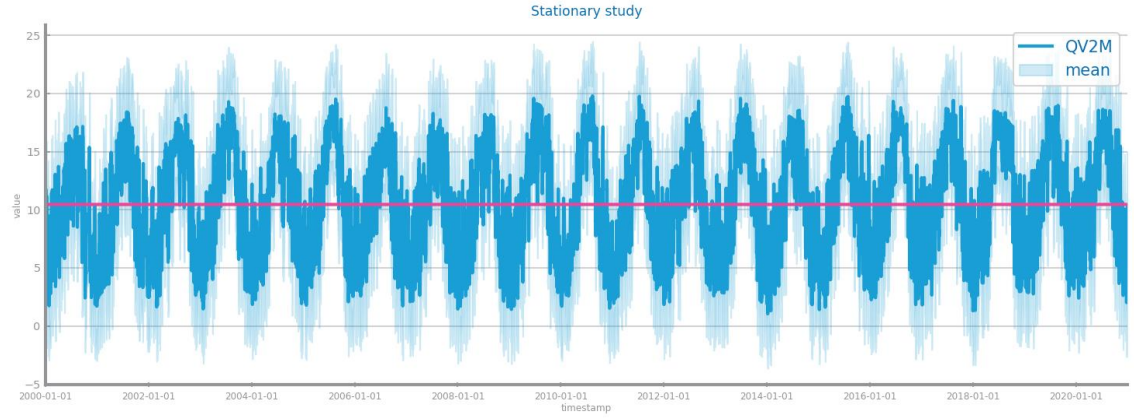



*Figure 86 Stationarity study for time series 2*

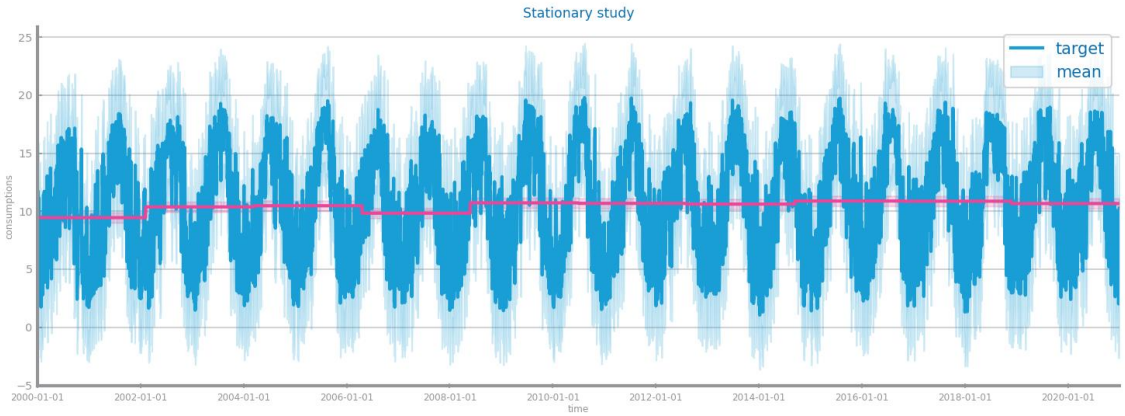



*Figure 87 Segmented stationarity study for time series 2*

# 6 DATA TRANSFORMATION

## *Aggregation*

**TS1-** We observe a lag between the prediction and the actual data, the lag is consistent in the training set and the test set.

**TS2-** Our dataset is quite persistent and the low values given in RMSE and MAE indicate a low accuracy

**Time Series 1 - Results**

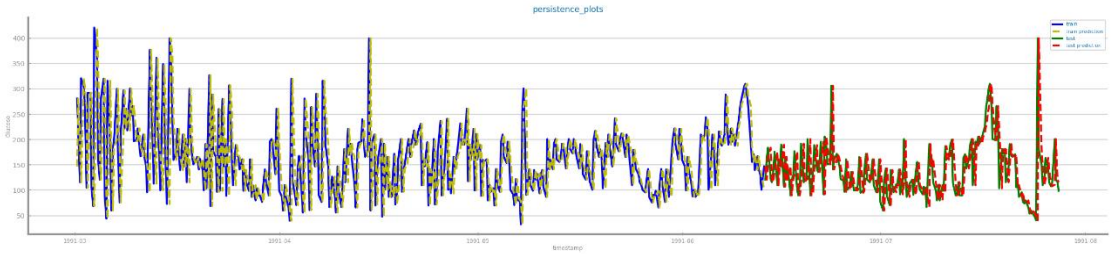



*Figure 88 Forecasting plots after **hourly** aggregations on time series 1*

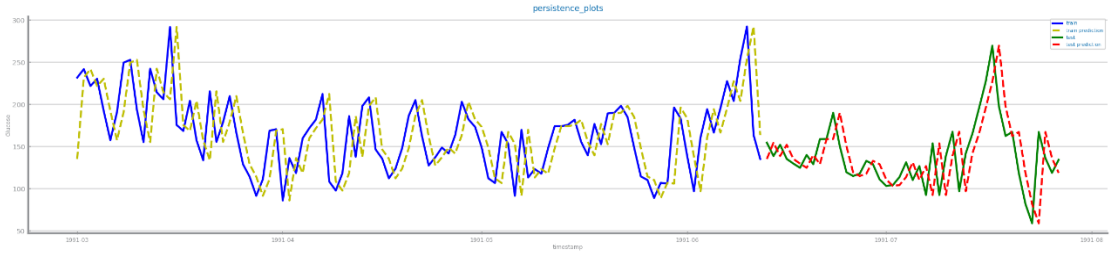



*Figure 89 Forecasting plots after **daily** aggregations on time series 1*

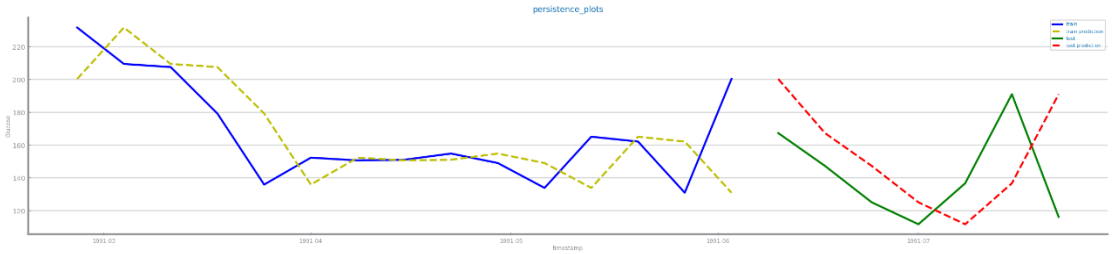



*Figure 90 Forecasting plots after **weekly** aggregations on time series 1*

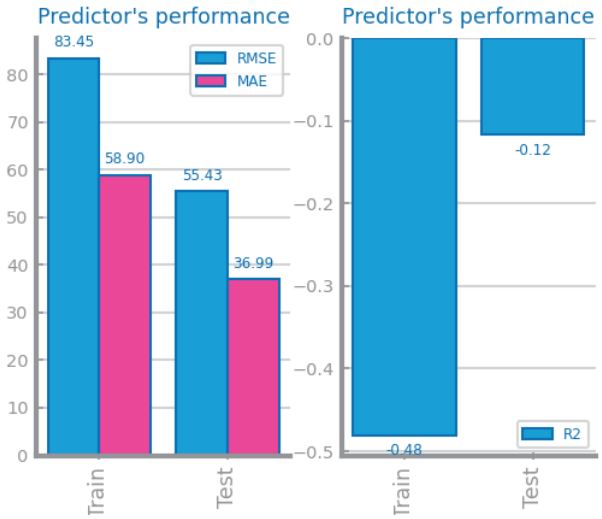



*Figure 91 Forecasting results after **hourly** aggregations on time series 1*

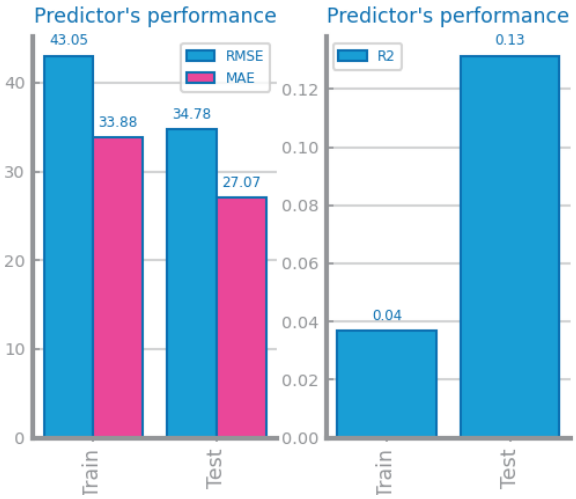



*Figure 92 Forecasting results after **daily** aggregations on time series 1*

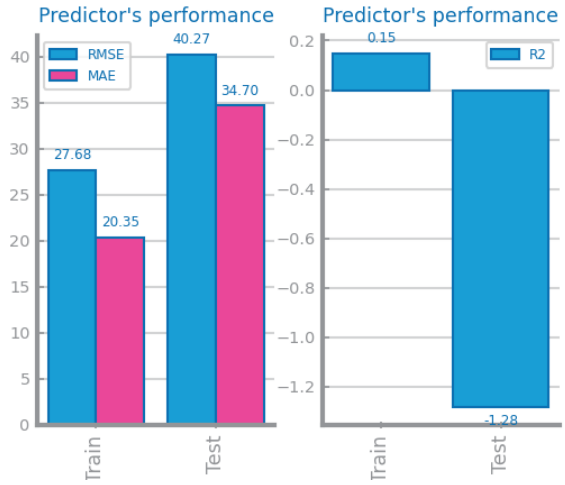



*Figure 93 Forecasting results after **weekly** aggregations on time series 1*

## Time Series 2 - Results

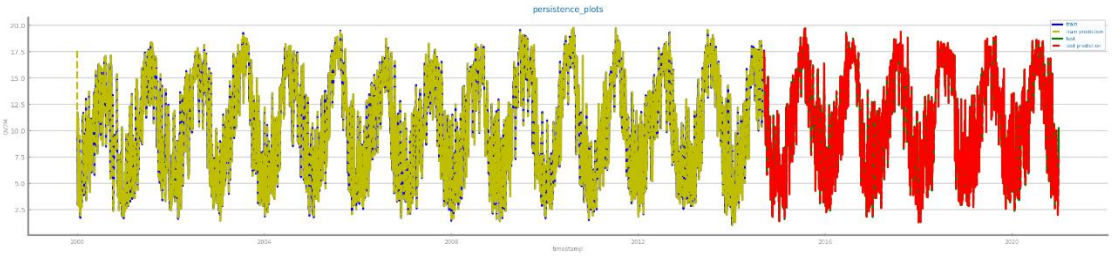



*Figure 94 Forecasting plots after daily aggregations on time series 2*

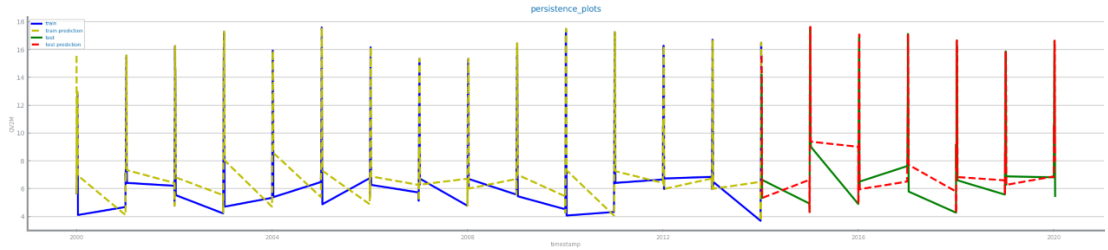



*Figure 95 Forecasting plots after weekly aggregations on time series 2*


*Figure 96 Forecasting plots after monthly aggregations on time series 2*

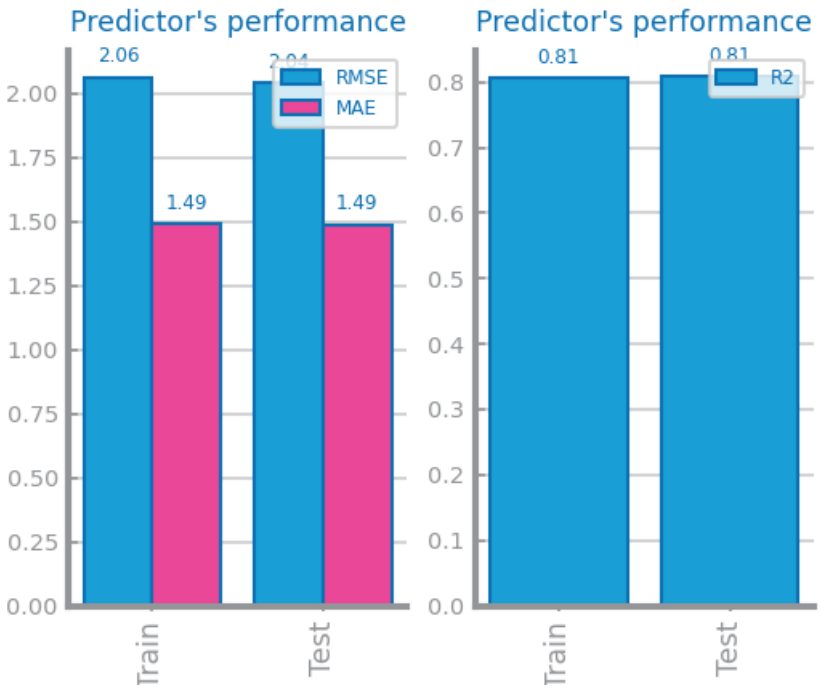



*Figure 97 Forecasting results after daily aggregations on time series 2*

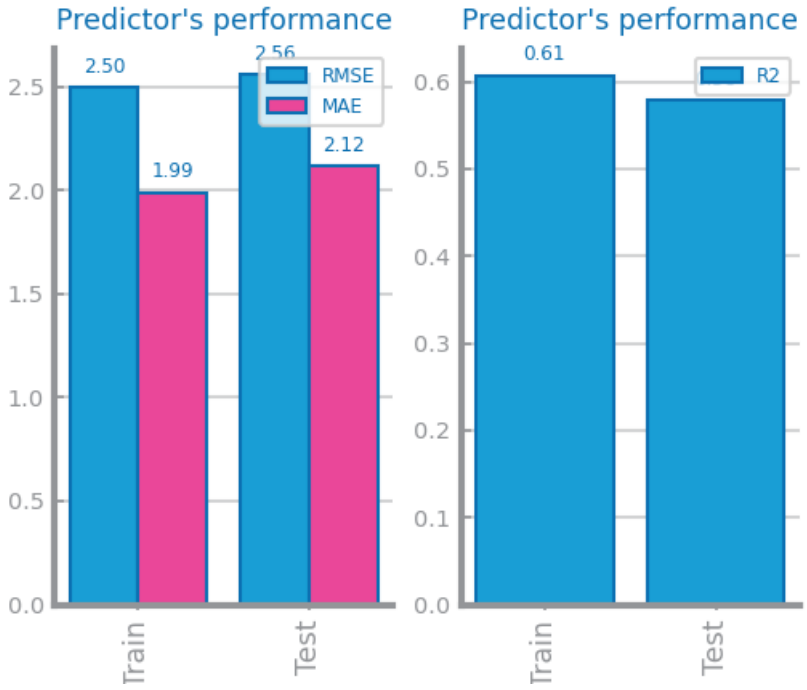



*Figure 98 Forecasting results after weekly aggregations on time series 2*

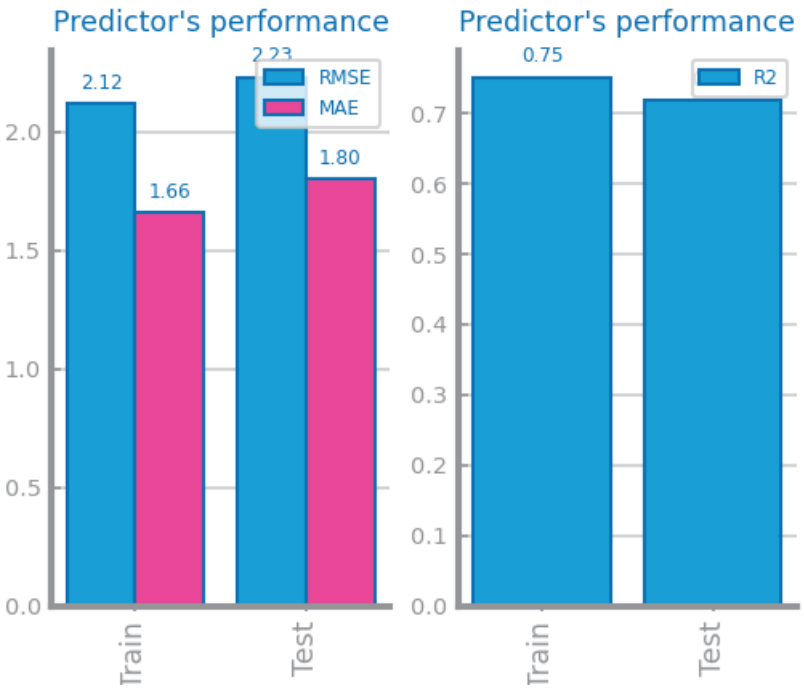



*Figure 99 Forecasting results after monthly aggregations on time series 2*

### *Smoothing*

The two datasets give us much more accuracy when the WIN-SIZE is bigger. For Dataset 1, With 100 WIN-SIZE around the 90% of the model predictions are correct and the variation in the errors is around 6 units for RMSE; and for dataset 2, we have 99% of the prediction model are right with a variation around 1 units.

**Time Series 1 - Results**

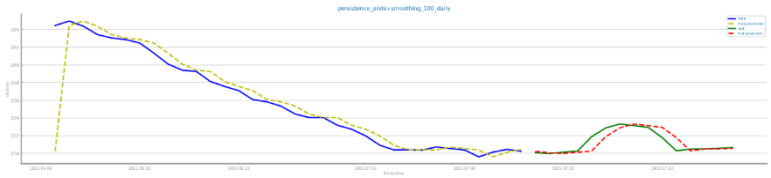


*Figure 100 Forecasting plots after smoothing: WIN_SIZE:10 parameterizations on time series 1*

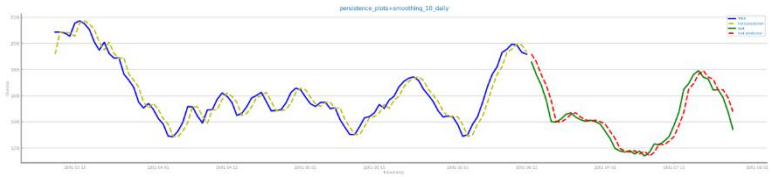


*Figure 101 Forecasting plots after smoothing: WIN_SIZE:100 parameterizations on time series 1*

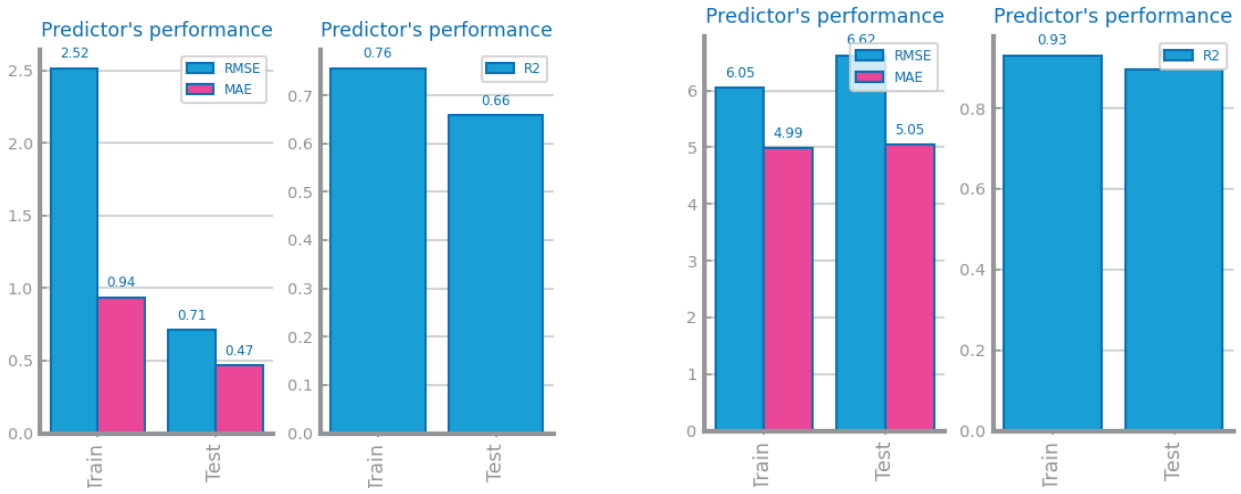



*Figure 102 Forecasting results after different smoothing parameterizations on time series 1 – WIN_SIZE:10 (left), WIN_SIZE:100 (right)*

## Time Series 2 - Results

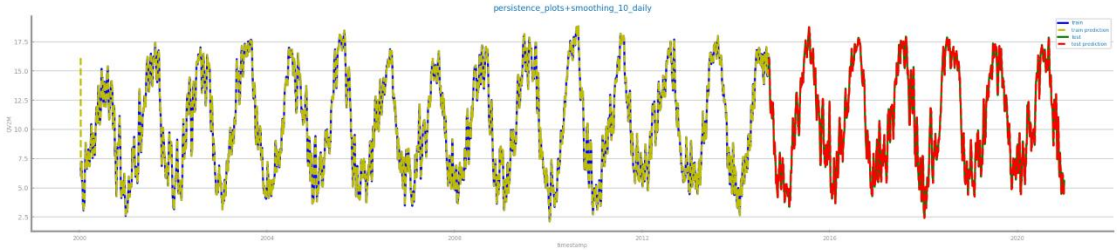



*Figure 103 Forecasting plots after smoothing: WIN_SIZE:10 parameterizations on time series 2*

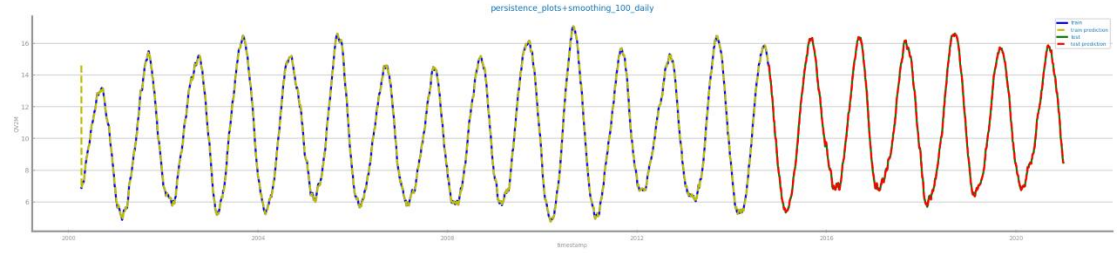



*Figure 104 Forecasting plots after smoothing: WIN_SIZE:100 parameterizations on time series 2*

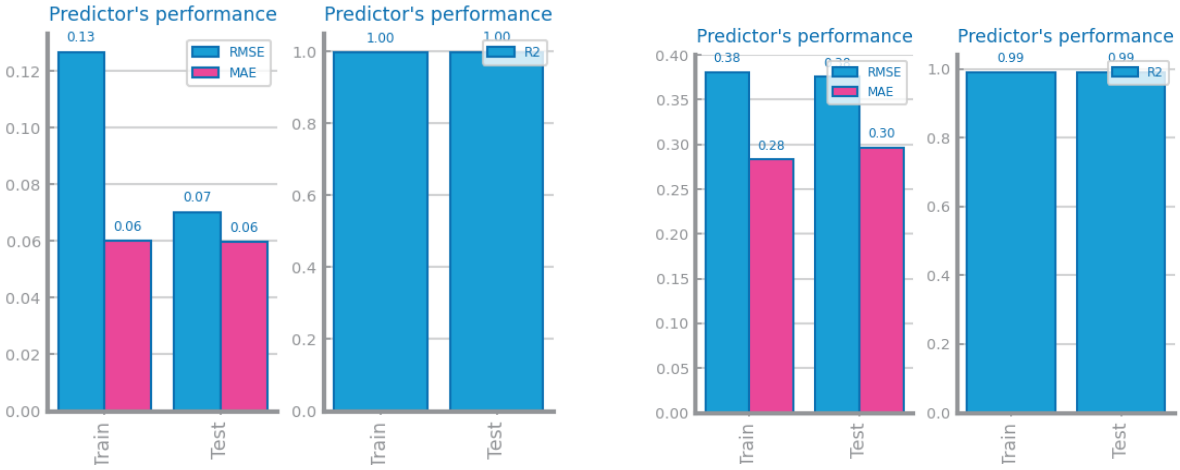



*Figure 105 Forecasting results after different smoothing parameterizations on time series 2 – WIN_SIZE:10 (left), WIN_SIZE:100 (right)*

## *Differentiation*

**TS1 –** In ts1 we identify that the best result was achieved by applying only one differentiation function.

**TS2 –** In the other hand, we identify that in ts2, the best result was achieved by applying two consecutive differentiation.

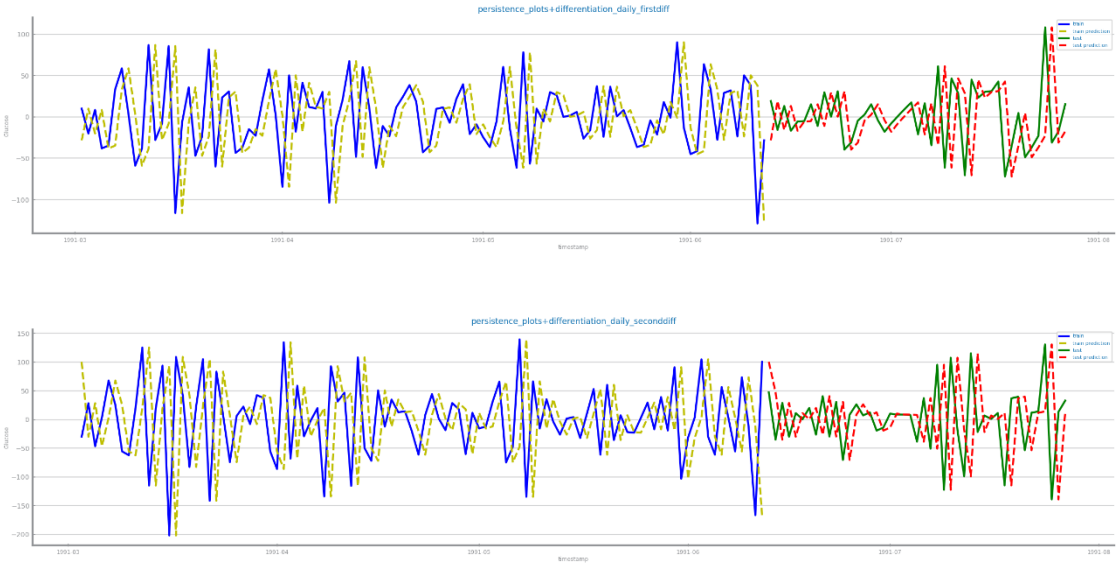



*Figure 106 Forecasting plots after first and second differentiation of time series 1 respectively*

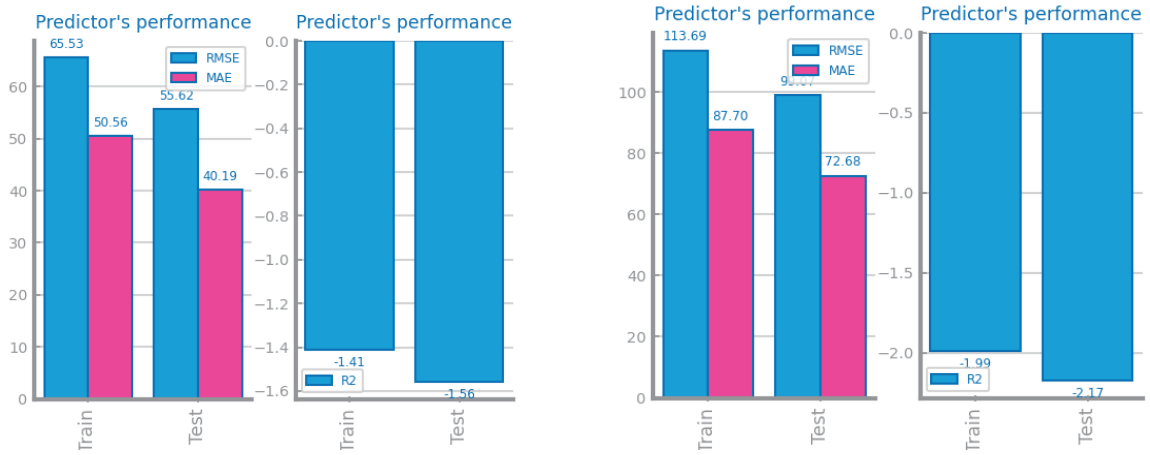



*Figure 107 Forecasting results after first and second differentiation of time series 1 – 1st (left), 2nd(right)*

**Time Series 2 - Results**

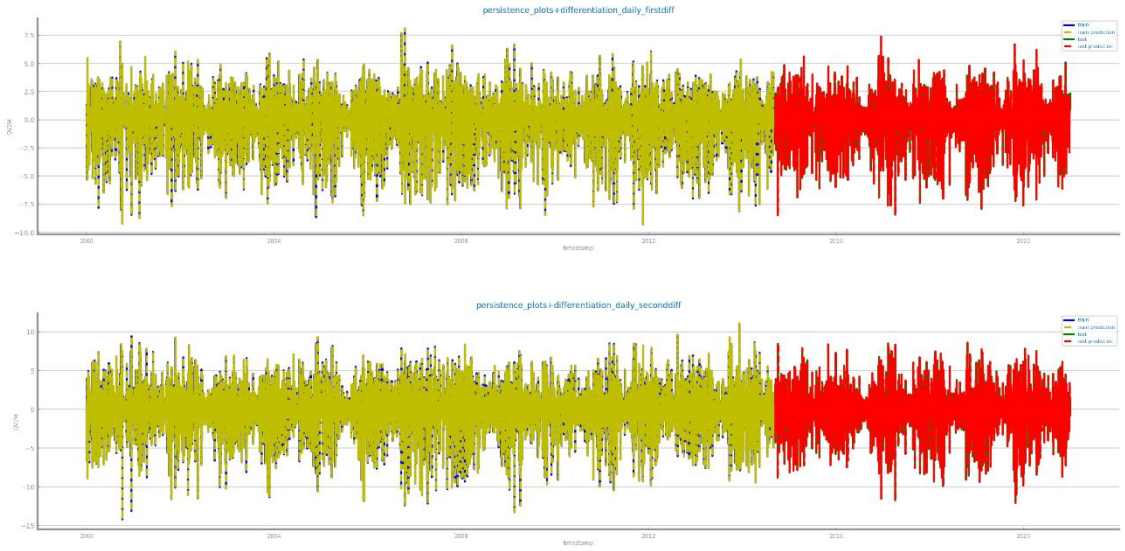



*Figure 108 Forecasting plots after first and second differentiation of time series 2 respectively*

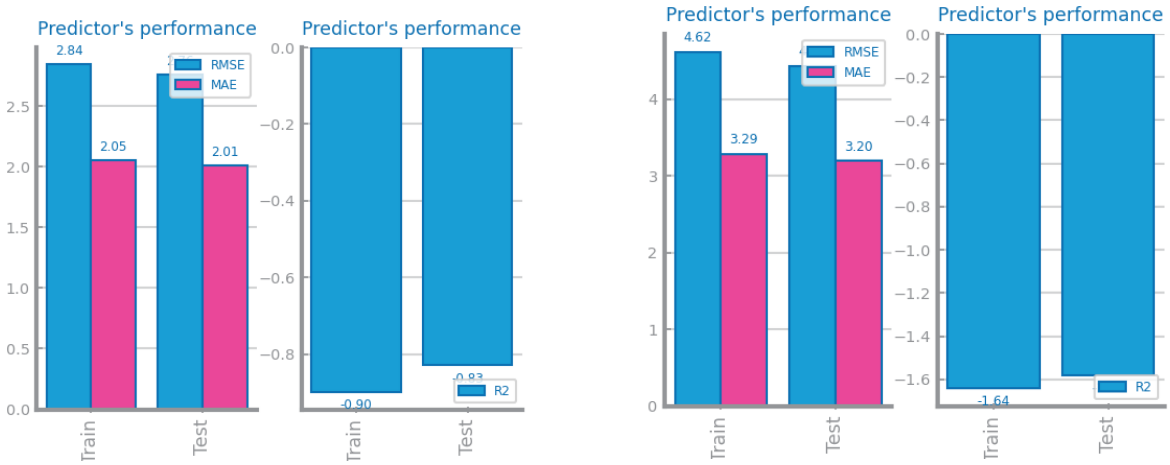



*Figure 109 Forecasting results after first and second differentiation of time series 2 – 1st (left), 2nd(right)*

# 7 MODELS' EVALUATION

For both dataset we are running the evaluation with split dataset 70% train and 30% test. Dataset 1 have low values in predictor model and Dataset 2 have high values in the predictor model.

## *Simple Average Model*

For the simple average we have the right average values for both dataset around 150 for dataset 1 and around 10 for dataset 2 in the plots.

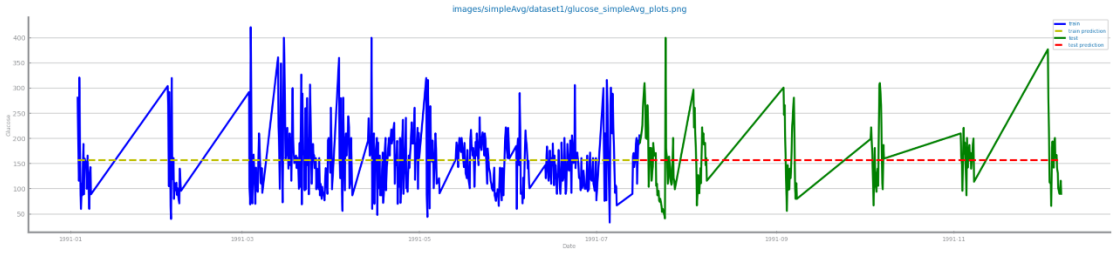



*Figure 110 Forecasting plots obtained with Simple Average model over time series 1*

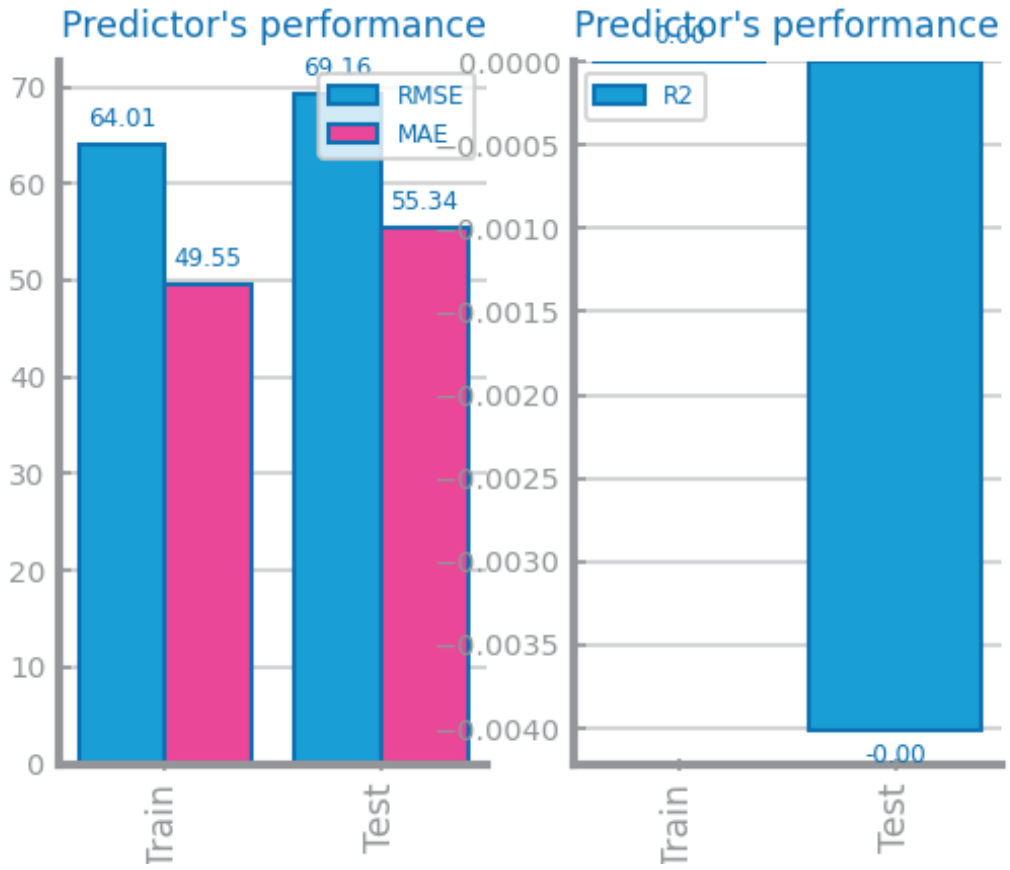



*Figure 111 Forecasting results obtained with Simple Average model over time series 1*

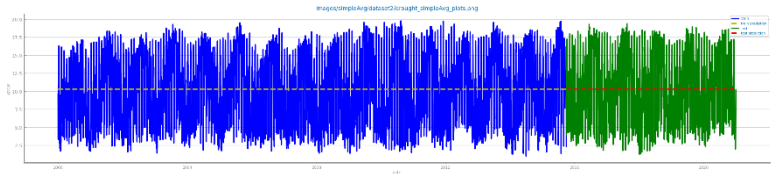


*Figure 112 Forecasting plots obtained with Simple Average model over time series 2*

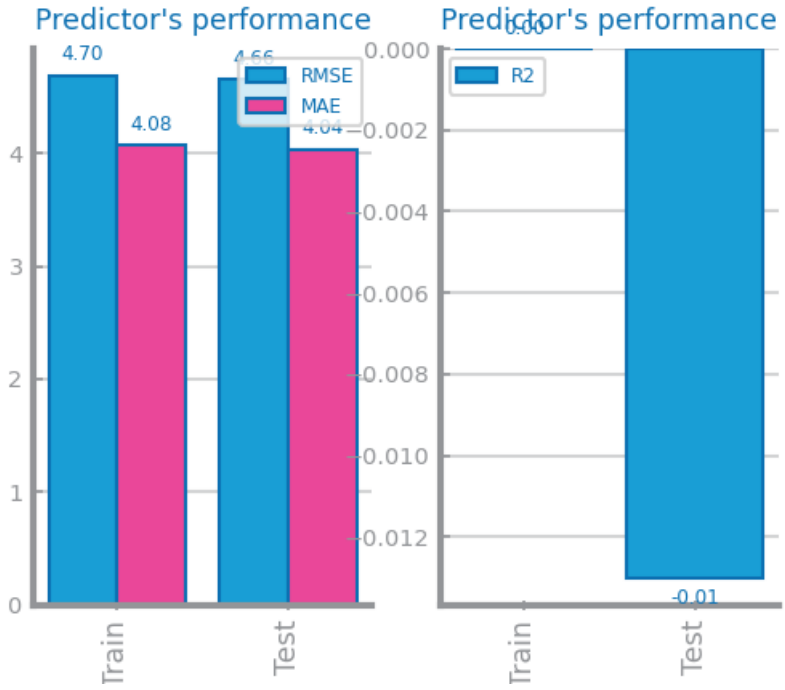



*Figure 113 Forecasting results obtained with Simple Average model over time series 2*

### *Persistence Model*

| Persistence model for dataset2 is better than dataset1. For dataset1, the R2 mesaure has 0.04(train) and 0.13 (test) values which in ideal model would be closer to 1. |
|---|

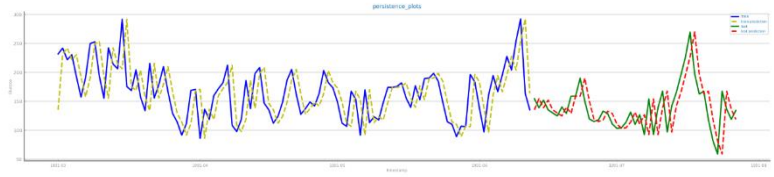


*Figure 109 Forecasting plots obtained with Persistence model over time series 1*

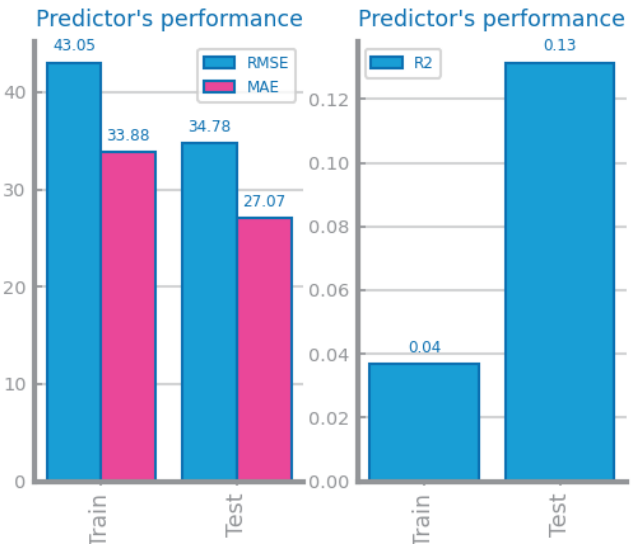



*Figure 110 Forecasting results obtained with Persistence model over time series 1*

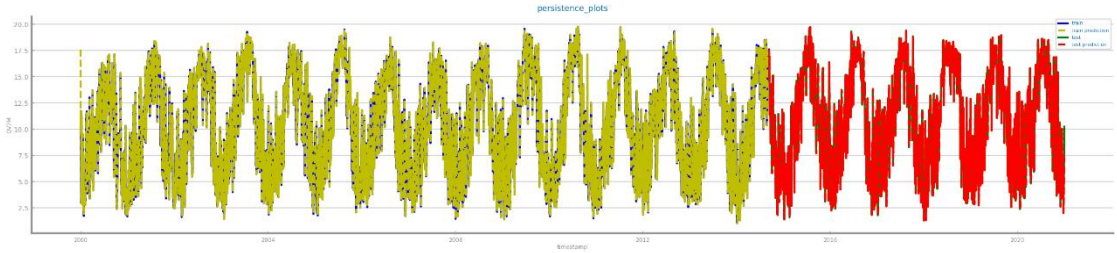



*Figure 114 Forecasting plots obtained with Persistence model over time series 2*

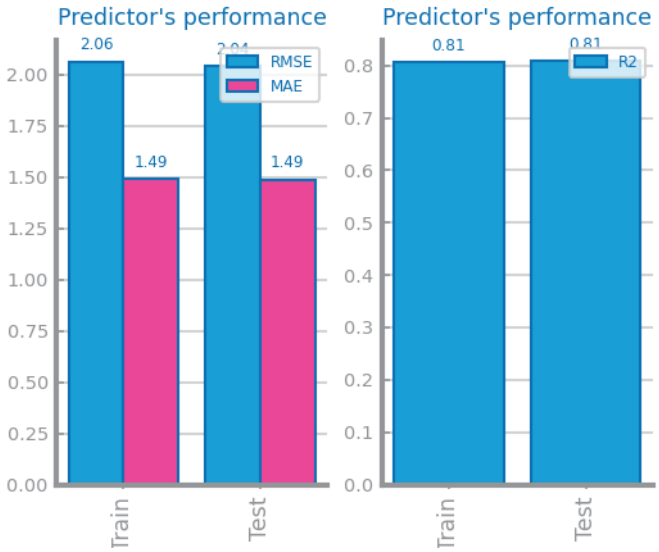



*Figure 115 Forecasting results obtained with Persistence model over time series 2*

### *Rolling Mean Model*

**TS1 –** In ts1 we tested the different window sizes (2,3,4). The best result was achieved with the windowSize = 2.

**TS2 -** In ts2 the same happened as with ts1, better results with windowSize = 2.

We observed that the more we increase the window size the worse the results are. WindowSize = 1 gave us the ideal results but is not considered in our study.

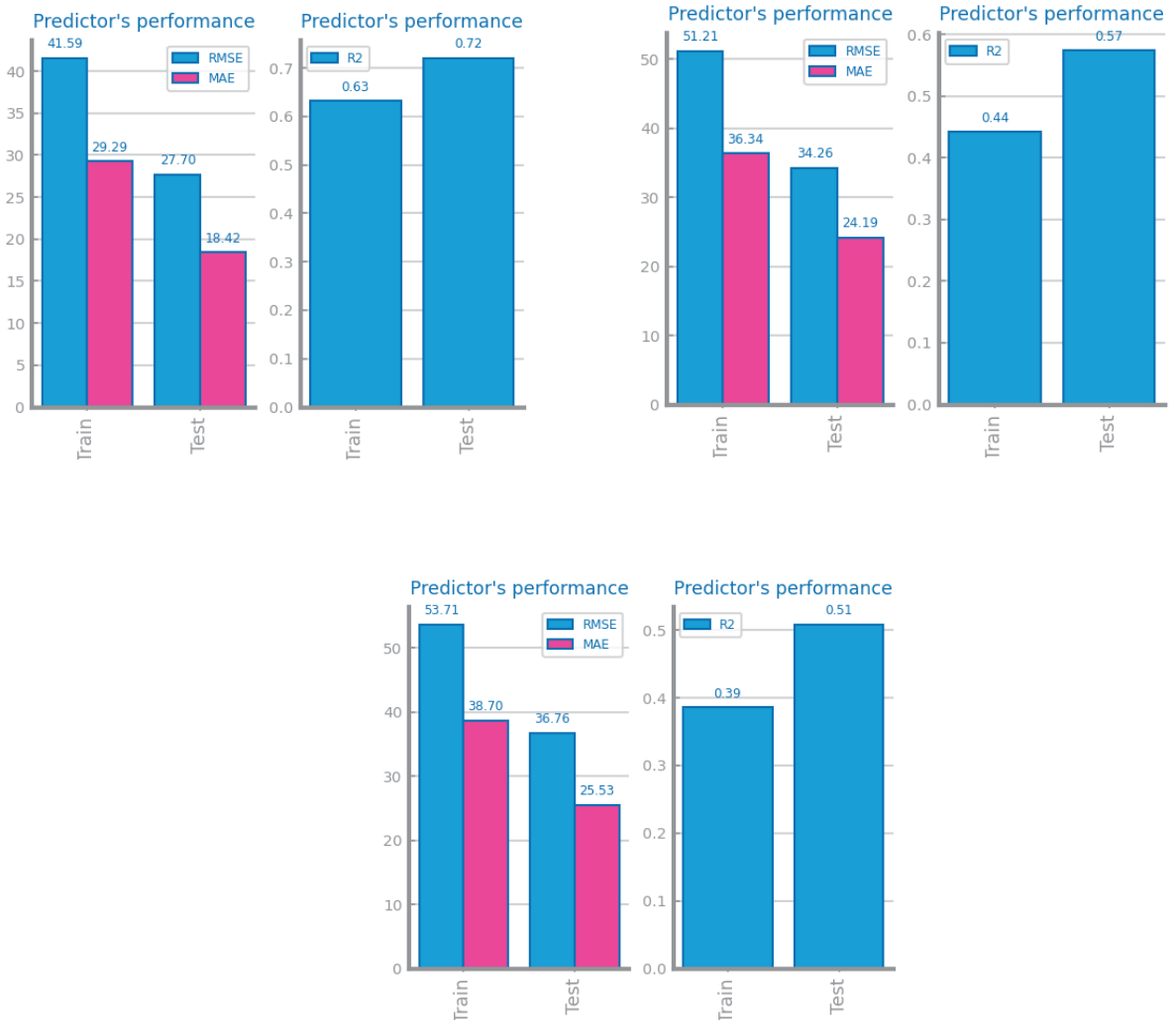



*Figure 116 Forecasting study over different parameterizations of the rolling mean algorithm over time series 1 – WIN_SIZE:2,3,4 respectively*

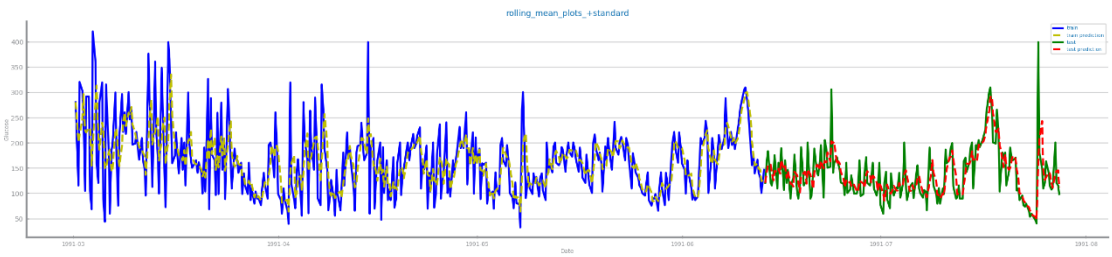


*Figure 117 Forecasting plots obtained with the best parameterization of rolling mean algorithm, over time series 1 - WIN_SIZE:2*

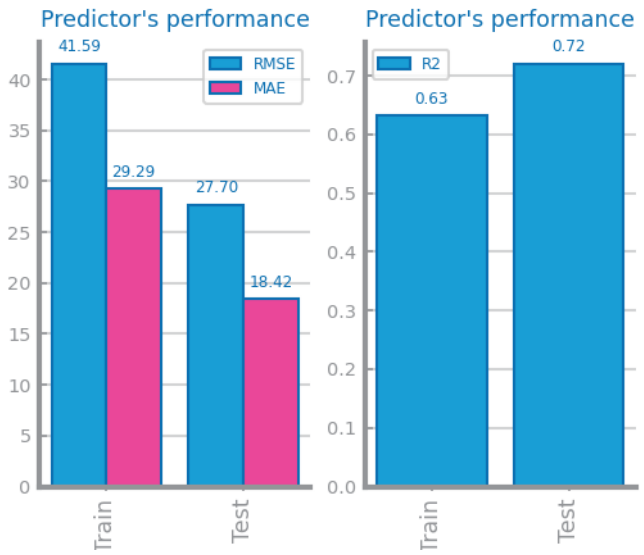



*Figure 118 Forecasting results obtained with the best parameterization of rolling mean algorithm, over time series 1 - WIN_SIZE:2*

**Time Series 2 Results**

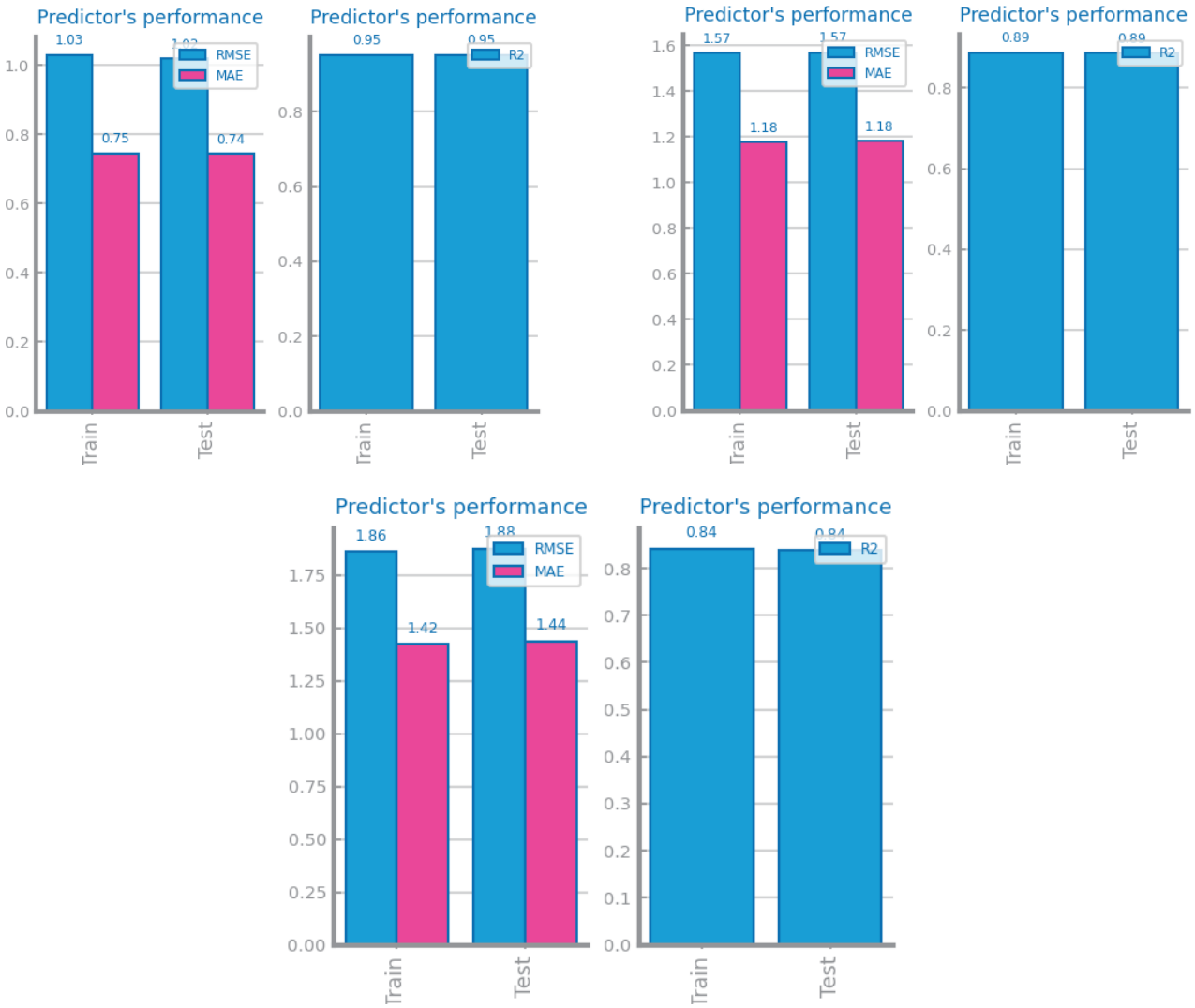



*Figure 119 Forecasting study over different parameterizations of the rolling mean algorithm over time series 2 - WIN_SIZE:2,3,4 respectively*

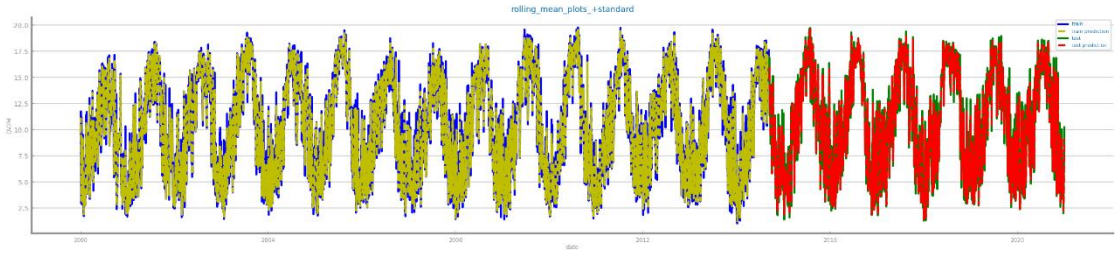



*Figure 120 Forecasting plots obtained with the best parameterization of rolling mean algorithm, over time series 2 - WIN_SIZE:2*

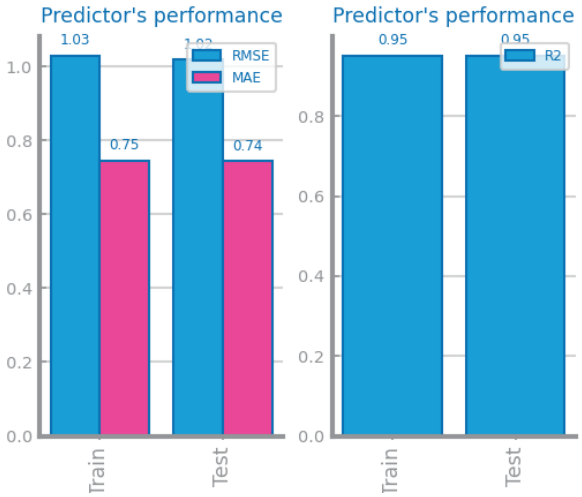



*Figure 121 Forecasting results obtained with the best parameterization of rolling mean algorithm, over time series 2 - WIN_SIZE:2*

## *ARIMA Model*

**TS1 –** We found that the best parameters for ts1 would be (1,1,3). In ts1 by increasing the number of times the raw observations are differenced (increasing d parameter) we got better results than with that parameter = 0. We tested it with these two sets of parameters, (p,d,q): (1,0,3), (1,1,3).

**TS2 –** We found that the best parameters for ts2 would be (1,2,1). In ts2 by incresing the same parameter as we did in ts1, we obtain better results too. We tested it with these to sets of parameters, (p,d,q): (1,0,1), (1,2,1)

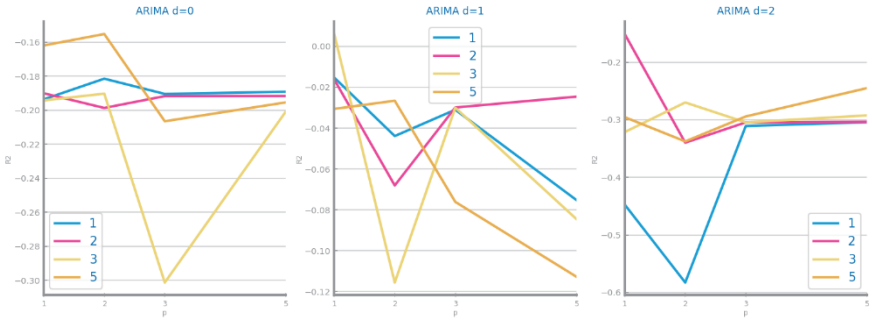



*Figure 122 Forecasting study over different parameterizations of the ARIMA algorithm over time series 1*

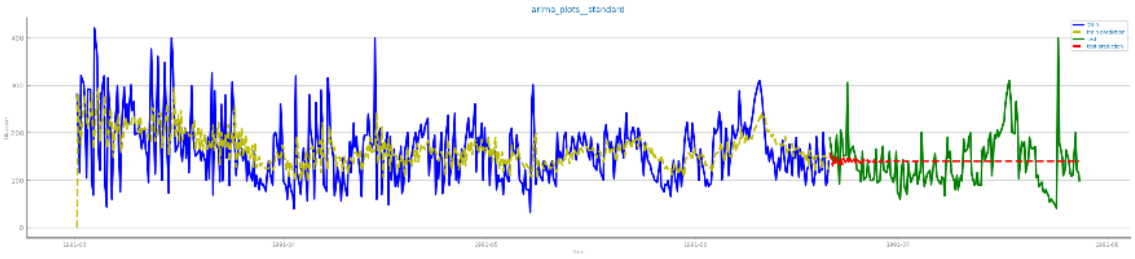


*Figure 123 Forecasting plots obtained with the best parameterization of ARIMA algorithm, over time series 1*

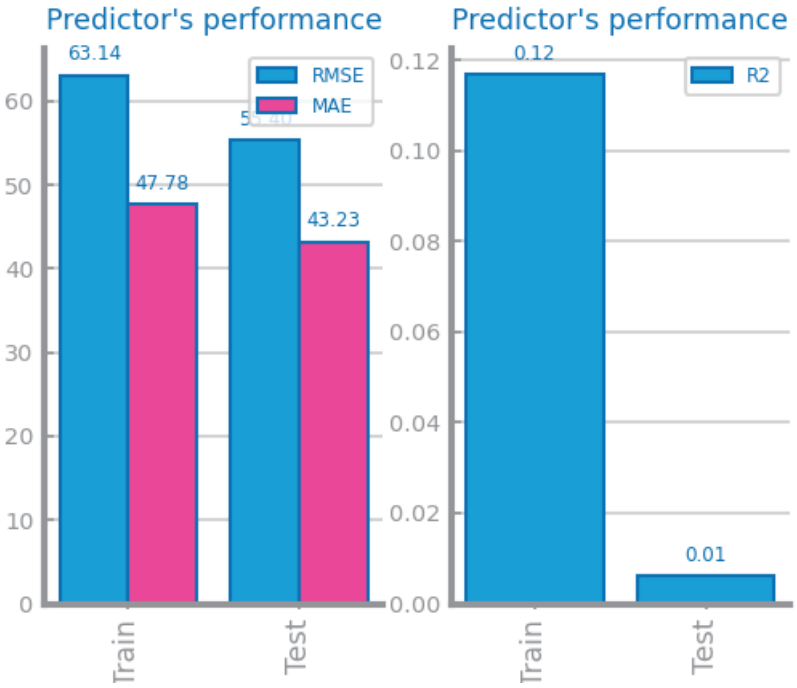



*Figure 124 Forecasting results obtained with the best parameterization of ARIMA algorithm, over time series 1*

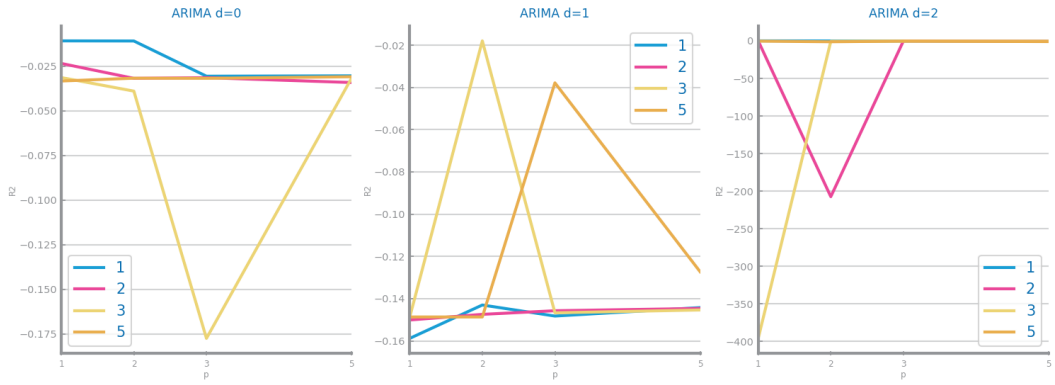



*Figure 125 Forecasting study over different parameterizations of the ARIMA algorithm over time series 2*

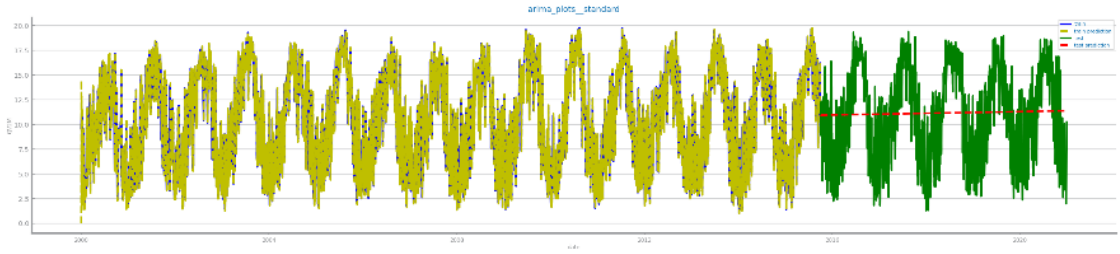


*Figure 126 Forecasting plots obtained with the best parameterization of ARIMA algorithm, over time series 2*

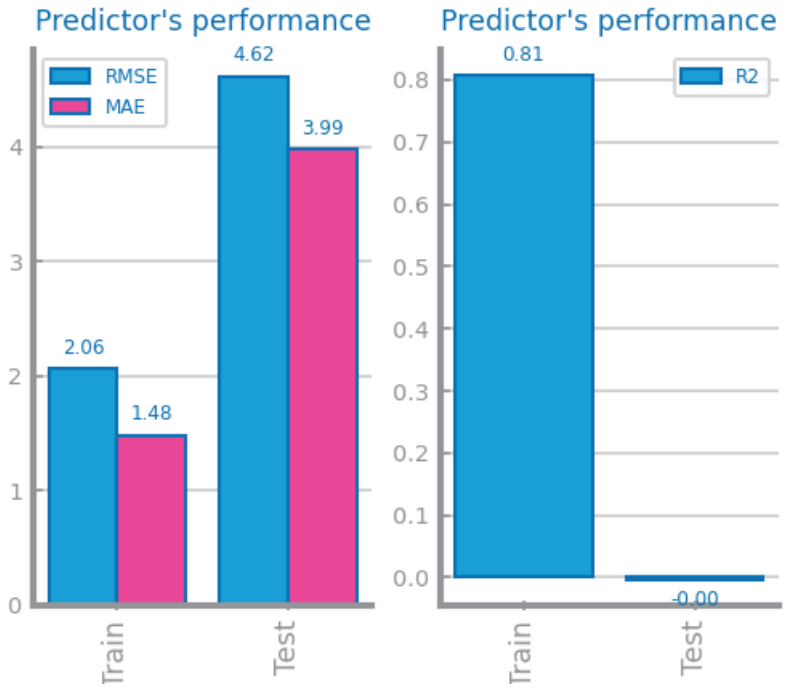



*Figure 127 Forecasting results obtained with the best parameterization of ARIMA algorithm, over time series 2*

## *LSTMs Model*

**Dataset1:** Best results with seq length=4 hidden=8 episodes=500 ==> measure=0.08
**Dataset2:** Best results with seq length=100 hidden=16 episodes=5000 ==> measure=0.65

In order to simplify the study we fixed the optimizer and loss metric to be Adam and MSE respectively.

**TimeSeries 1 - Results**

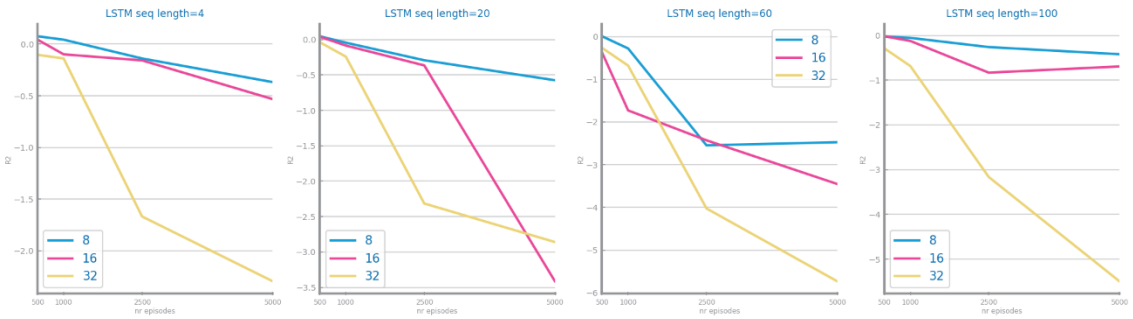



*Figure 128 Forecasting study over different parameterizations of LSTMs over time series 1*

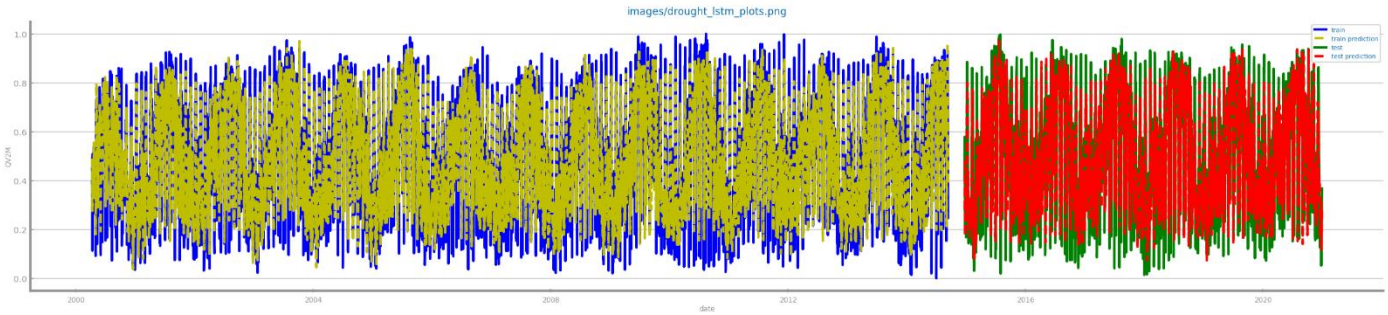



*Figure 129 Forecasting plots obtained with the best parameterization of LSTMs, over time series 1*

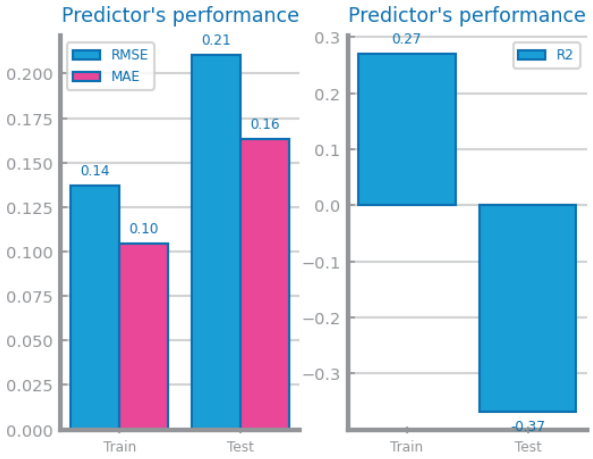



*Figure 130 Forecasting results obtained with the best parameterization of LSTMs, over time series 1*

## TimeSeries 2 - Results

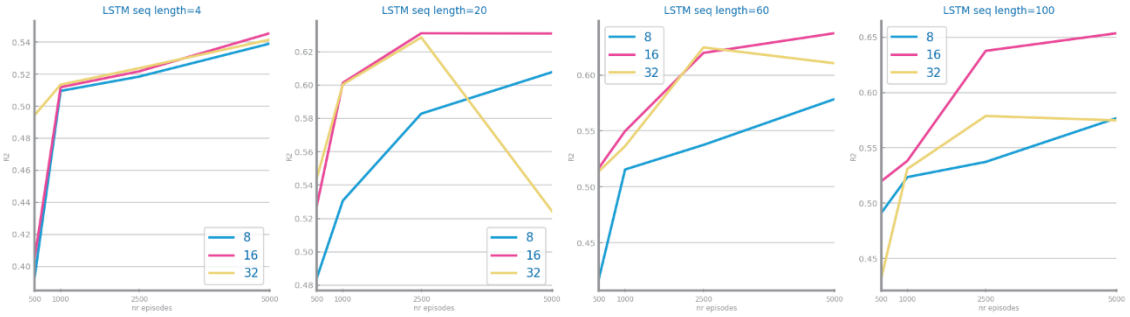



*Figure 131 Forecasting study over different parameterizations of the LSTMs over time series 2*

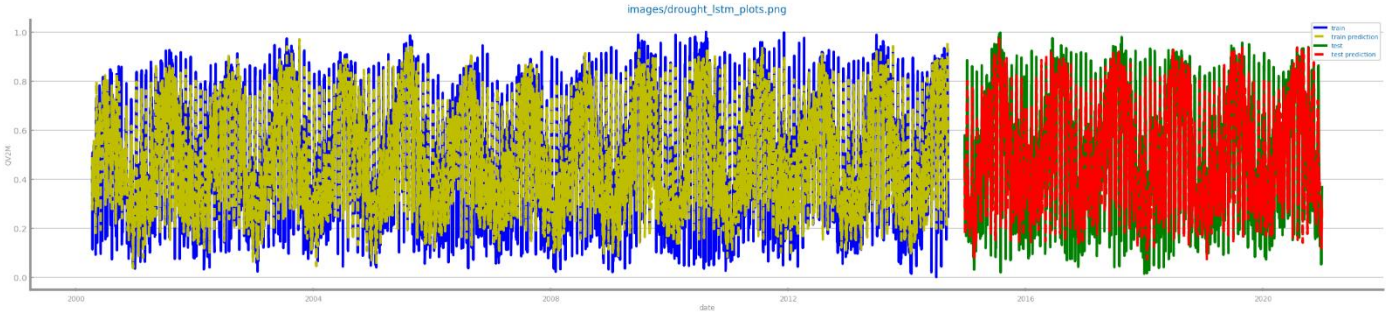



*Figure 132 Forecasting plots obtained with the best parameterization of LSTMs, over time series 2*

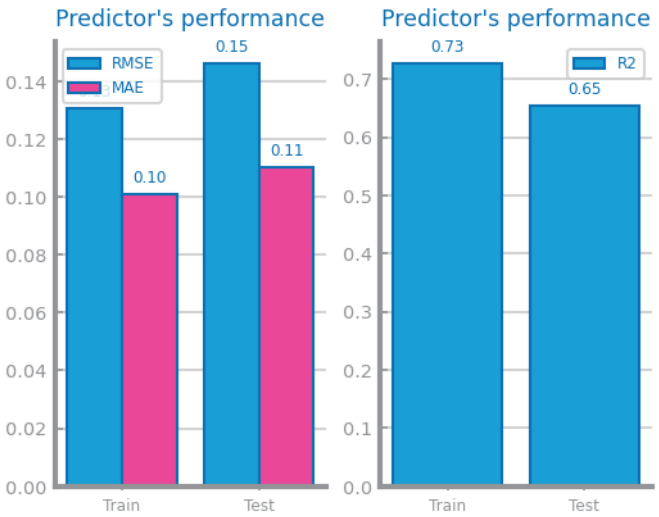



*Figure 133 Forecasting results obtained with the best parameterization of LSTMs, over time series 2*

# 8 CRITICAL ANALYSIS

In forecasting we have a low prediction in both models, the predictor performance RMSE and MAE gives high values in most of the predictions with R-squared (R2) value far from 100% giving a much higher error.

Based on the results we can say that the best predictor for dataset 2 with daily granularity is LSTM since it develops better results with a low error, where RMSE corresponds to 0.14 and RME with value 0.10 both for the training dataset, and with an R2 close to 0.73 which is closer to 100%.

Although the predictor for dataset 1 is not so good, we believe that Rolling Mean is the predictor that gives us the best results, even though we are aware that it is not the best and the one we would have wished for. We believe that the low predictability is due to an error in the initial stages of the process or maybe we selected the wrong granularity, for dataset 1 we noticed in the graphs that we have a lag which may cause that our prediction is not correctly aligned.