TÉCNICO LISBOA

DS 2023

Data Science Project

Team nr: 11	**Student 1 :** Guilherme Pascoal	**IST nr:** 99079
	Student 2 : Inês Garcia	**IST nr:** 99083
	Student 3 : José Cutileiro	**IST nr:** 99097
	Student 4 : Miguel Vale	**IST nr:** 99133

CLASSIFICATION

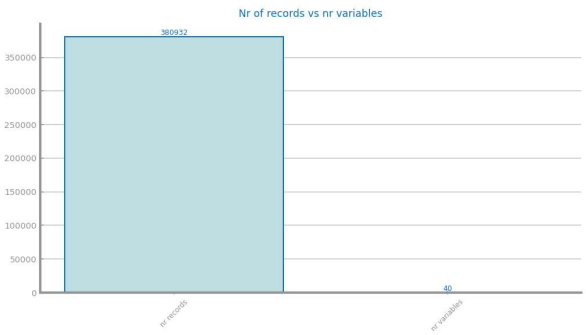
1 DATA PROFILING

Telephone surveys as collection methods of historical health-related fields hint at missing values in COVID dataset. Customers' financial disparities cause income-related outliers in CREDIT dataset.

Data Dimensionality

The first set is (100000, 28), and the second is (380932, 40). No Curse of Dimensionality issues as we have ample records. The dataset isn't sparse, benefiting distance-based algorithms. Yet, high dimensionality challenges efficiency and interpretation. Individuals may find it challenging to recall specific details about tetanus shots received over the past 10 years, having the most missing values. In the credit dataset, numeric typing is common as money is measured on a numerical scale.

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



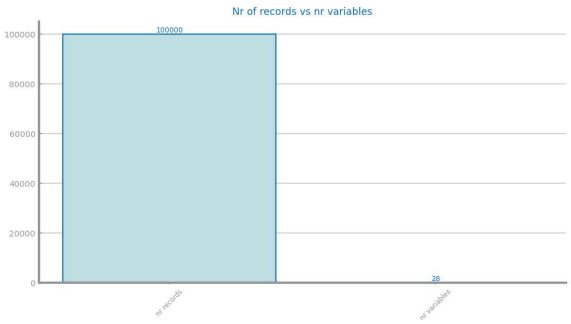
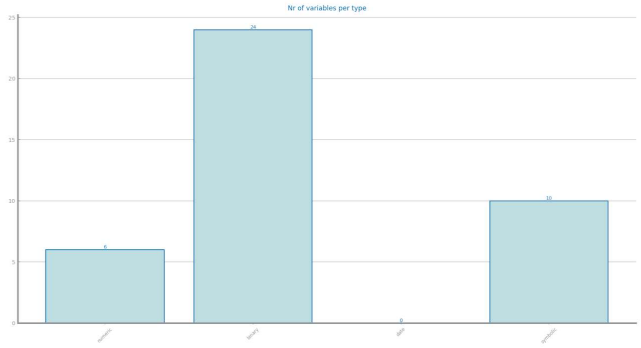


Figure 2 Nr variables per type for dataset 1 (left) and dataset 2 (right)



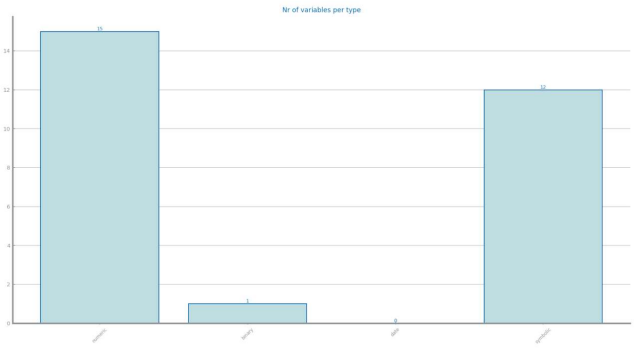
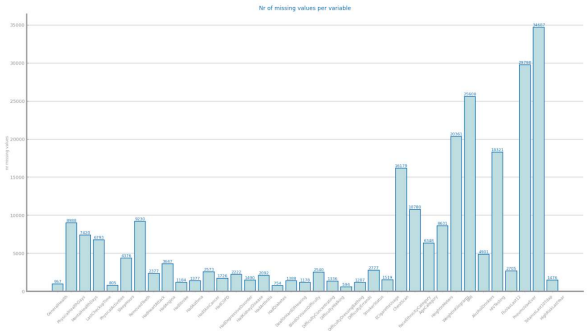
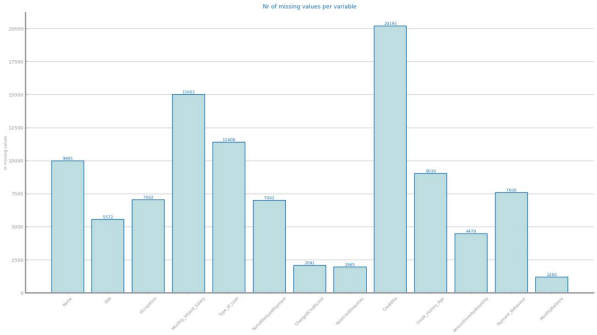


Figure 3 Nr missing values for dataset 1 (left) and dataset 2 (right)





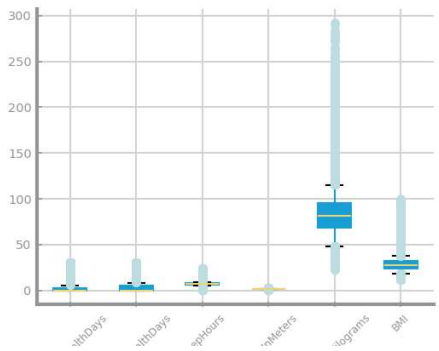
Data Distribution

Global boxplots are inconclusive due to variable scale differences; normalization is necessary to make global comparisons.

The biometric variables demonstrated a more uniform and clustered distribution. On the other hand, the variables related to well-being exhibited a dispersed pattern, suggesting wider variation in values across the dataset.

Because the datasets have such a difference between one values' frequency to the other, we might consider techniques such as resampling to balance them out.

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



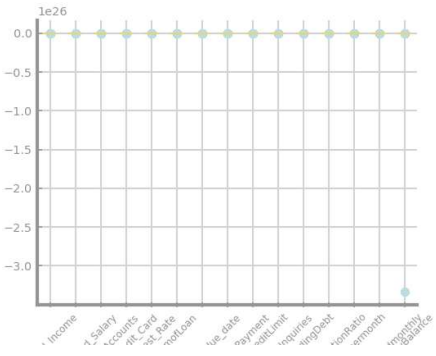


Figure 5 Single variable boxplots for dataset 1

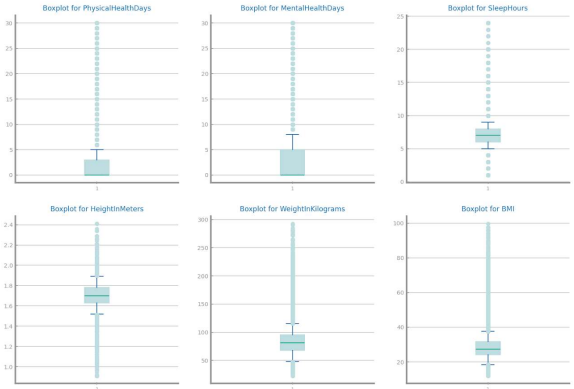


Figure 6 Single variable boxplots for dataset 2

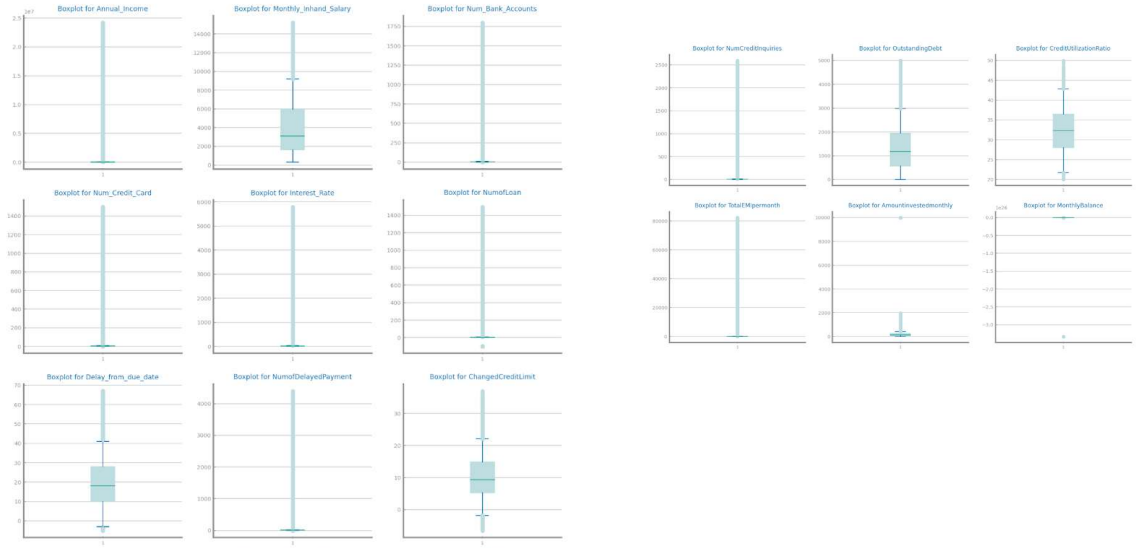


Figure 7 Histograms for dataset 1 (with distributions is enough)

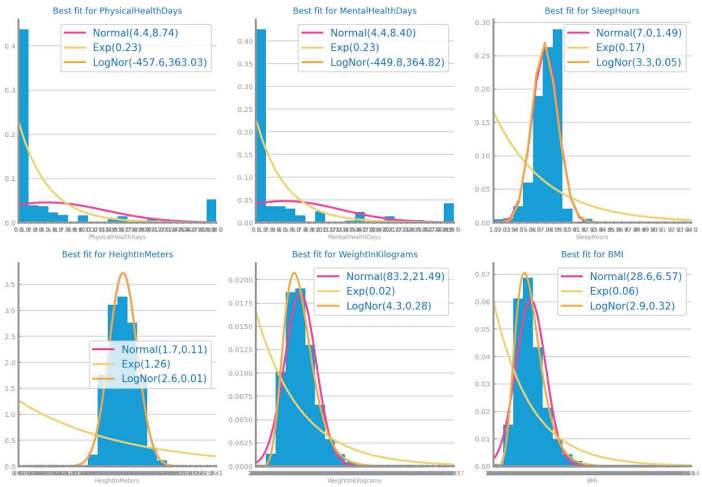


Figure 8 Histograms for dataset 2

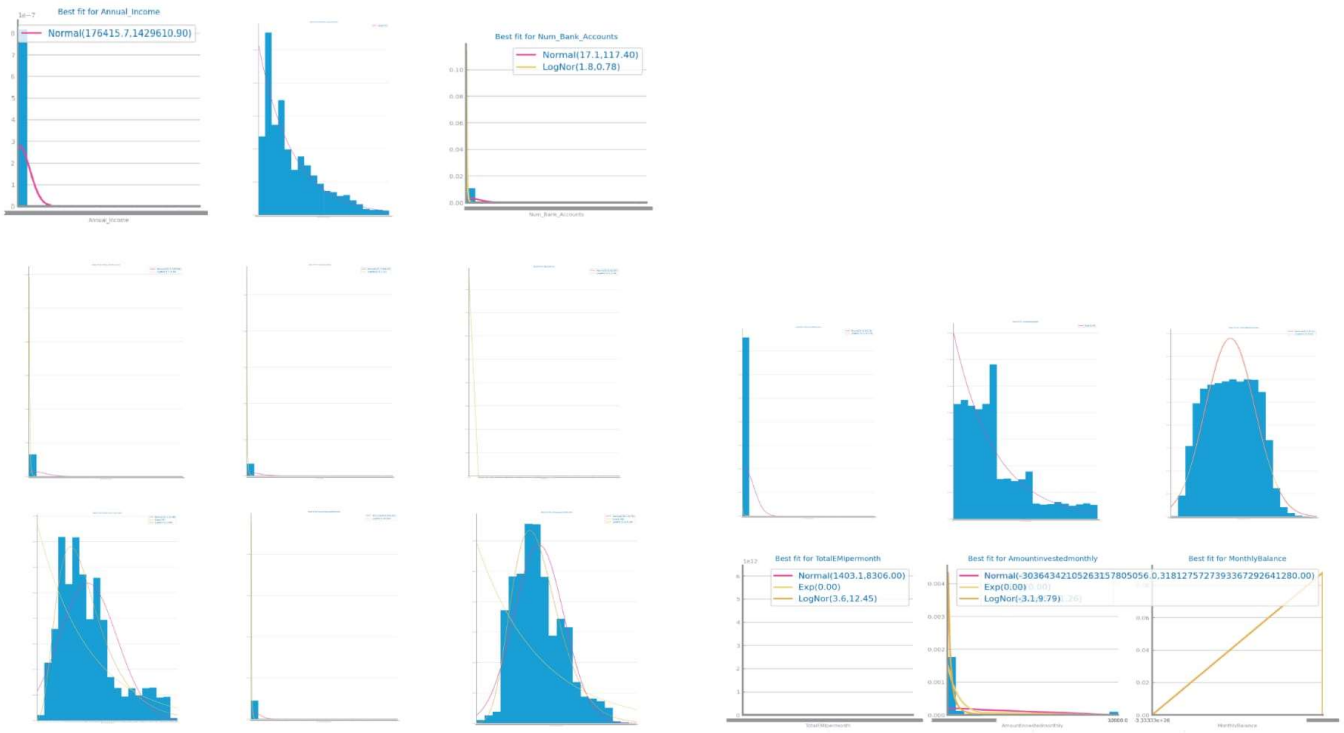


Figure 9 Outliers study dataset 1

Figure 10 Outliers study for dataset 2

Figure 11 Class distribution for dataset 1

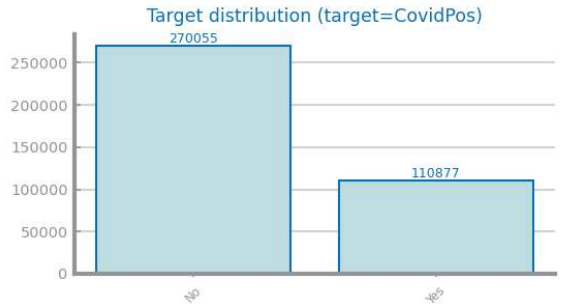
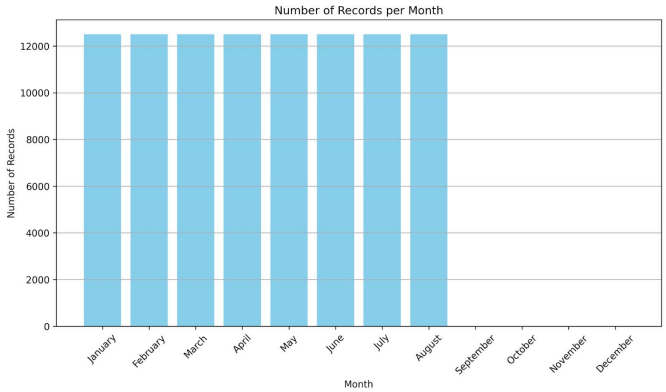


Figure 12 Class distribution for dataset 2

Data Granularity

Understanding data granularity is vital for informed decisions, exposing regional disparities. Analyzing states individually reveals uniqueness in population, economics, and policies. Regional scrutiny unveils commonalities and disparities among neighbors. Granularity in months captures fluctuations, while quarterly views aid long-term comparisons. This detailed data is crucial for managing public health crises, enabling targeted actions and effective strategies to mitigate epidemic impacts.

Figure 13 Granularity analysis for dataset 1



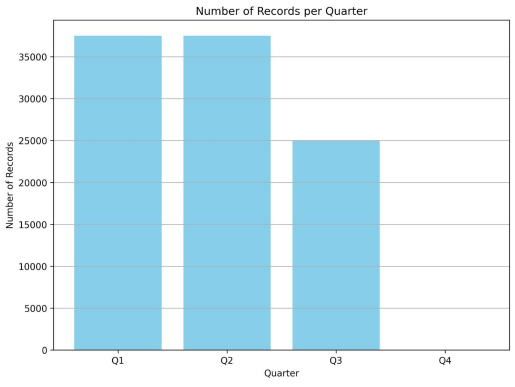
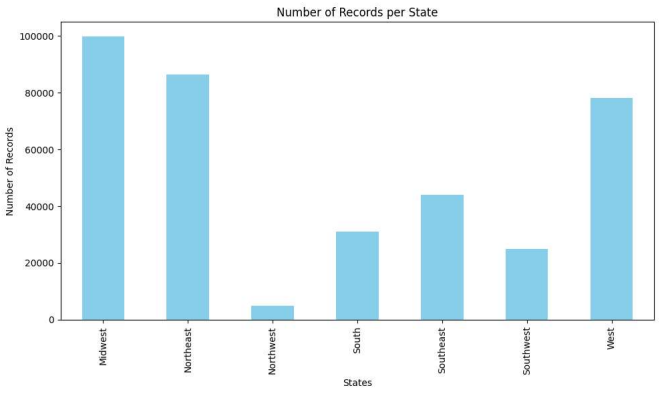
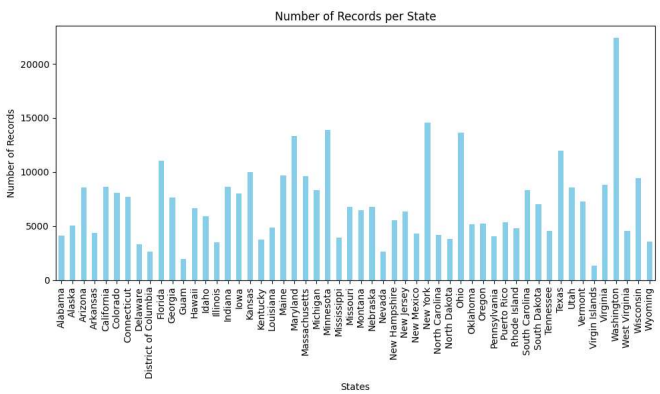


Figure 14 Granularity analysis for dataset 2





Data Sparsity

The covid dataset may exhibit data sparsity in certain variables because of missing information in pre-existing conditions and vaccination status. Similarly, some credit attributes might be less frequently reported or collected in the credit-related dataset, contributing to sparsity.
A correlation between weight and BMI was observed as expected, given the BMI formula's reliance on weight.
A notable linear relation was also verified in annual_income and the rest of the fields.

Figure 15 Sparsity analysis for dataset 1

Figure 16 Sparsity analysis for dataset 2

(This graphic was corrupted)

Figure 17 Correlation analysis for dataset 1

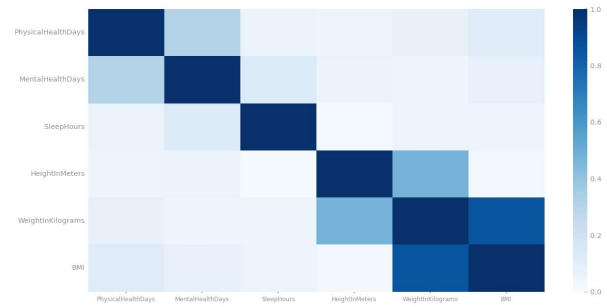
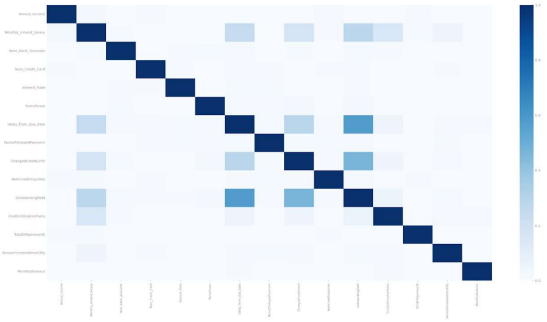


Figure 18 Correlation analysis for dataset 2



2 Data Preparation

Variables Encoding

Credit score (variable encoding):

Original data set: Can't be used because we have nominal variables, this is incompatible with future algorithms.

Drop variables: ID, Customer_id, SSN

Cyclic encoding: Months (used a harmonic function)

Ordinal linear encoding: CreditMix, Credit_History_age, Payment_Behaviour, Binary variables

Dummification: Occupation

Taxonomy: Type_of_Loan

Pos covid (variable encoding):

Original data set: Can't be used because we have nominal variables, this is incompatible with future algorithms.

Ordinal linear encoding: Binary variables, GeneralHealth, LastCheckUpTime, AgeCategory

Dummification: SmokerStatus, ECigaretteUsage, RaceEthnicityCategory

Other: TetanusLast10Tdap: In this encoding system, we use a dictionary where each response is associated with a unique integer. However, if the person does not know the type of vaccine they received, we assign 'None' to represent this uncertainty.

Other: State: We choose to use the population density by state.

Missing Value Imputation

The first strategy we used involved filling **missing values with zeros**, while the second strategy focused on imputing **missing values with either the mean** (for numeric variables) **or the mod**e (for binary or symbolic variables). Upon implementation yielded comparable outcomes when applied to Naive Bayes and KNN models. In our final processing steps, we opted to proceed with the mean/mode imputation approach, as it helps maintain the distribution of the original data.

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

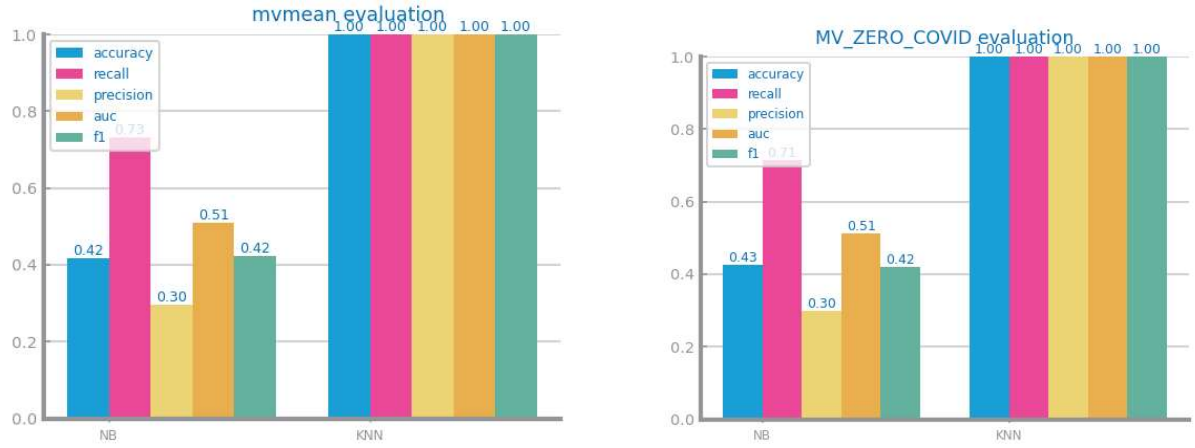
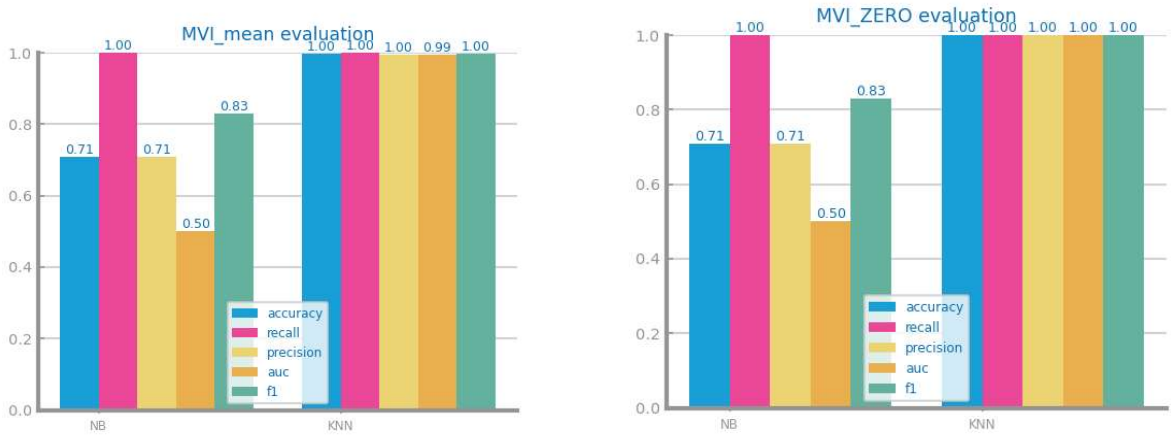


Figure 20 Missing values imputation results with different approaches for dataset 2



Outliers Treatment

Opting to remove outliers enhances performance, yet sacrifices significant data. Replacing outliers doesn't boost performance, in fact, worsens it. Consequently, we refrain from treating outliers to preserve data integrity.

Note:

No Outliers: remove the record if it is detected to be an outlier (with a threshold in order to don't remove a big percentage of the original data)

Replace outliers: Replace the values we consider an outlier with the mean value (we tried with another values but the mean presented best results)

Figure 21 Outliers imputation results with different approaches for dataset 1

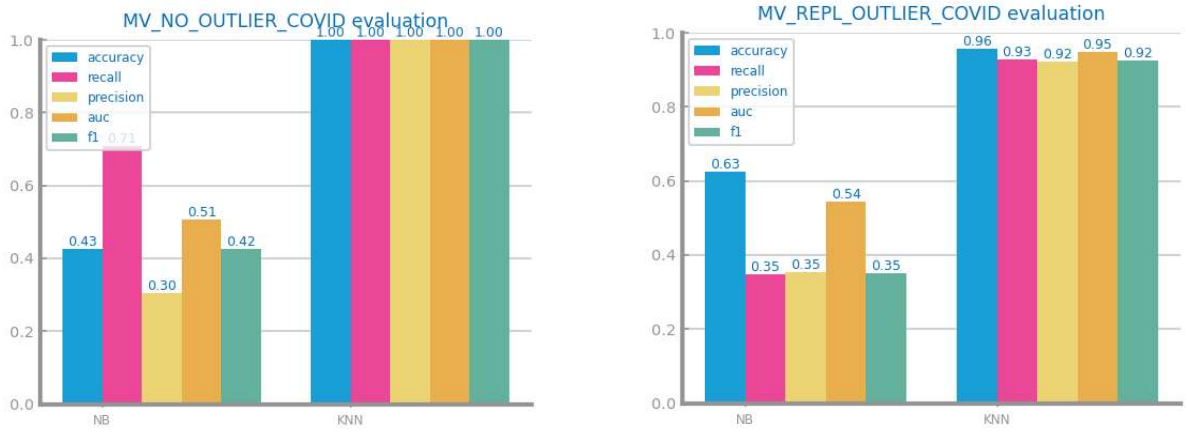
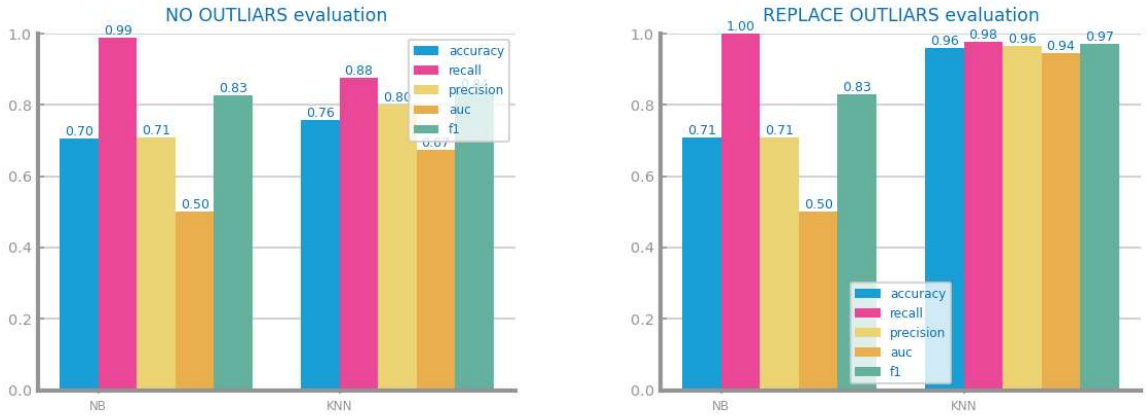


Figure 22 Outliers imputation results with different approaches for dataset 2



Scaling

Minmax scaling disrupts KNN and Z-score, too. So we opted for original unscaled data due to their impact on KNN evaluation in both datasets.

Figure 23 Scaling results with different approaches for dataset 1

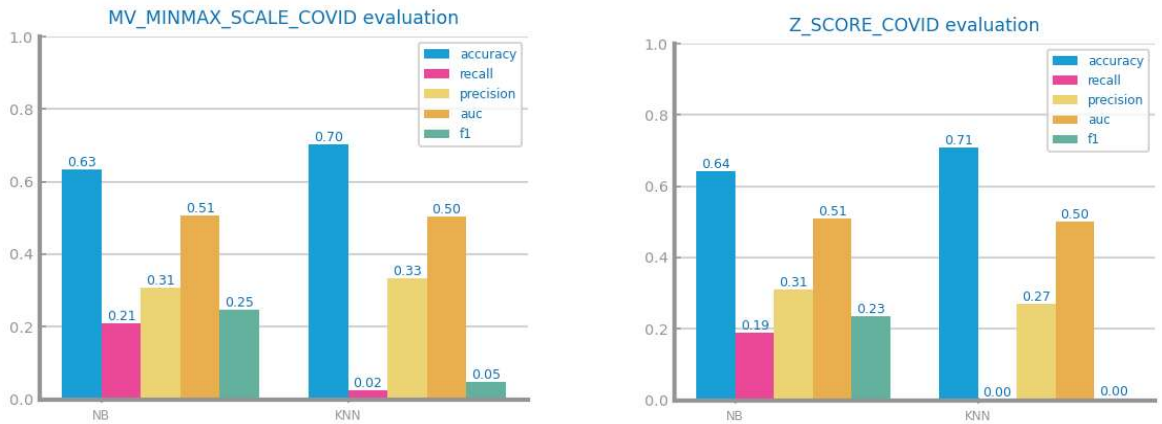
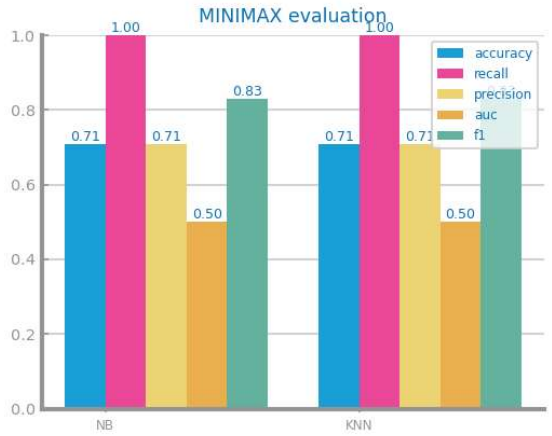
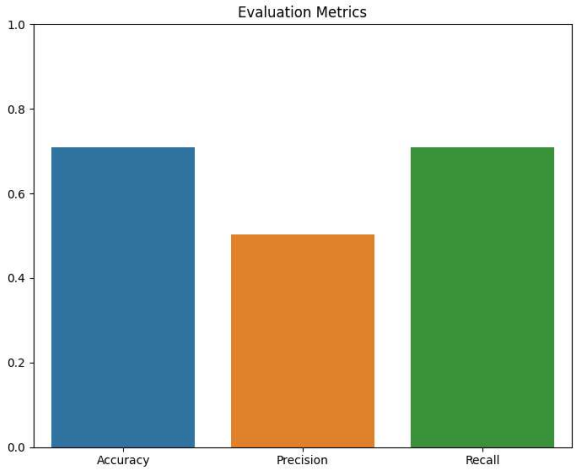


Figure 24 Scaling results with different approaches for dataset 2 (min-max image 1 and z-score KNN image 2)

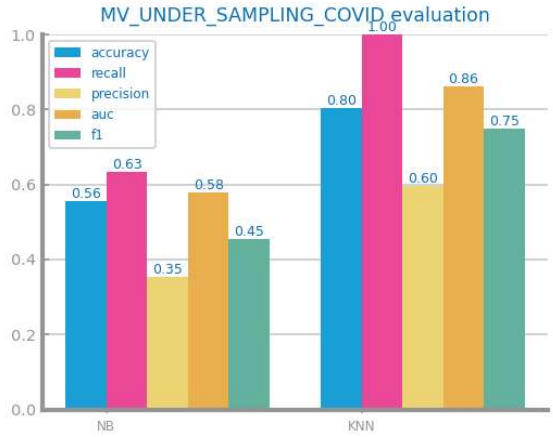


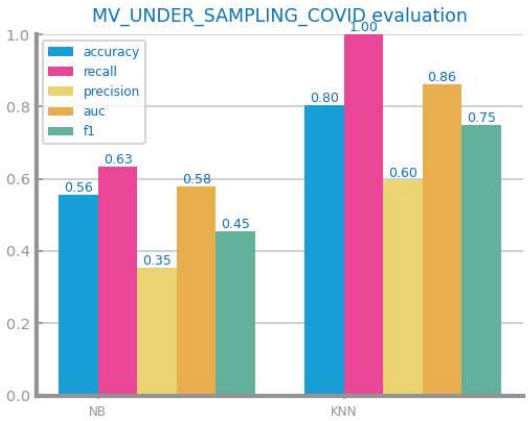


Balancing

After experimenting with under sampling, over sampling, and SMOTE techniques, we observed certain metrics showing improvement. These methods tended to deteriorate the overall performance of our models. However, considering the holistic performance metrics they all exhibited declines post-balancing. Consequently, we've decided to continue with our original, unbalanced dataset to maintain the overall model integrity and performance.

Figure 25 Balancing results with different approaches for dataset 1





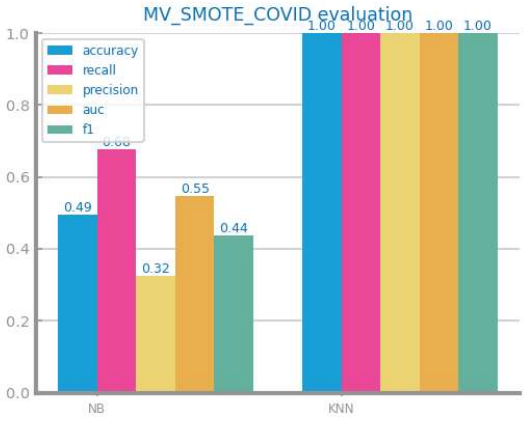
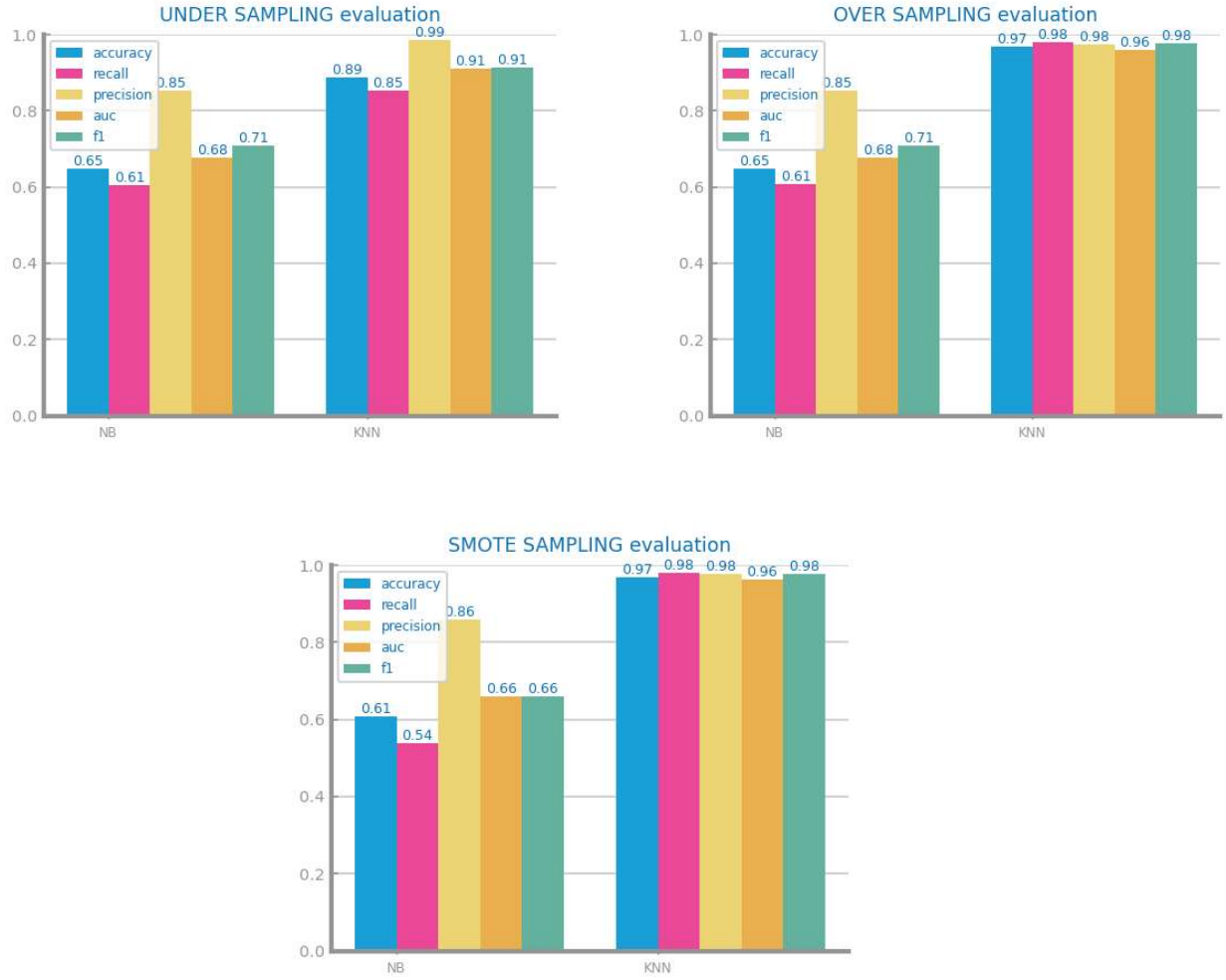


Figure 26 Balancing results with different approaches for dataset 2



Feature Selection

The decision to drop certain features, specifically the ID, customer ID, and SSN (Social Security Number), was a strategic step taken in consideration of the algorithms utilized for generalization purposes within the dataset. Knowing precisely the individual behind certain characteristics would not only fail to add value but could also worsen the results and affect the algorithms' performance. Hence, we made the decision to overlook individual characteristics that might uniquely identify the data.

Feature Extraction (optional)

This was done to standardize the values of these variables, thereby enhancing the numerical consistency to improve the performance of the algorithms slated for subsequent application.

Additional Feature Generation (optional)

We chose to refrain from feature generation to ensure result integrity and transparency.

3 Models' Evaluation

In our model assessment, we partitioned our dataset into 70% for training and 30% for testing. This deliberate split aims to mitigate overfitting tendencies in certain algorithms, while emphasizing the importance of generalization in algorithms, allowing them to apply learned insights to scenarios beyond the immediate scope of our dataset. With these algorithms, we'll be able to assess the strength of our data representation.

Naïve Bayes

Covid: Gaussian Naive Bayes assumes that continuous features follow a normal distribution. Bernoulli Naive Bayes is typically more effective with sparse datasets. Since our dataset consists of continuous features, Gaussian Naive Bayes is the preferred choice.

Credit score: Gaussian Naive Bayes assumes a normal distribution of continuous features. Bernoulli NB tends to perform well with sparse datasets, where most features are zeros which is the case.

Figure 33 Naïve Bayes alternatives comparison for dataset 1

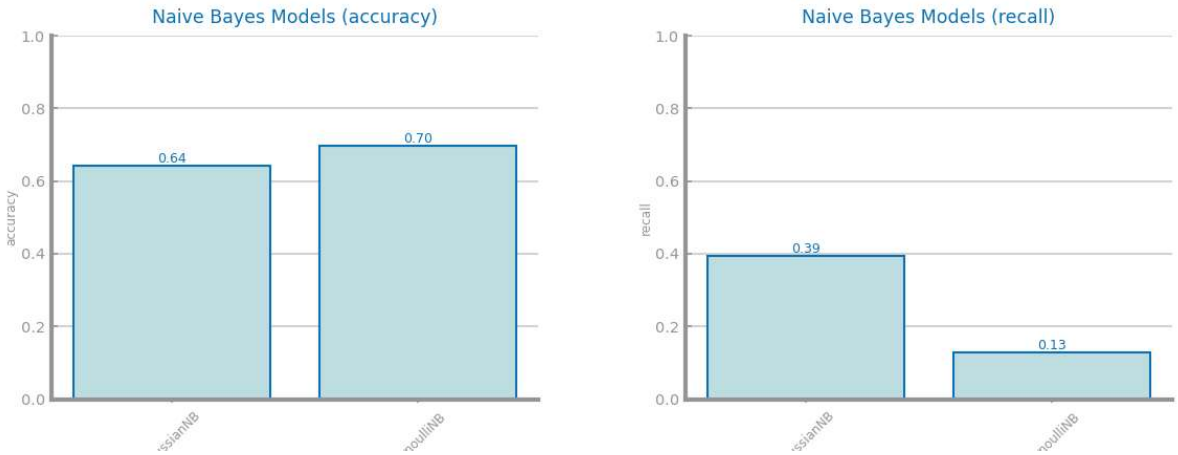


Figure 34 Naïve Bayes alternative comparison for dataset 2

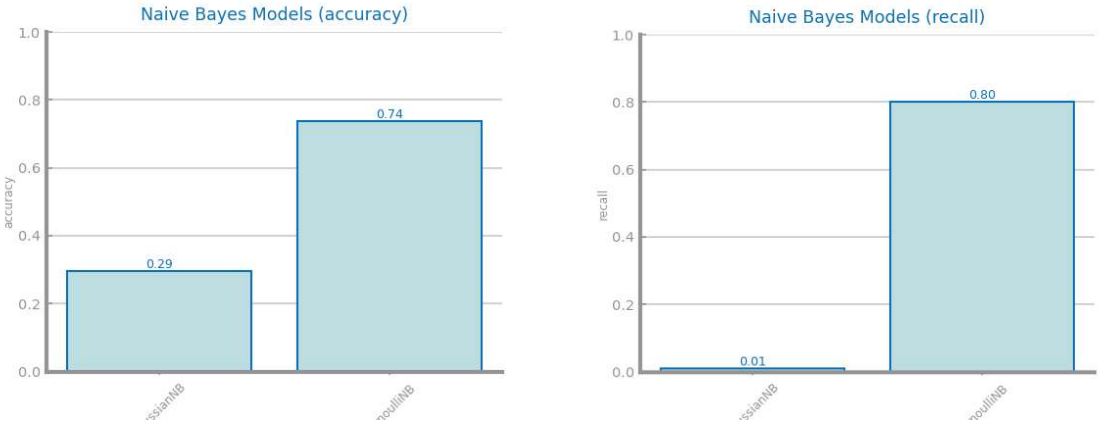
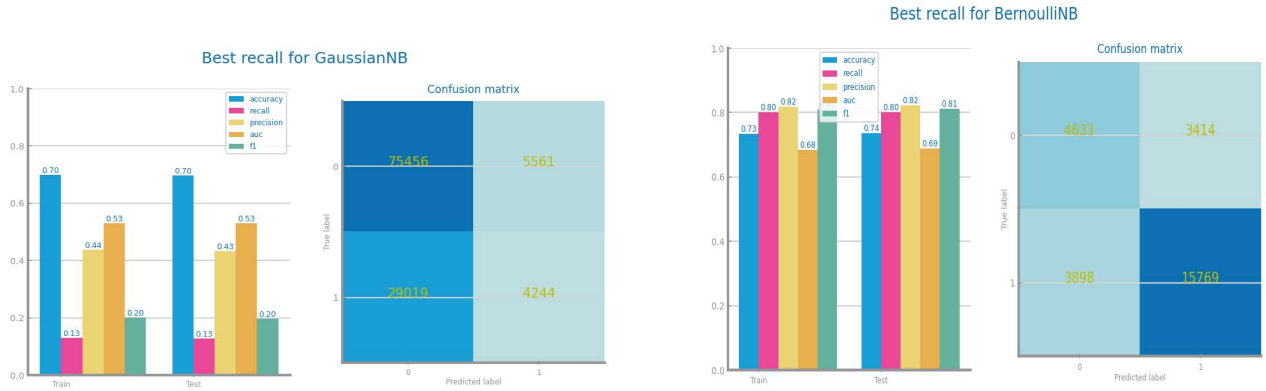


Figure 35 Naïve Bayes best model results for dataset 1 (left) and dataset 2 (right)



KNN

KNN algorithm relies heavily on the choice of distance metric, where Manhattan distance, although similar in performance, showcased a slight edge over Euclidean and Chebyshev distances in our evaluation. This could stem from Manhattan's robustness to scale differences and resilience to outliers in high-dimensional spaces efficiency compared to the others. Additionally, this may show that the features are correlated in magnitude but not direction, Manhattan distance might capture these nuances better. But we can't prove this because the results were very similar to the other distances.

Figure 36 KNN different parameterisations comparison for dataset 1

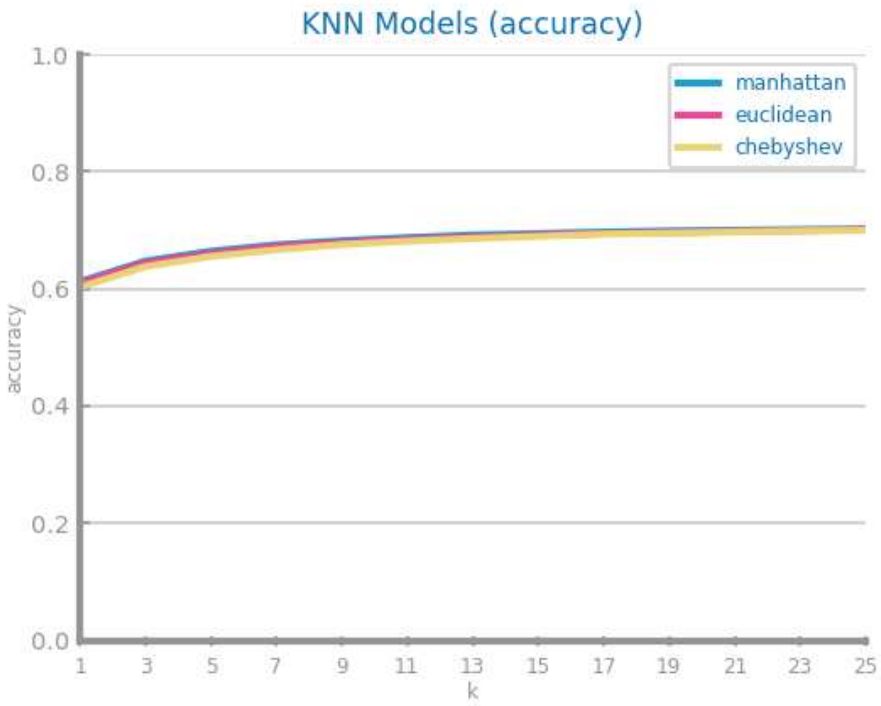


Figure 37 KNN different parameterisations comparison for dataset 2

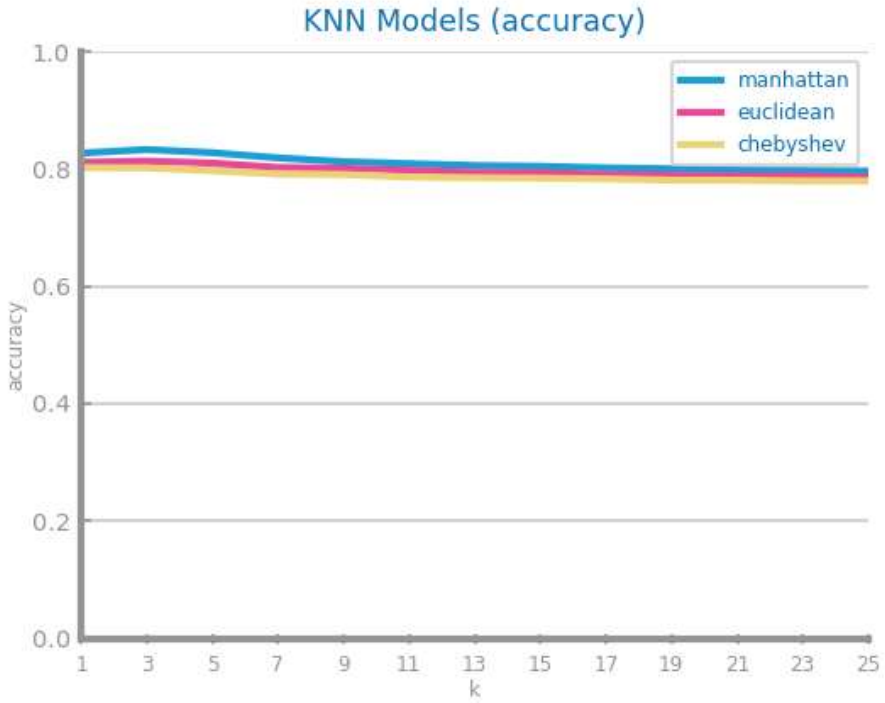


Figure 38 KNN overfitting analysis for dataset 1 (left) and dataset 2 (right)

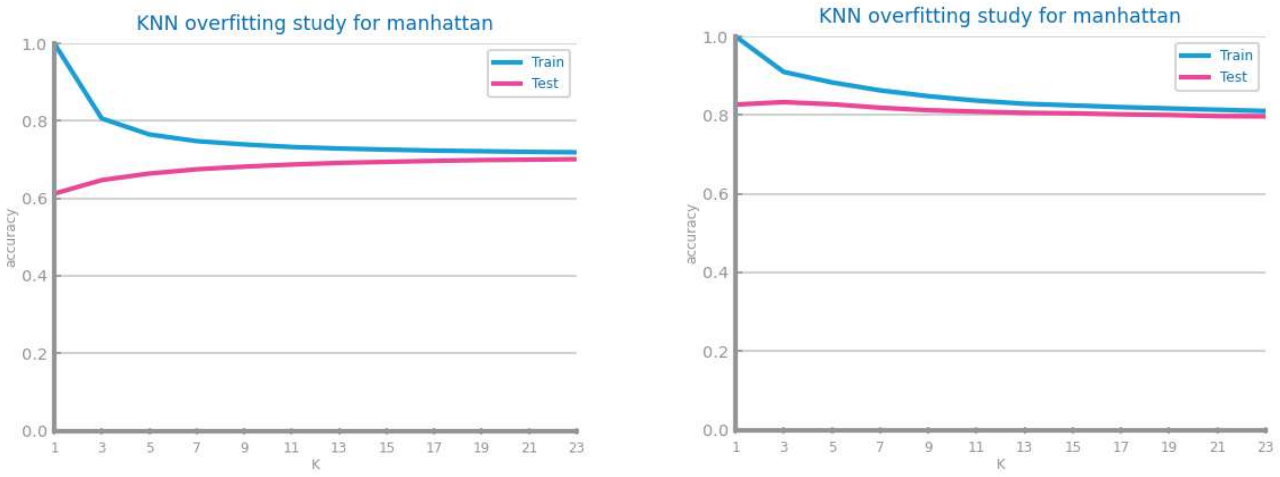
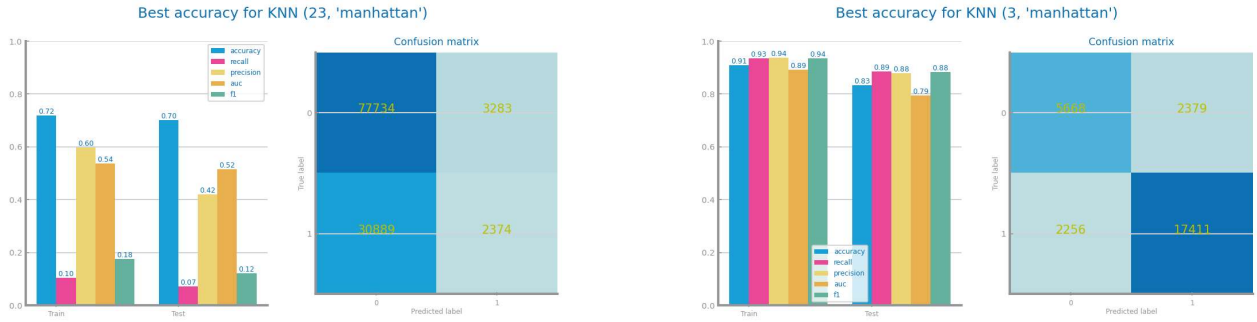


Figure 39 KNN best model results for dataset 1 (left) and dataset 2 (right)



Decision Trees

Decision trees serve as a straightforward means of understanding data. We utilized two criteria to determine the best variable: entropy gauged the information gained with a variable, while Gini assessed its impurity level. These measures helped in evaluating the significance and purity of variables for optimal tree construction. Through these criteria, we conducted an overfitting study by comparing training and test results. Additionally, we were able to save the best trees for potential manual analysis, providing a simplified overview if needed.

Figure 40 Decision Trees different parameterisations comparison for dataset 1

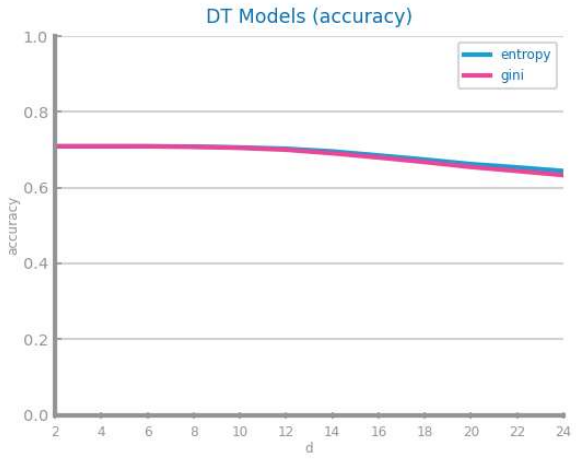


Figure 41 Decision Trees different parameterisations comparison for dataset 2

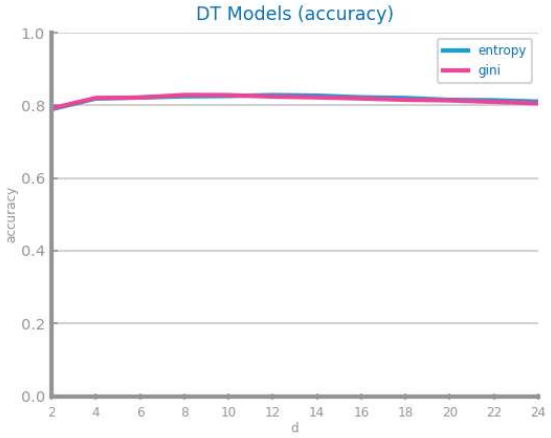


Figure 42 Decision Trees overfitting analysis for dataset 1 (left) and dataset 2 (right)

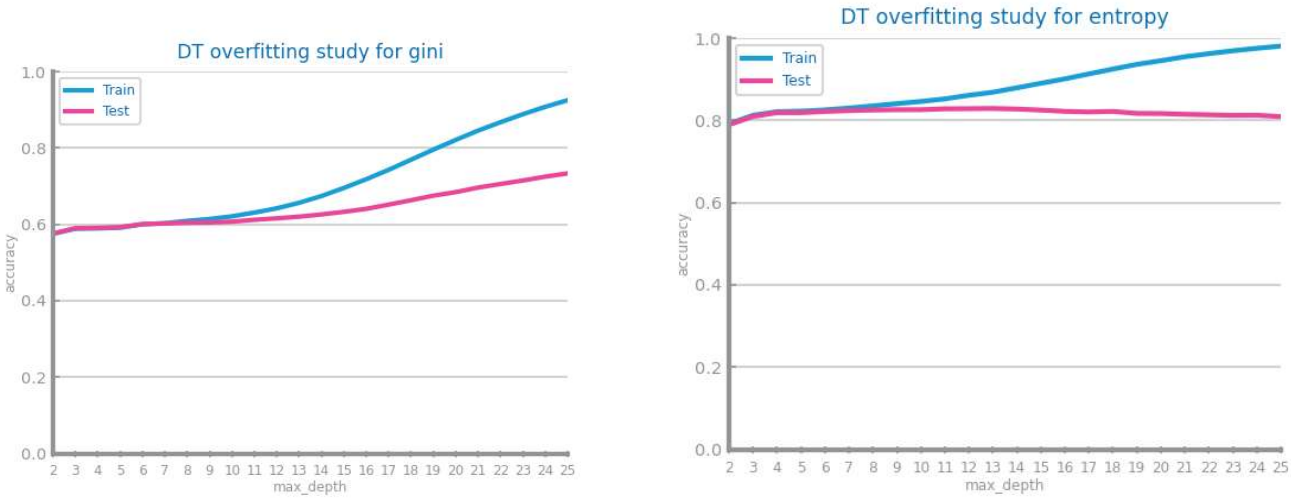


Figure 43 Decision trees best model results for dataset 1 (left) and dataset 2 (right)

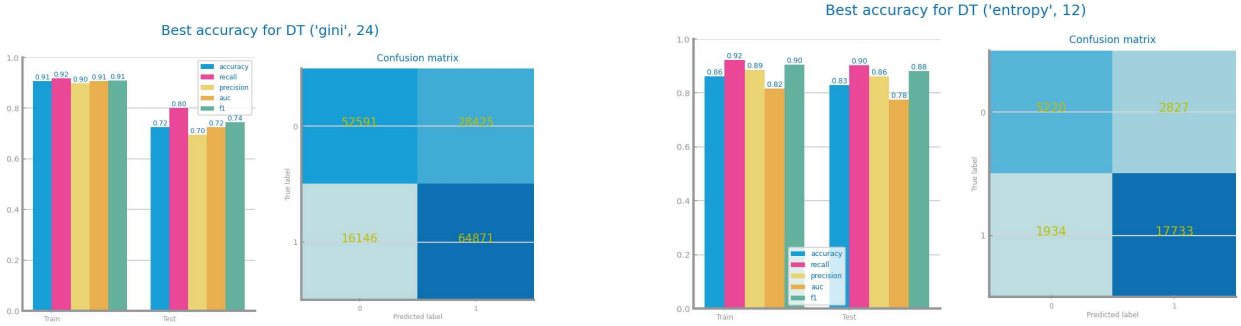


Figure 44 Best tree for dataset 1

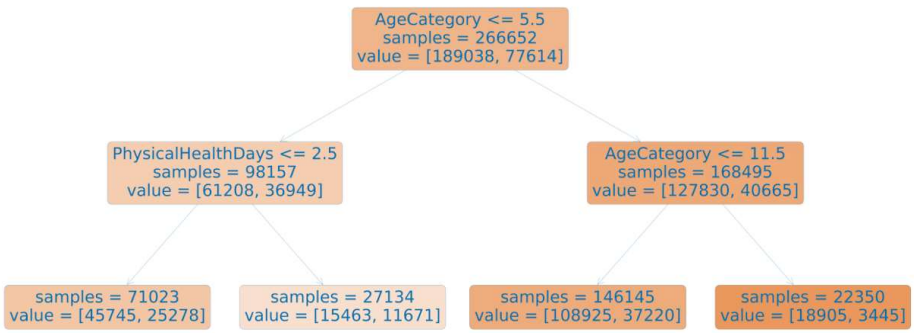
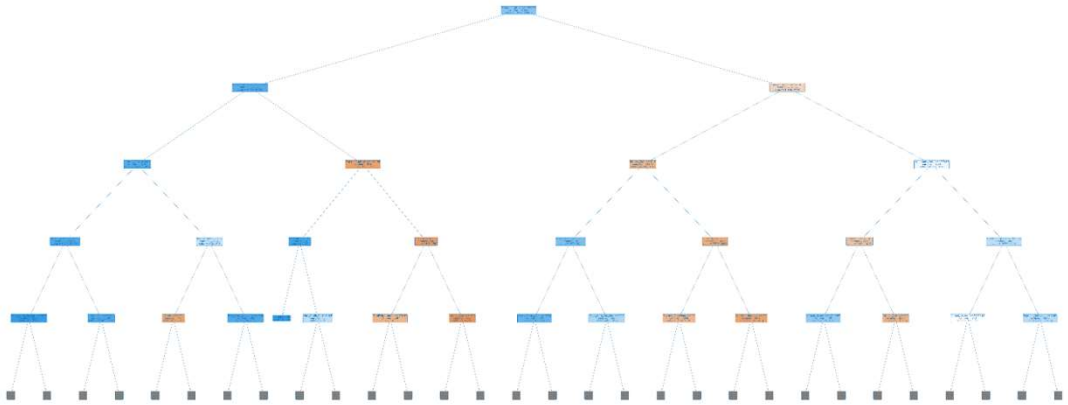


Figure 45 Best trees for dataset 2



Random Forests

Random forests have the particularity of providing the importance of each variable in the global model

pos_covid: category

credit_score: OutstandingDebt

Moreover, random forests are highly efficient, and their inherent randomness leads to a lower risk of overfitting. This is evident in our results, as the difference between the train set and test set outcomes is nearly imperceptible. We were also able to save the parameters yielding the best results for each dataset. (figure 49)

Figure 46 Random Forests different parameterisations comparison for dataset 1

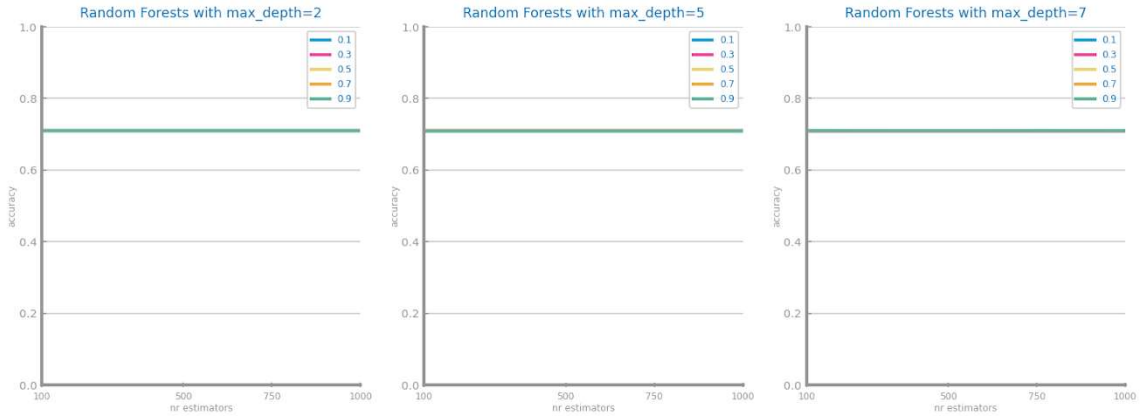


Figure 47 Random Forests different parameterisations comparison for dataset 2

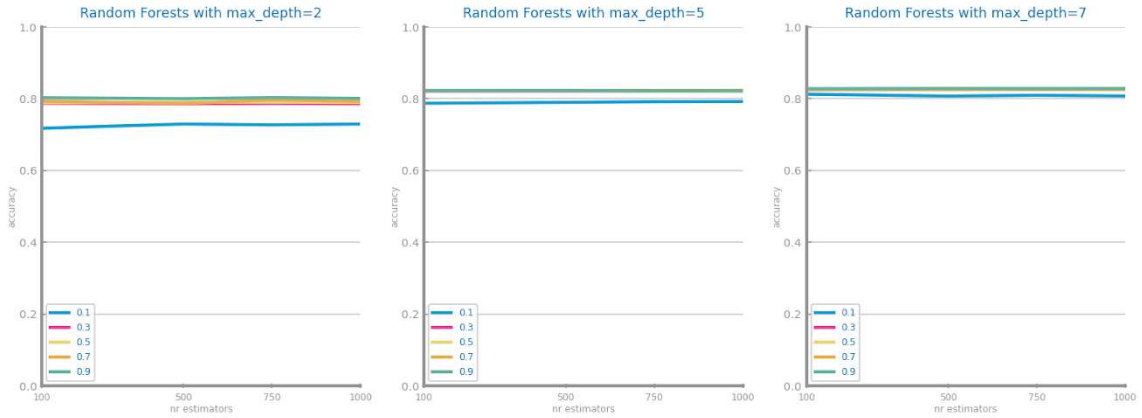


Figure 48 Random Forests overfitting analysis for dataset 1 (left) and dataset 2 (right)

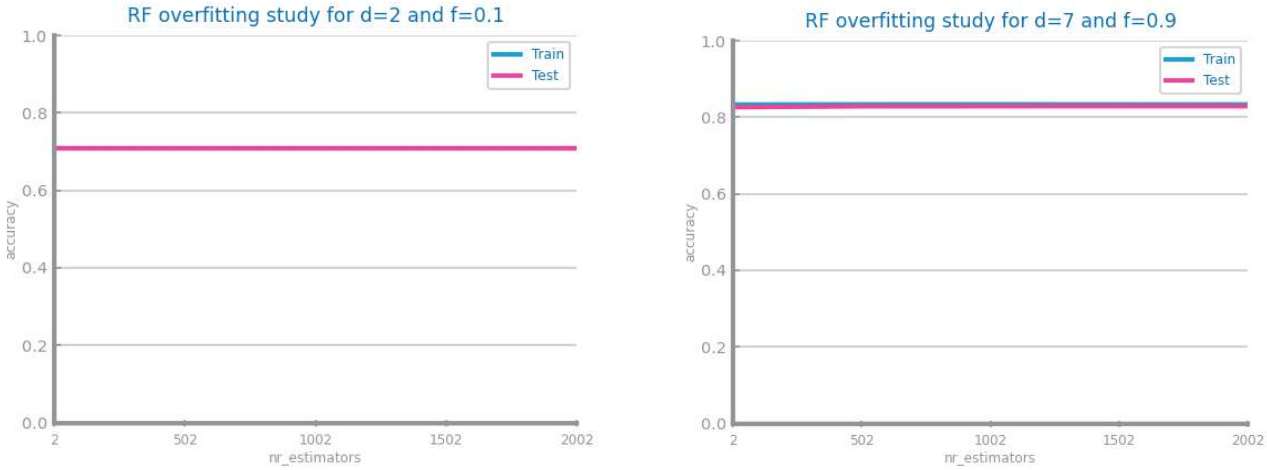


Figure 49 Random Forests best model results for dataset 1 (left) and dataset 2 (right)

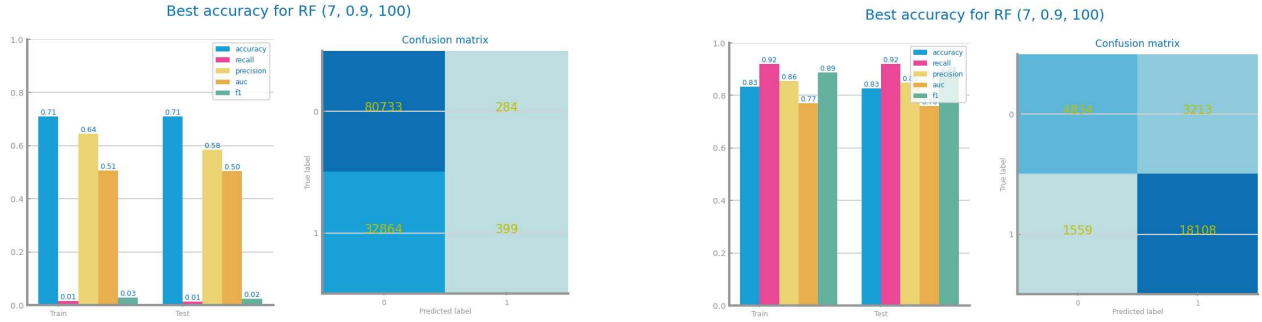
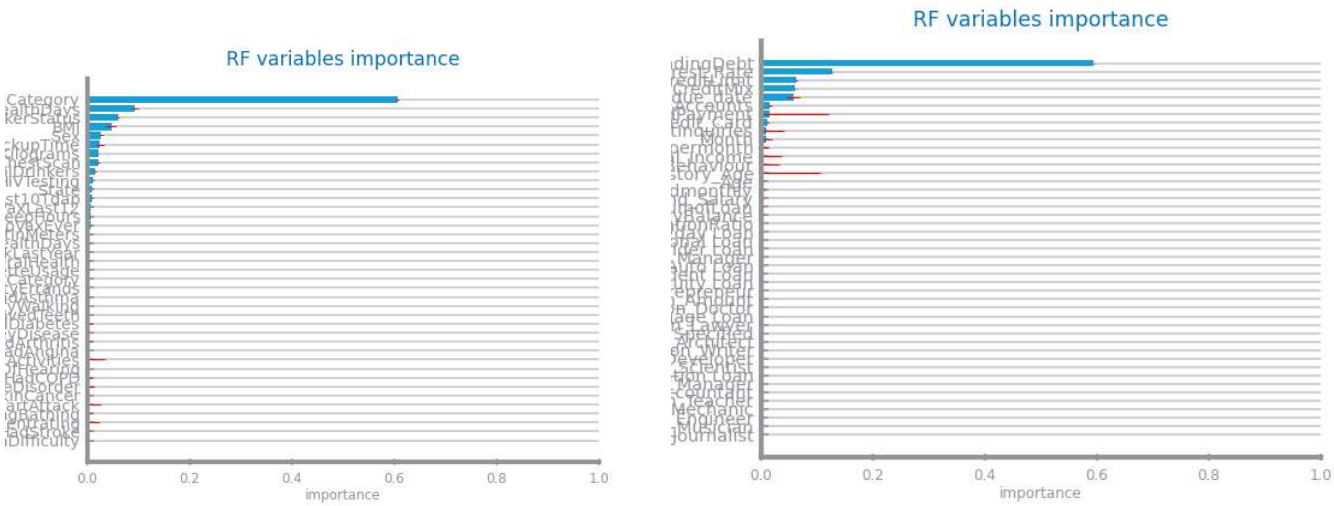


Figure 50 Random Forests variables importance for dataset 1 (left) and dataset 2 (right)



Gradient Boosting

For dataset 1, the observed outcomes seem to deviate from the expected results. This discrepancy is evident in the confusion matrix, where there are almost no observed true negatives (very low recall), indicating a certain fragility in the model's performance. However, dataset 2 does not exhibit any apparent irregularities in the results, so we can infer that overfitting becomes apparent beyond 500 estimators for d = 7 and lr = 0.3 for the second dataset.

Figure 51 Gradient boosting different parameterisations comparison for dataset 1

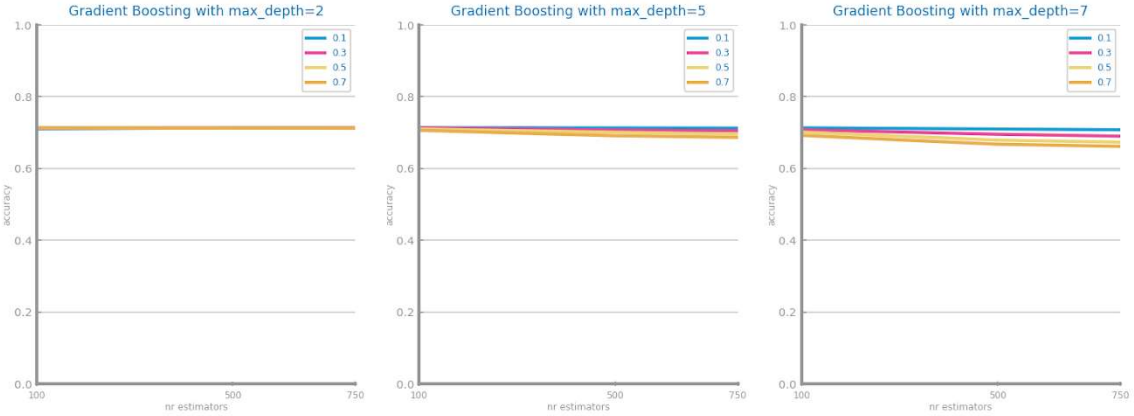


Figure 52 Gradient boosting different parameterisations comparison for dataset 2

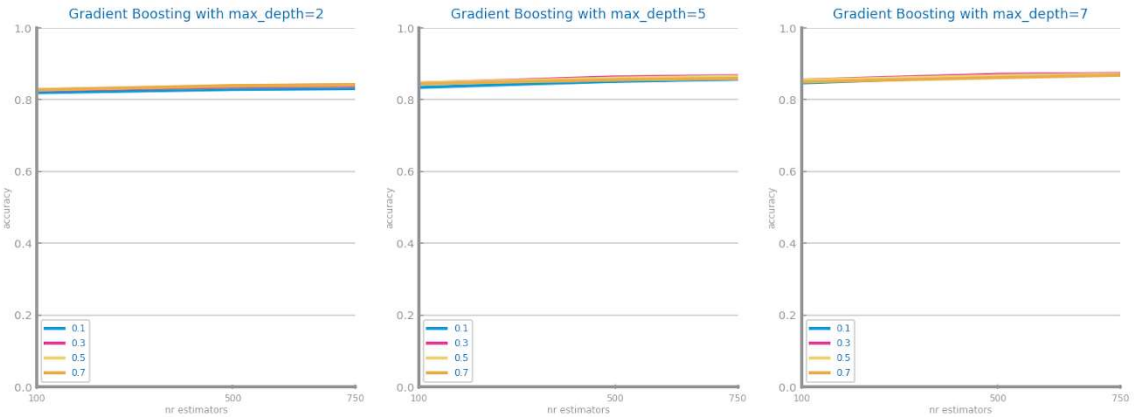


Figure 53 Gradient boosting overfitting analysis for dataset 1 (left) and dataset 2 (right)

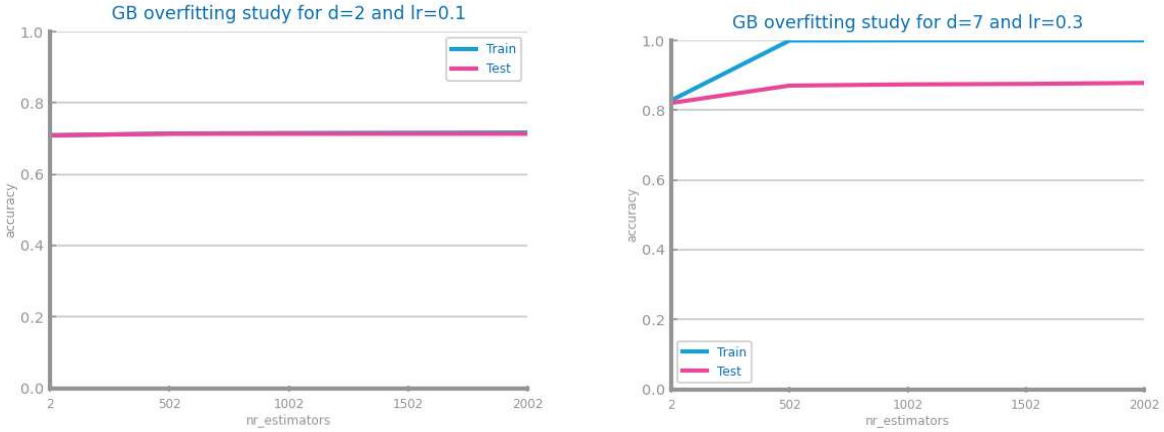


Figure 54 Gradient boosting best model results for dataset 1 (left) and dataset 2 (right)

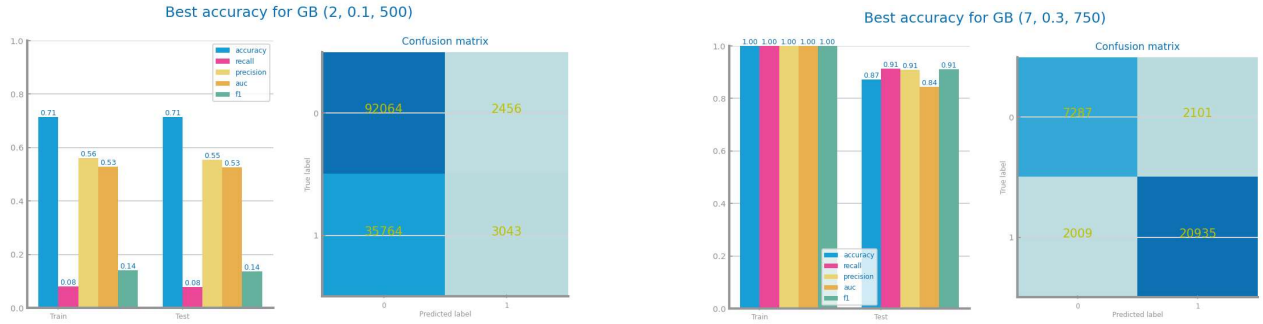
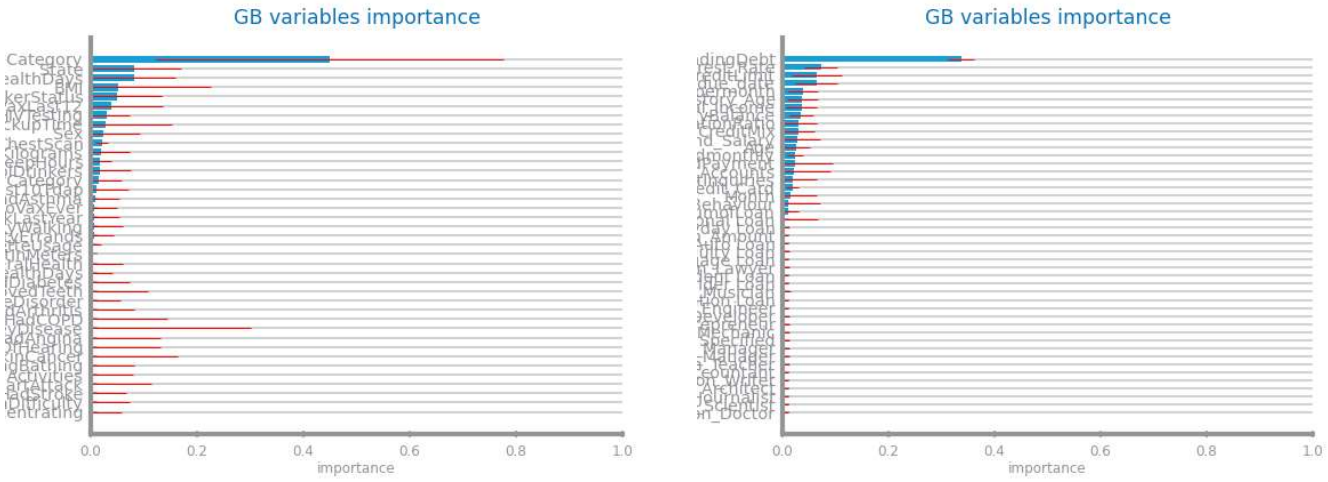


Figure 55 Gradient boosting variables importance for dataset 1 (left) and dataset 2 (right)



Multi-Layer Perceptrons

The results for the MLP did not meet our expectations, particularly concerning parameterization. Despite testing various learning rates, types and values, the outcomes remained remarkably stable, deviating from the typical behavior. This consistency might indicate an irregularity in the dataset that we might have overlooked or an error in the code during model execution. Consequently, we refrain from drawing definitive conclusions based on these results.

This becomes further evident when evaluating the confusion matrices, where one lacks true positives entirely, and the other exhibits an almost negligible count of true negatives (very low recall).

Figure 56 MLP different parameterisations comparison for dataset 1

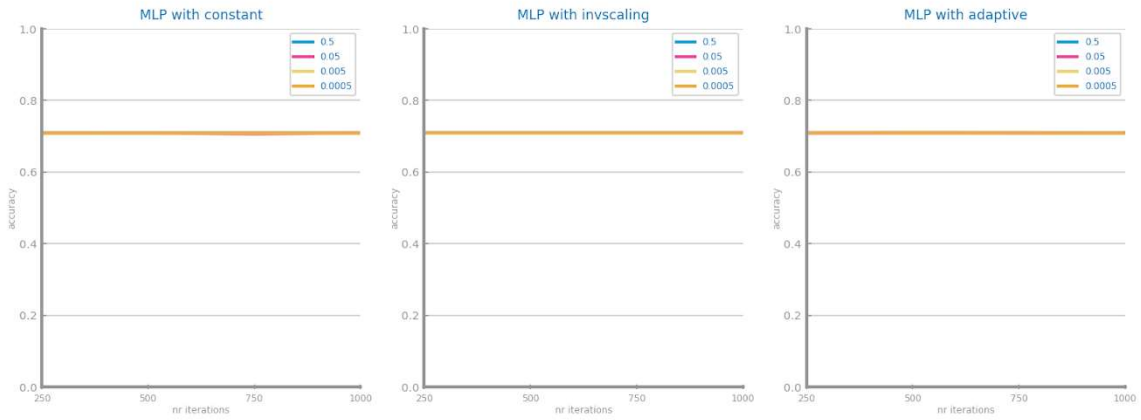


Figure 57 MLP different parameterisations comparison for dataset 2

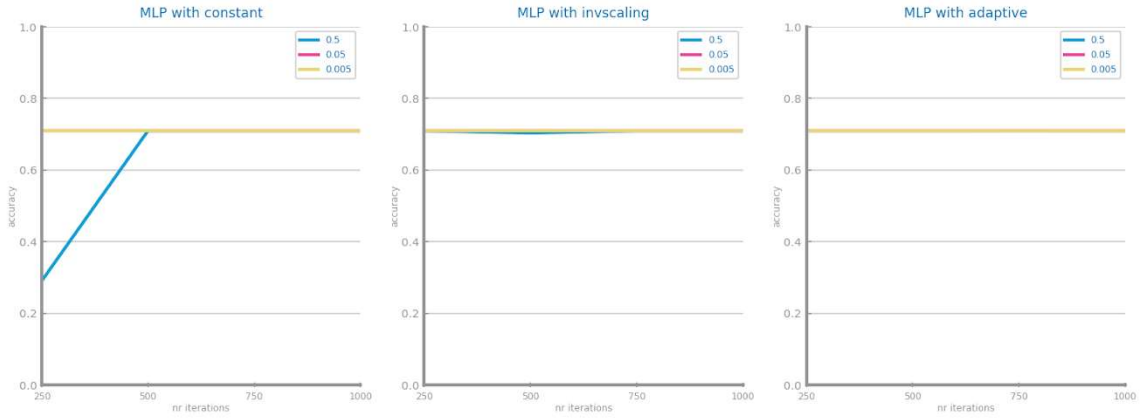


Figure 58 MLP overfitting analysis for dataset 1 (left) and dataset 2 (right)

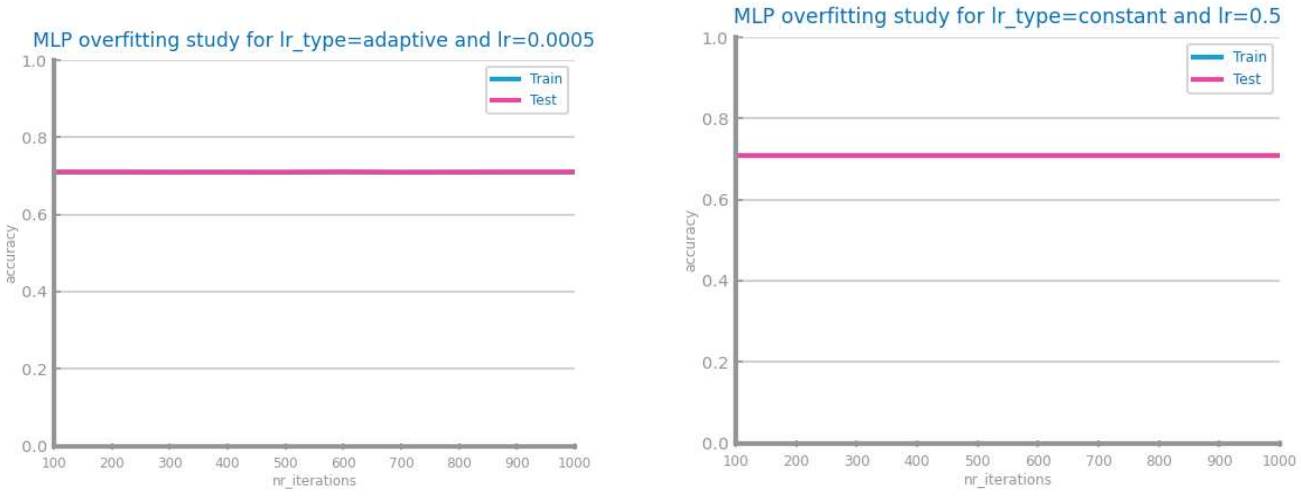
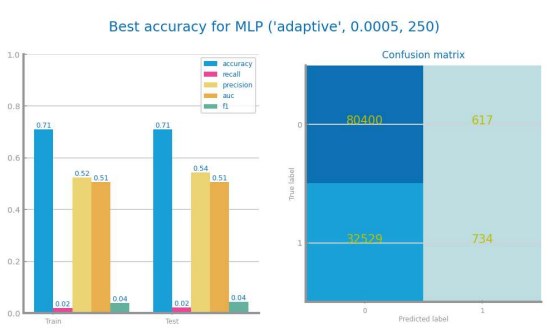
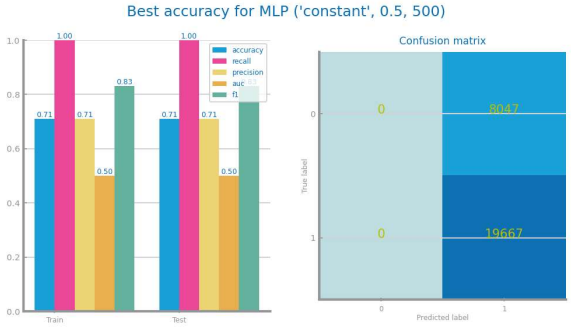


Figure 59 MLP best model results for dataset 1 (left) and dataset 2 (right)





4 Critical Analysis

Shall be used to present a summary of the results achieved with the different modeling techniques, and the impact of the different preparation tasks on their performance.

A cross-analysis of the different models may also be presented, identifying the most relevant variables common to all of them (when possible) and the relation among the patterns identified within the different classifiers.

A critical assessment of the best models shall be presented, clearly stating if the models seem to be good enough for the problem at hand.

Additional charts may be presented here. Shall not exceed 2000 characters.

In order to correctly apply the models to our data, we had to make some modifications to the original datasets, converting all types of nominal variables into numeric ones. (see section 2).

(dataset 1)

It's important to note that for many model evaluation algorithms, **dataset 1** shows very low recall values but high precision values. This indicates that either these algorithms are not suitable for the dataset in question, or the dataset is somehow misaligned. However, given the positive outcomes obtained for the **decision tree** algorithm, it seems more likely that the first option is true. The best results for dataset 1 were achieved with the decision tree. Despite the results not being as appealing as we'd prefer, the decision tree model can still be used as an additional method for performing classification tasks. However, considering its performance isn't as high as desired, it should be used with proper caution.

(dataset 2)

Meanwhile, **dataset 2** doesn't seem to have significant issues while handling different algorithms, except for lower outcomes in Gaussian approaches. However, it's crucial to note that both Gradient Boosting and Decision Trees exhibit signs of overfitting at higher depths, which could pose an issue if we intend to deploy these models in real-life scenarios. It's important to mention this observation to emphasize that the models are not foolproof. Nevertheless, the results for the test dataset are high, and by using various models, it's possible to obtain highly reliable outcomes. For the dataset 2, the model with the best results is the Gradient Boosting, with all evaluation parameters exceeding 80% on the test set. The results were generally very good. Therefore, we can effectively use some of the models with the best performance and combine them to increase our confidence in the classification task outcomes. By leveraging the strengths of each model, we can achieve very high levels of confidence. However, it's important to note that these models are predictive and, as such, prone to potential failures.

TIME SERIES FORECASTING

1 DATA PROFILING

Data Dimensionality and Granularity

In the covid dataset, the 'date' variable is formatted as yyyy-mm-dd, providing a daily level of granularity as the highest. Additionally, the data allows for analysis at monthly and yearly levels.

For the credit dataset, the 'Timestamp' variable records values in yyyy-mm-dd hh:mm format. The most atomic granularity is at the minute level, and further analysis can be conducted at hourly and daily levels, given that both year and month remain constant across the dataset.

Figure 62 Original time series 1 (the most atomic detail)

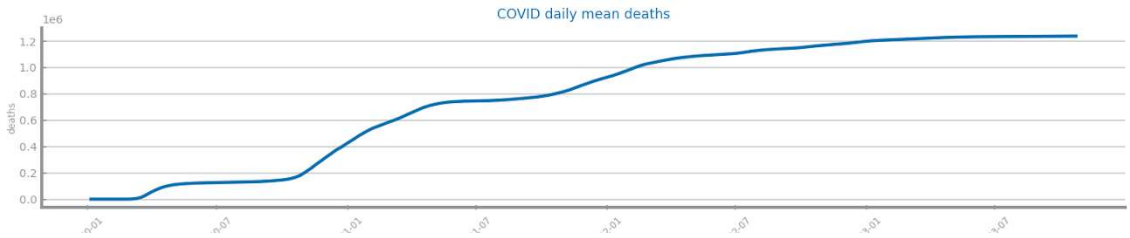


Figure 63 Time series 1 at the second chosen granularity

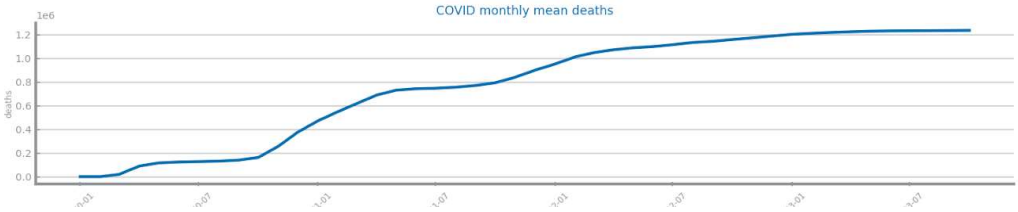


Figure 64 Time series 1 at the third chosen granularity

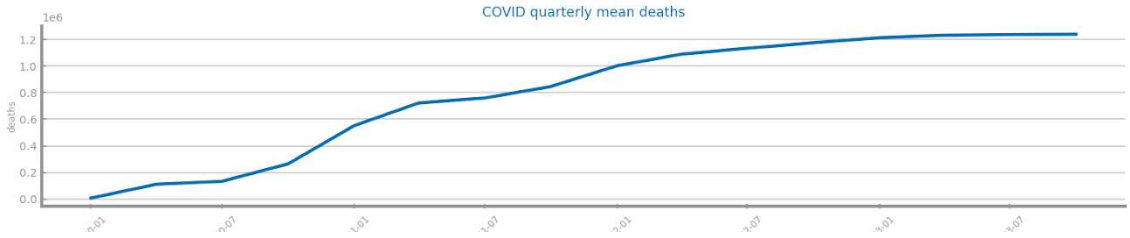


Figure 65 Original time series 2 (the most atomic detail)

Figure 66 Time series 2 at the second chosen granularity

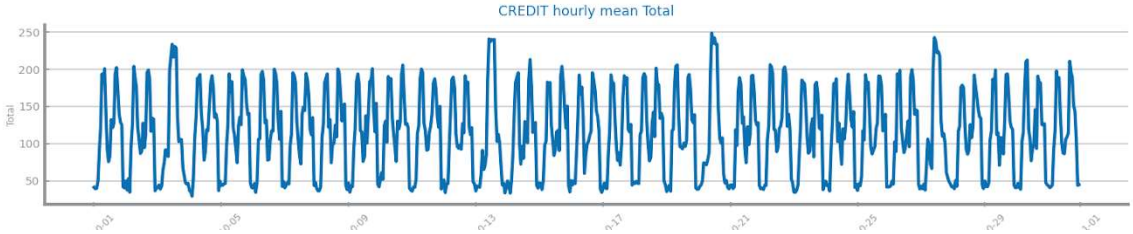


Figure 67 Time series 2 at the third chosen granularity

Data Distribution

Minute-level outliers (Fig. 69) need attention as outliers can skew statistical measures. The first series lacks values in a specific range (Fig. 70), requiring consideration. Figure 72 suggests distribution patterns, prompting further investigation. No significant autocorrelation at any lag for the first time series (Fig. 74) suggesting independence of past values. Peaked pattern with symmetric rise and fall in autocorrelation values of time series 2 (Fig. 75) points at periodic distribution.

Figure 68 Boxplots for time series 1 at different granularities

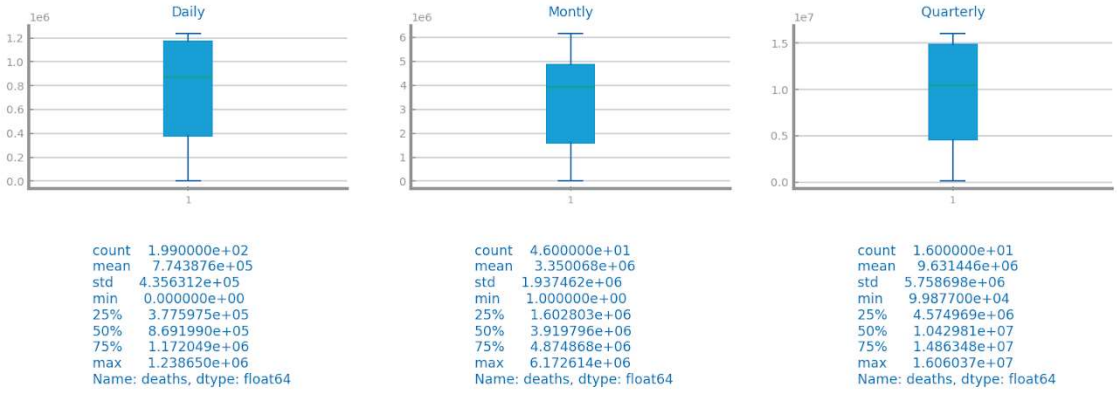


Figure 69 Boxplots for time series 2 at different granularities

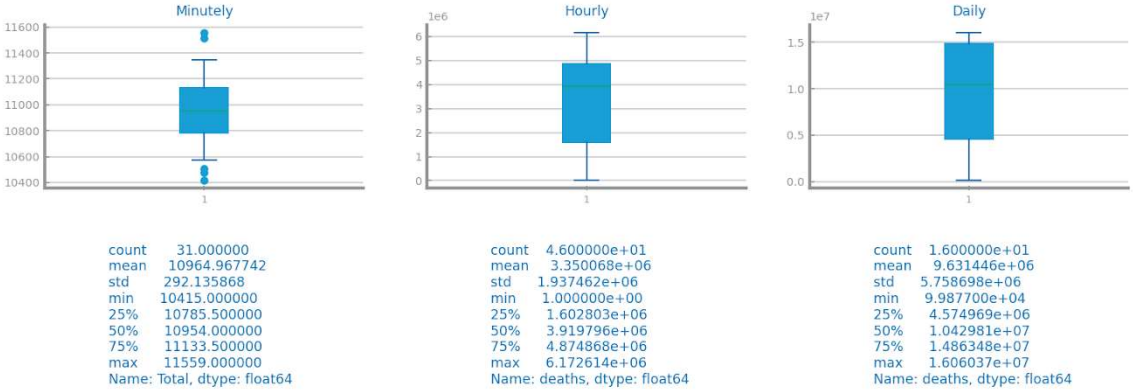


Figure 70 Histograms for time series 1 at different granularities

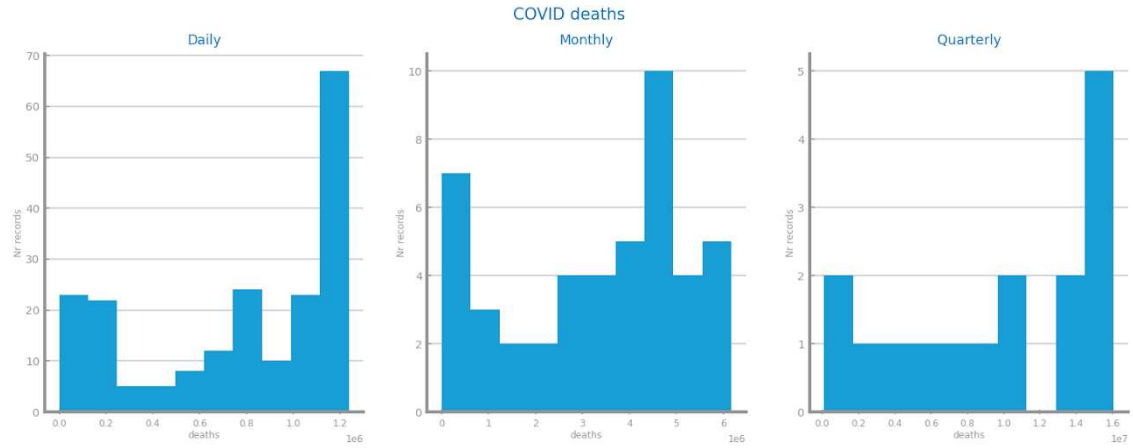


Figure 71 Histograms for time series 2 at different granularities

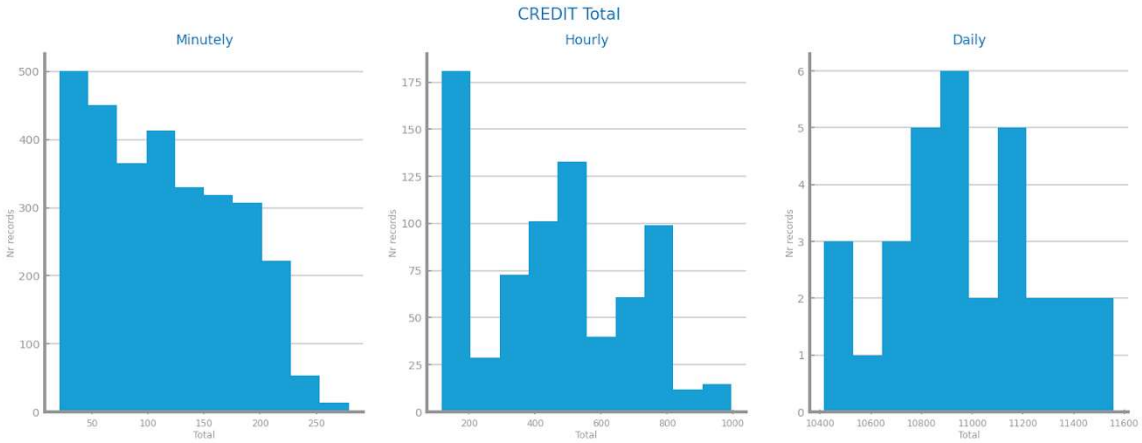


Figure 72 Autocorrelation lag-plots for original time series 1

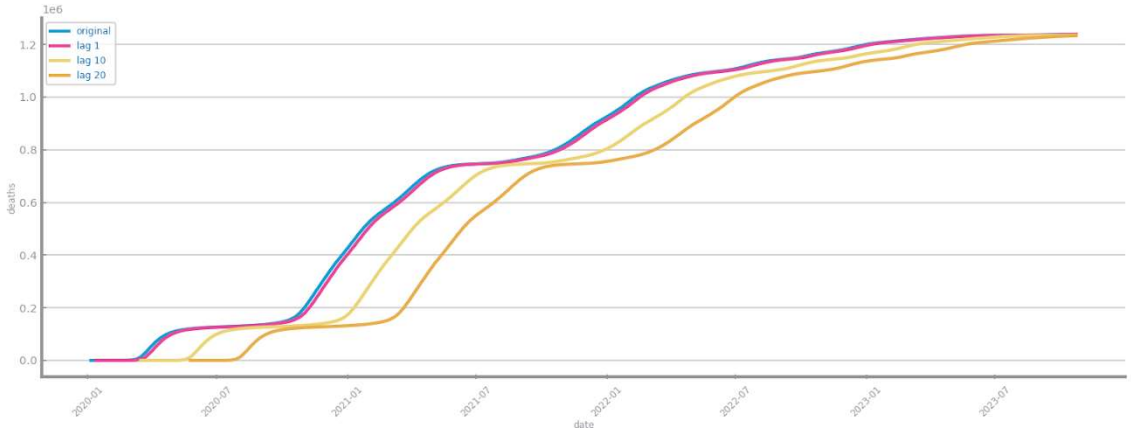


Figure 73 Autocorrelation lag-plots for original time series 2

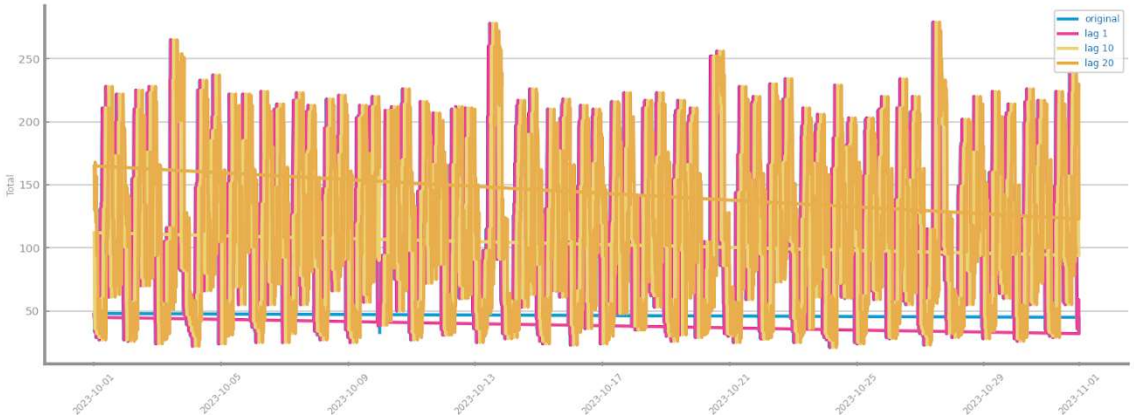


Figure 74 Autocorrelation correlogram for original time series 1

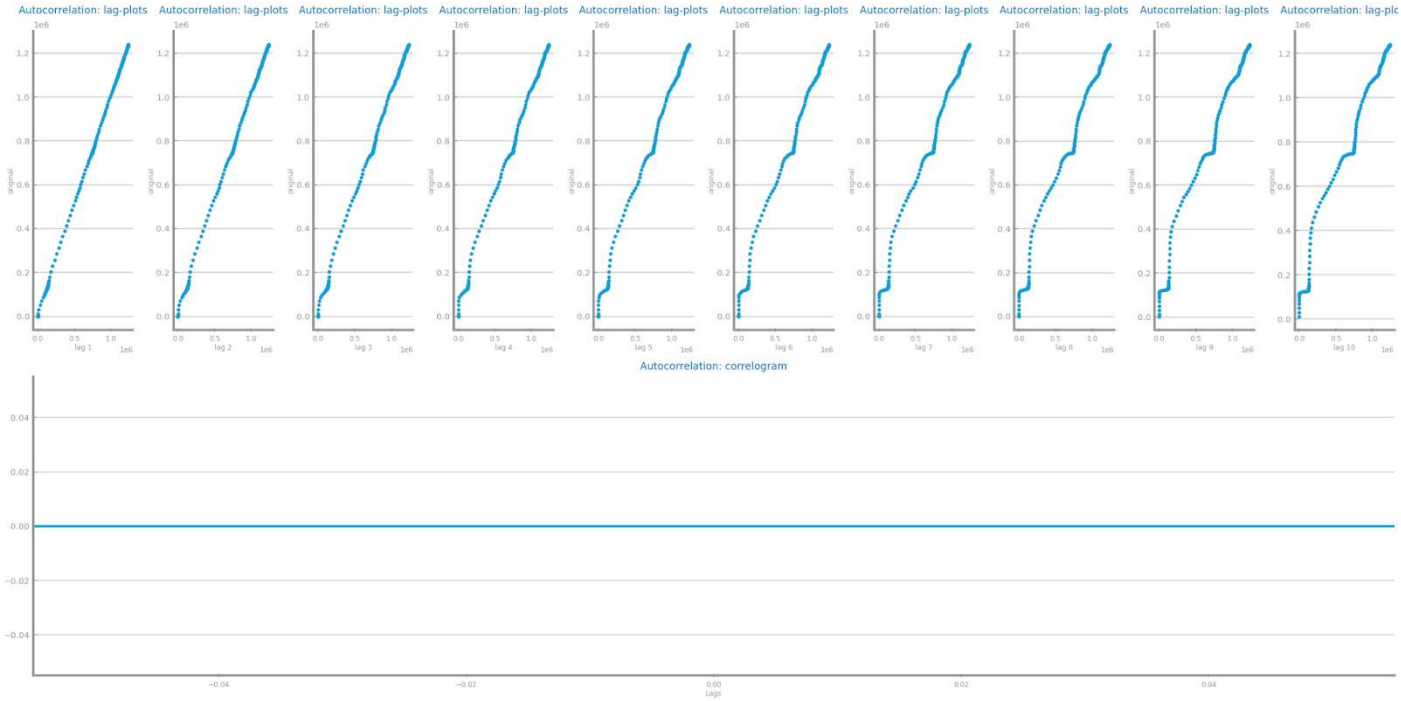
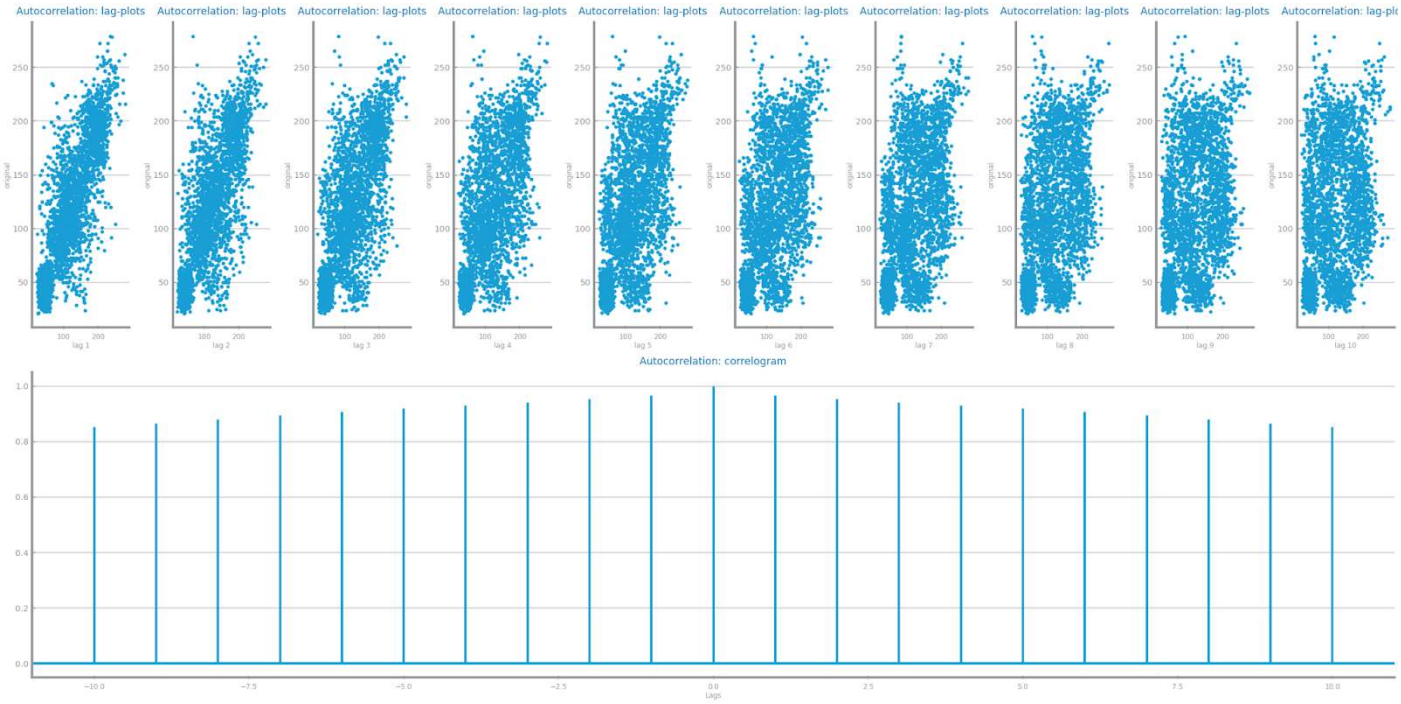


Figure 75 Autocorrelation correlogram for original time series 2



Data Stationarity

1:Despite an initial rise in observed and trend components, it begins to level off. The seasonal exhibits a distinct pattern, and residuals start adhering to a specific trend.The mean is moderately increasing over time.

2:While the mean remains constant, the seasonal displays a distinct structure.

Considering that ADF Statistic>critical values, p-value exceeds 0.05, there is no evidence to reject the null hypothesis of non-stationarity.Thus, the conclusion is that both datasets are non-stationary.

Figure 76 Components study for time series 1

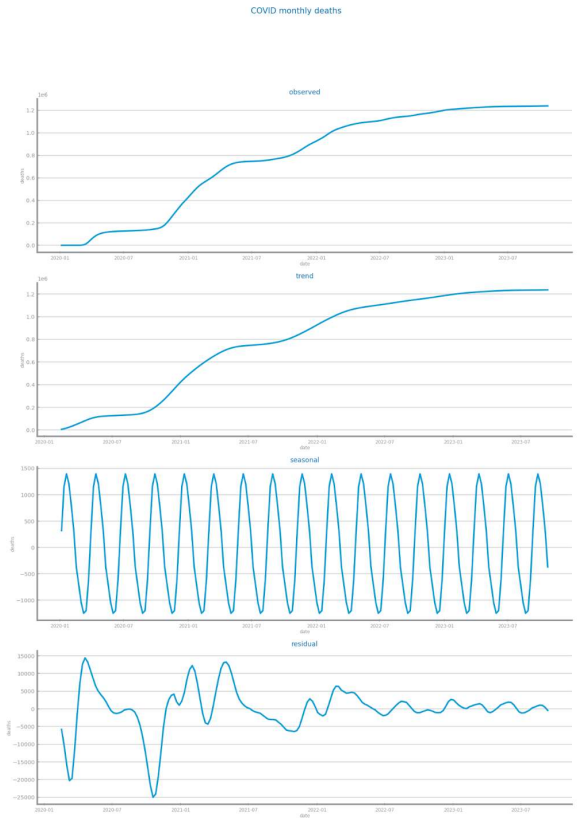


Figure 77 Stationarity study for time series 1

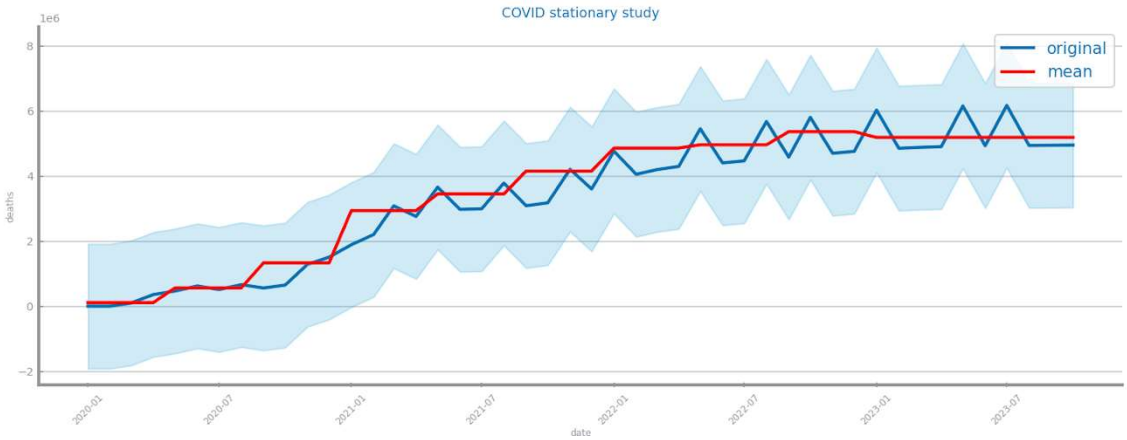


Figure 78 Components study for time series 2

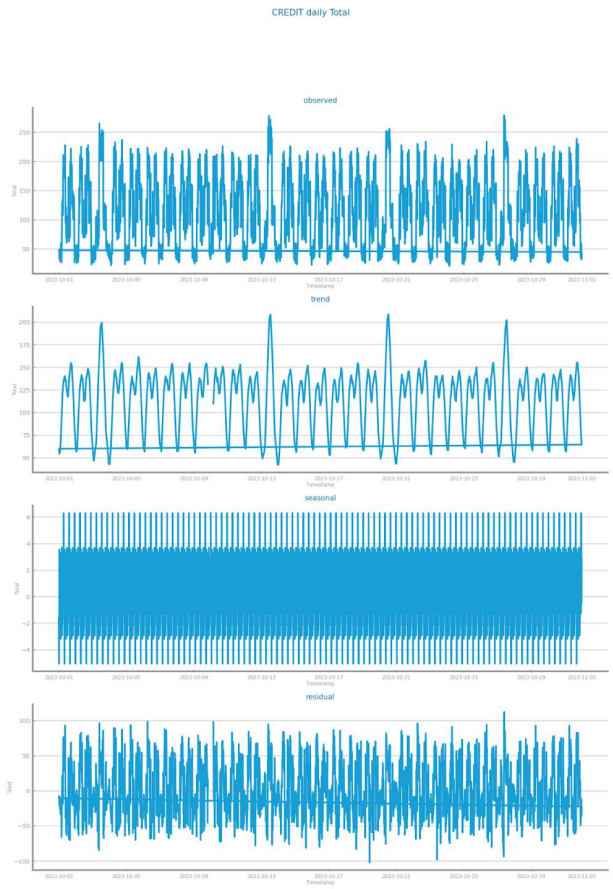
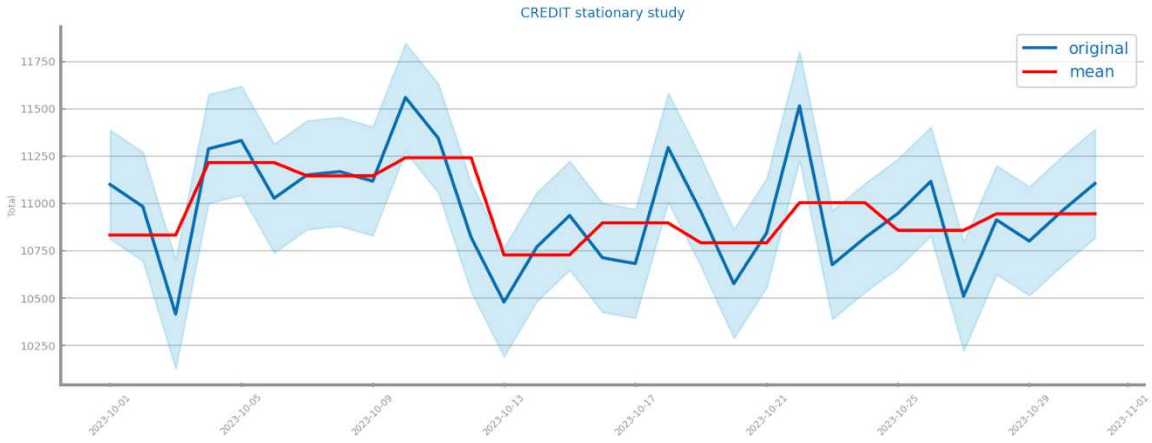


Figure 79 Stationarity study for time series 2



2 Data Transformation

Aggregation

For the Covid dataset, we compared daily, monthly, and quarterly aggregations, and found that daily aggregation yielded the optimal results.

For the Credit Score dataset, we examined minute, hourly, and daily aggregations, and determined that minute-by-minute aggregation was the most appropriate.

Figure 80 Forecasting plots after different aggregations on time series 1

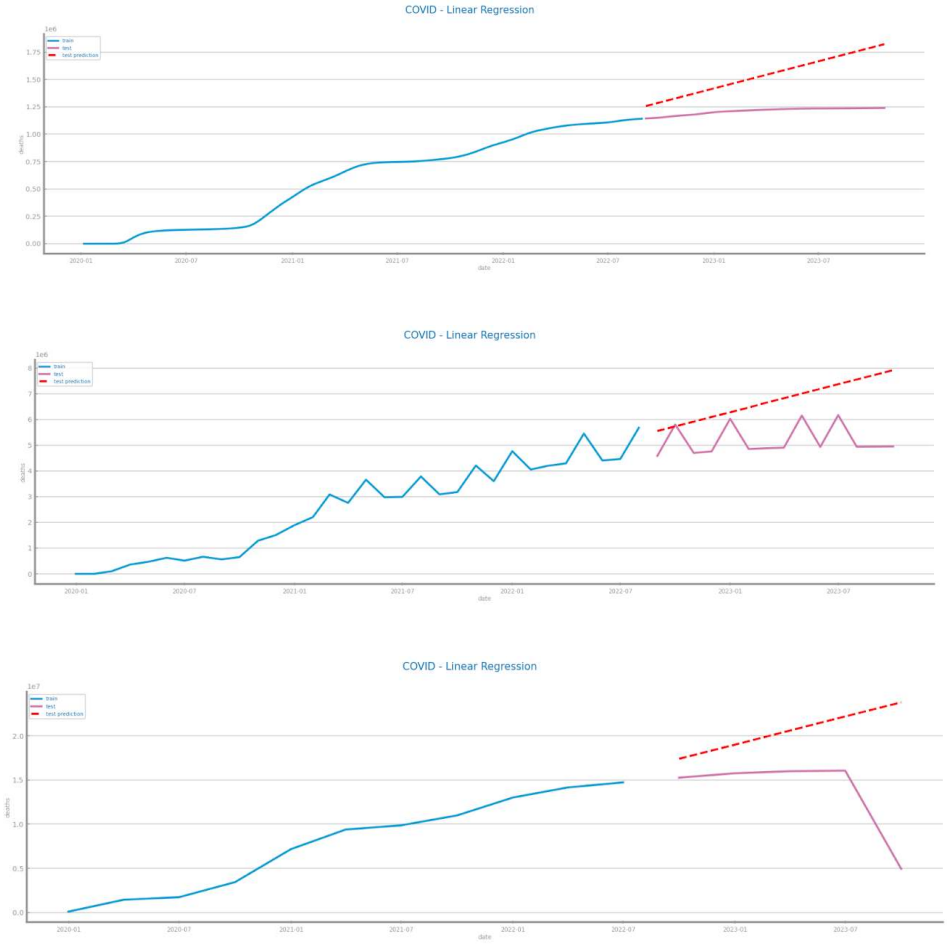


Figure 81 Forecasting results after different aggregations on time series 1

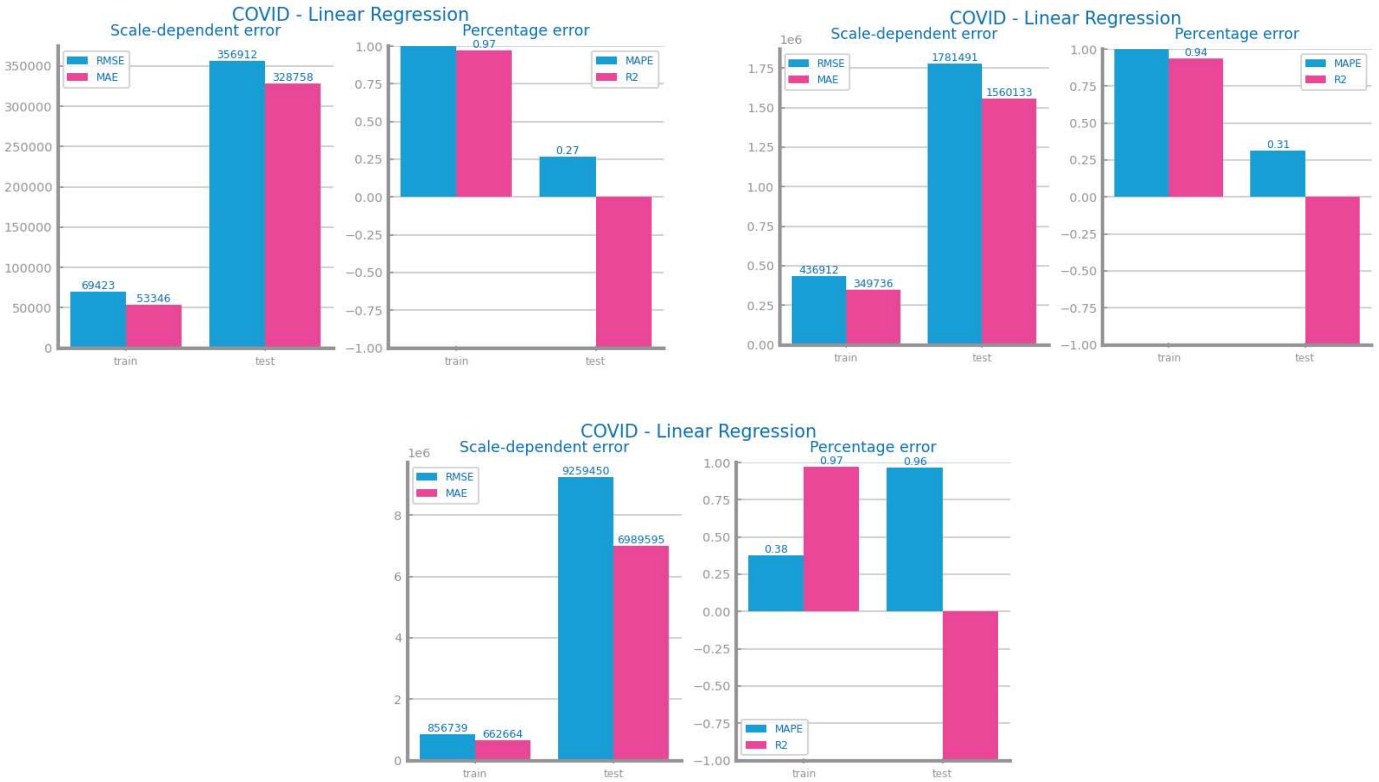


Figure 82 Forecasting plots after different aggregations on time series 2

Figure 83 Forecasting results after different aggregations on time series 2

Smoothing

For the Covid dataset, a window size of 50 was found to be optimal after applying smoothing techniques. In the Credit Score dataset, the best results were achieved with a window size of 25.

Figure 84 Forecasting plots after different smoothing parameterisations on time series 1

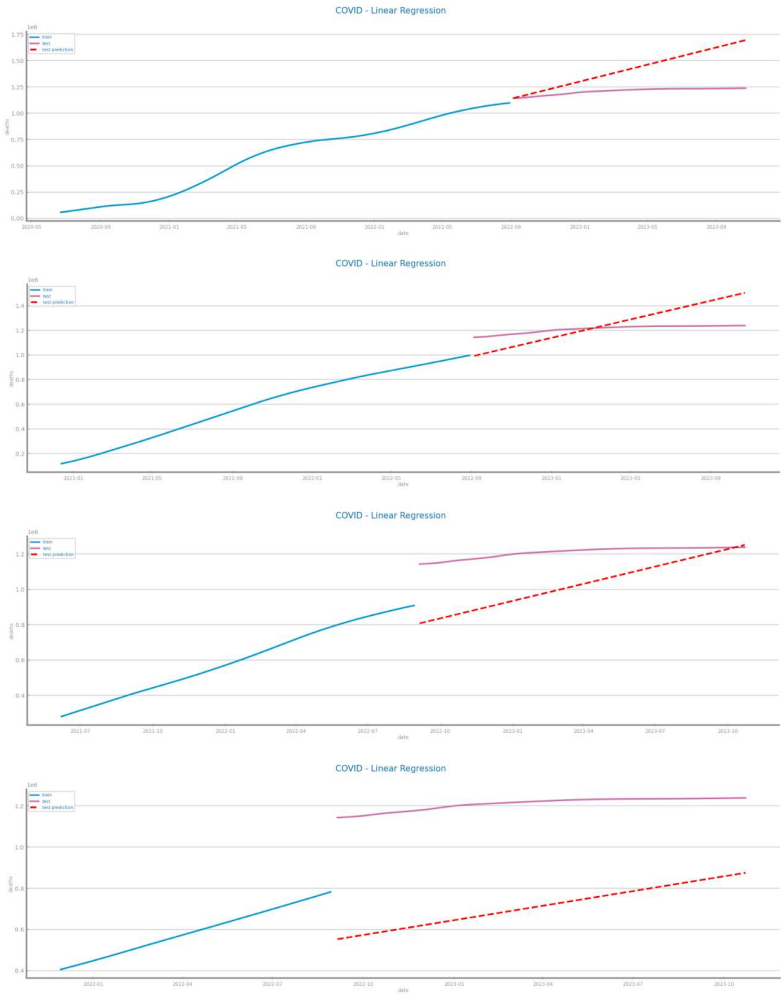


Figure 85 Forecasting results after different smoothing parameterisations on time series 1

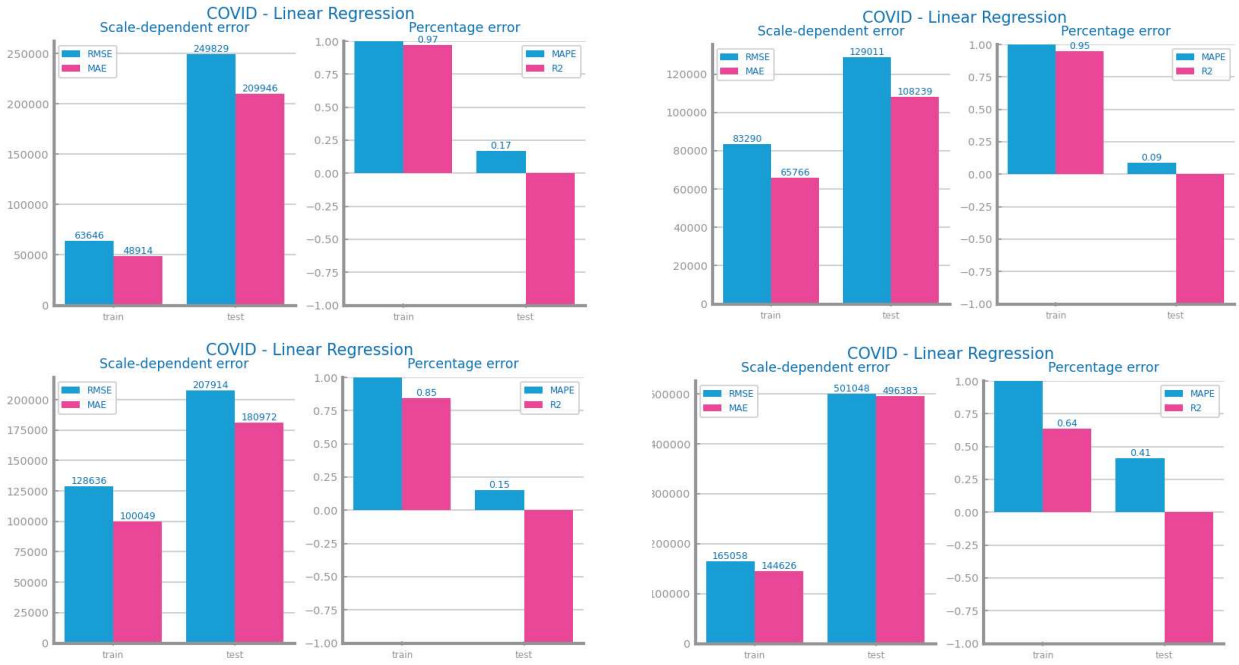
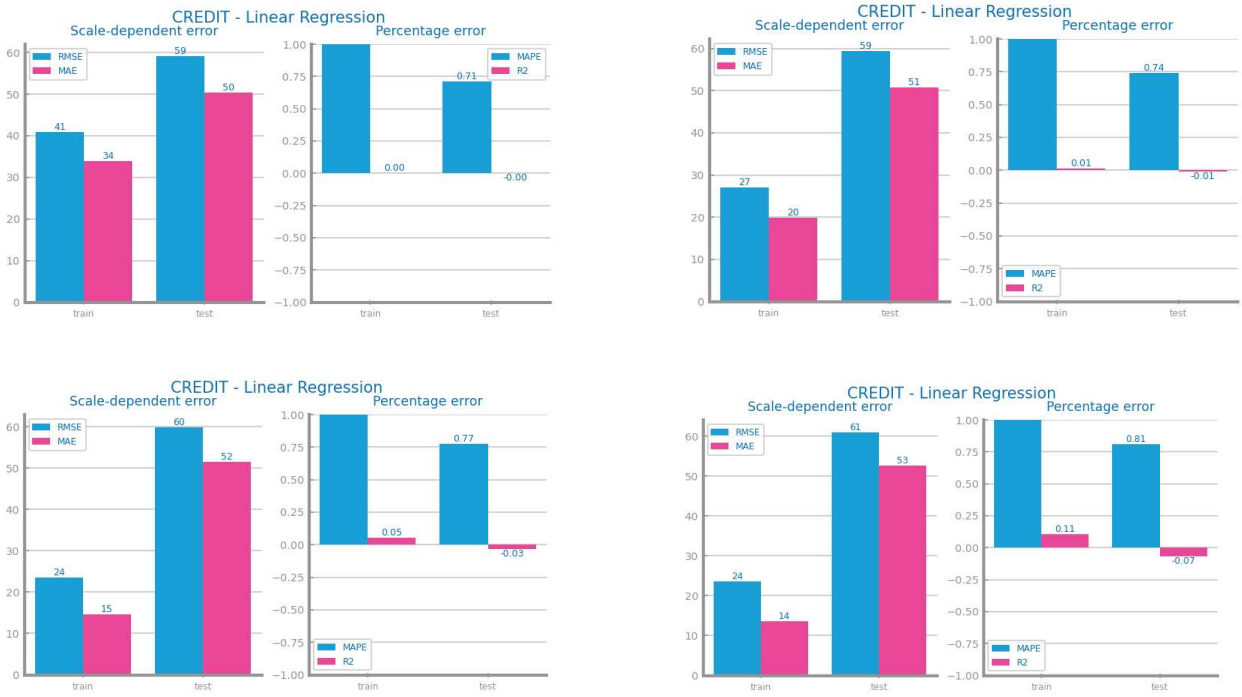


Figure 86 Forecasting plots after different smoothing parameterisations on time series 2

Figure 87 Forecasting results after different smoothing parameterisations on time series 2



Differentiation

We noted that differencing both time series brought volatility, probably because of the introduction of noise. Therefore we opted for keeping the original time series after transformation without any differentiation.

Figure 88 Forecasting plots after first and second differentiation of time series 1

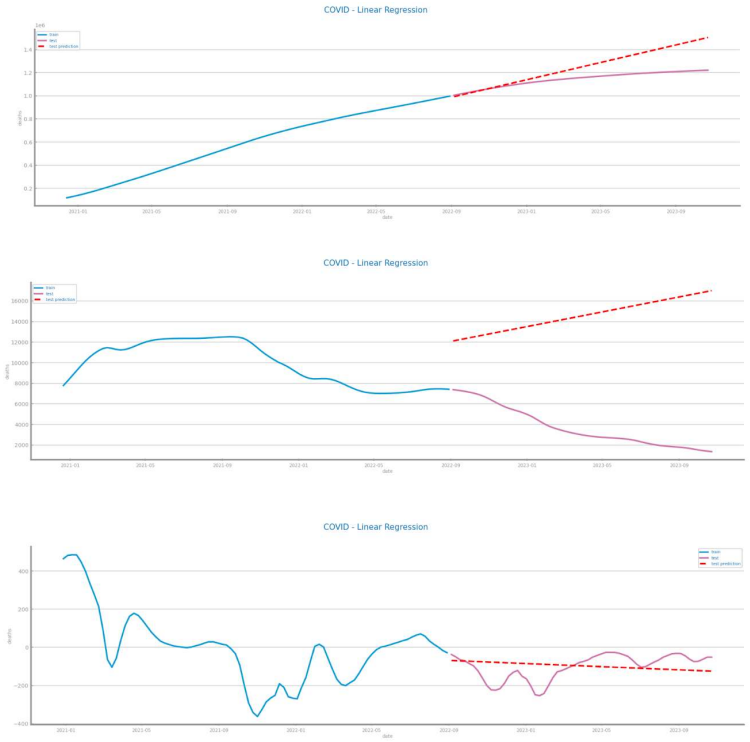


Figure 89 Forecasting results after first and second differentiation of time series 1

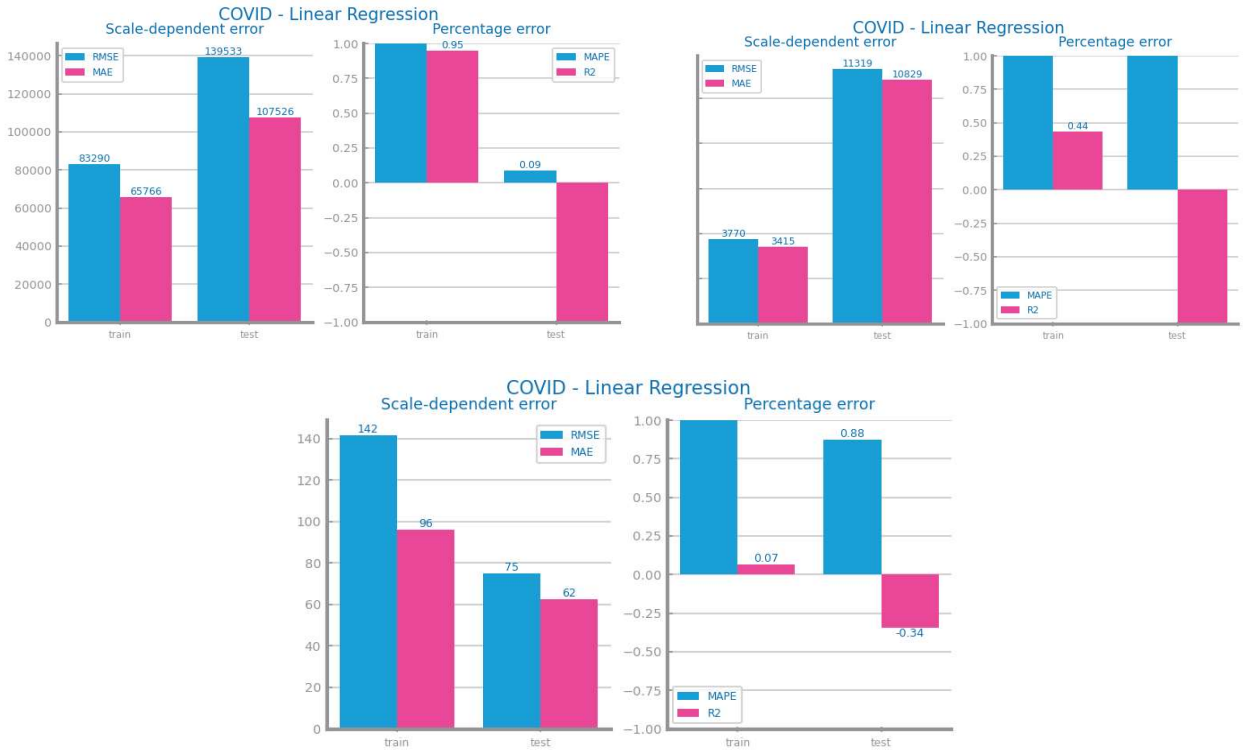


Figure 90 Forecasting plots after first and second differentiation of time series 2

Figure 91 Forecasting results after first and second differentiation of time series 2

3 Models' Evaluation

After applying the data transformation, we came to realize that for the **COVID dataset**, daily aggregation yields better results, with a window size of 50 for smoothing, and differentiation ultimately proved to be not beneficial. For the **credit score** analysis, the aggregation is done on a minute-to-minute basis, with a window size set at 25, and differentiation also does not yield improved results.

Simple Average Model

The test and test prediction lines are widely separated, indicating that this model is not suitable for our transformed dataset 1. And it also fails to capture the sasonality of the dataset 2.

Figure 96 Forecasting plots obtained with Simple Average model over time series 1

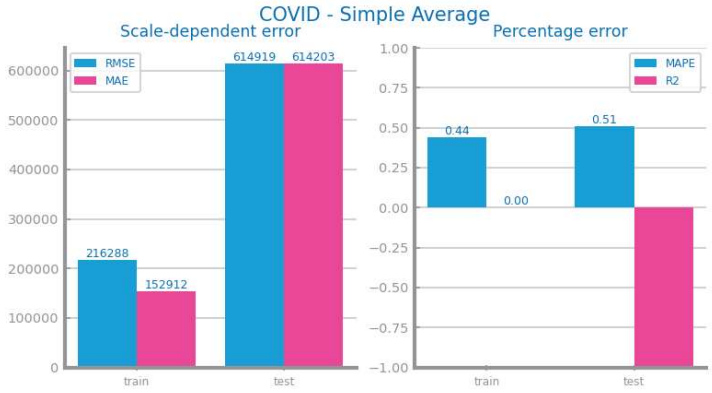


Figure 97 Forecasting results obtained with Simple Average model over time series 1

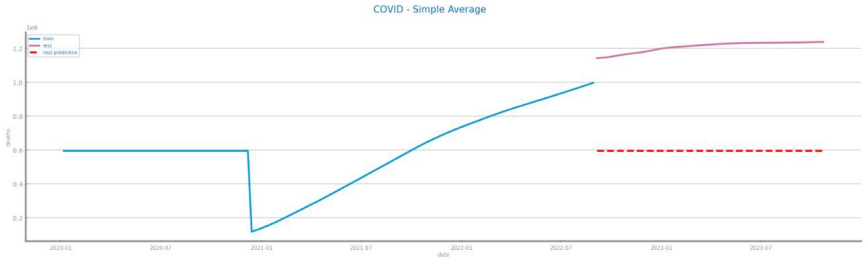


Figure 98 Forecasting plots obtained with Simple Average model over time series 2

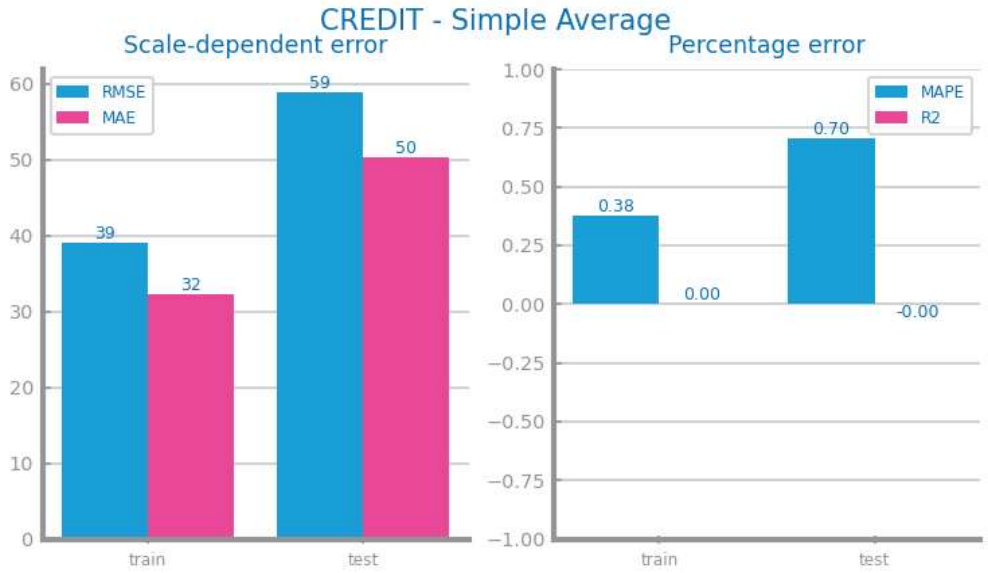
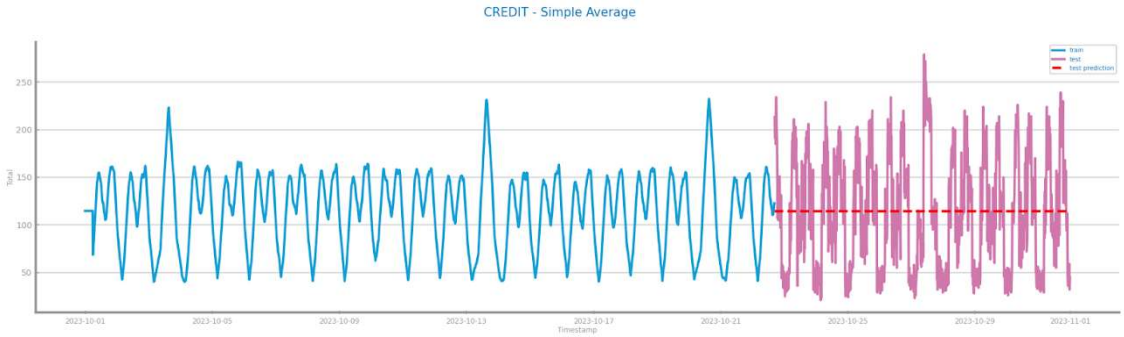


Figure 99 Forecasting results obtained with Simple Average model over time series 2



Persistence Model

The persistence model manages to achieve very good results in dataset 1, both with optimistic and realistic persistence, showing only a slight distance between the predicted and tested lines. Additionally, considering that the seasonal pattern in dataset 2 is not very complex and rather repetitive, the model also captures this seasonality quite well (but only when the persistence is optimistic) (however the error percentage is very high).

Figure 100 Forecasting plots obtained with Persistence model (long term) over time series 1

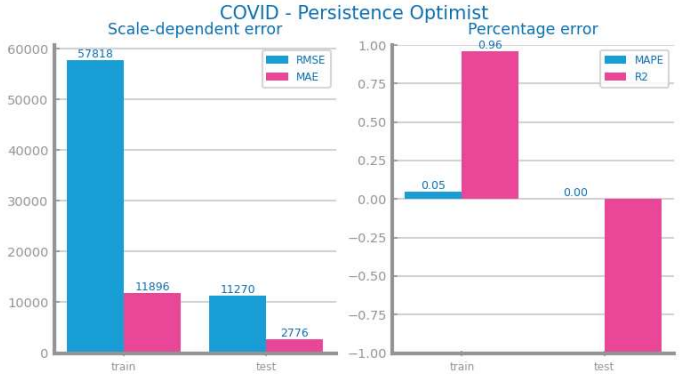


Figure 101 Forecasting plots obtained with Persistence model (next point) over time series 1

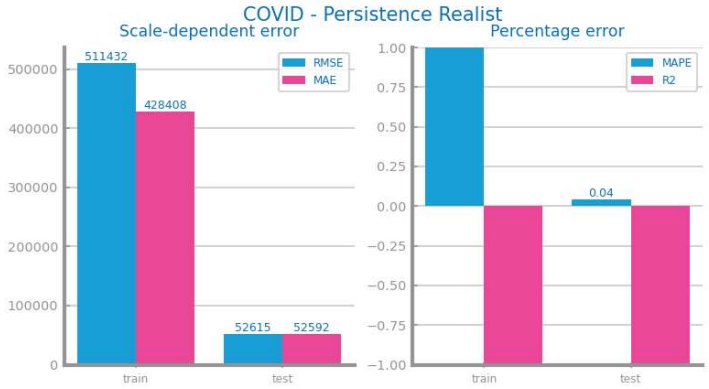


Figure 102 Forecasting results obtained with Persistence model in both situations over time series 1

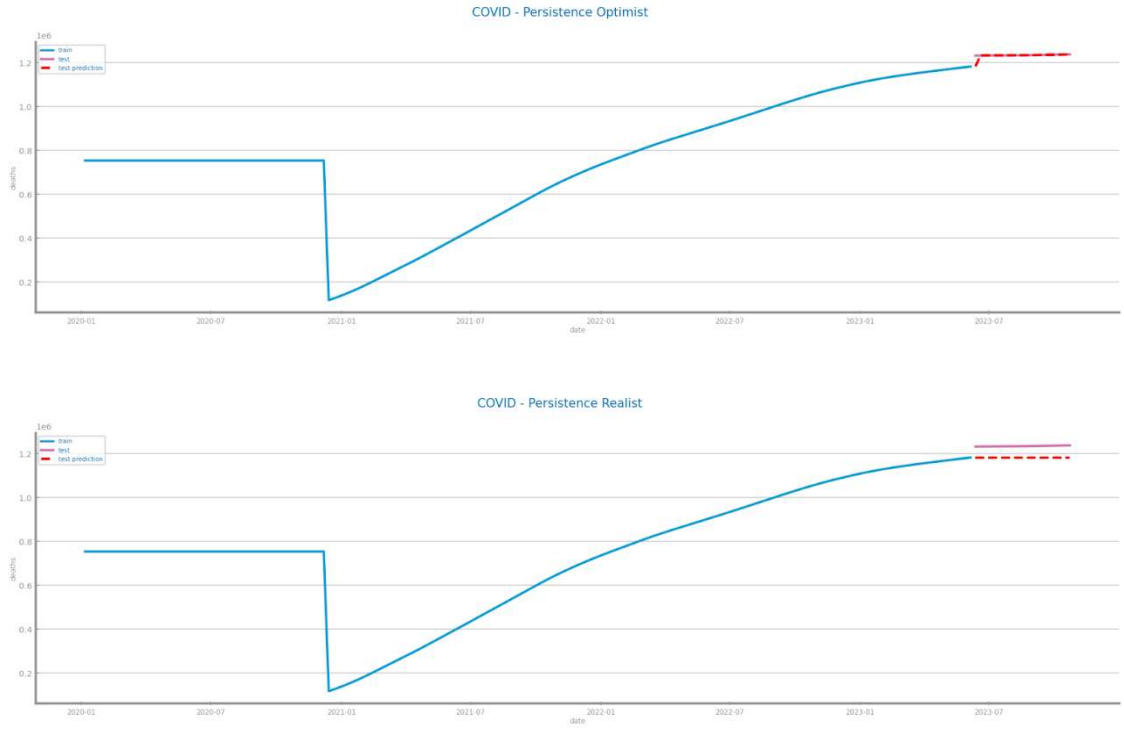
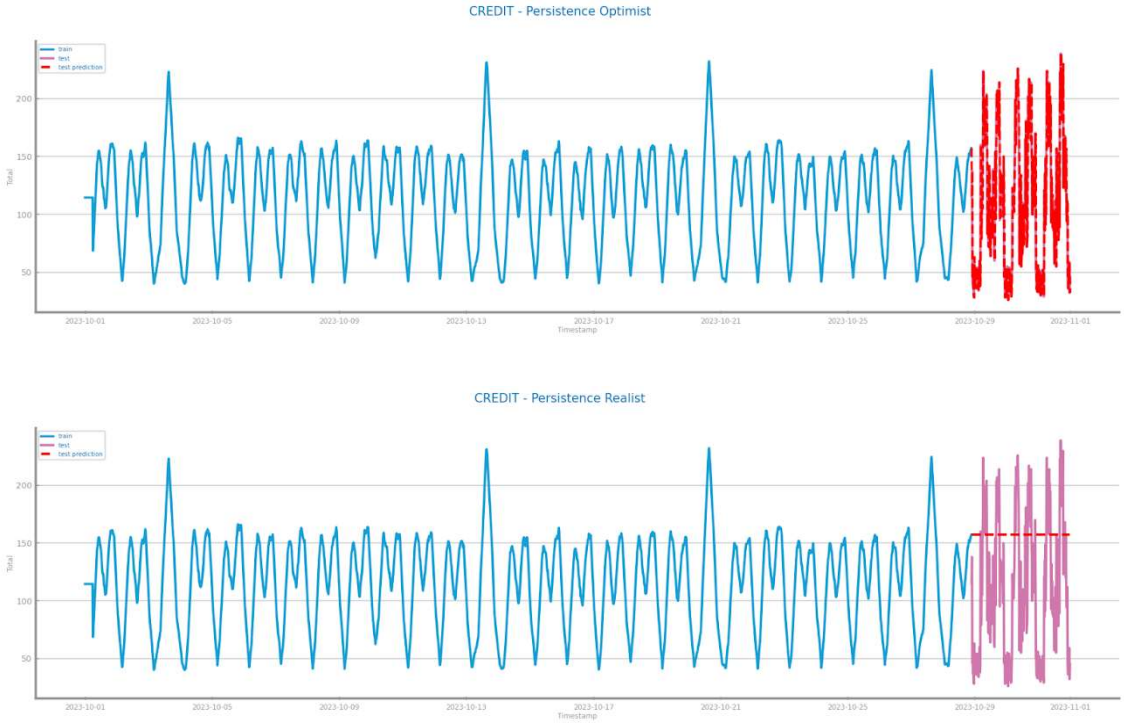


Figure 103 Forecasting plots obtained with Persistence model (long term) over time series 2

Figure 104 Forecasting plots obtained with Persistence model (next point) over time series 2

Figure 105 Forecasting results obtained with Persistence model in both situation over time series 2



Rolling Mean Model

For dataset 2, this approach ultimately didn't yield useful results as the rolling mean failed to capture the seasonality of our data, rendering it ineffective for this dataset. However, similar to the persistence model, it also achieved very good results close to the test data for dataset 1.

Figure 106 Forecasting study over different parameterisations of the rolling mean algorithm over time series 1

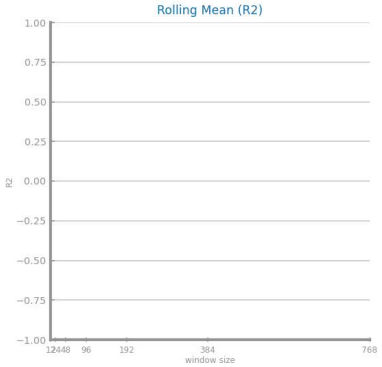


Figure 107 Forecasting plots obtained with the best parameterisation of rolling mean algorithm, over time series 1

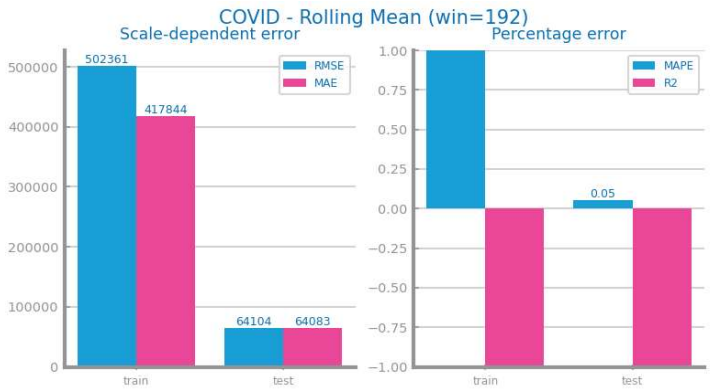


Figure 108 Forecasting results obtained with the best parameterisation of rolling mean algorithm, over time series 1

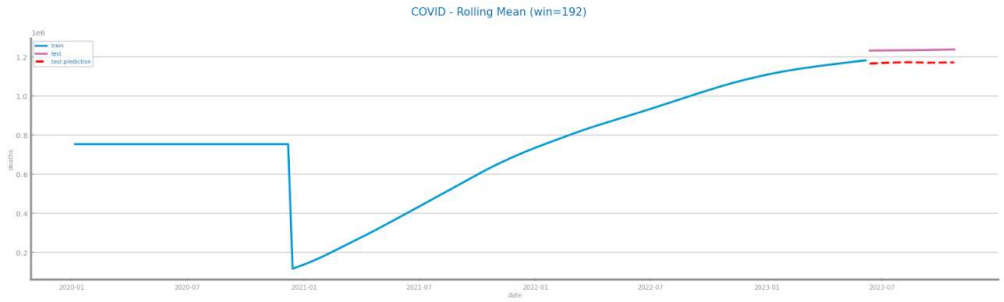


Figure 109 Forecasting study over different parameterisations of the rolling mean algorithm over time series 2

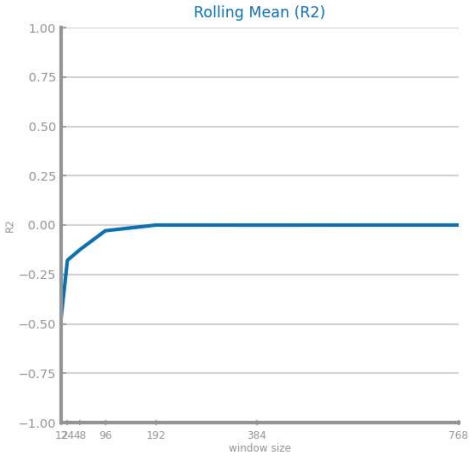


Figure 110 Forecasting plots obtained with the best parameterisation of rolling mean algorithm, over time series 2

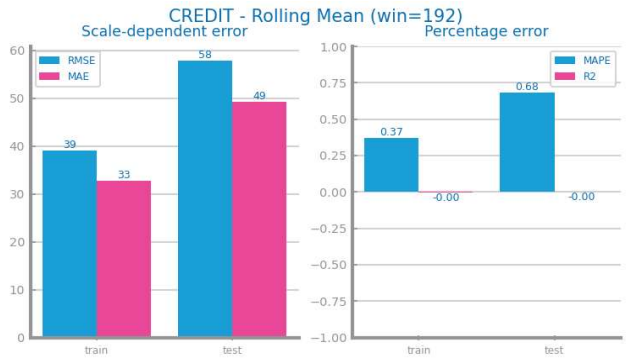
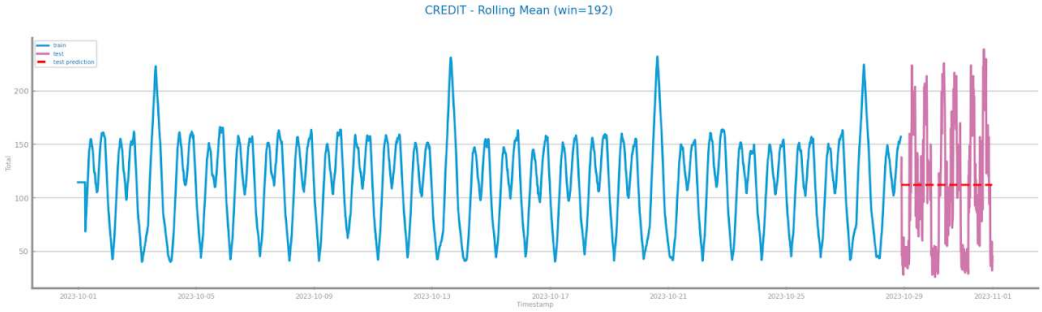


Figure 111 Forecasting results obtained with the best parameterisation of rolling mean algorithm, over time series 2



ARIMA Model

ARIMA assumes that the time series is stationary, which can be a limitation in some situations, including our case. Despite yielding results with a high percentage of error, ARIMA serves to detect patterns and seasonality, but unfortunately, it gradually loses information over time for dataset 2, resulting in weak outcomes. However, for dataset 1, despite producing results close to reality, it tends to capture something unintended, compromising its credibility slightly.

Figure 112 Forecasting study over different parameterisations of the ARIMA algorithm over time series 1

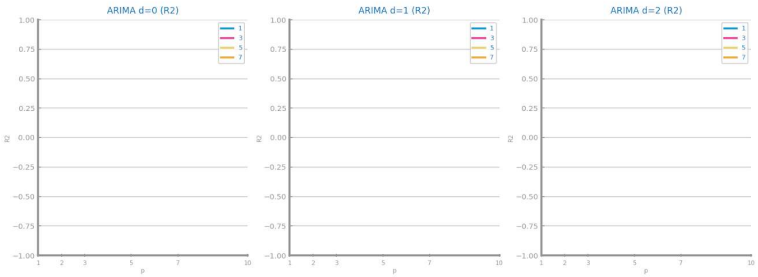


Figure 113 Forecasting plots obtained with the best parameterisation of ARIMA algorithm, over time series 1

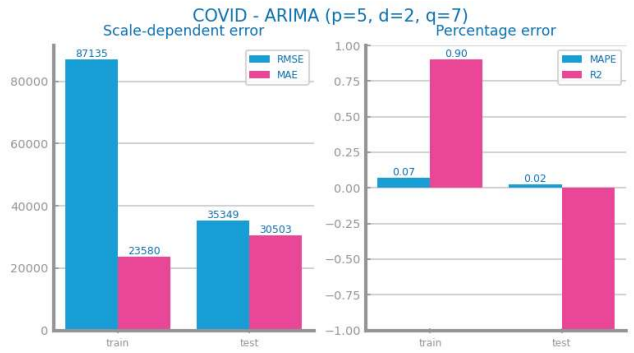


Figure 114 Forecasting results obtained with the best parameterisation of ARIMA algorithm, over time series 1

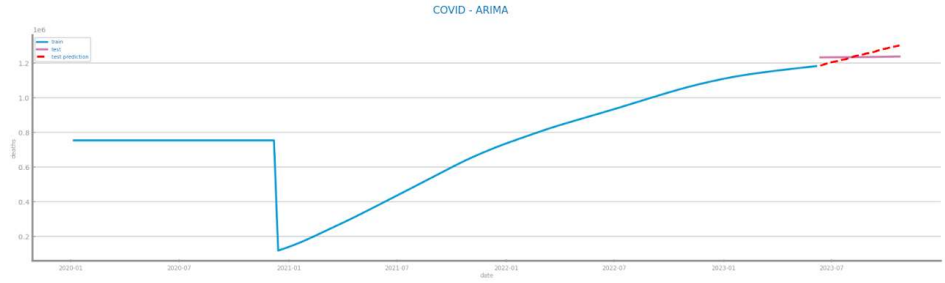


Figure 115 Forecasting study over different parameterisations of the ARIMA algorithm over time series 2

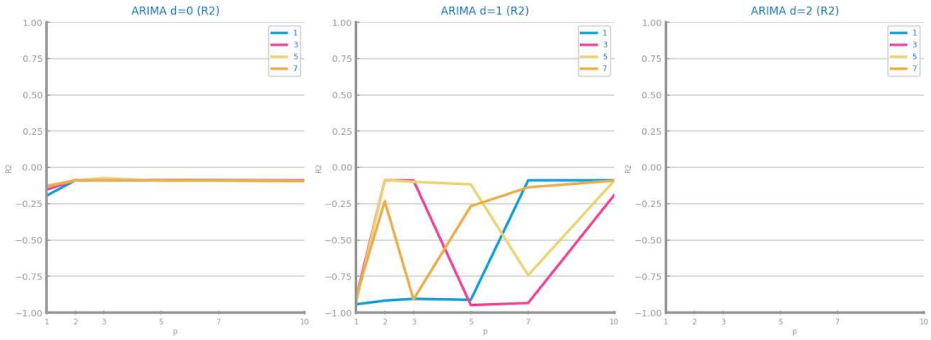


Figure 116 Forecasting plots obtained with the best parameterisation of ARIMA algorithm, over time series 2

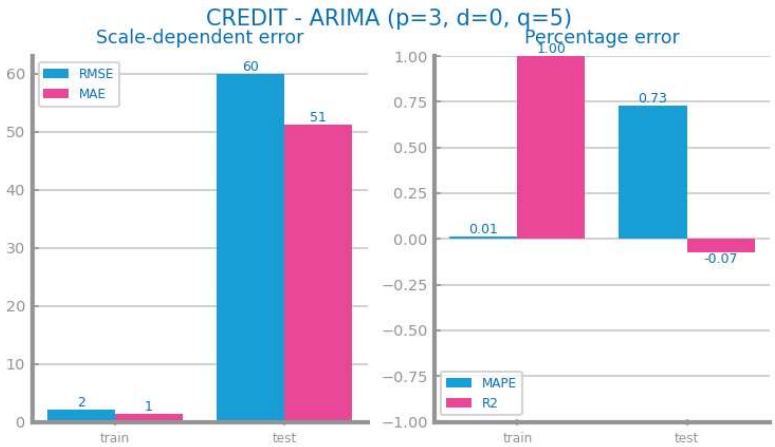
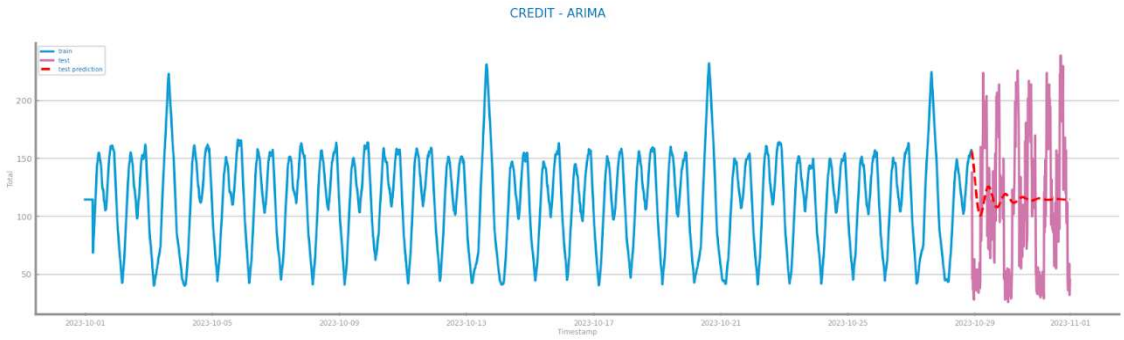


Figure 117 Forecasting results obtained with the best parameterisation of ARIMA algorithm, over time series 2



LSTMs Model

Due to some irregularity caused by the group, it was not possible to generate the results for dataset 1, hence the analysis is solely for dataset 2. This was the model with the best results, but it's important to note some issues. Testing this type of model is very expensive, and they can suffer from overfitting if not regularized properly, which seems to be the case in our results due to the significant difference between the training and testing errors.

Figure 121 Forecasting study over different parameterisations of the LSTMs over time series 2

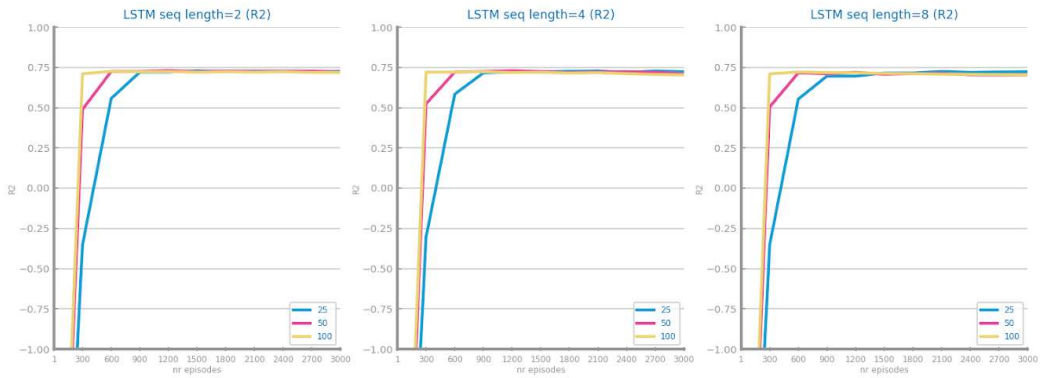


Figure 122 Forecasting plots obtained with the best parameterisation of LSTMs, over time series 2

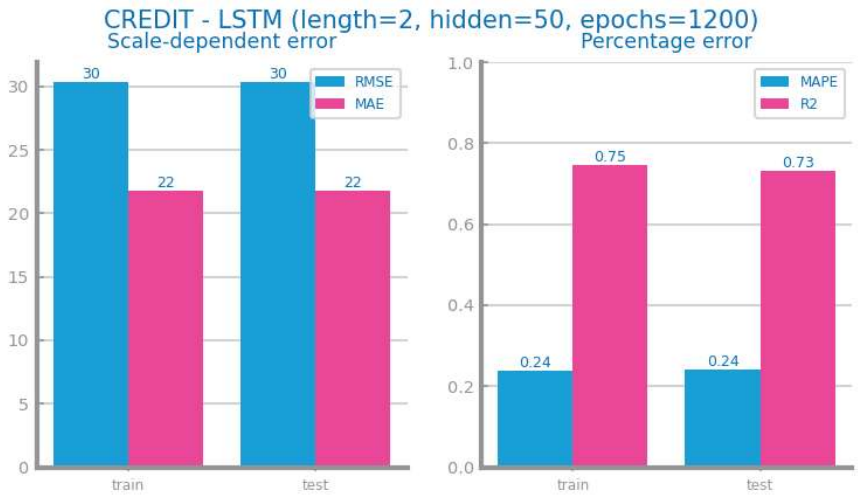
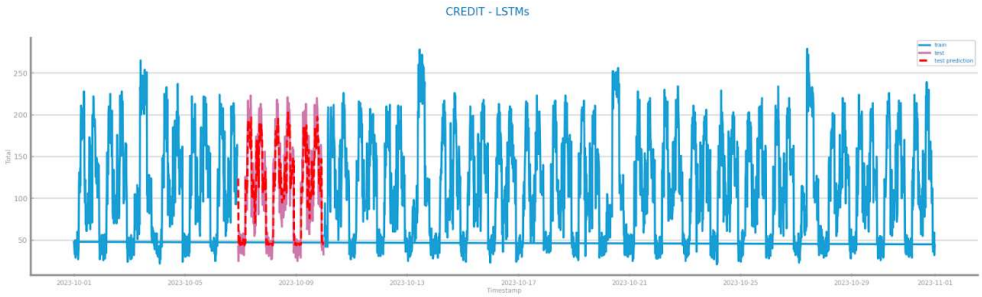


Figure 123 Forecasting results obtained with the best parameterisation of LSTMs, over time series 2



4 Critical Analysis

(dataset 1)

For dataset 1, some models proved dysfunctional. The LSTM model couldn't be successfully generated due to the insertion of values without significance in the original dataset. All training sets showed stability in the mean value. This stability was also a result of the insertion of initially insignificant values. The simple average model was highly insufficient due to this issue. Given that it's a dataset with few abrupt changes, the rolling mean with a very large window size ended up achieving very good results. Similarly, the persistence model worked well, probably due to the low noise in the dataset and minimal fluctuation. We can conclude that the results were in line with expectations overall, except for the simple average model. However, as mentioned earlier, this was likely caused by the insertion of values in the NA fields.

(dataset 2)

Dataset 2 appears to be more complex than dataset 1, but this doesn't necessarily mean it will be more difficult to find a fitting model. It will probably require a more complex model capable of capturing numerous oscillations within a short time interval, while also identifying the seasonality in the results. It's obvious that the simple average model isn't capable of handling such dynamics since it can't capture oscillations. The persistence model managed to capture the oscillation, which might indicate that the data is likely clean (i.e., with little noise). This is because if there were noise, the significant oscillations would result in highly disparate outcomes. Nonetheless, the error percentage was quite high, revealing the limitations of this model. The LSTMs yielded very good results even for a dataset with as much turbulence as this one, which is expected as these models are generally very powerful. However, it's important to note that these models are often overkill for the given task and can potentially have negative long-term impacts when dealing with chaotic situations (for example, the stock market), as they are prone to overfitting.