TÉCNICO LISBOA

DS 2024

Data Science Project

Team nr: 01	**Student 1:** Luis Miguel Lourenço da Cruz	**IST nr:** 110979
	Student 2: Ricardo de Jesus Vicente Tavares	**IST nr:** 113368

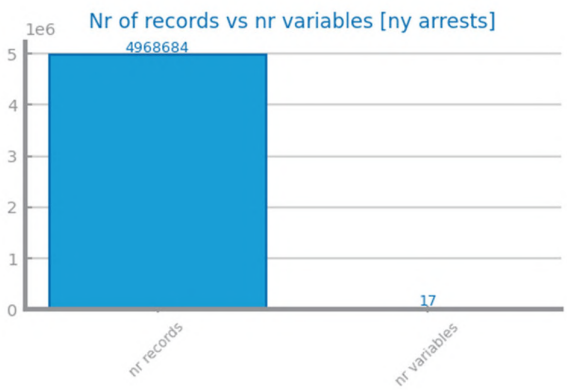
CLASSIFICATION

1 DATA PROFILING

Dataset 1 includes non-numeric variables, missing values, and hierarchical features, such as time and geographic coordinates, requiring encoding and preprocessing. It has a high record-to-variable ratio and imbalanced targets. Dataset 2, with fewer variables, has no missing values or hierarchical features but shows correlations among variables. Both datasets present outliers and target imbalances, demanding tailored preprocessing to address their unique challenges and ensure effective modeling.

Data Dimensionality

In Dataset 1, the variable ARREST_KEY, a unique key, was set as the “index_col” (read_csv). Dataset 1 has a very high records/variables ratio (292,276), while dataset 2's ratio is much lower (42.21). Only Dataset 1 contains non-numeric and non-binary variables, specifically 1 Date and 5 symbolic, requiring posterior extraction and encoding. While Dataset 2 has no missing values, Dataset 1 has 12 variables with missing values (NaN or Unknown), the highest corresponding to 0.97% of the dataset, which is low, so these items can be discarded, without significant impact.



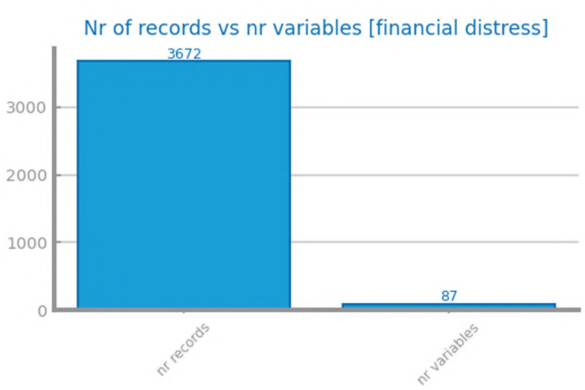


Figure 1 Nr Records x Nr variables for dataset 1 (left) and dataset 2 (right)

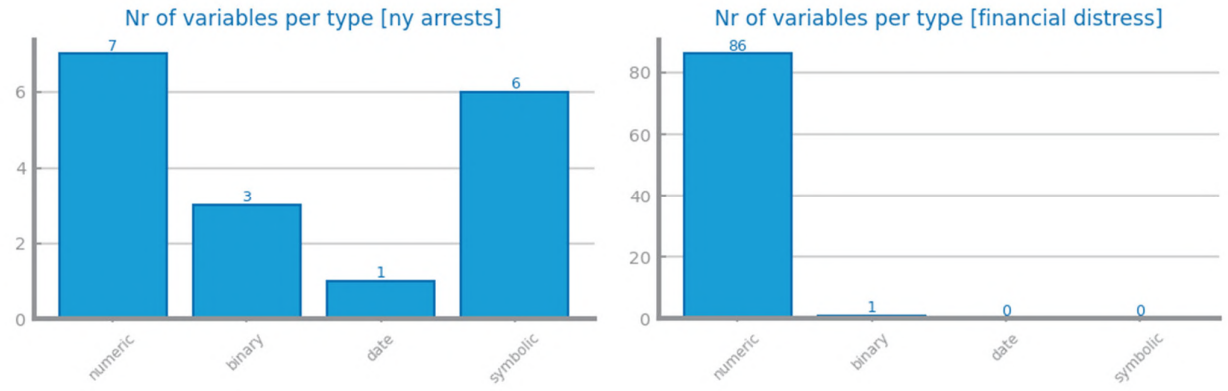


Figure 2 Nr variables per type for dataset 1 (left) and dataset 2 (right)

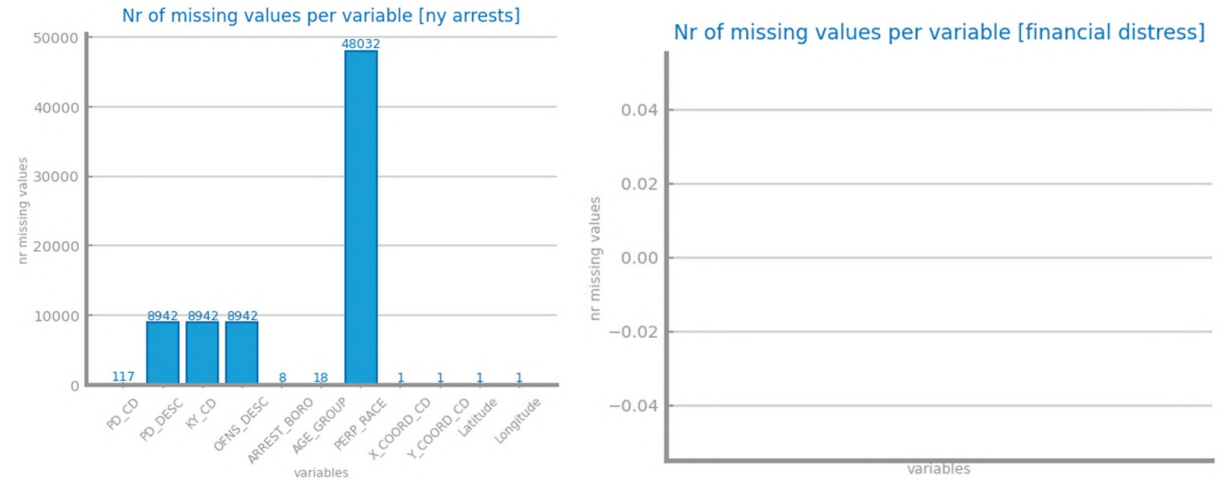


Figure 3 Nr missing values for dataset 1 (left) and dataset 2 (right)

Data Distribution

In Dataset 1, the binary target LAW_CAT_CD and the binary variable PERP_SEX were converted to numeric values ("M": 0, "F": 1 and "M": 1, "F": 0, respectively); the most frequent values are encoded as 1 (“male” and “misdemeanor,” respectively). Several numeric variables in Dataset 1 lack meaningful numerical interpretation. Outliers in both datasets were identified using NR_STDEV = 2 and IQR_FACTOR = 2. The targets LAW_CAT_CD in Dataset 1 and, especially, CLASS in Dataset 2 are imbalanced.

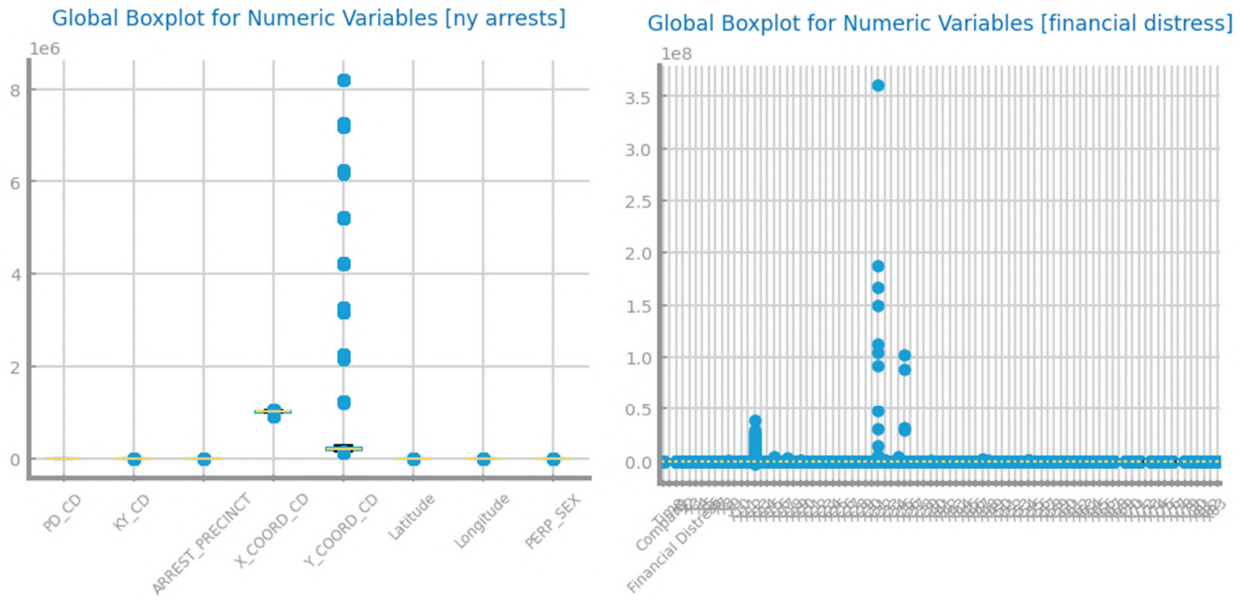
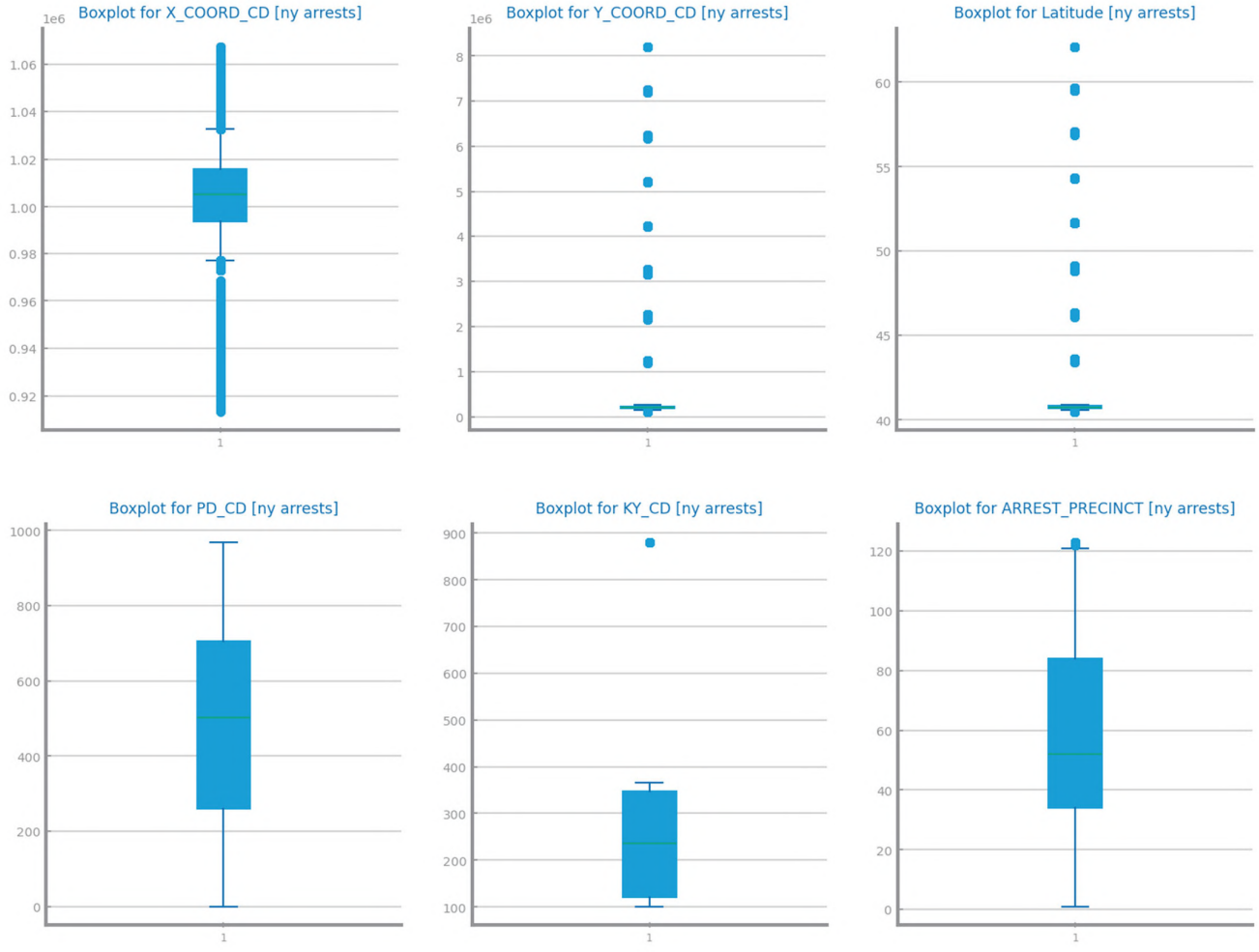


Figure 4 Global boxplots for dataset 1 (left) and dataset 2 (right)



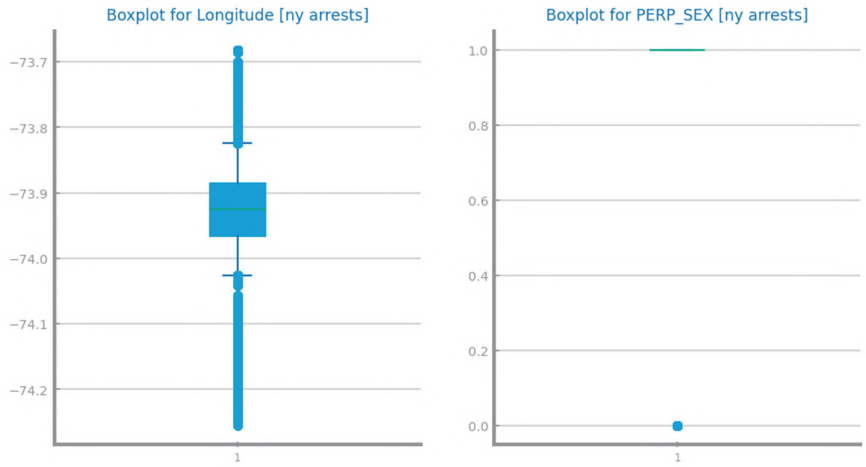
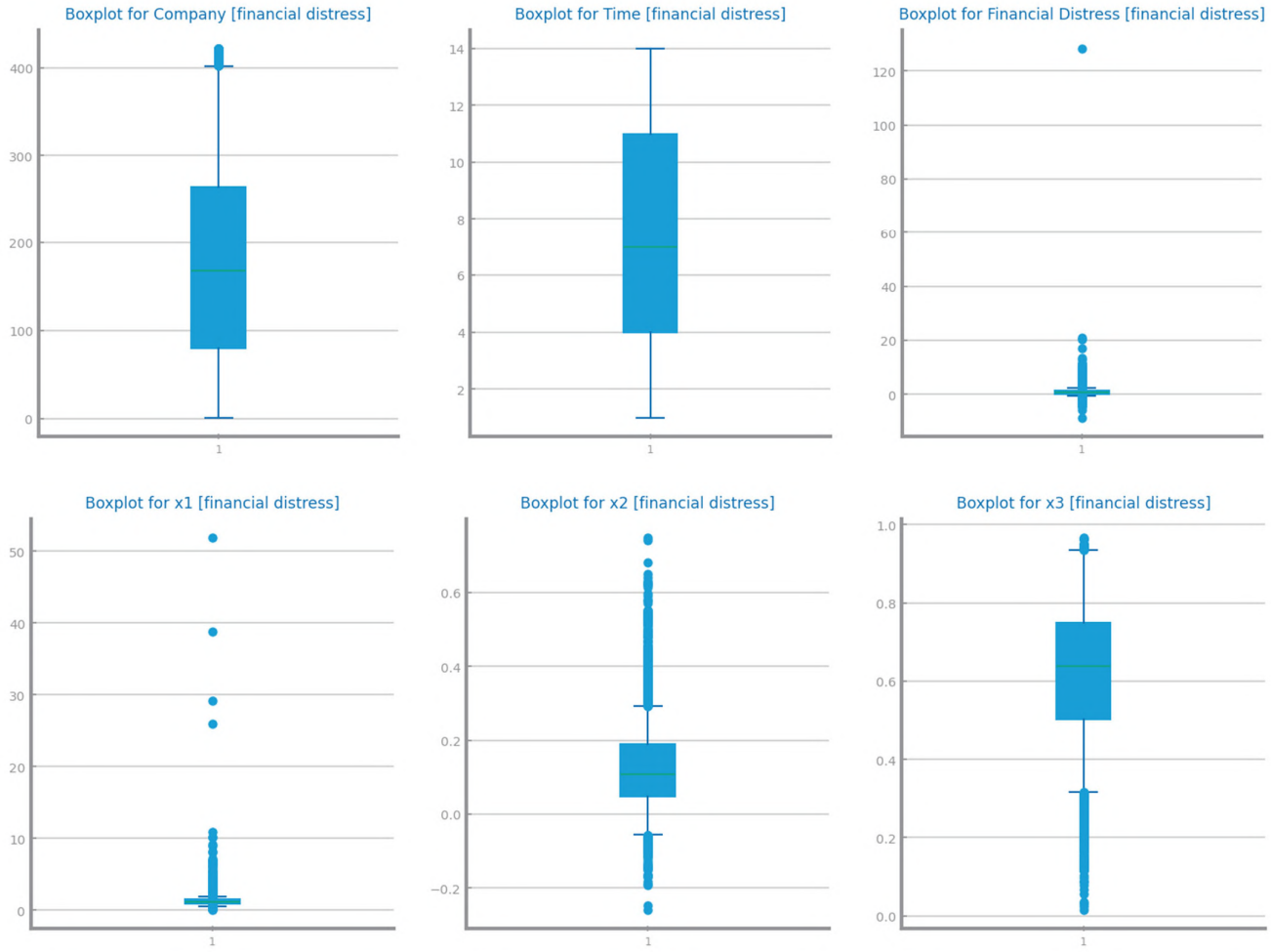
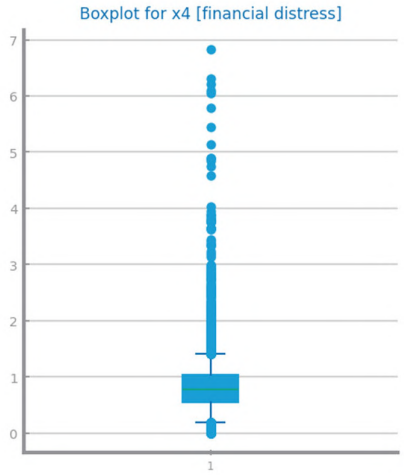
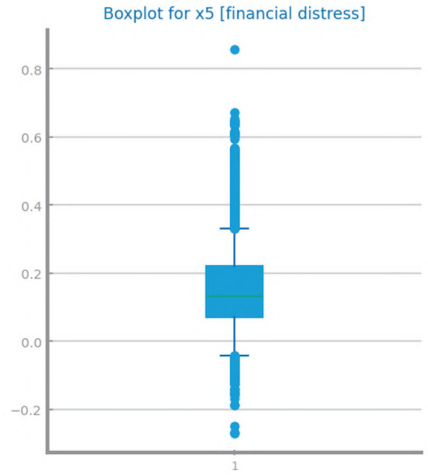
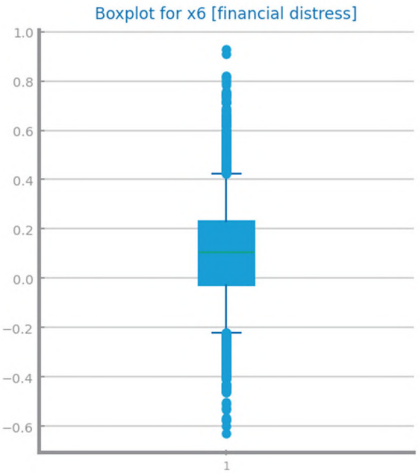


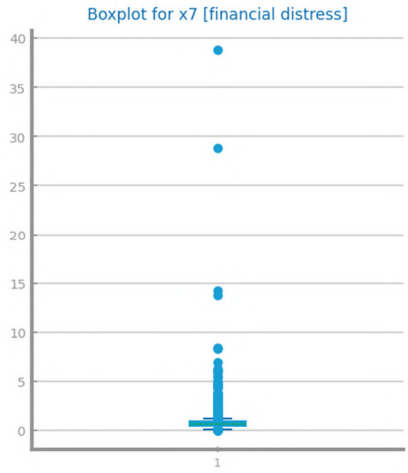
Figure 5 Single variable boxplots for dataset 1

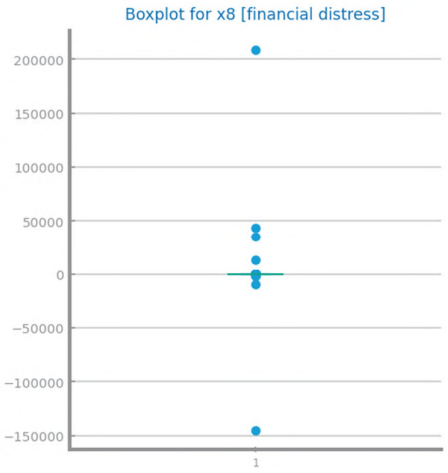


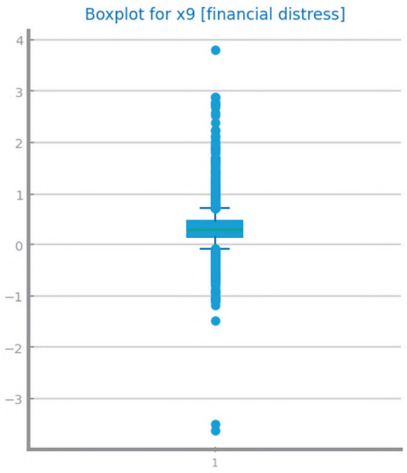

Boxplot for x4 [financial distress]
7
6
5
4
3
2
1
0
1

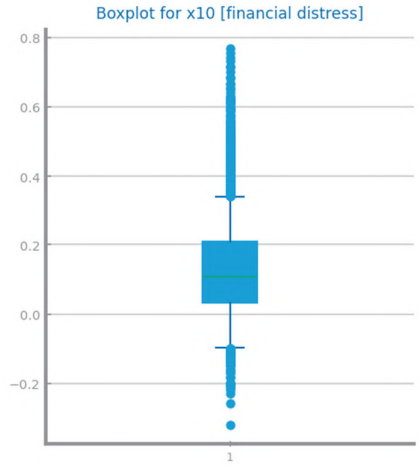

Boxplot for x5 [financial distress]
0.8
0.6
0.4
0.2
0.0
−0.2
1

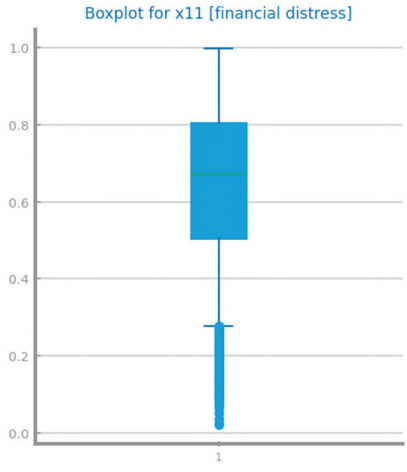

Boxplot for x6 [financial distress]
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
1

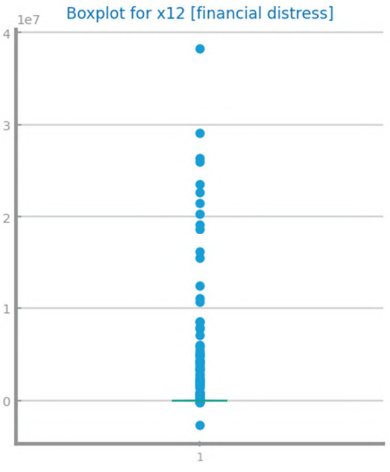

Boxplot for x7 [financial distress]
40
35
30
25
20
15
10
5
0
1

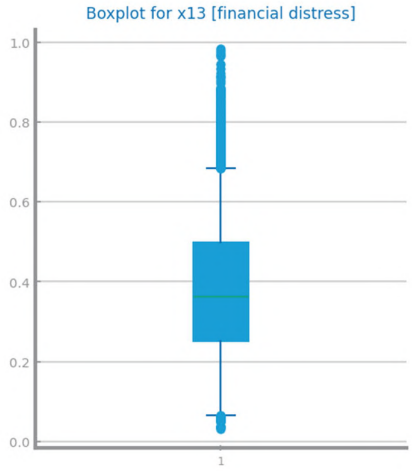

Boxplot for x8 [financial distress]
200000
150000
100000
50000
0
−50000
−100000
−150000
1

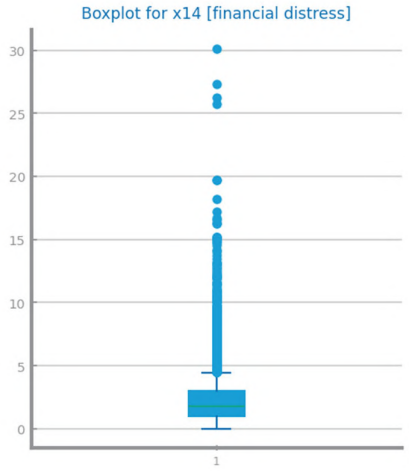

Boxplot for x9 [financial distress]
4
3
2
1
0
−1
−2
−3
1

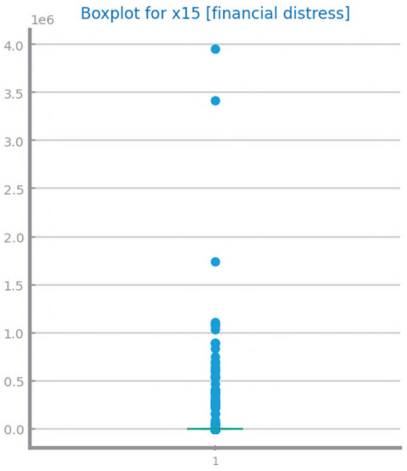

Boxplot for x10 [financial distress]
0.8
0.6
0.4
0.2
0.0
−0.2
1

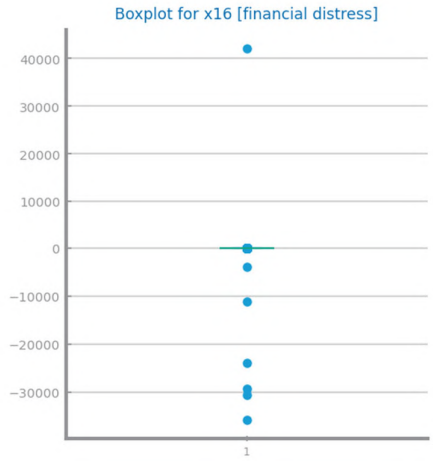

Boxplot for x11 [financial distress]
1.0
0.8
0.6
0.4
0.2
0.0
1

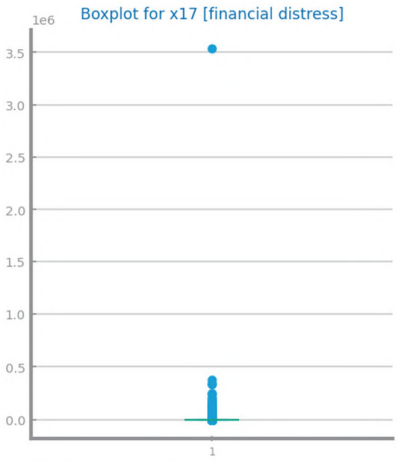

Boxplot for x12 [financial distress]
1e7
4
3
2
1
0
1

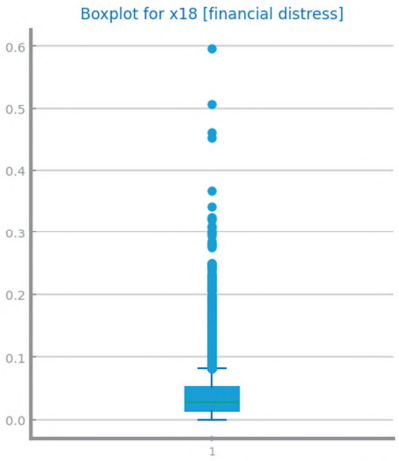

Boxplot for x13 [financial distress]
1.0
0.8
0.6
0.4
0.2
0.0
1

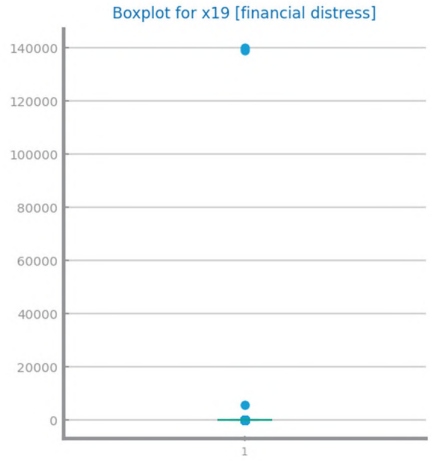

Boxplot for x14 [financial distress]
30
25
20
15
10
5
0
1

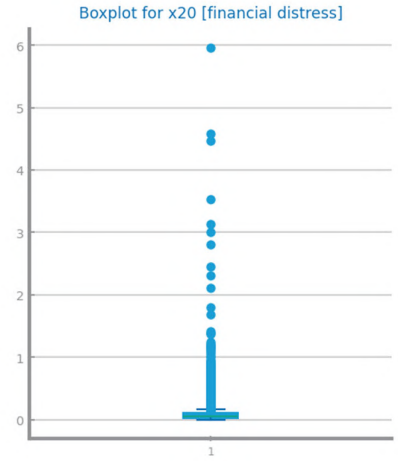

Boxplot for x15 [financial distress]
1e6
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
1

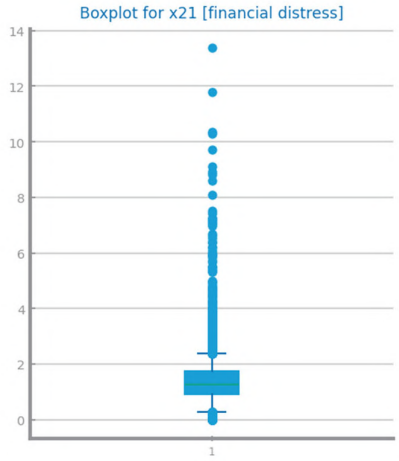

Boxplot for x16 [financial distress]
40000
30000
20000
10000
0
−10000
−20000
−30000
1

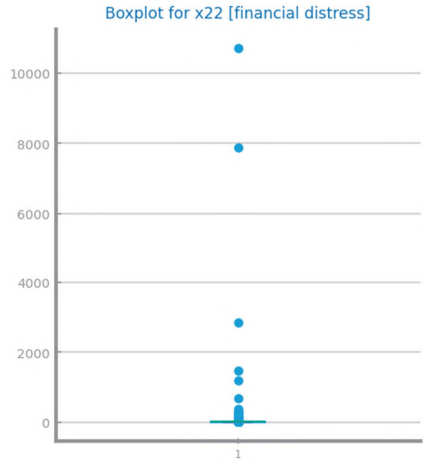

Boxplot for x17 [financial distress]
1e6
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
1

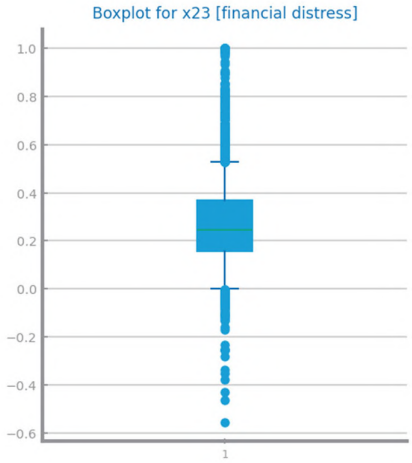

Boxplot for x18 [financial distress]
0.6
0.5
0.4
0.3
0.2
0.1
0.0
1

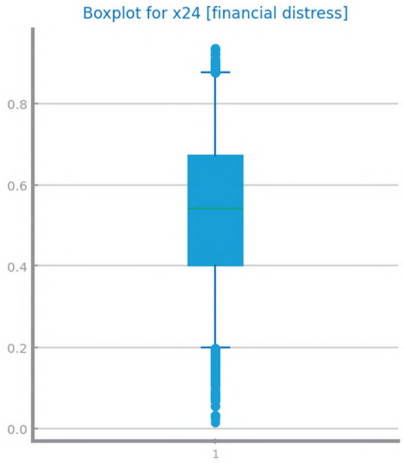

Boxplot for x19 [financial distress]
140000
120000
100000
80000
60000
40000
20000
0
1

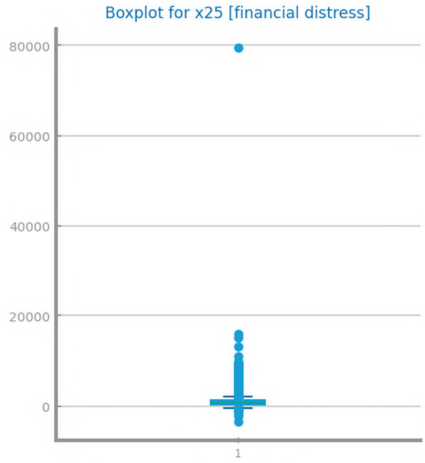

Boxplot for x20 [financial distress]
6
5
4
3
2
1
0
1

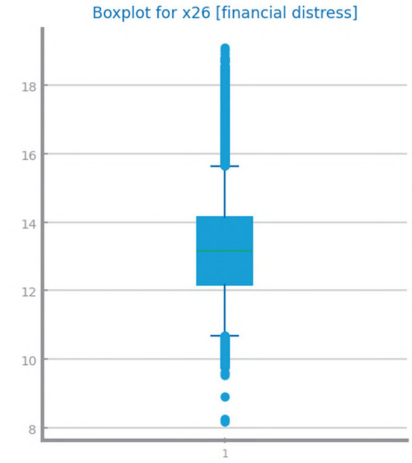

Boxplot for x21 [financial distress]
14
12
10
8
6
4
2
0
1

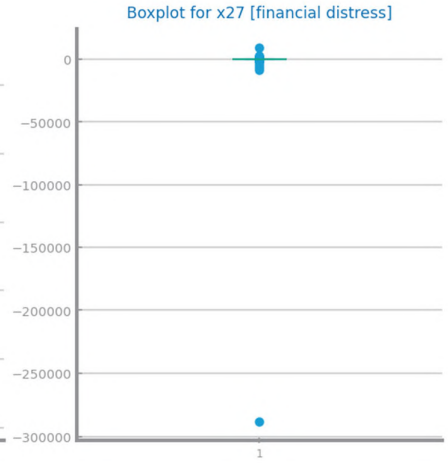

Boxplot for x22 [financial distress]
10000
8000
6000
4000
2000
0
1

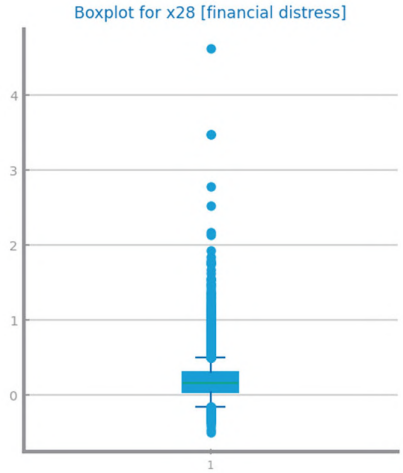

Boxplot for x23 [financial distress]
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
1

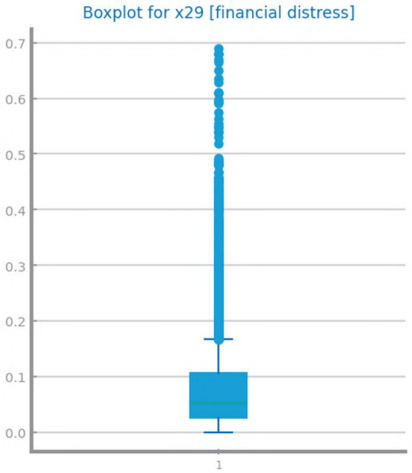

Boxplot for x24 [financial distress]
0.8
0.6
0.4
0.2
0.0
1

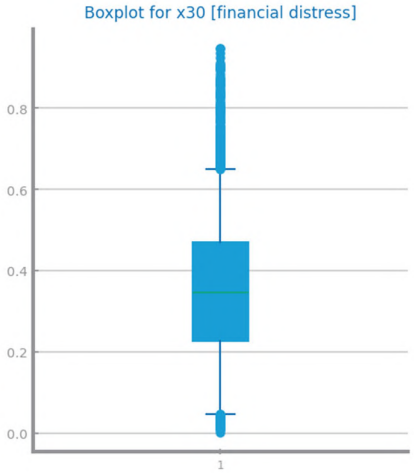

Boxplot for x25 [financial distress]
80000
60000
40000
20000
0
1

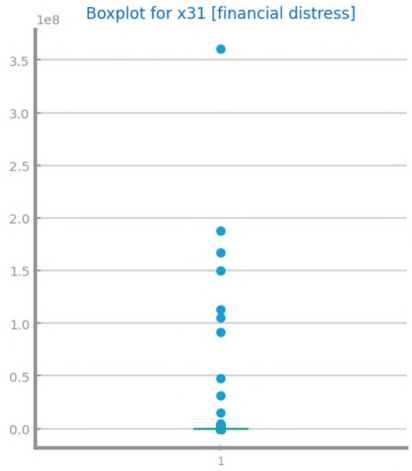

Boxplot for x26 [financial distress]
18
16
14
12
10
8
1

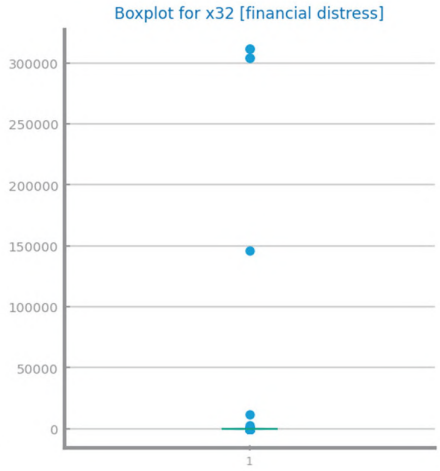

Boxplot for x27 [financial distress]
0
−50000
−100000
−150000
−200000
−250000
−300000
1

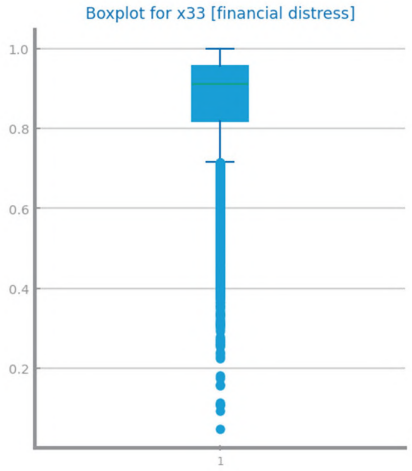

Boxplot for x28 [financial distress]
4
3
2
1
0
1

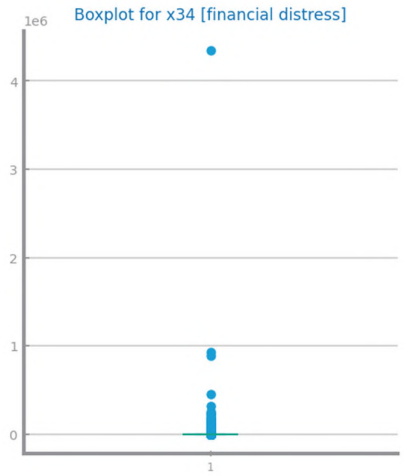

Boxplot for x29 [financial distress]
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
1

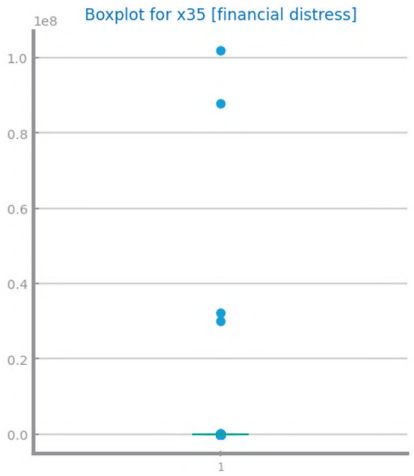

Boxplot for x30 [financial distress]
0.8
0.6
0.4
0.2
0.0
1

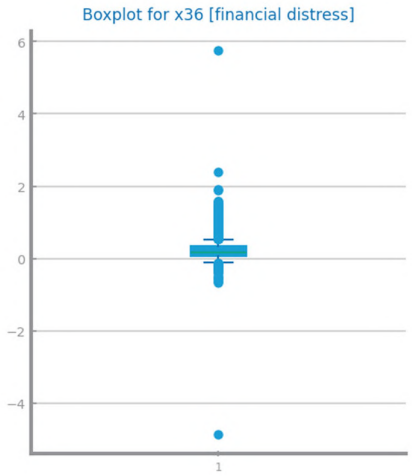

Boxplot for x31 [financial distress]
1e8
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
1

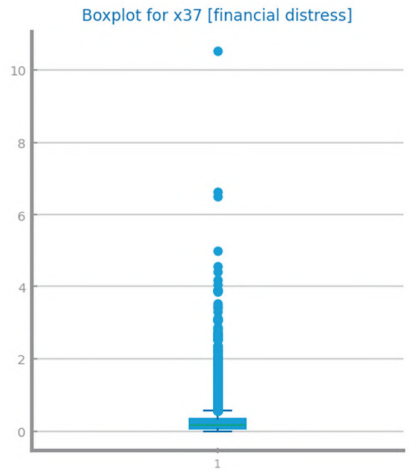

Boxplot for x32 [financial distress]
300000
250000
200000
150000
100000
50000
0
1

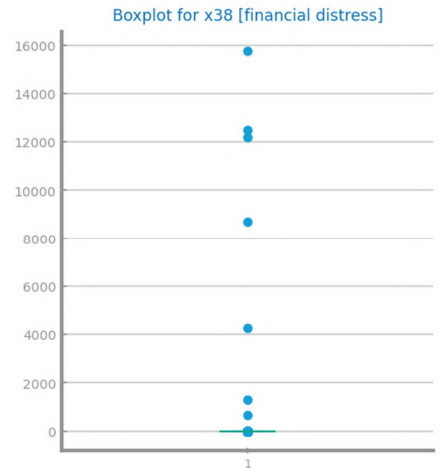

Boxplot for x33 [financial distress]
1.0
0.8
0.6
0.4
0.2
1

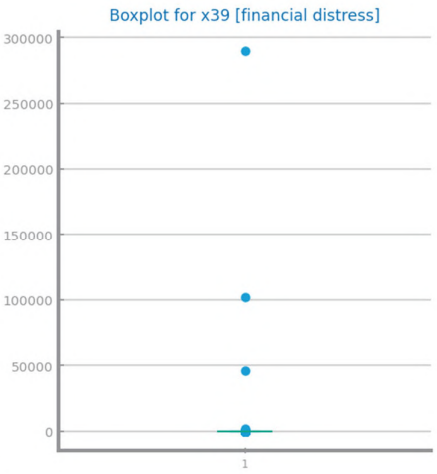

Boxplot for x34 [financial distress]
1e6
4
3
2
1
0
1

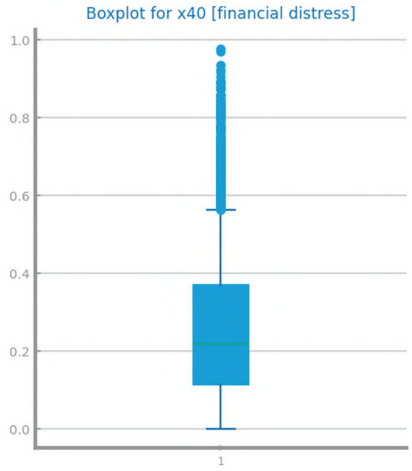

Boxplot for x35 [financial distress]
1e8
1.0
0.8
0.6
0.4
0.2
0.0
1

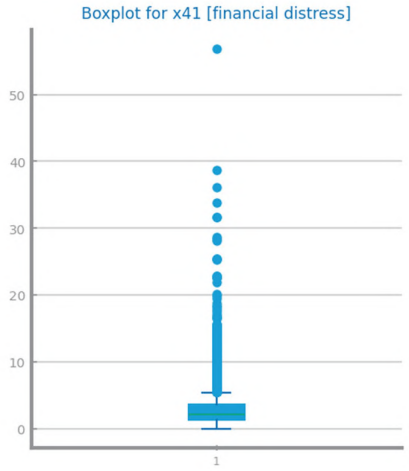

Boxplot for x36 [financial distress]
6
4
2
0
−2
−4
1

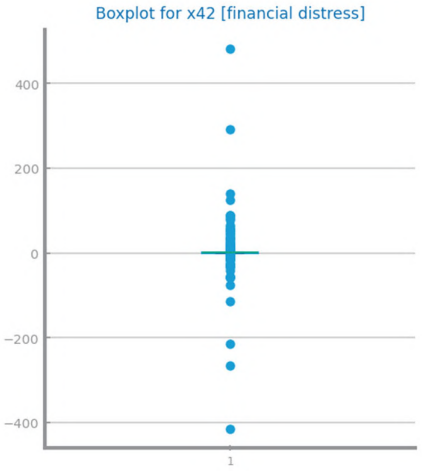

Boxplot for x37 [financial distress]
10
8
6
4
2
0
1

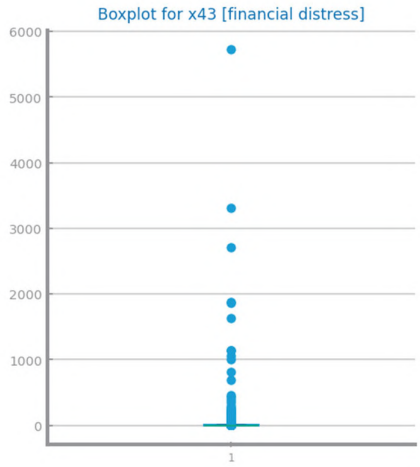

Boxplot for x38 [financial distress]
16000
14000
12000
10000
8000
6000
4000
2000
0
1

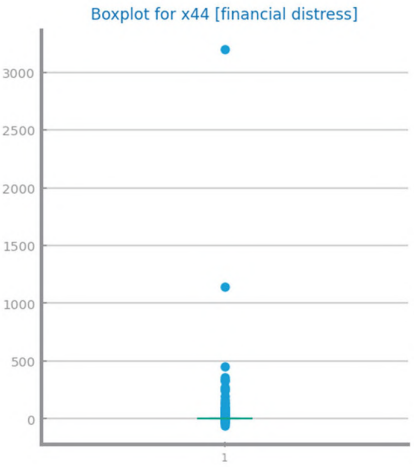

Boxplot for x39 [financial distress]
300000
250000
200000
150000
100000
50000
0
1

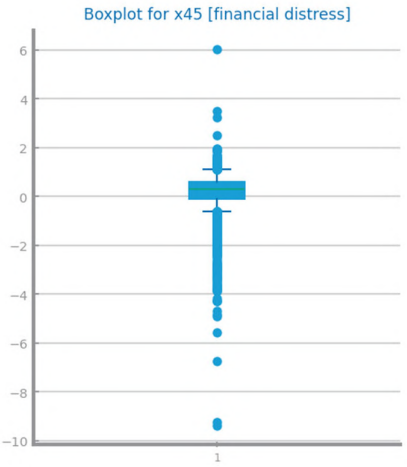

Boxplot for x40 [financial distress]
1.0
0.8
0.6
0.4
0.2
0.0
1

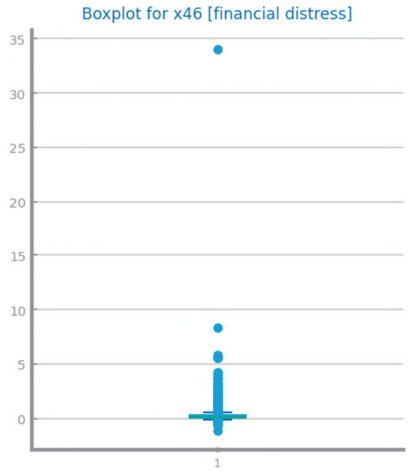

Boxplot for x41 [financial distress]
50
40
30
20
10
0
1

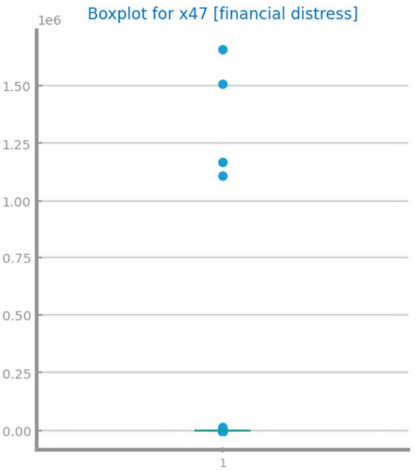

Boxplot for x42 [financial distress]
400
200
0
−200
−400
1

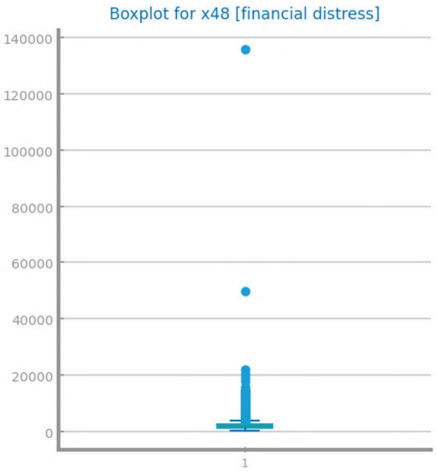

Boxplot for x43 [financial distress]
6000
5000
4000
3000
2000
1000
0
1

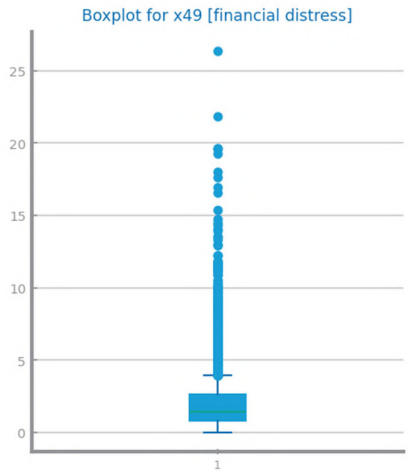

Boxplot for x44 [financial distress]
3000
2500
2000
1500
1000
500
0
1

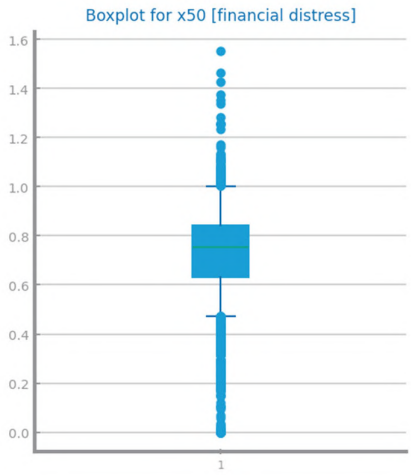

Boxplot for x45 [financial distress]
6
4
2
0
−2
−4
−6
−8
−10
1

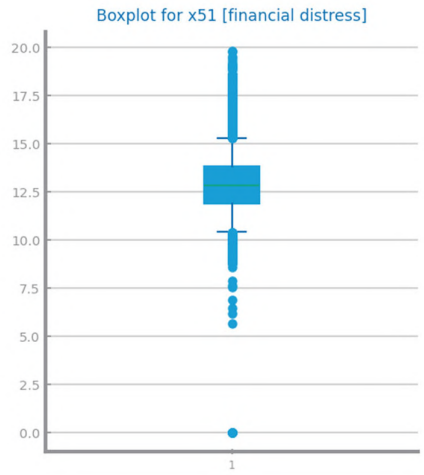

Boxplot for x46 [financial distress]
35
30
25
20
15
10
5
0
1

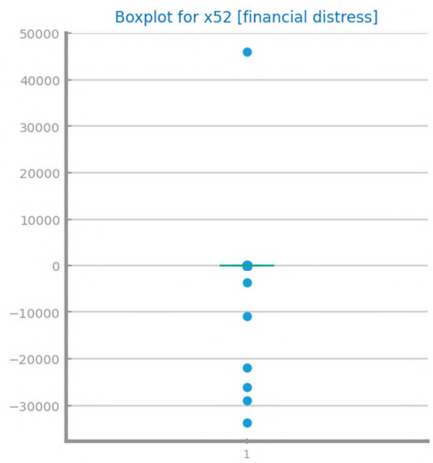

Boxplot for x47 [financial distress]
1e6
1.50
1.25
1.00
0.75
0.50
0.25
0.00
1

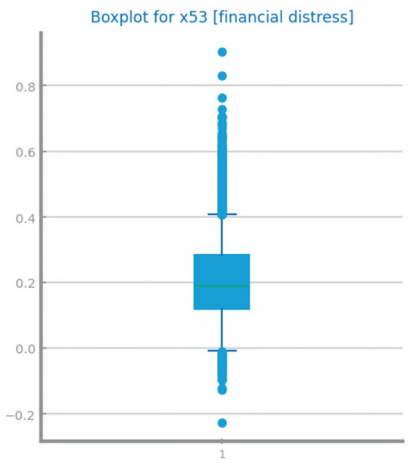

Boxplot for x48 [financial distress]
140000
120000
100000
80000
60000
40000
20000
0
1

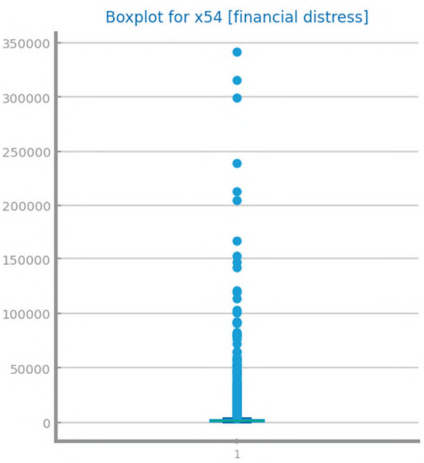

Boxplot for x49 [financial distress]
25
20
15
10
5
0
1

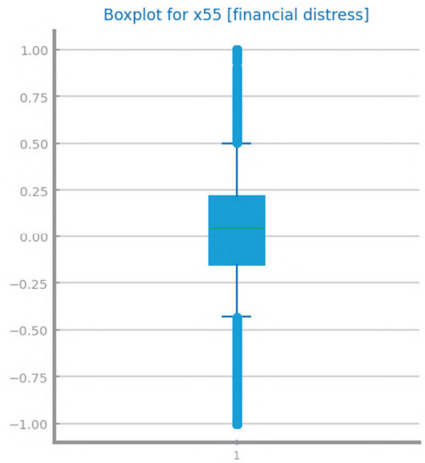

Boxplot for x50 [financial distress]
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
1

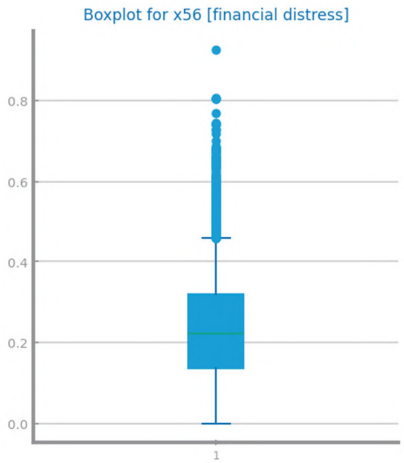

Boxplot for x51 [financial distress]
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
1

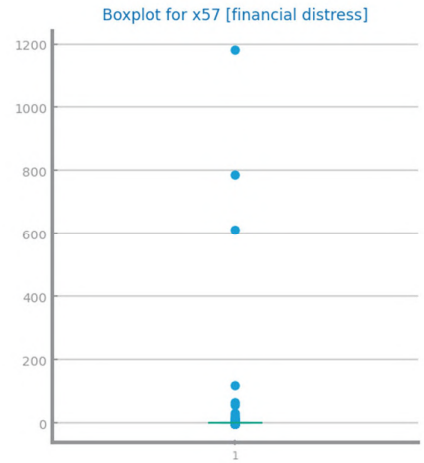

Boxplot for x52 [financial distress]
50000
40000
30000
20000
10000
0
−10000
−20000
−30000
1

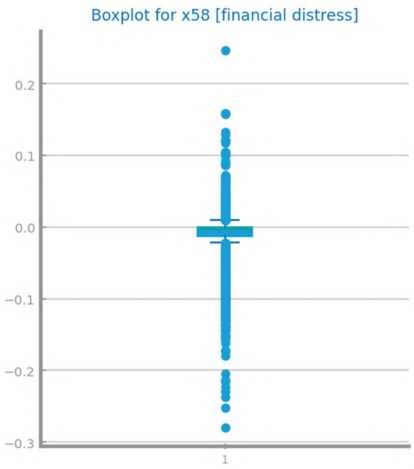

Boxplot for x53 [financial distress]
0.8
0.6
0.4
0.2
0.0
−0.2
1

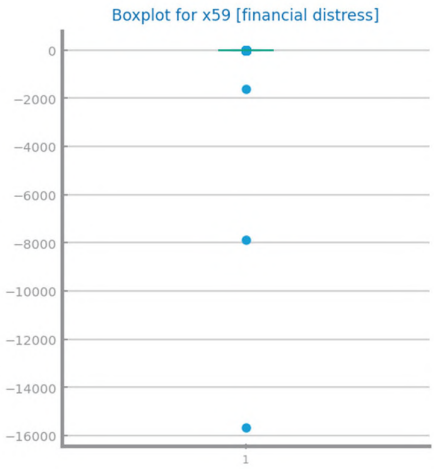

Boxplot for x54 [financial distress]
350000
300000
250000
200000
150000
100000
50000
0
1

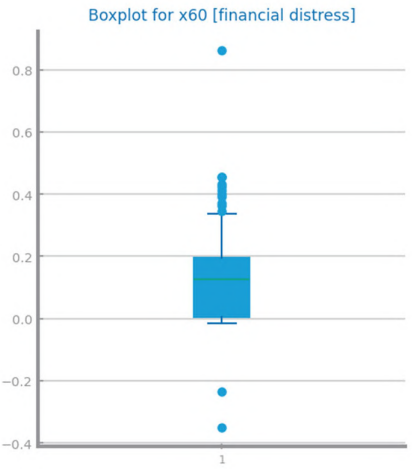

Boxplot for x55 [financial distress]
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
1

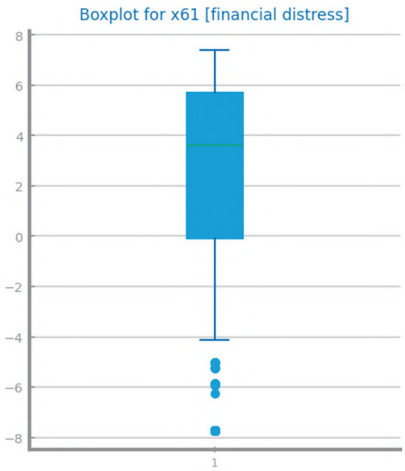

Boxplot for x56 [financial distress]
0.8
0.6
0.4
0.2
0.0
1

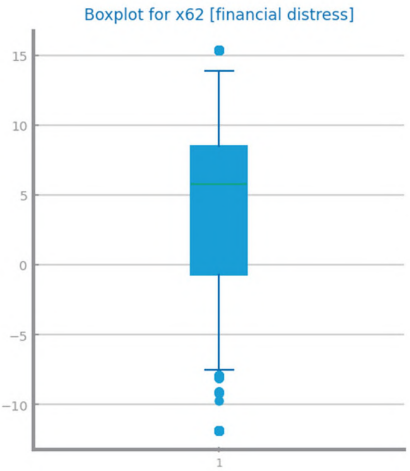

Boxplot for x57 [financial distress]
1200
1000
800
600
400
200
0
1

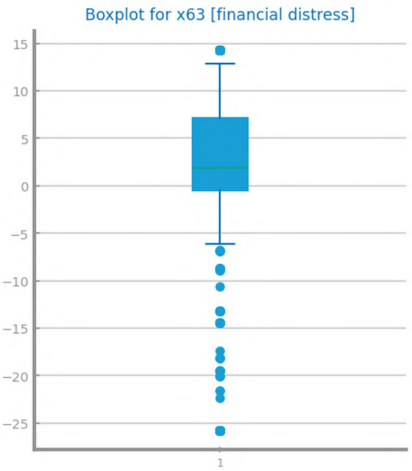

Boxplot for x58 [financial distress]
0.2
0.1
0.0
−0.1
−0.2
−0.3
1

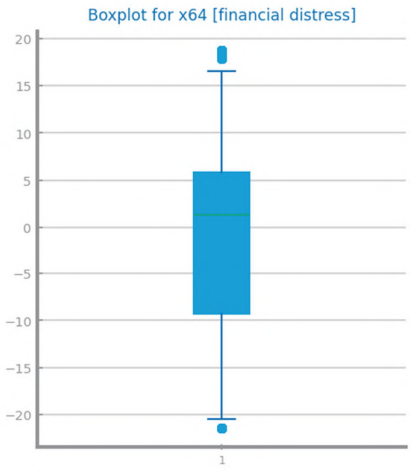

Boxplot for x59 [financial distress]
0
−2000
−4000
−6000
−8000
−10000
−12000
−14000
−16000
1

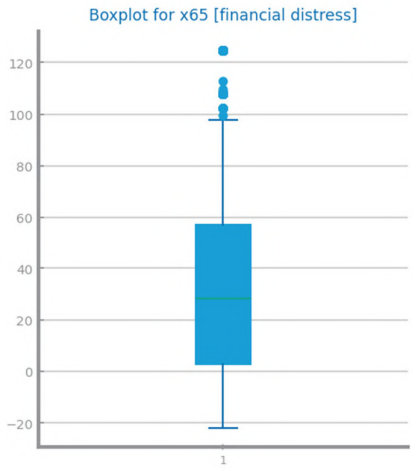

Boxplot for x60 [financial distress]
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
1

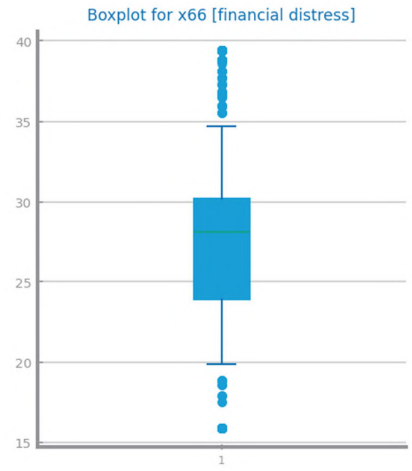

Boxplot for x61 [financial distress]
8
6
4
2
0
−2
−4
−6
−8
1

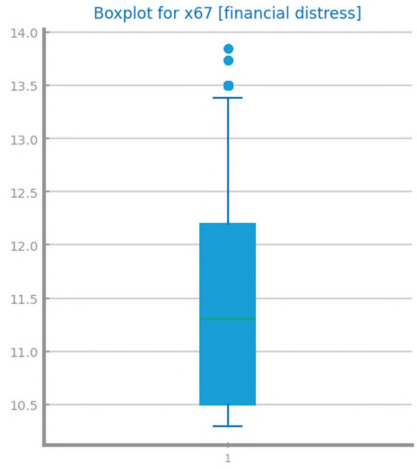

Boxplot for x62 [financial distress]
15
10
5
0
−5
−10
1

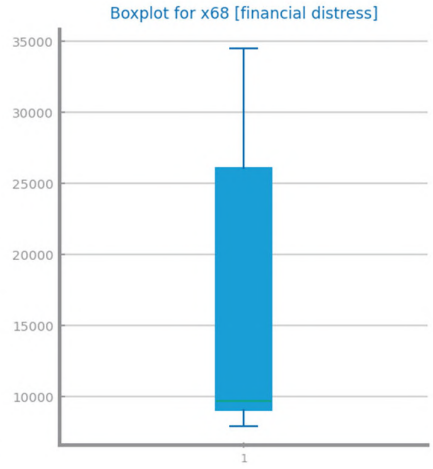

Boxplot for x63 [financial distress]
15
10
5
0
−5
−10
−15
−20
−25
1

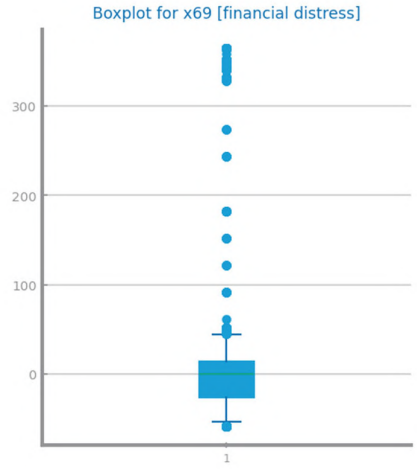

Boxplot for x64 [financial distress]
20
15
10
5
0
−5
−10
−15
−20
1

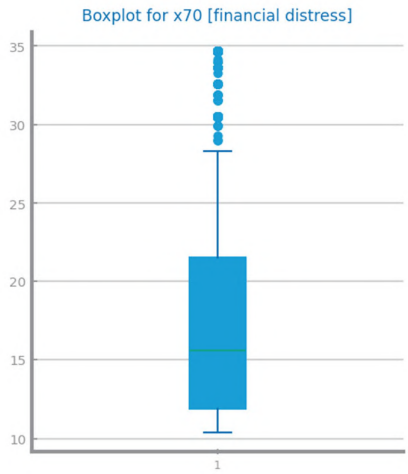

Boxplot for x65 [financial distress]
120
100
80
60
40
20
0
−20
1

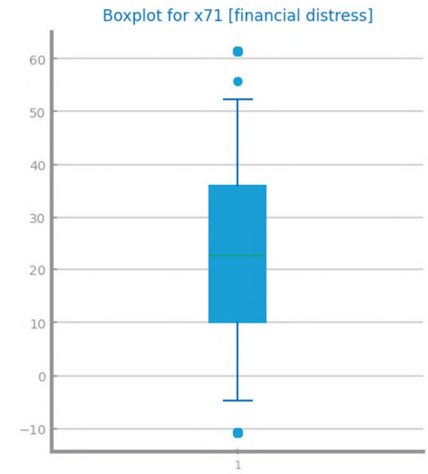

Boxplot for x66 [financial distress]
40
35
30
25
20
15
1

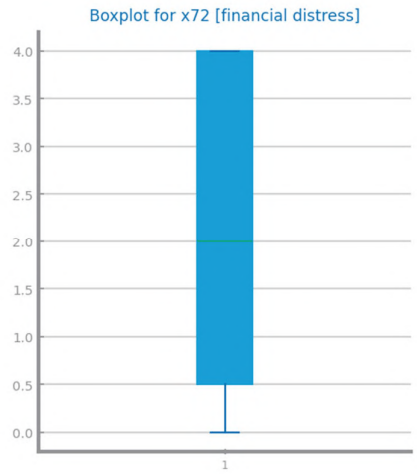

Boxplot for x67 [financial distress]
14.0
13.5
13.0
12.5
12.0
11.5
11.0
10.5
1

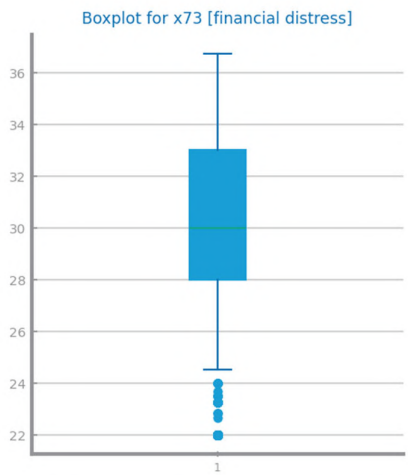

Boxplot for x68 [financial distress]
35000
30000
25000
20000
15000
10000
1

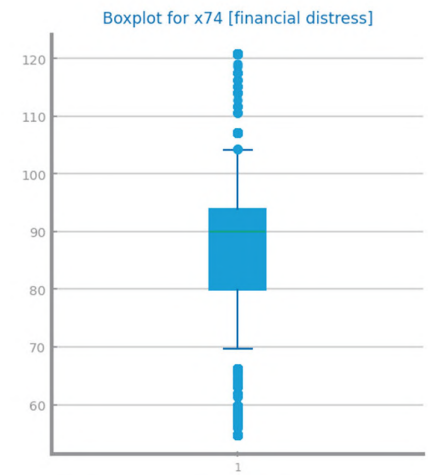

Boxplot for x69 [financial distress]
300
200
100
0
1

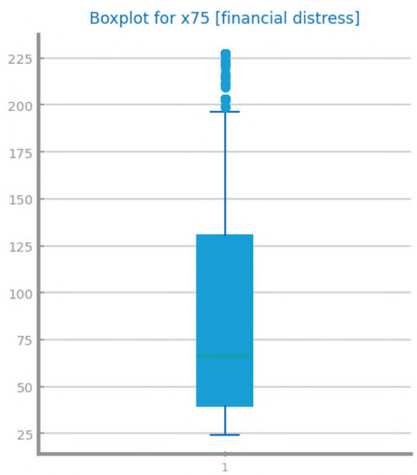

Boxplot for x70 [financial distress]
35
30
25
20
15
10
1


Boxplot for x71 [financial distress]
60
50
40
30
20
10
0
−10
1


Boxplot for x72 [financial distress]
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
1


Boxplot for x73 [financial distress]
36
34
32
30
28
26
24
22
1


Boxplot for x74 [financial distress]
120
110
100
90
80
70
60
1


Boxplot for x75 [financial distress]
225
200
175
150
125
100
75
50
25
1

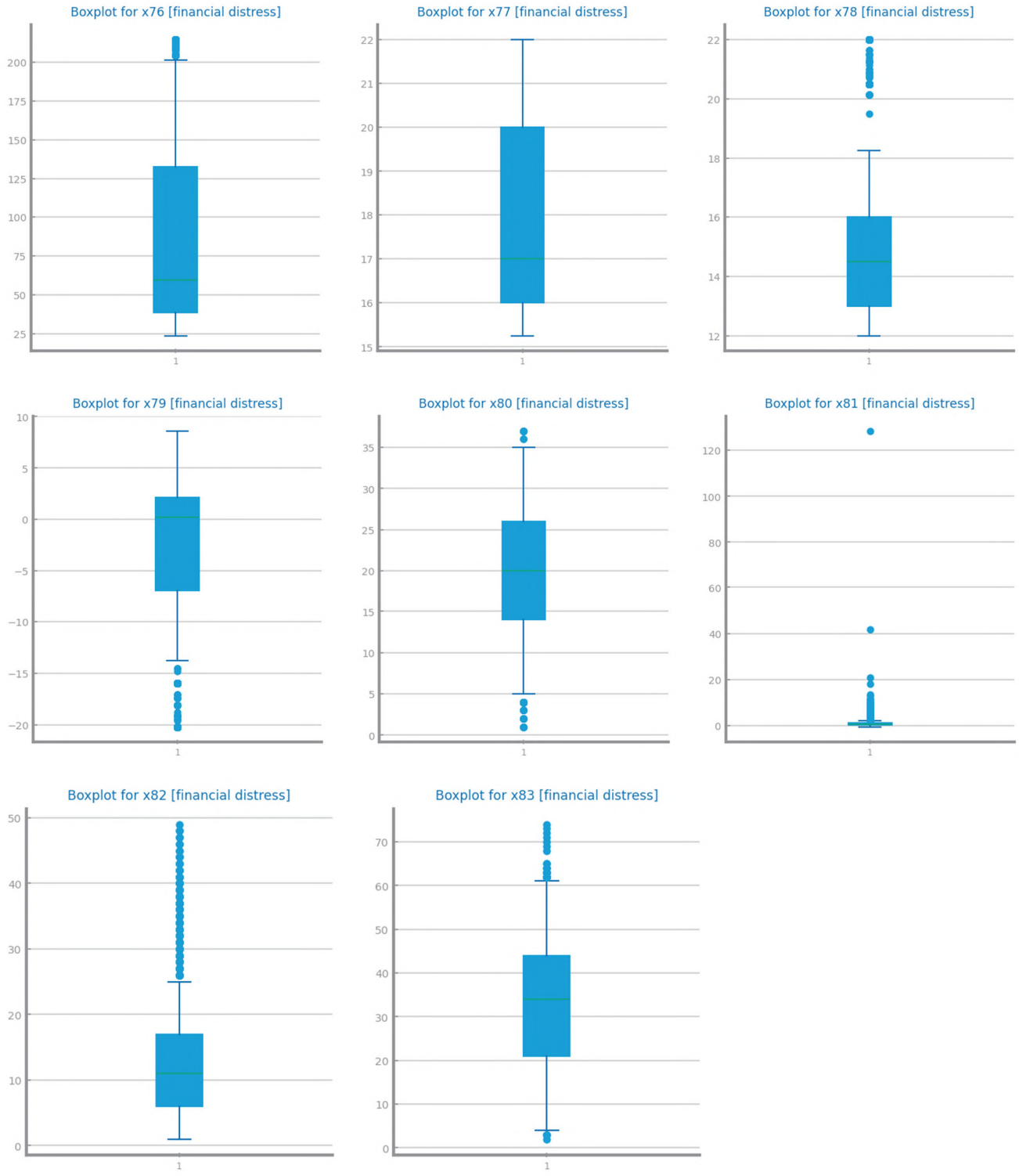
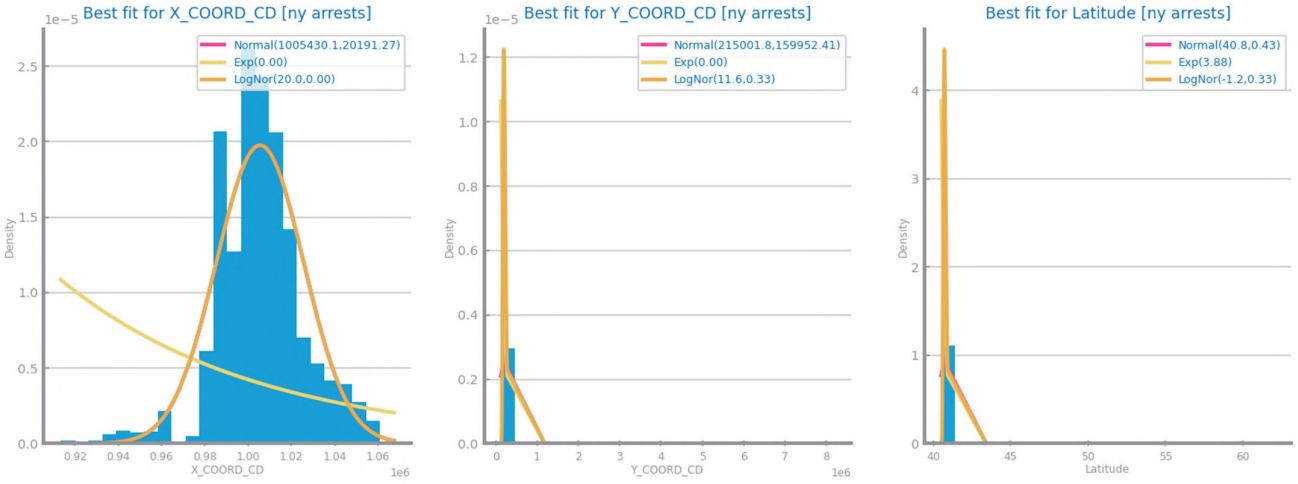
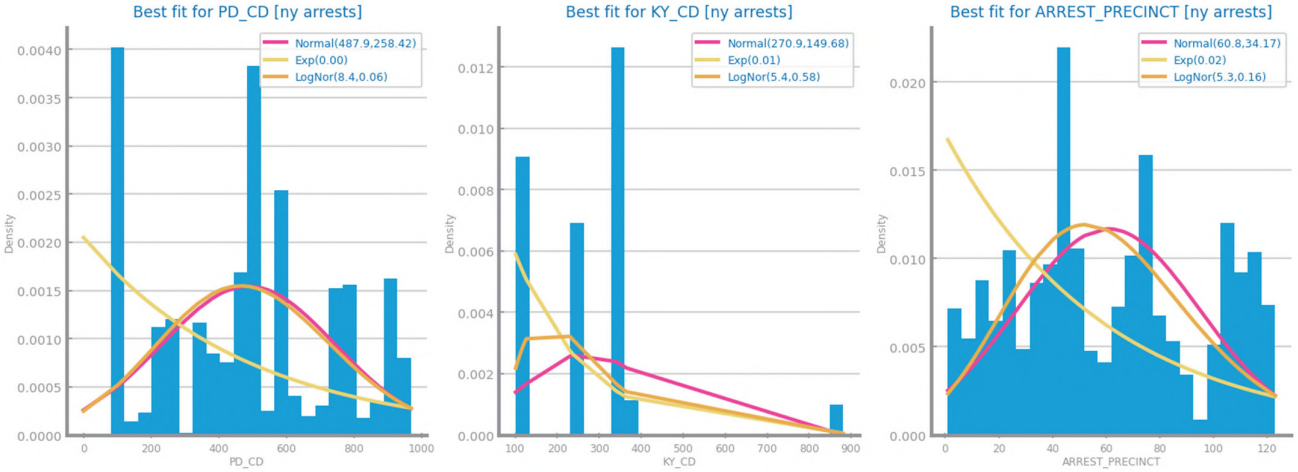
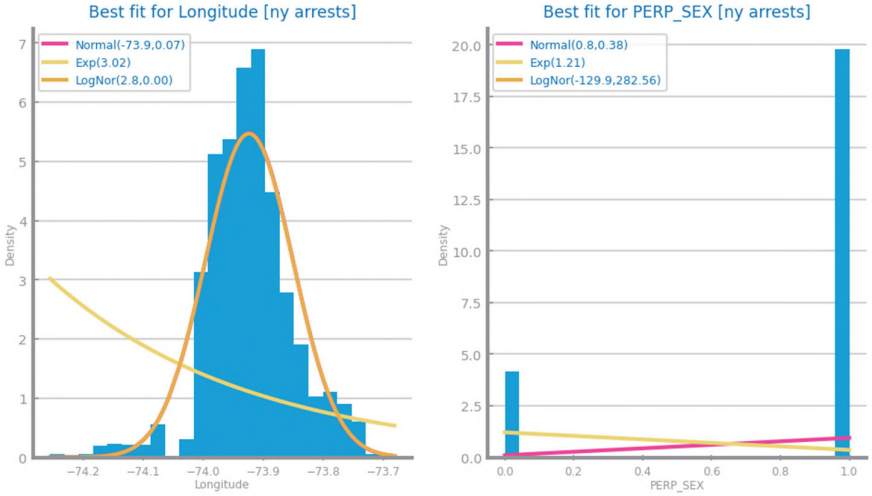


Figure 6 Single variable boxplots for dataset 2


Best fit for X_COORD_CD [ny arrests]
Normal(1005430.1,20191.27)
Exp(0.00)
LogNor(20.0,0.00)
Density
X_COORD_CD
1e6
Best fit for Y_COORD_CD [ny arrests]
Normal(215001.8,159952.41)
Exp(0.00)
LogNor(11.6,0.33)
Y_COORD_CD
Best fit for Latitude [ny arrests]
Normal(40.8,0.43)
Exp(3.88)
LogNor(-1.2,0.33)
Latitude


Best fit for PD_CD [ny arrests]
Normal(487.9,258.42)
Exp(0.00)
LogNor(8.4,0.06)
PD_CD
Best fit for KY_CD [ny arrests]
Normal(270.9,149.68)
Exp(0.01)
LogNor(5.4,0.58)
KY_CD
Best fit for ARREST_PRECINCT [ny arrests]
Normal(60.8,34.17)
Exp(0.02)
LogNor(5.3,0.16)
ARREST_PRECINCT


Best fit for Longitude [ny arrests]
Normal(-73.9,0.07)
Exp(3.02)
LogNor(2.8,0.00)
Longitude
Best fit for PERP_SEX [ny arrests]
Normal(0.8,0.38)
Exp(1.21)
LogNor(-129.9,282.56)
PERP_SEX

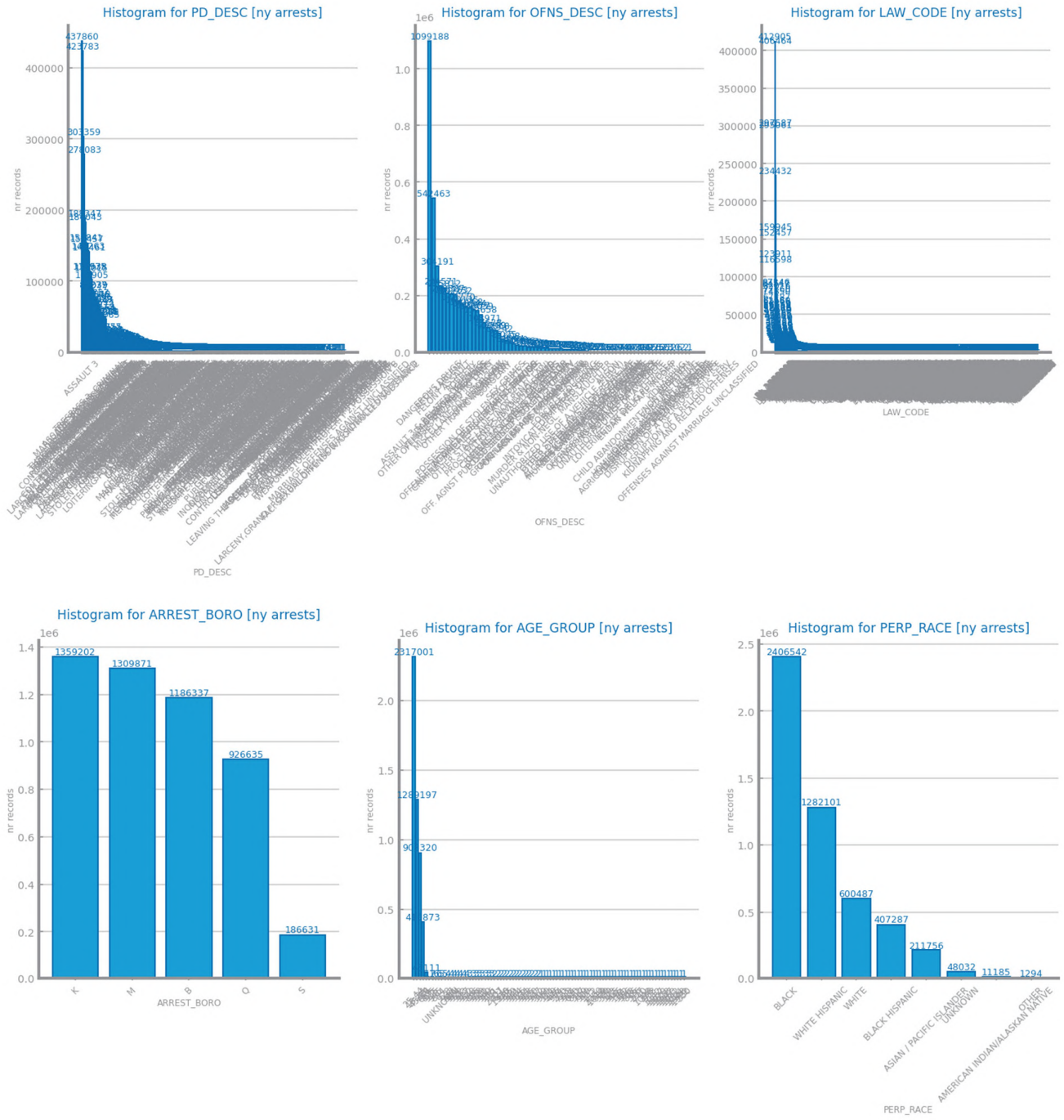
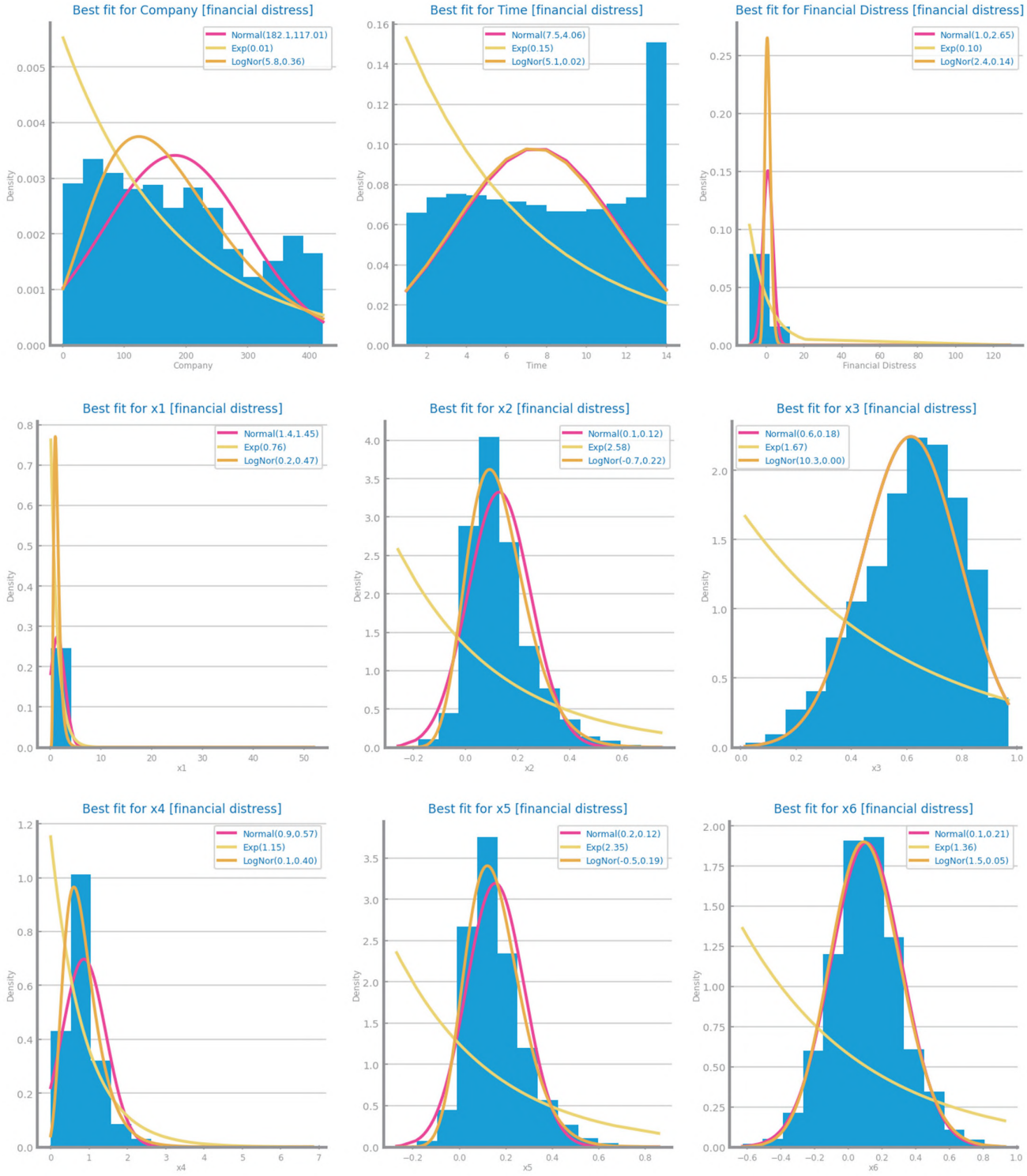
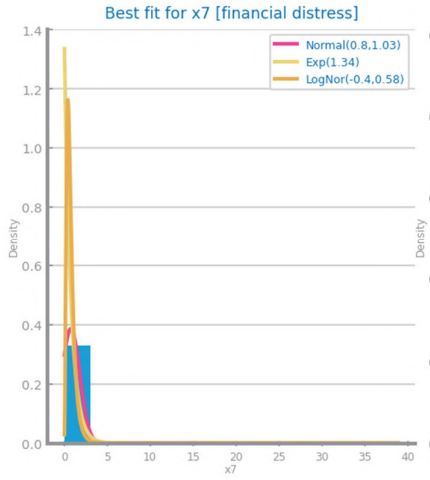
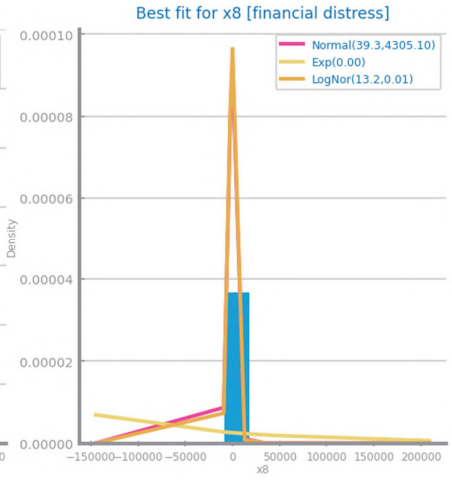
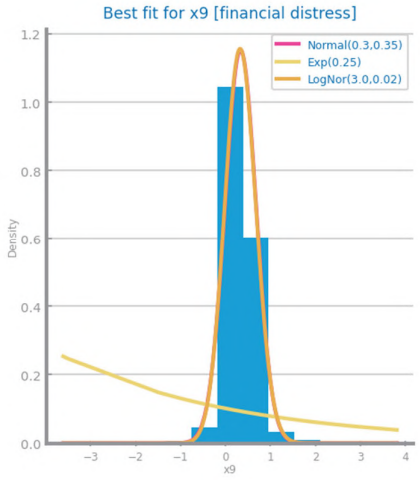


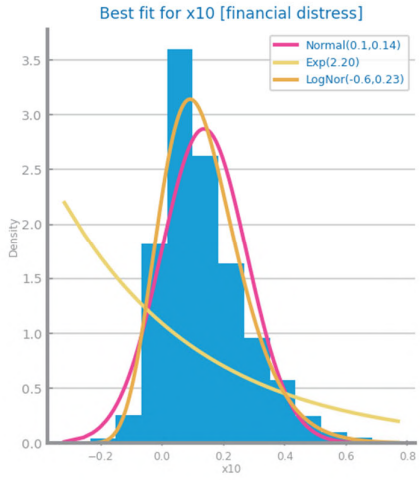
Figure 7 Histograms for dataset 1 (numeric and binary with approximations, and symbolic)

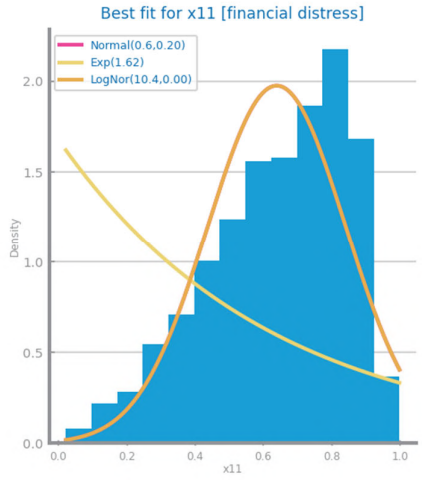

Best fit for Company [financial distress]
Normal(182.1,117.01)
Exp(0.01)
LogNor(5.8,0.36)
Density
Company
Best fit for Time [financial distress]
Normal(7.5,4.06)
Exp(0.15)
LogNor(5.1,0.02)
Density
Time
Best fit for Financial Distress [financial distress]
Normal(1.0,2.65)
Exp(0.10)
LogNor(2.4,0.14)
Density
Financial Distress
Best fit for x1 [financial distress]
Normal(1.4,1.45)
Exp(0.76)
LogNor(0.2,0.47)
Density
x1
Best fit for x2 [financial distress]
Normal(0.1,0.12)
Exp(2.58)
LogNor(-0.7,0.22)
Density
x2
Best fit for x3 [financial distress]
Normal(0.6,0.18)
Exp(1.67)
LogNor(10.3,0.00)
Density
x3
Best fit for x4 [financial distress]
Normal(0.9,0.57)
Exp(1.15)
LogNor(0.1,0.40)
Density
x4
Best fit for x5 [financial distress]
Normal(0.2,0.12)
Exp(2.35)
LogNor(-0.5,0.19)
Density
x5
Best fit for x6 [financial distress]
Normal(0.1,0.21)
Exp(1.36)
LogNor(1.5,0.05)
Density
x6

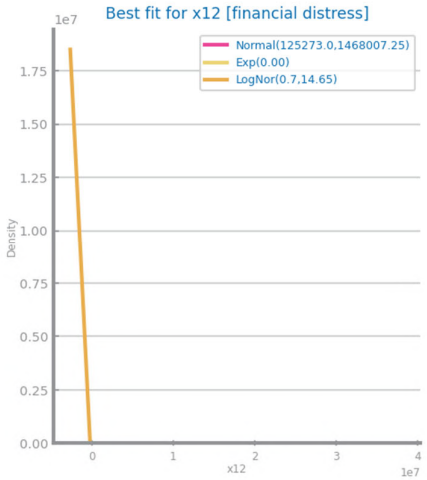

Best fit for x7 [financial distress]
Normal(0.8,1.03)
Exp(1.34)
LogNor(-0.4,0.58)
Density
x7

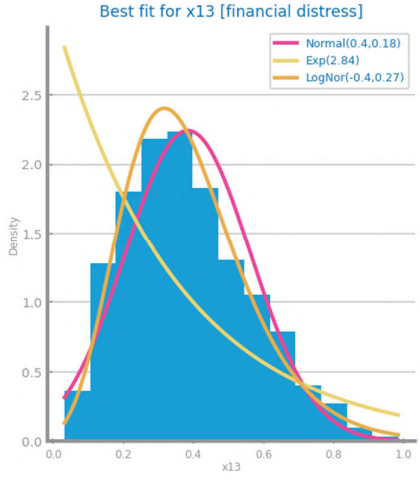

Best fit for x8 [financial distress]
Normal(39.3,4305.10)
Exp(0.00)
LogNor(13.2,0.01)
Density
x8

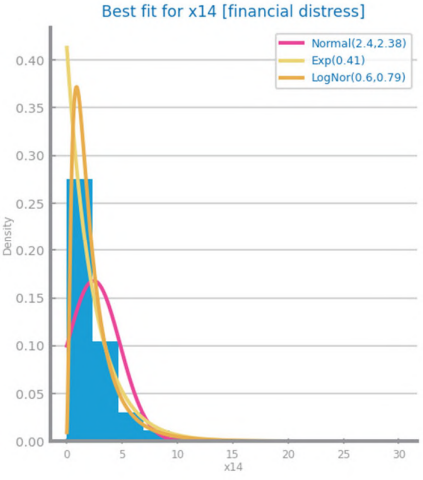

Best fit for x9 [financial distress]
Normal(0.3,0.35)
Exp(0.25)
LogNor(3.0,0.02)
Density
x9

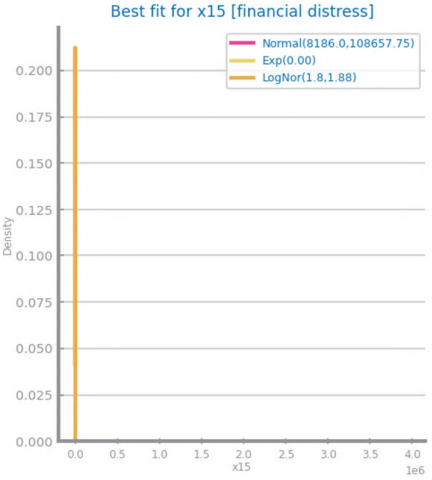

Best fit for x10 [financial distress]
Normal(0.1,0.14)
Exp(2.20)
LogNor(-0.6,0.23)
Density
x10

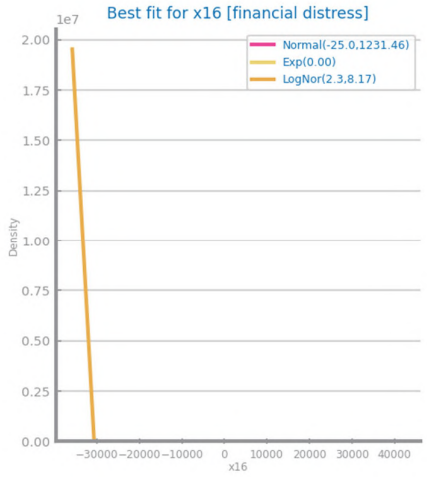

Best fit for x11 [financial distress]
Normal(0.6,0.20)
Exp(1.62)
LogNor(10.4,0.00)
Density
x11

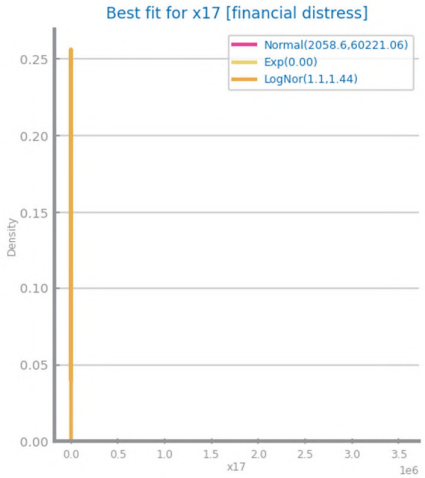

Best fit for x12 [financial distress]
Normal(125273.0,1468007.25)
Exp(0.00)
LogNor(0.7,14.65)
1e7
Density
x12
1e7

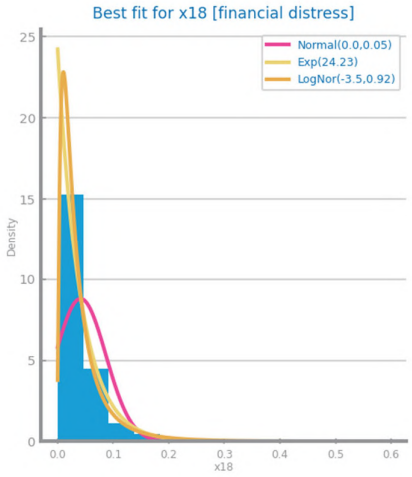

Best fit for x13 [financial distress]
Normal(0.4,0.18)
Exp(2.84)
LogNor(-0.4,0.27)
Density
x13

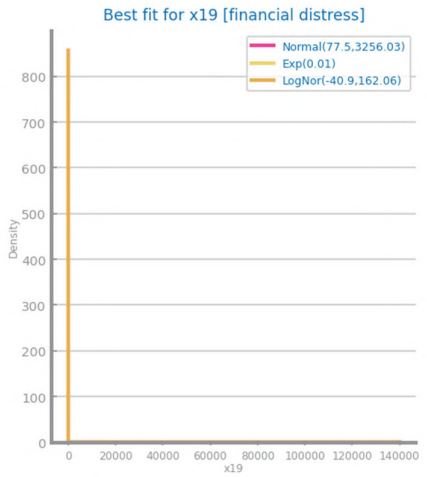

Best fit for x14 [financial distress]
Normal(2.4,2.38)
Exp(0.41)
LogNor(0.6,0.79)
Density
x14

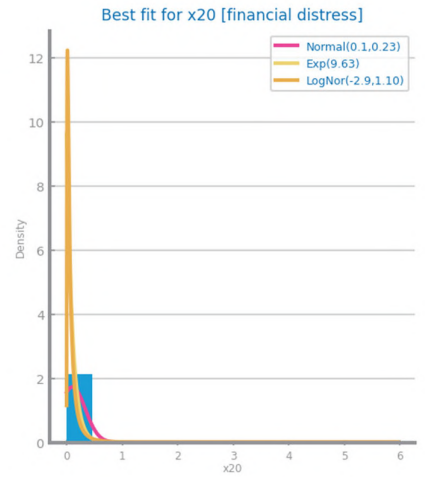

Best fit for x15 [financial distress]
Normal(8186.0,108657.75)
Exp(0.00)
LogNor(1.8,1.88)
Density
x15
1e6

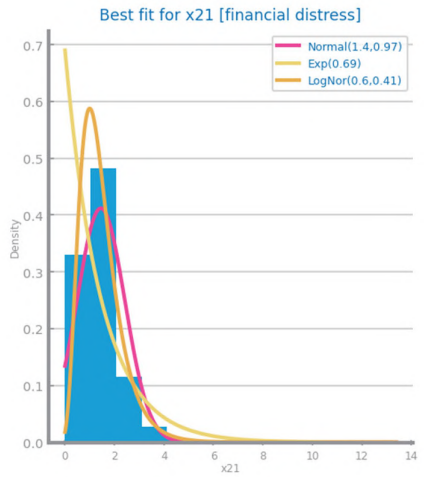

Best fit for x16 [financial distress]
1e7
Normal(-25.0,1231.46)
Exp(0.00)
LogNor(2.3,8.17)
Density
x16

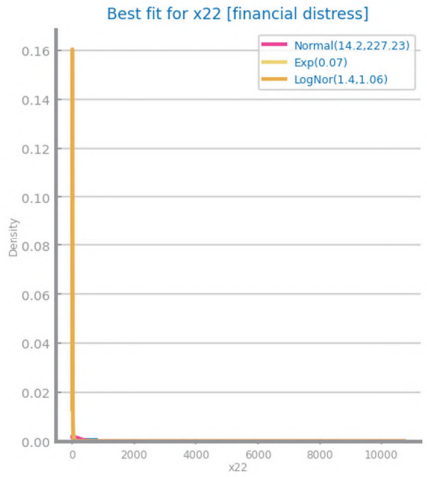

Best fit for x17 [financial distress]
Normal(2058.6,60221.06)
Exp(0.00)
LogNor(1.1,1.44)
Density
x17
1e6

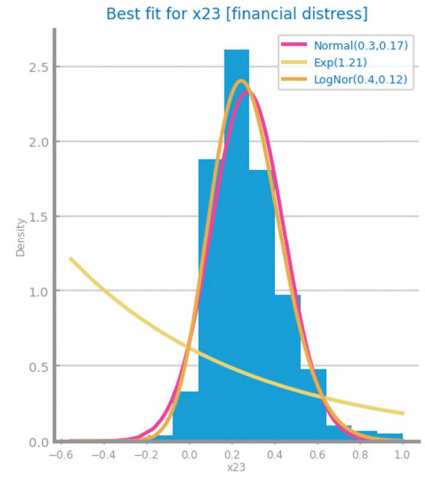

Best fit for x18 [financial distress]
Normal(0.0,0.05)
Exp(24.23)
LogNor(-3.5,0.92)
Density
x18

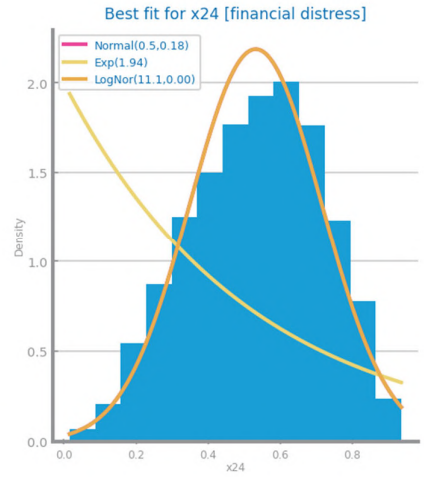

Best fit for x19 [financial distress]
Normal(77.5,3256.03)
Exp(0.01)
LogNor(-40.9,162.06)
Density
x19

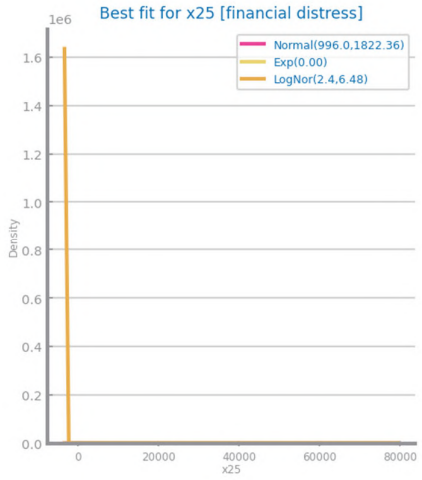

Best fit for x20 [financial distress]
Normal(0.1,0.23)
Exp(9.63)
LogNor(-2.9,1.10)
Density
x20

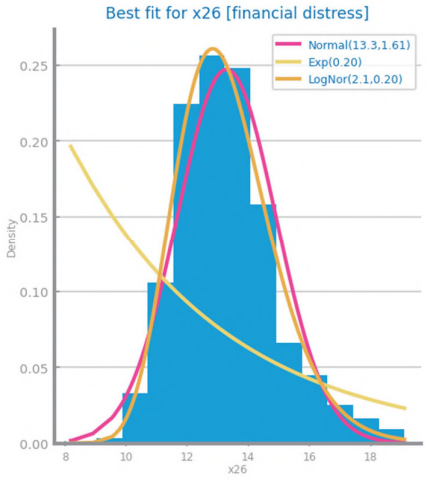

Best fit for x21 [financial distress]
Normal(1.4,0.97)
Exp(0.69)
LogNor(0.6,0.41)
Density
x21

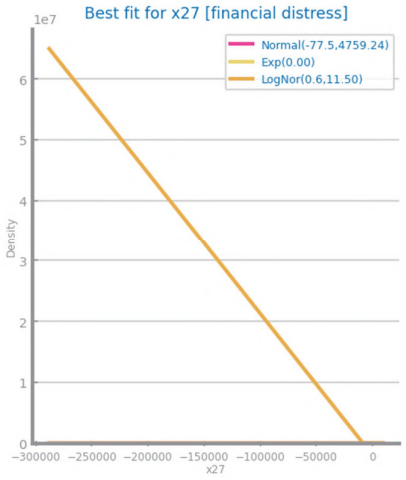

Best fit for x22 [financial distress]
Normal(14.2,227.23)
Exp(0.07)
LogNor(1.4,1.06)
Density
x22

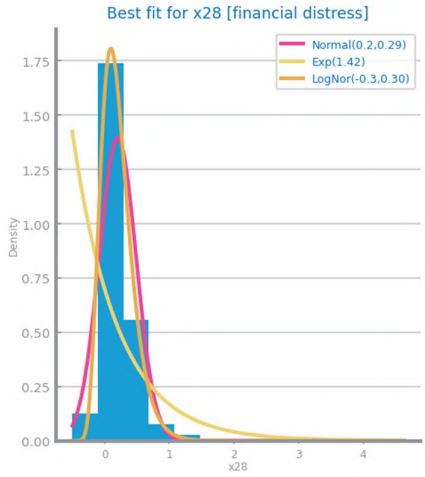

Best fit for x23 [financial distress]
Normal(0.3,0.17)
Exp(1.21)
LogNor(0.4,0.12)
Density
x23

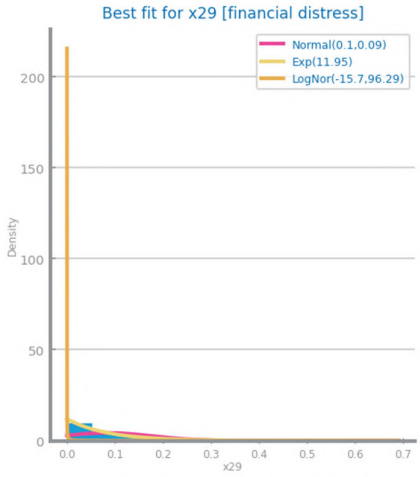

Best fit for x24 [financial distress]
Normal(0.5,0.18)
Exp(1.94)
LogNor(11.1,0.00)
Density
x24

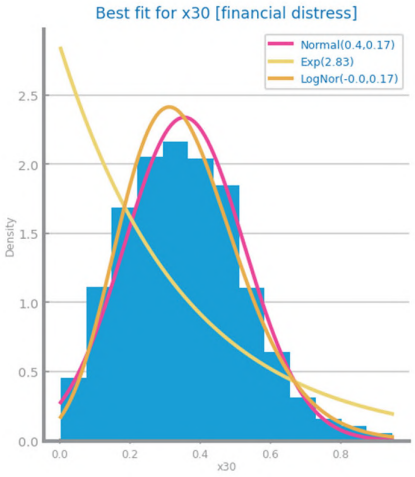

Best fit for x25 [financial distress]
Normal(996.0,1822.36)
Exp(0.00)
LogNor(2.4,6.48)
1e6
Density
x25

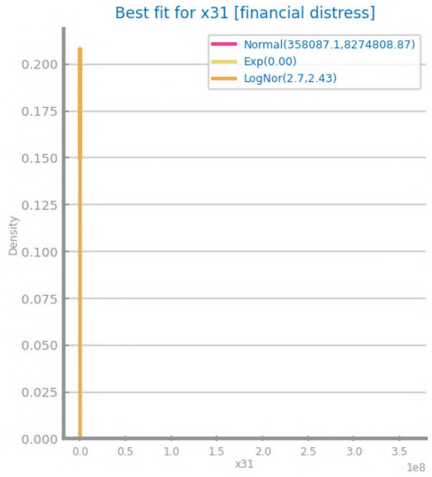

Best fit for x26 [financial distress]
Normal(13.3,1.61)
Exp(0.20)
LogNor(2.1,0.20)
Density
x26

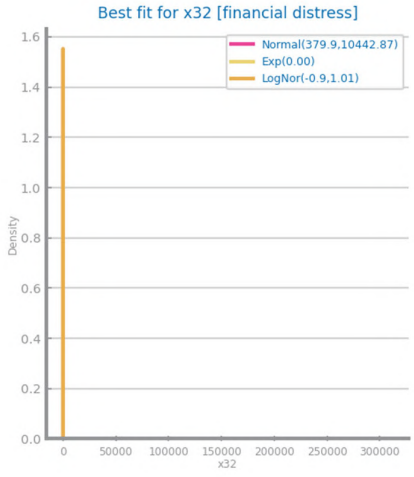

Best fit for x27 [financial distress]
Normal(-77.5,4759.24)
Exp(0.00)
LogNor(0.6,11.50)
1e7
Density
x27

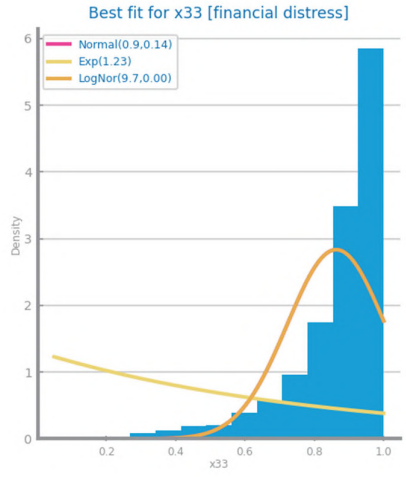

Best fit for x28 [financial distress]
Normal(0.2,0.29)
Exp(1.42)
LogNor(-0.3,0.30)
Density
x28

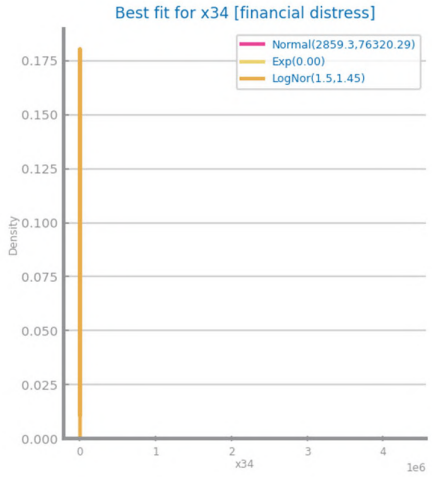

Best fit for x29 [financial distress]
Normal(0.1,0.09)
Exp(11.95)
LogNor(-15.7,96.29)
Density
x29

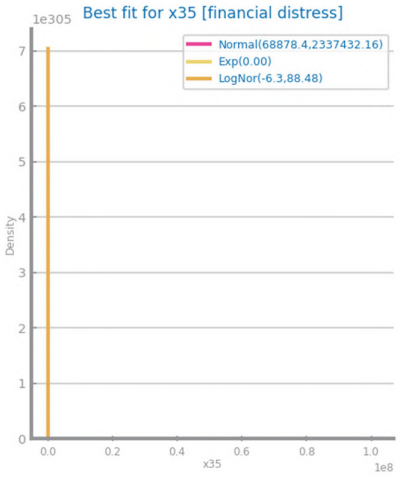

Best fit for x30 [financial distress]
Normal(0.4,0.17)
Exp(2.83)
LogNor(-0.0,0.17)
Density
x30

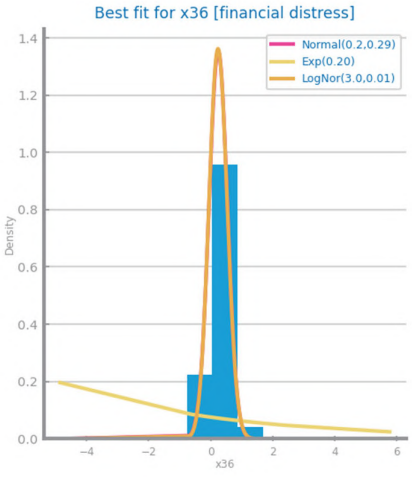

Best fit for x31 [financial distress]
Normal(358087.1,8274808.87)
Exp(0.00)
LogNor(2.7,2.43)
Density
x31
1e8

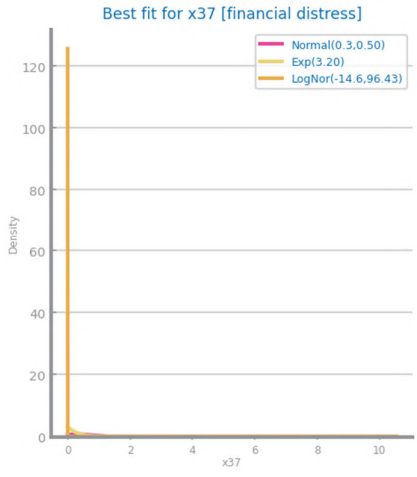

Best fit for x32 [financial distress]
Normal(379.9,10442.87)
Exp(0.00)
LogNor(-0.9,1.01)
Density
x32

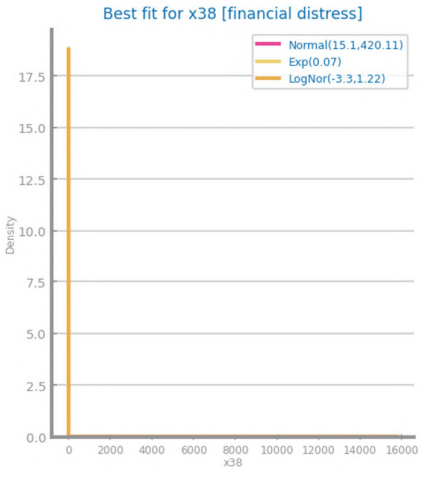

Best fit for x33 [financial distress]
Normal(0.9,0.14)
Exp(1.23)
LogNor(9.7,0.00)
Density
x33

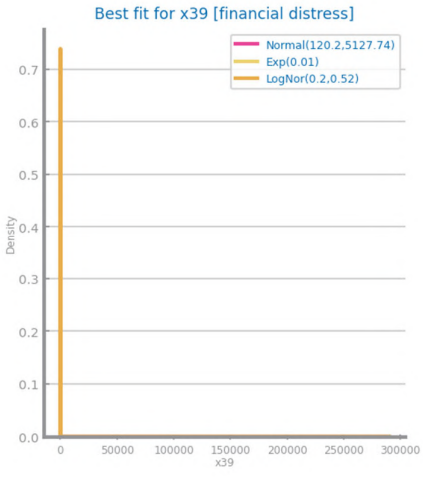

Best fit for x34 [financial distress]
Normal(2859.3,76320.29)
Exp(0.00)
LogNor(1.5,1.45)
Density
x34
1e6

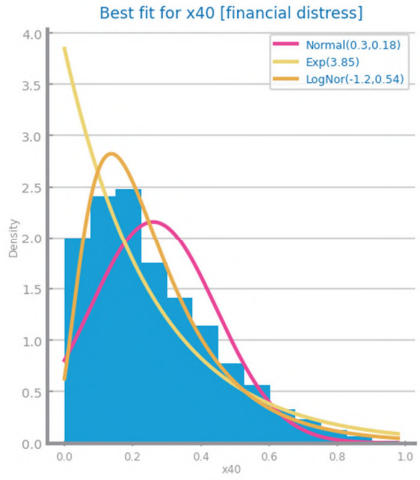

Best fit for x35 [financial distress]
1e305
Normal(68878.4,2337432.16)
Exp(0.00)
LogNor(-6.3,88.48)
Density
x35
1e8

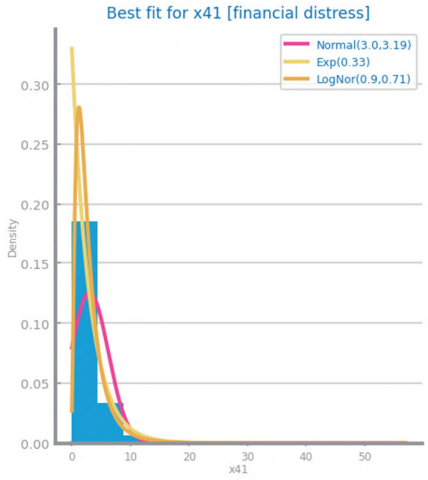

Best fit for x36 [financial distress]
Normal(0.2,0.29)
Exp(0.20)
LogNor(3.0,0.01)
Density
x36

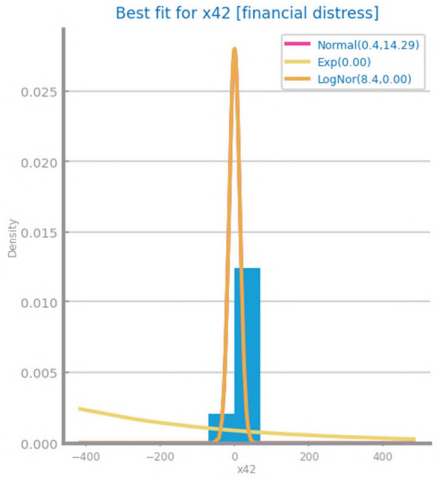

Best fit for x37 [financial distress]
Normal(0.3,0.50)
Exp(3.20)
LogNor(-14.6,96.43)
Density
x37

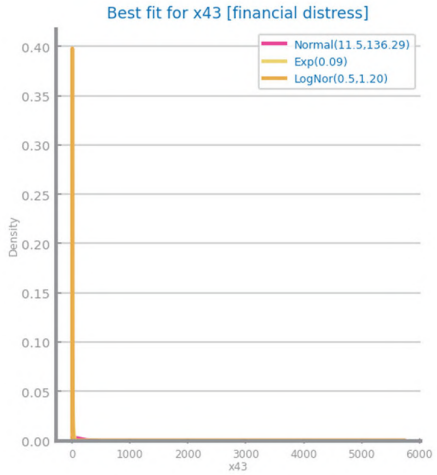

Best fit for x38 [financial distress]
Normal(15.1,420.11)
Exp(0.07)
LogNor(-3.3,1.22)
Density
x38

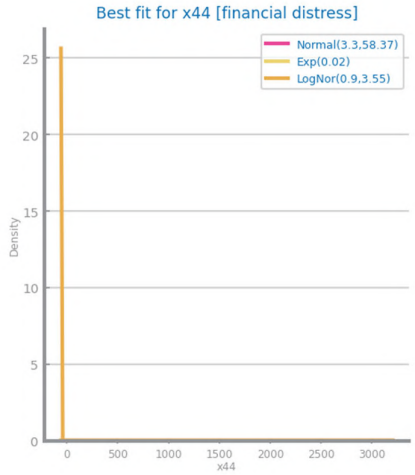

Best fit for x39 [financial distress]
Normal(120.2,5127.74)
Exp(0.01)
LogNor(0.2,0.52)
Density
x39

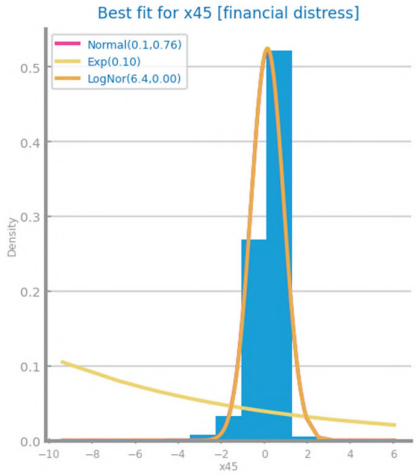

Best fit for x40 [financial distress]
Normal(0.3,0.18)
Exp(3.85)
LogNor(-1.2,0.54)
Density
x40

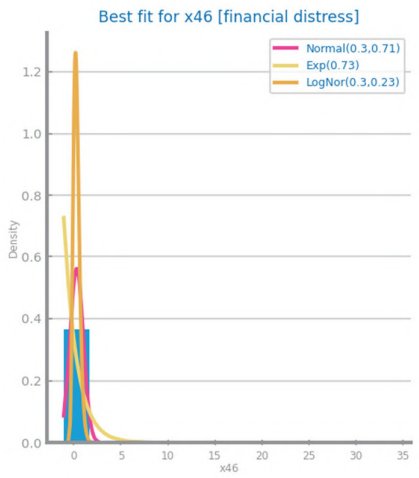

Best fit for x41 [financial distress]
Normal(3.0,3.19)
Exp(0.33)
LogNor(0.9,0.71)
Density
x41

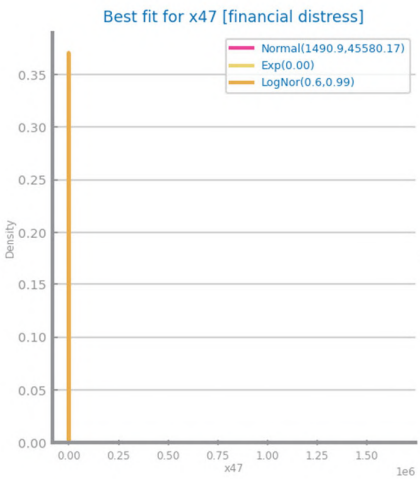

Best fit for x42 [financial distress]
Normal(0.4,14.29)
Exp(0.00)
LogNor(8.4,0.00)
Density
x42

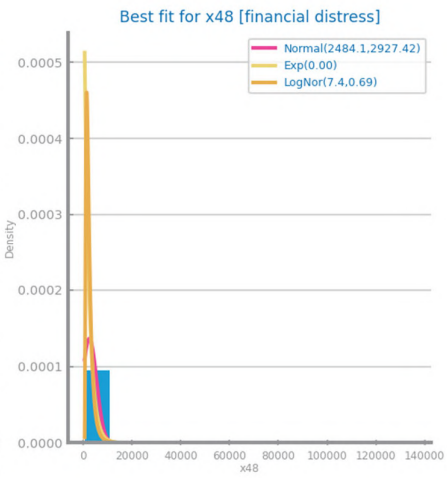

Best fit for x43 [financial distress]
Normal(11.5,136.29)
Exp(0.09)
LogNor(0.5,1.20)
Density
x43

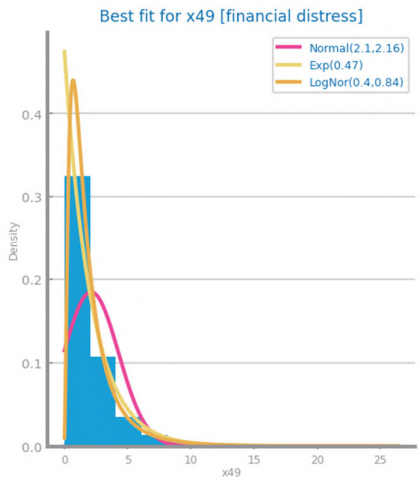

Best fit for x44 [financial distress]
Normal(3.3,58.37)
Exp(0.02)
LogNor(0.9,3.55)
Density
x44

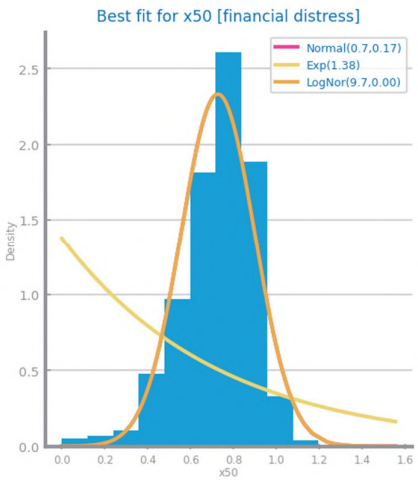

Best fit for x45 [financial distress]
Normal(0.1,0.76)
Exp(0.10)
LogNor(6.4,0.00)
Density
x45

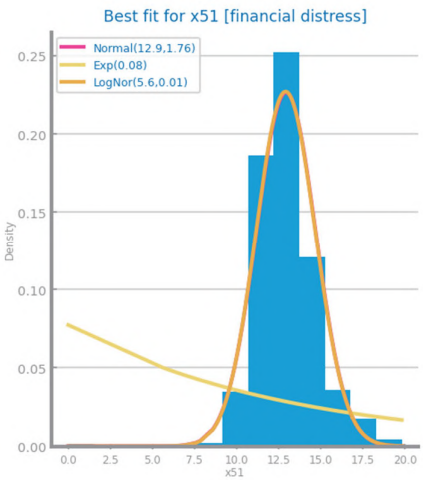

Best fit for x46 [financial distress]
Normal(0.3,0.71)
Exp(0.73)
LogNor(0.3,0.23)
Density
x46

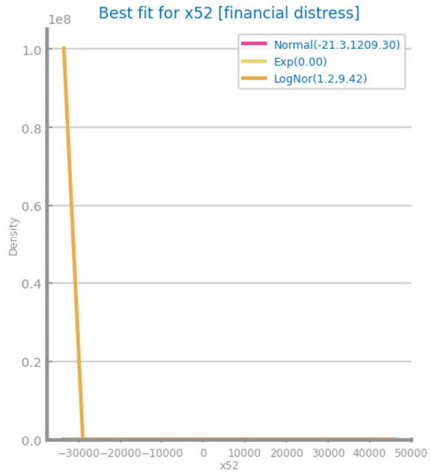

Best fit for x47 [financial distress]
Normal(1490.9,45580.17)
Exp(0.00)
LogNor(0.6,0.99)
Density
x47
1e6

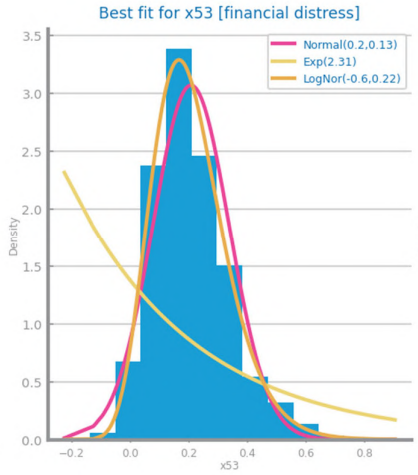

Best fit for x48 [financial distress]
Normal(2484.1,2927.42)
Exp(0.00)
LogNor(7.4,0.69)
Density
x48

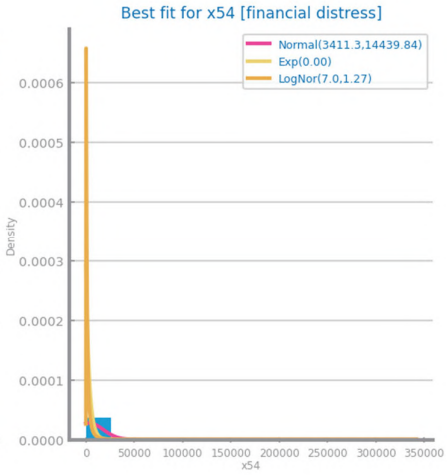

Best fit for x49 [financial distress]
Normal(2.1,2.16)
Exp(0.47)
LogNor(0.4,0.84)
Density
x49

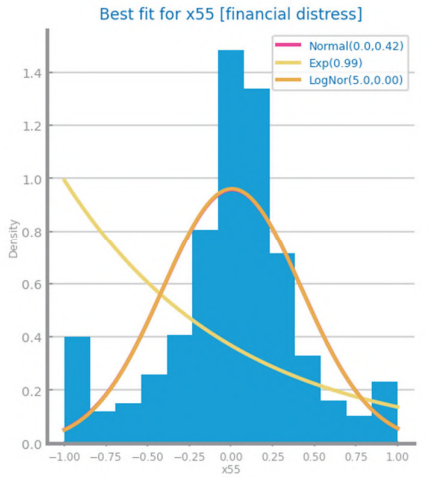

Best fit for x50 [financial distress]
Normal(0.7,0.17)
Exp(1.38)
LogNor(9.7,0.00)
Density
x50

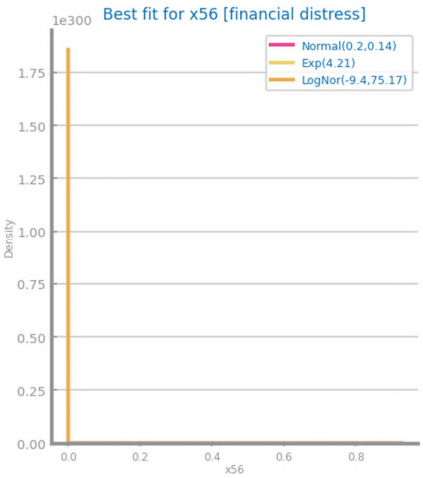

Best fit for x51 [financial distress]
Normal(12.9,1.76)
Exp(0.08)
LogNor(5.6,0.01)
Density
x51

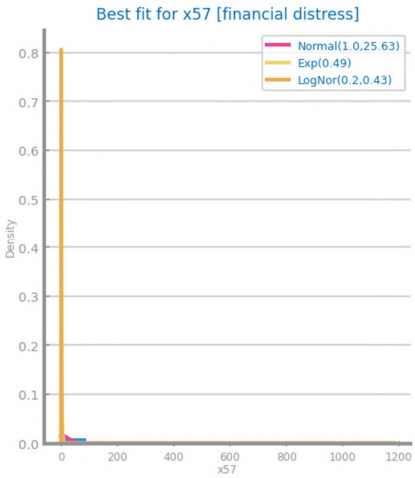

Best fit for x52 [financial distress]
1e8
Normal(-21.3,1209.30)
Exp(0.00)
LogNor(1.2,9.42)
Density
x52

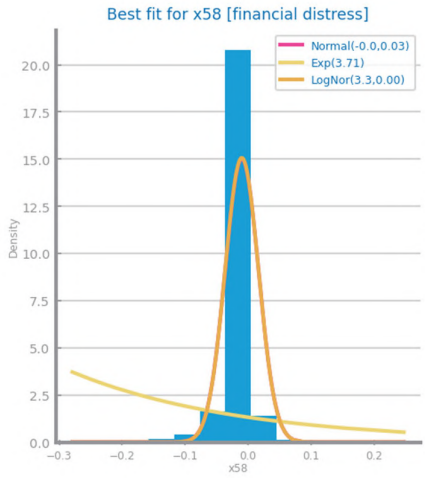

Best fit for x53 [financial distress]
Normal(0.2,0.13)
Exp(2.31)
LogNor(-0.6,0.22)
Density
x53

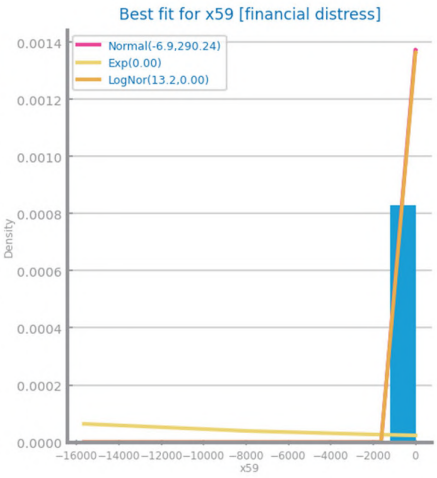

Best fit for x54 [financial distress]
Normal(3411.3,14439.84)
Exp(0.00)
LogNor(7.0,1.27)
Density
x54

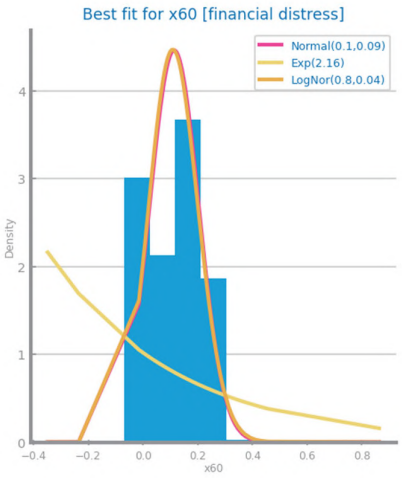

Best fit for x55 [financial distress]
Normal(0.0,0.42)
Exp(0.99)
LogNor(5.0,0.00)
Density
x55

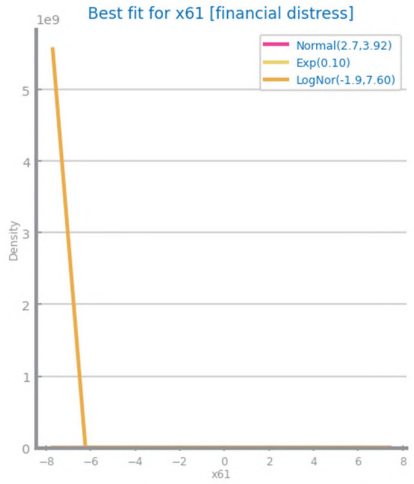

Best fit for x56 [financial distress]
1e300
Normal(0.2,0.14)
Exp(4.21)
LogNor(-9.4,75.17)
Density
x56

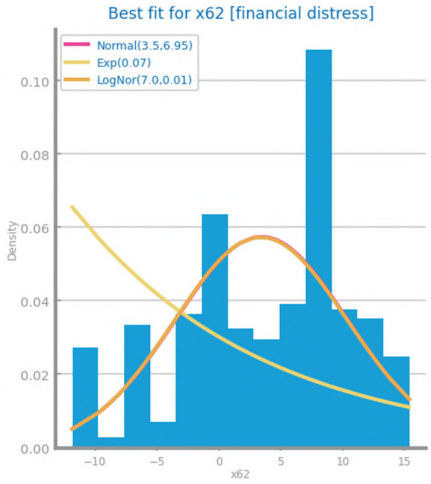

Best fit for x57 [financial distress]
Normal(1.0,25.63)
Exp(0.49)
LogNor(0.2,0.43)
Density
x57

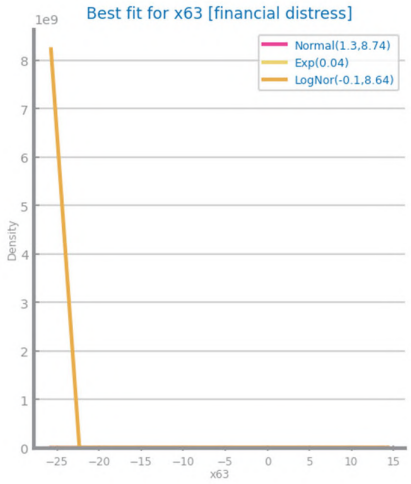

Best fit for x58 [financial distress]
Normal(-0.0,0.03)
Exp(3.71)
LogNor(3.3,0.00)
Density
x58

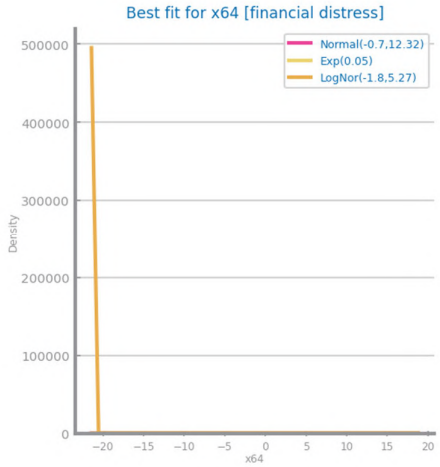

Best fit for x59 [financial distress]
Normal(-6.9,290.24)
Exp(0.00)
LogNor(13.2,0.00)
Density
x59

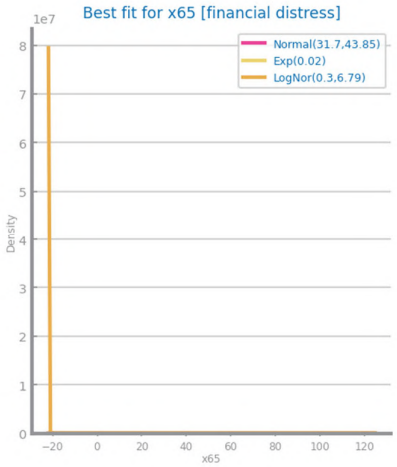

Best fit for x60 [financial distress]
Normal(0.1,0.09)
Exp(2.16)
LogNor(0.8,0.04)
Density
x60

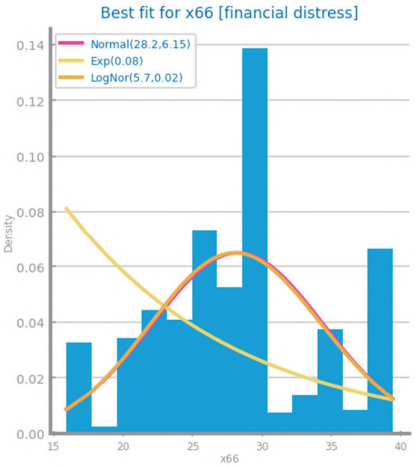

Best fit for x61 [financial distress]
Normal(2.7,3.92)
Exp(0.10)
LogNor(-1.9,7.60)
Density
x61

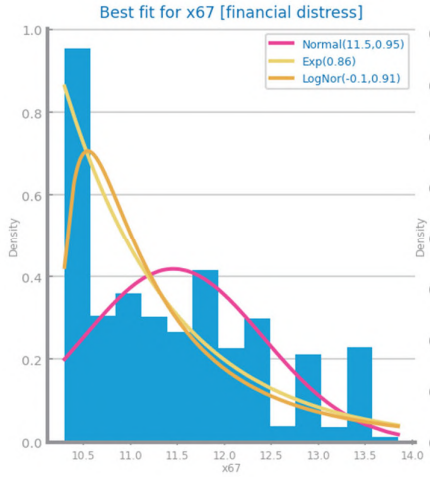

Best fit for x62 [financial distress]
Normal(3.5,6.95)
Exp(0.07)
LogNor(7.0,0.01)
Density
x62

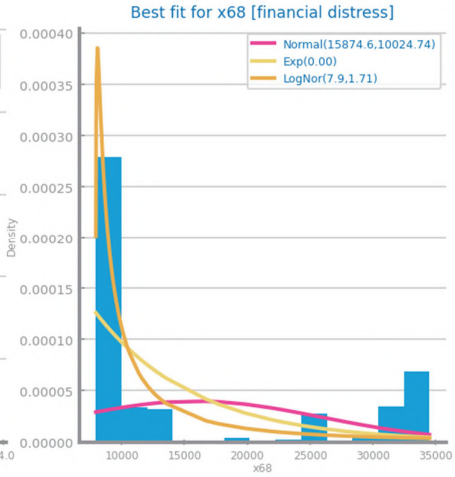

Best fit for x63 [financial distress]
Normal(1.3,8.74)
Exp(0.04)
LogNor(-0.1,8.64)
Density
x63

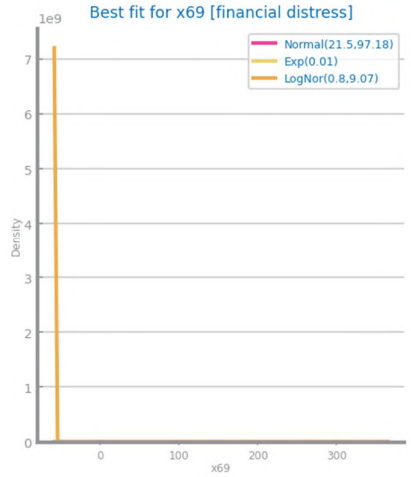

Best fit for x64 [financial distress]
Normal(-0.7,12.32)
Exp(0.05)
LogNor(-1.8,5.27)
Density
x64

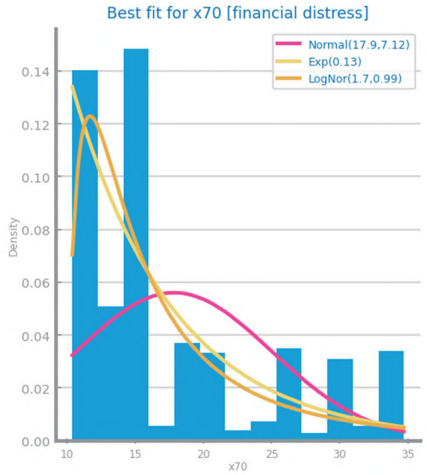

Best fit for x65 [financial distress]
Normal(31.7,43.85)
Exp(0.02)
LogNor(0.3,6.79)
Density
x65

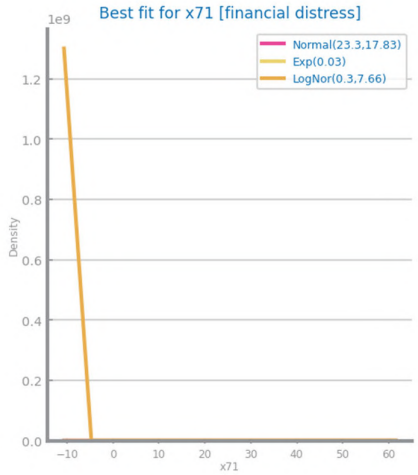

Best fit for x66 [financial distress]
Normal(28.2,6.15)
Exp(0.08)
LogNor(5.7,0.02)
Density
x66

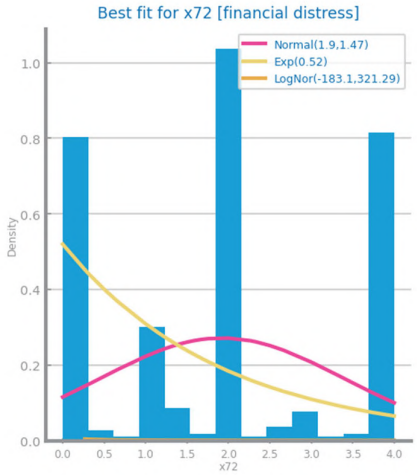

Best fit for x67 [financial distress]
Normal(11.5,0.95)
Exp(0.86)
LogNor(-0.1,0.91)
Density
x67

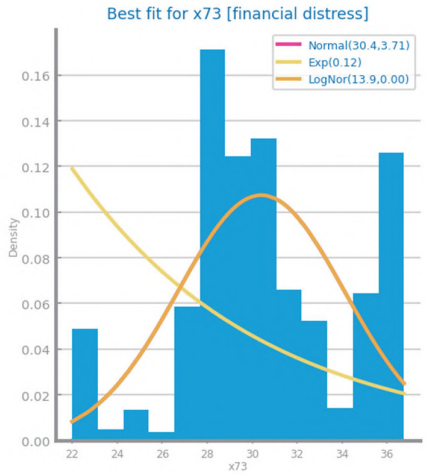

Best fit for x68 [financial distress]
Normal(15874.6,10024.74)
Exp(0.00)
LogNor(7.9,1.71)
Density
x68

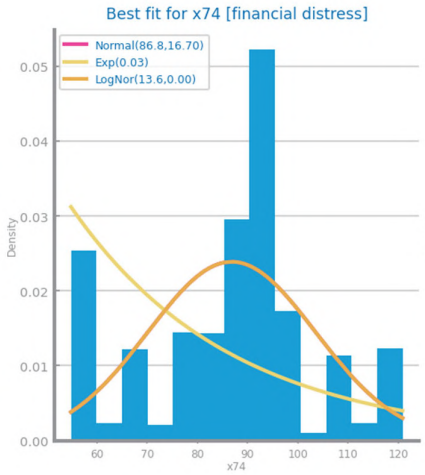

Best fit for x69 [financial distress]
Normal(21.5,97.18)
Exp(0.01)
LogNor(0.8,9.07)
Density
x69

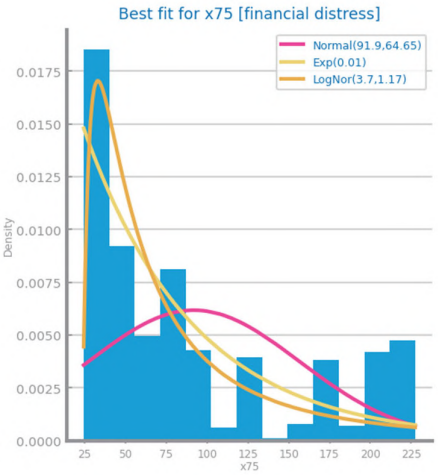

Best fit for x70 [financial distress]
Normal(17.9,7.12)
Exp(0.13)
LogNor(1.7,0.99)
Density
x70

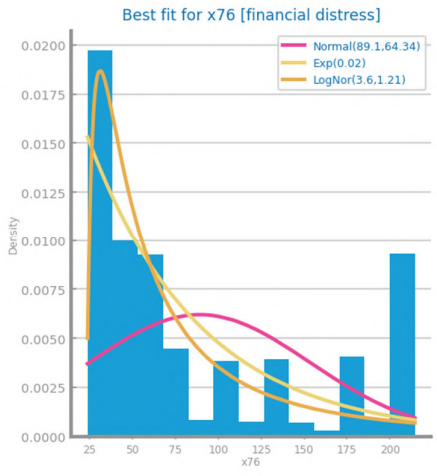

Best fit for x71 [financial distress]
1e9
Normal(23.3,17.83)
Exp(0.03)
LogNor(0.3,7.66)
Density
x71

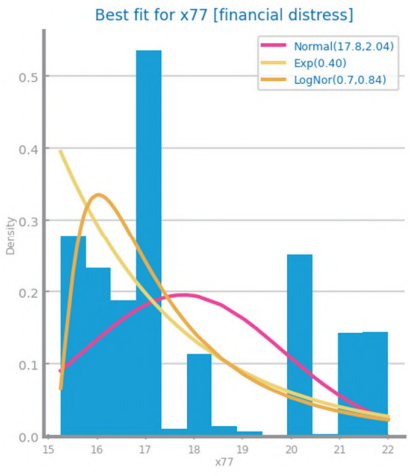

Best fit for x72 [financial distress]
Normal(1.9,1.47)
Exp(0.52)
LogNor(-183.1,321.29)
Density
x72

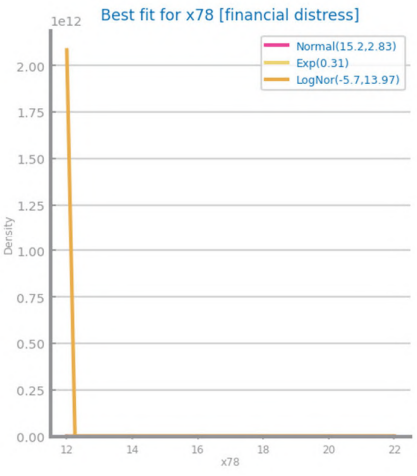

Best fit for x73 [financial distress]
Normal(30.4,3.71)
Exp(0.12)
LogNor(13.9,0.00)
Density
x73


Best fit for x74 [financial distress]
Normal(86.8,16.70)
Exp(0.03)
LogNor(13.6,0.00)
Density
x74


Best fit for x75 [financial distress]
Normal(91.9,64.65)
Exp(0.01)
LogNor(3.7,1.17)
Density
x75


Best fit for x76 [financial distress]
Normal(89.1,64.34)
Exp(0.02)
LogNor(3.6,1.21)
Density
x76


Best fit for x77 [financial distress]
Normal(17.8,2.04)
Exp(0.40)
LogNor(0.7,0.84)
Density
x77


Best fit for x78 [financial distress]
1e12
Normal(15.2,2.83)
Exp(0.31)
LogNor(-5.7,13.97)
Density
x78

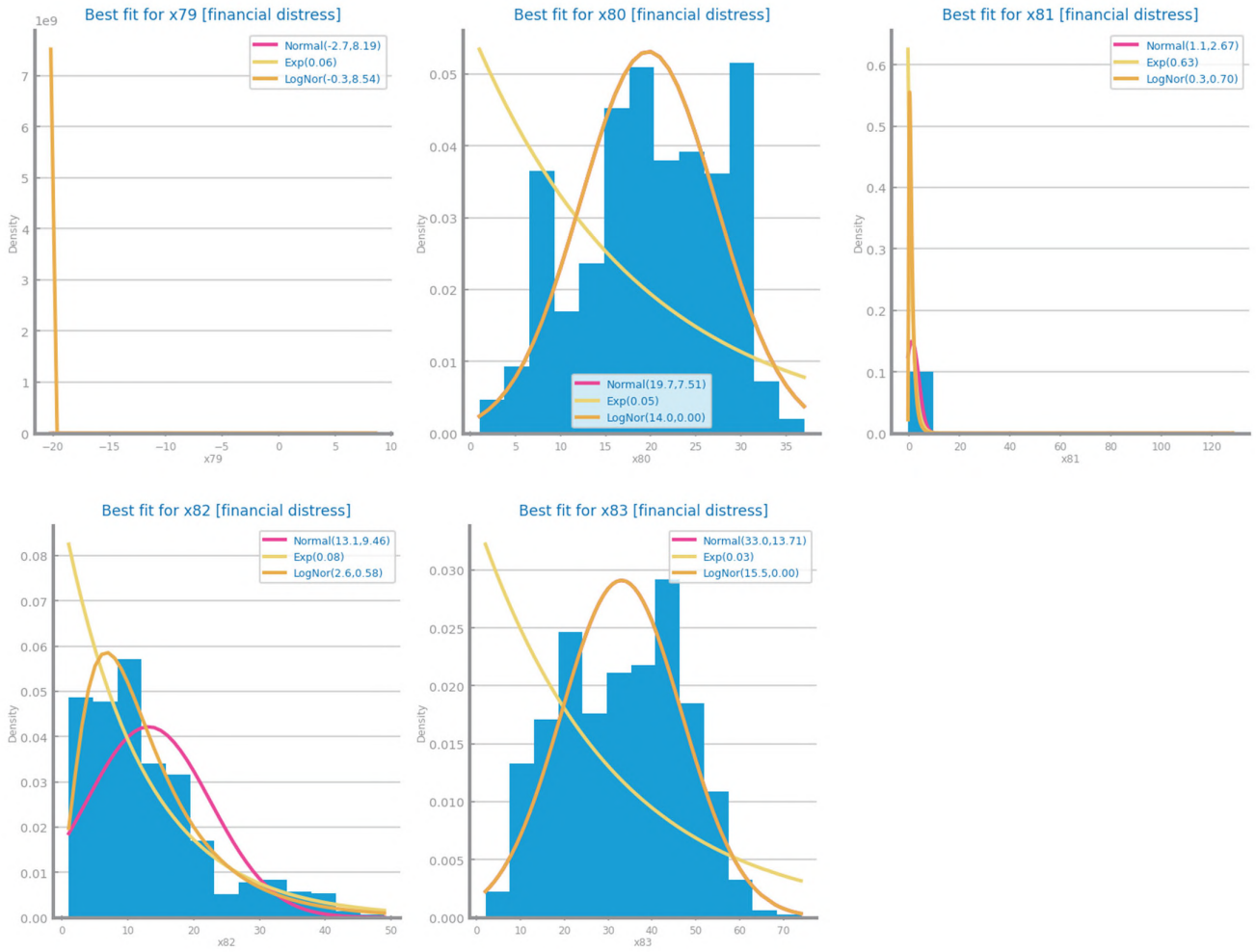


Figure 8 Histograms for dataset 2

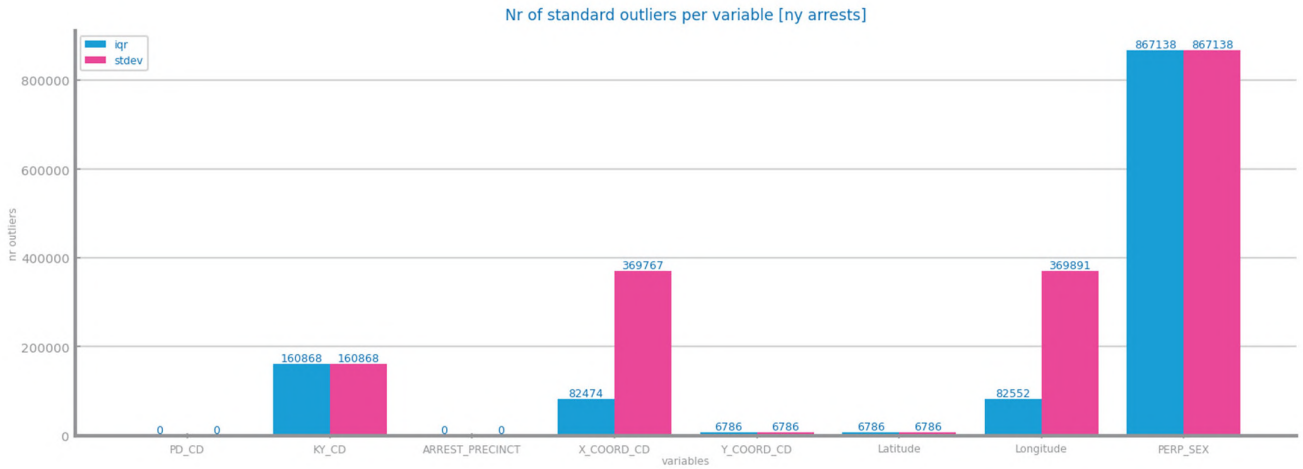


Figure 9 Outliers study dataset 1

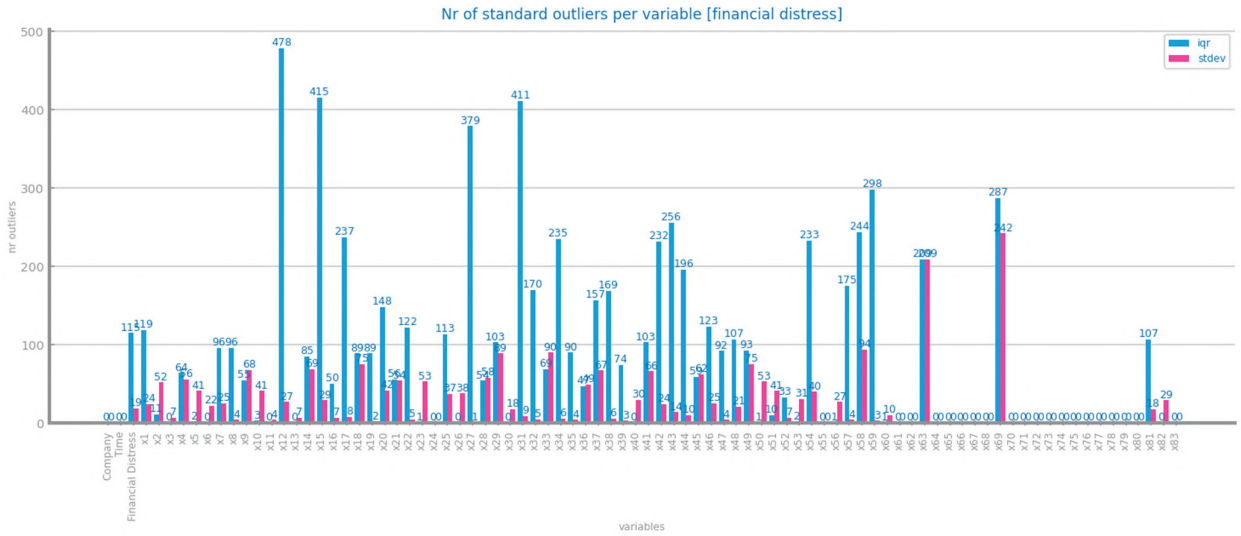


Figure 9 Outliers study for dataset 2

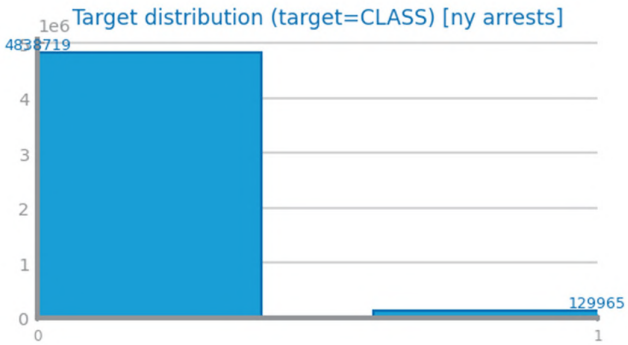


Figure 10 Class distribution for dataset 1

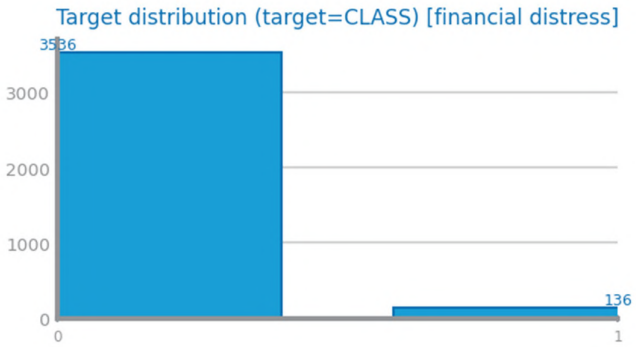
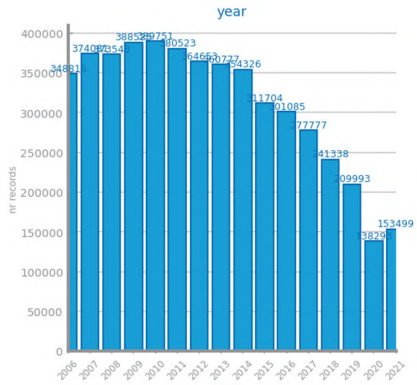


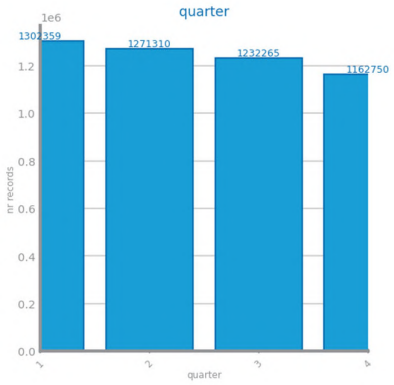
Figure 11 Class distribution for dataset 2

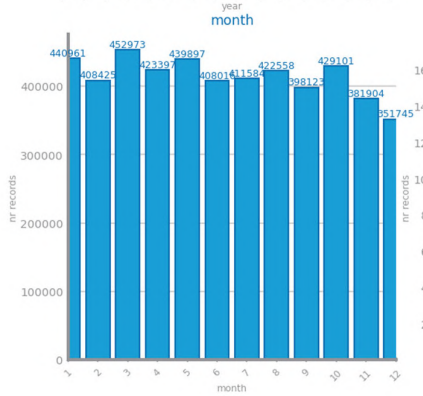
Data Granularity

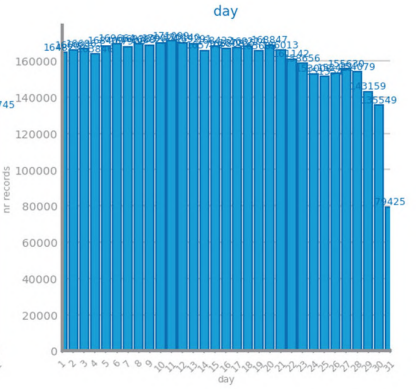
While no variables in Dataset 2 exhibited hierarchical characteristics, Dataset 1 contained two features with levels of granularity, namely time (ARREST_DATE) and geographic coordinates (Latitude and Longitude); the coordinates, in decimal values, were converted to degrees, minutes, and seconds.

Granularity study for ARREST_DATE [ny arrests]

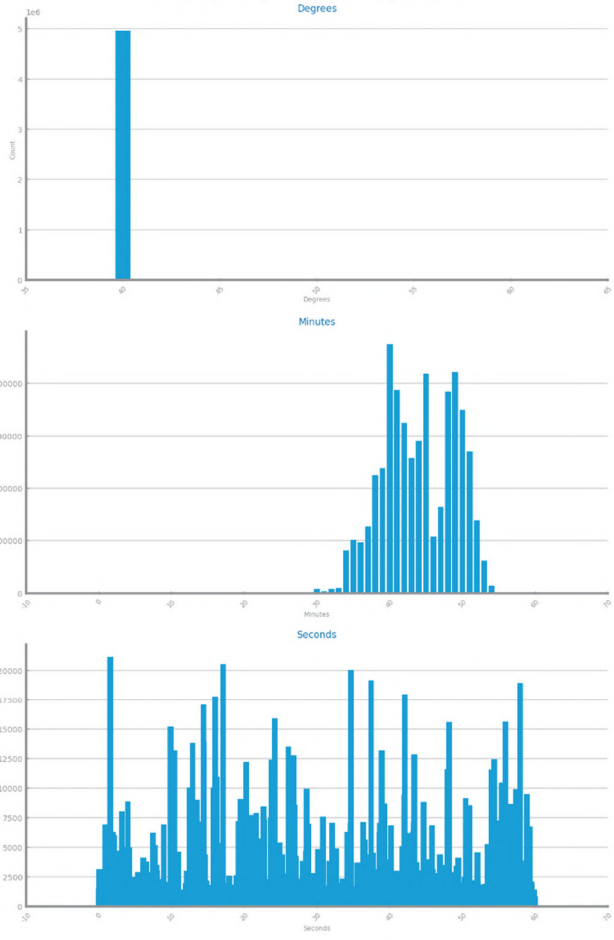








Granularity study for Latitude [ny arrests]



Granularity study for Longitude [ny arrests]

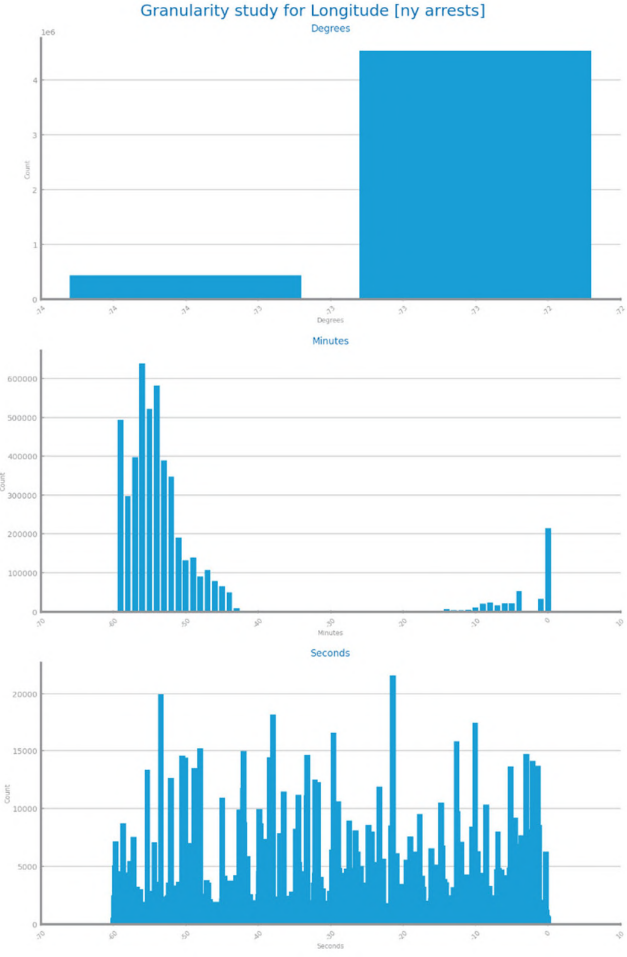


Figure 12 Granularity analysis for dataset 1

From the analysis of Dataset 2, it was found that no variable exhibits hierarchical characteristics, making it impossible to perform a granular level analysis.

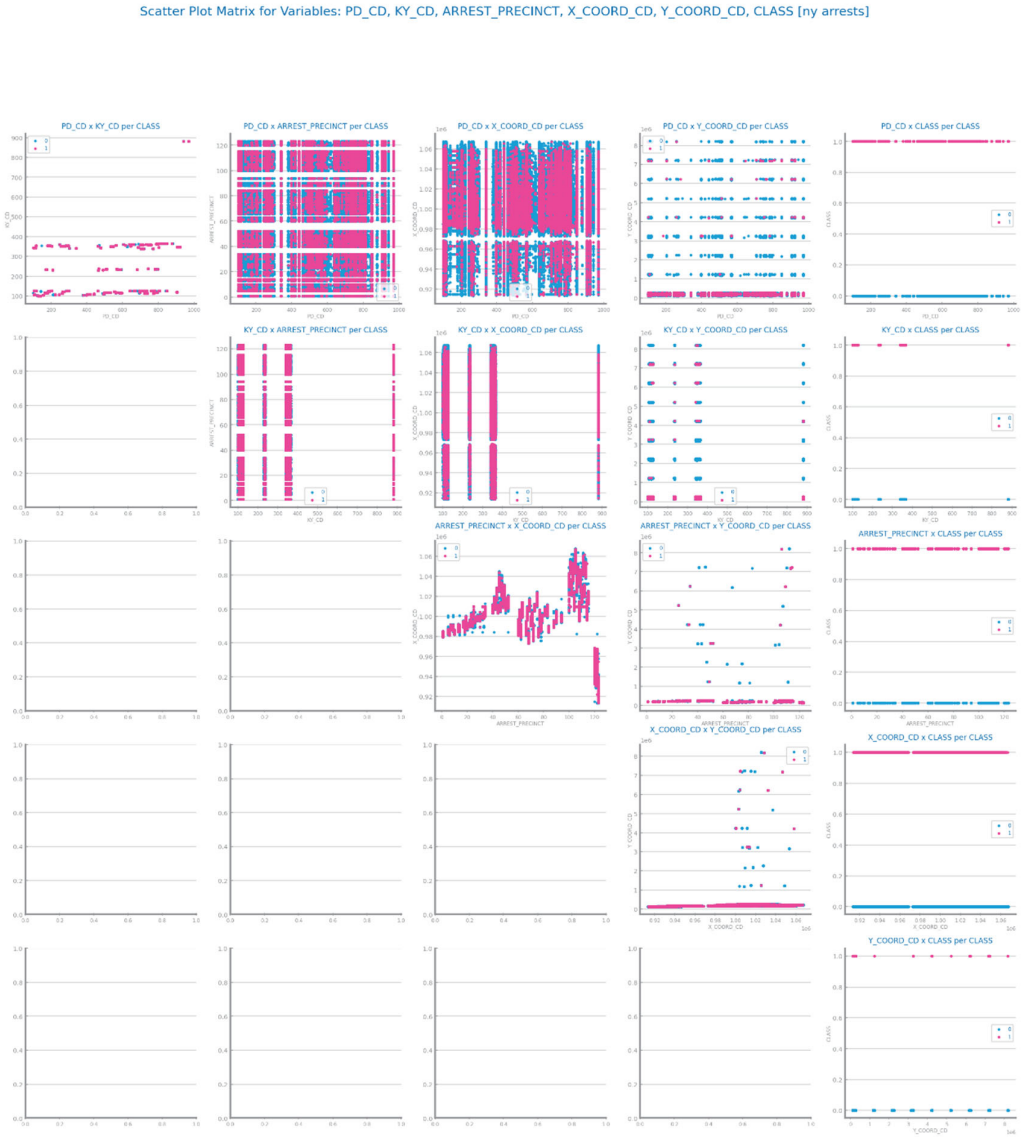
Figure 13 Granularity analysis for dataset 2

Data Sparsity

The following charts show the range of values in which the data are found, as well as how it is spread within that range, with different levels of domain coverage.

In Dataset 1, the binary symbolic target CLASS (target) was converted to binary numeric ("nonNY": 1, "NY": 0}). The charts reveal potential links between variables and with the class. In Dataset 2, some correlations among the 86 variables, including the class, were displayed.

From the heatmaps, no significant correlations with the class in both datasets were observed.



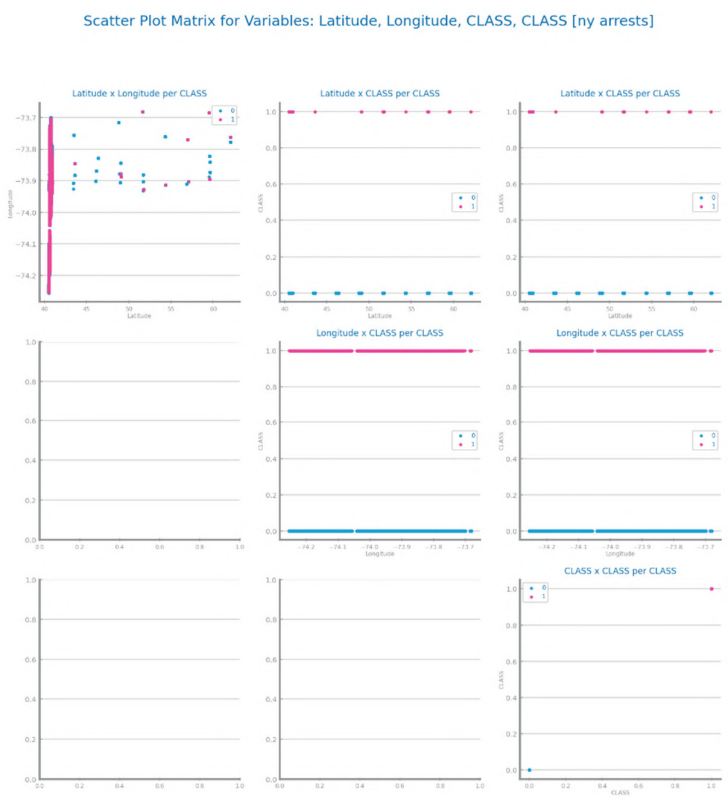


Figure 14 Sparsity analysis for dataset 1

Figure 15 Sparsity analysis for dataset 2

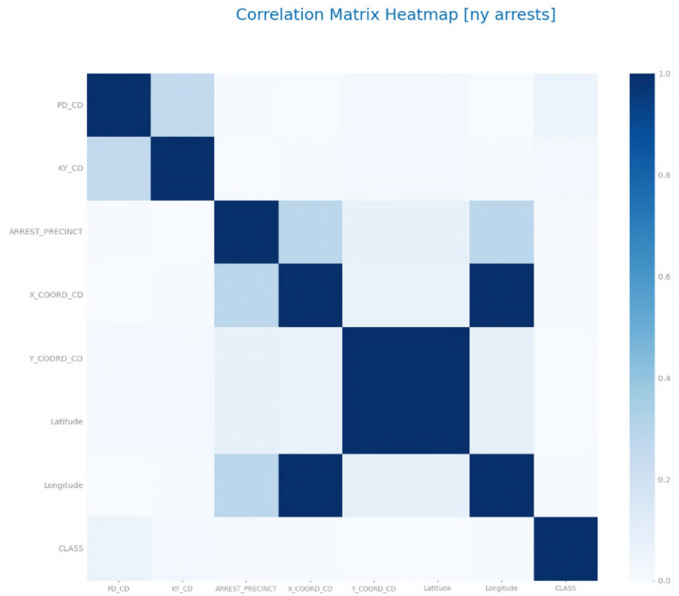


Figure 16 Correlation analysis for dataset 1

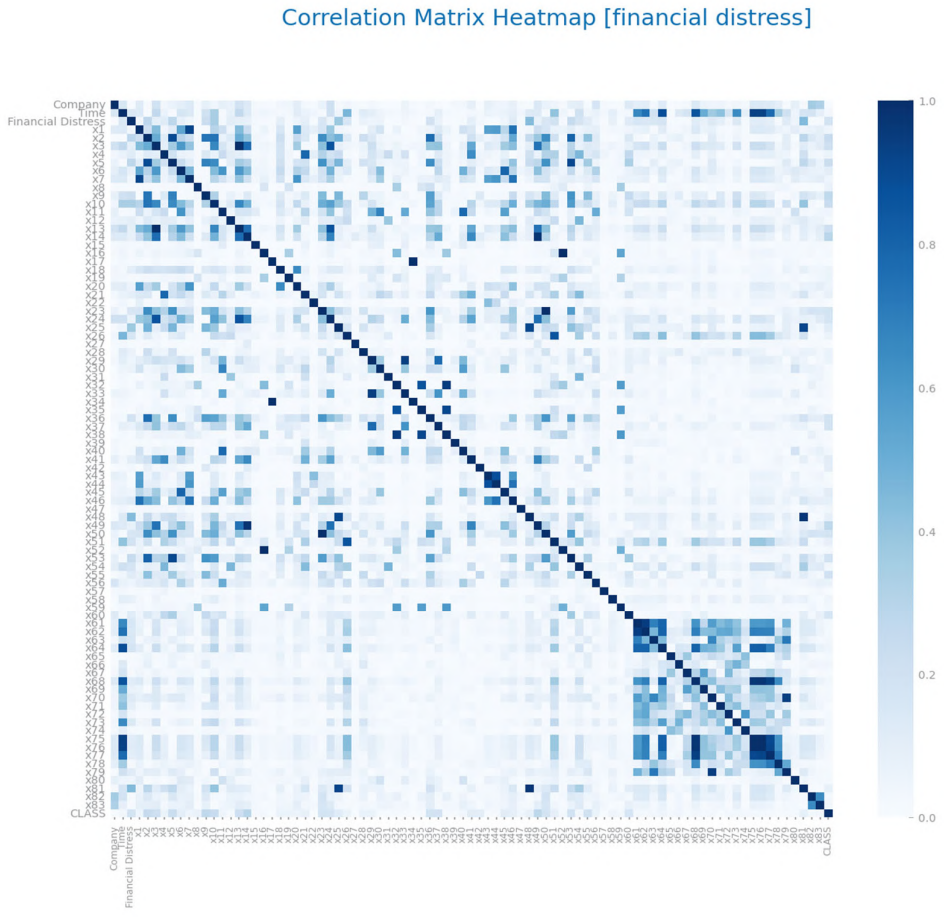


Figure 17 Correlation analysis for dataset 2

2 DATA PREPARATION

Dataset 1 required extensive preprocessing, including encoding, scaling, and missing data handling, while Dataset 2 required minimal adjustments due to its numerical nature. Outliers were removed in Dataset 2 to improve model performance but retained in Dataset 1 due to symbolic meanings. Both datasets used scaling to enhance algorithm efficiency, especially for Neural Networks. Oversampling was applied to address class imbalance, though it poses risks like overfitting. Variable reduction improved efficiency but required careful thresholds to avoid losing critical information.

Variables Encoding

In Dataset 1, the symbolic variables PD_DESC and OFNS_DESC are already encoded by the numerical variables PD_CD and KY_CD, respectively.

The ARREST_DATE variable underwent a feature extraction technique (year, quarter and month). The years were encoded ordinally, and the quarters and months were encoded using sine and cosine. The variable ARREST_DATE was dropped.

The binary variables PERP_SEX and LAW_CAT_CD have been encoded with the values "M": 1, "F": 0.

The unique values of the variables ARREST_BORO and LAW_CODE (legal codes), due to limitations in domain understanding, have been alphabetically ordered and mapped ordinally with integers.

The variable AGE_GROUP has been mapped ordinally with integers according to the ascending order of age intervals.

The unique values of the variable PERP_RACE, which indicates the ethnicity of the perpetrators, have been mapped with integers based on a logical order derived from historical dispersion and human evolution, considering that, according to various studies, human behavior is strongly influenced by genetics and culture, shaped over the course of human history.

Since Dataset 2 only contains numerical variables and a binary target variable with integer values, no encoding was required.

Missing Value Imputation

Except for the variables PD_DESC and OFNS_DESC, which will later be removed for being redundant in relation to PD_CD and KY_CD, respectively, the Dataset 1 has a number of rows with at least one missing value of 101.048. Approaches to impute missing values with the mean or median, or simply eliminate their items, resulted in the conclusion that the best approach was:for the Dataset 1, to eliminate items with missing values; for the Dataset2, to impute with the mean or median, so we opted for the median. However, it is important to note that imputing with the mean or median can reduce data variability and introduce bias, especially if the missing data is not random. It also ignores correlations between variables and may not be ideal for temporal or sequential data.

Dataset 2 does not have any missing values.

The elimination of missing values improves the performance of sensitive algorithms such as Naive Bayes and KNN. In general, there is a potential bias if the missing values are not random.

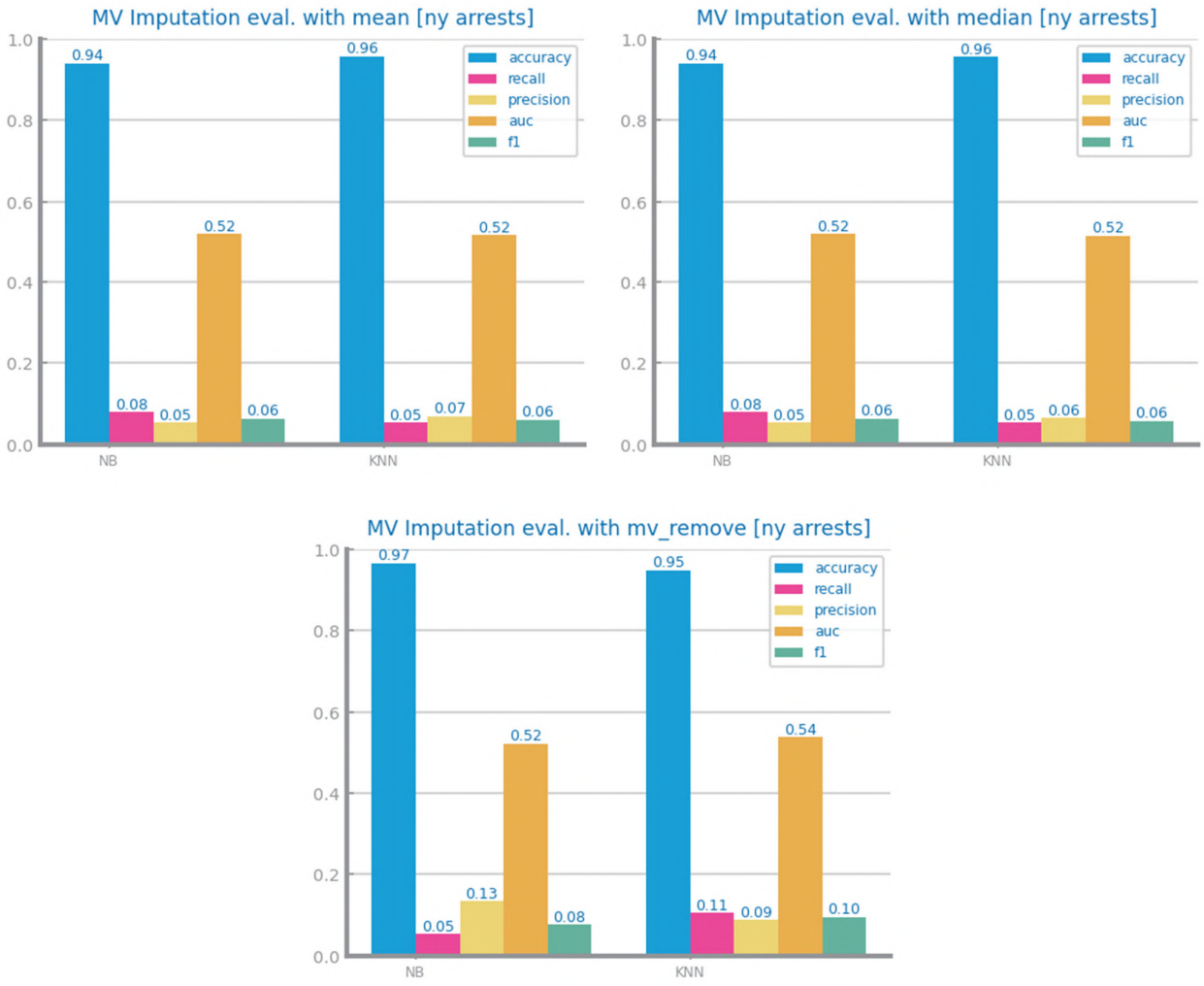


Figure 18 Missing values imputation results with different approaches for dataset 1

No approaches were performed for missing values imputation on Dataset 2, as this dataset does not have missing values.

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

Outliers Treatment

In Dataset 1, none of the numerical variables were subjected to outlier removal, as the values of these variables have symbolic meaning (codes).

In Dataset 2, several outlier imputation techniques were tested (mean, median and outliers removal). This technique was applied to the numeric variables, except for Company and Time, which have symbolic and ordinal meanings, respectively. The results showed that, overall, the best metrics result from the removal of outliers.

Eliminating outliers can improve the accuracy of sensitive algorithms such as KNN, Naive Bayes, Decision Trees, Random Forests, Gradient Boosting, and Multilayer Perceptrons, but it may lead to a loss of variability, removal of rare patterns, and introduction of bias if the definition of outliers is not appropriate.

N/A

Figure 20 Outliers imputation results with different approaches for dataset 1

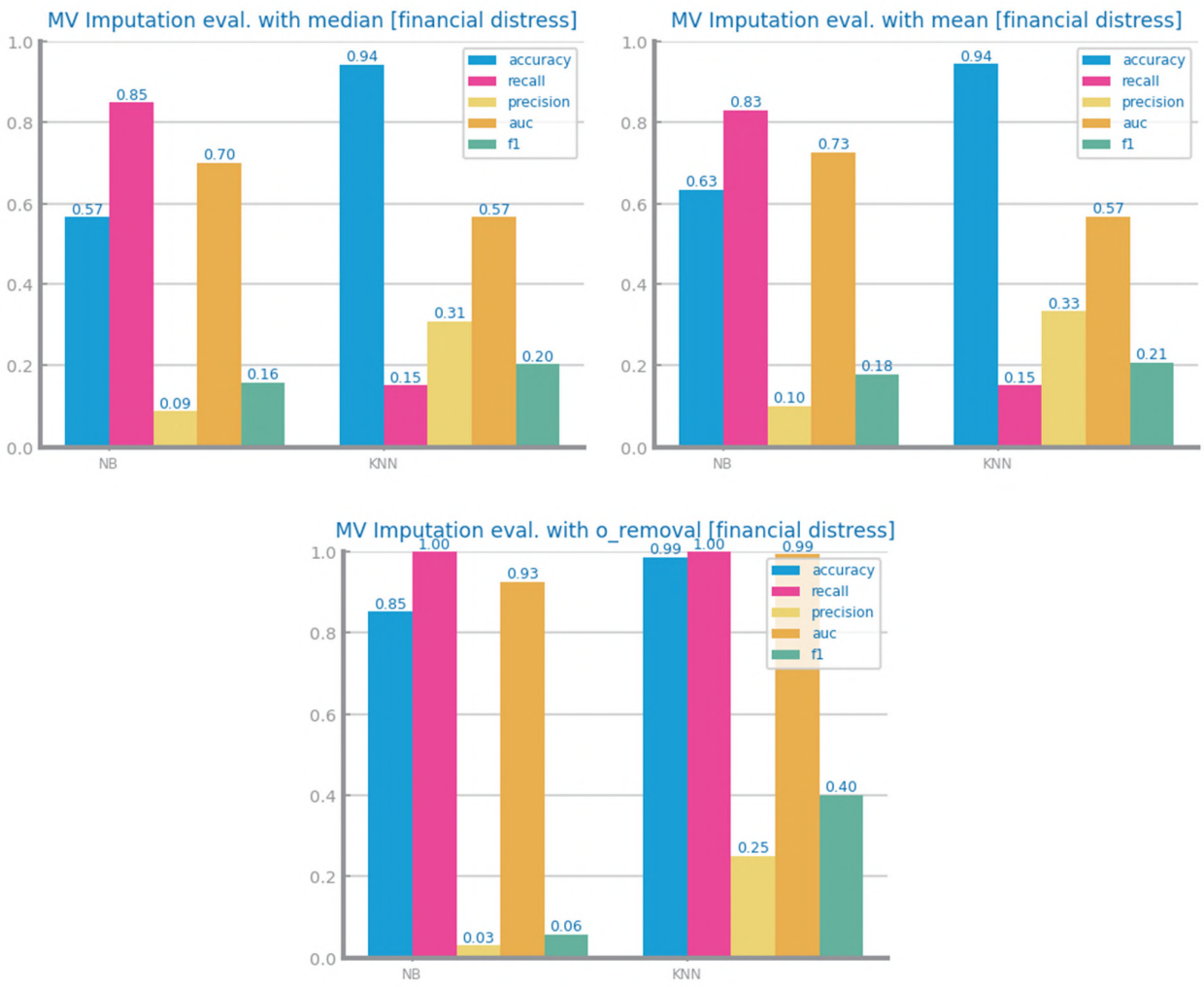


Figure 21 Outliers imputation results with different approaches for dataset 2

Scaling

In Dataset 1, Z-Score scaling (StandardScaler) was chosen due to its significantly better metrics. In Dataset 2, MinMax scaling was used because of the slightly better metrics compared to Z-Score.

Although it may be unnecessary for Naïve Bayes, Random Forests, and Decision Trees, scaling the data improves the performance of KNN, Gradient Boosting, and Multilayer Perceptrons, accelerating convergence and avoiding bias from variables with large magnitudes. Especially, scaling prevents extreme values that could negatively affect convergence in neural networks, one of the algorithms we will use, particularly when using activation functions like ReLU, which can be sensitive to large input values.

It is true that scaling is necessary to run algorithms like KNN and Neural Networks; however, it is also important to note that while scaling assigns the same importance to each variable, it also removes the original significance of the most relevant ones.

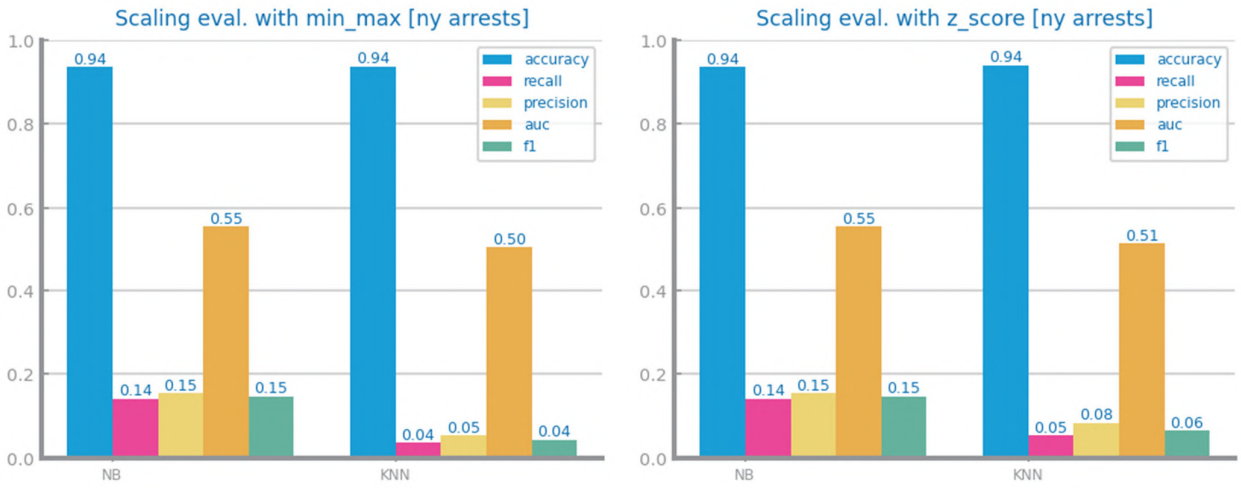


Figure 22 Scaling results with different approaches for dataset 1

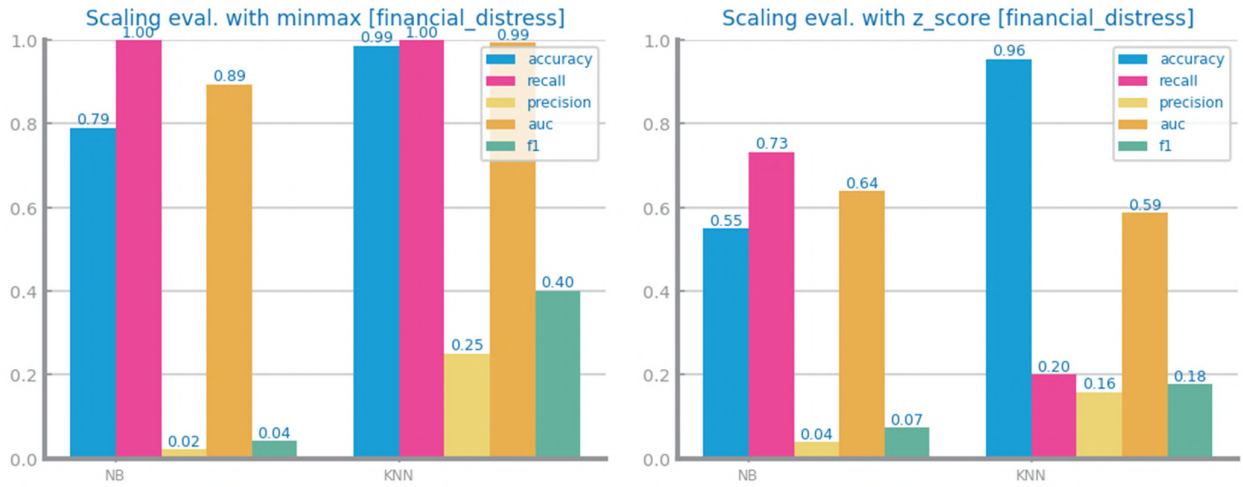


Figure 23 Scaling results with different approaches for dataset 2

Balancing

In both datasets, oversampling the trainset was chosen, as the metrics are more favorable to this technique, especially in the case of the KNN algorithm.

This technique is advantageous in that the model can better learn the patterns of the minority class, improving its ability to correctly predict those samples. However, it is important to be cautious about the risk of overfitting due to the duplication of instances or the creation of very similar examples, increased training time, loss of variability, and the increase of noise in the minority class.

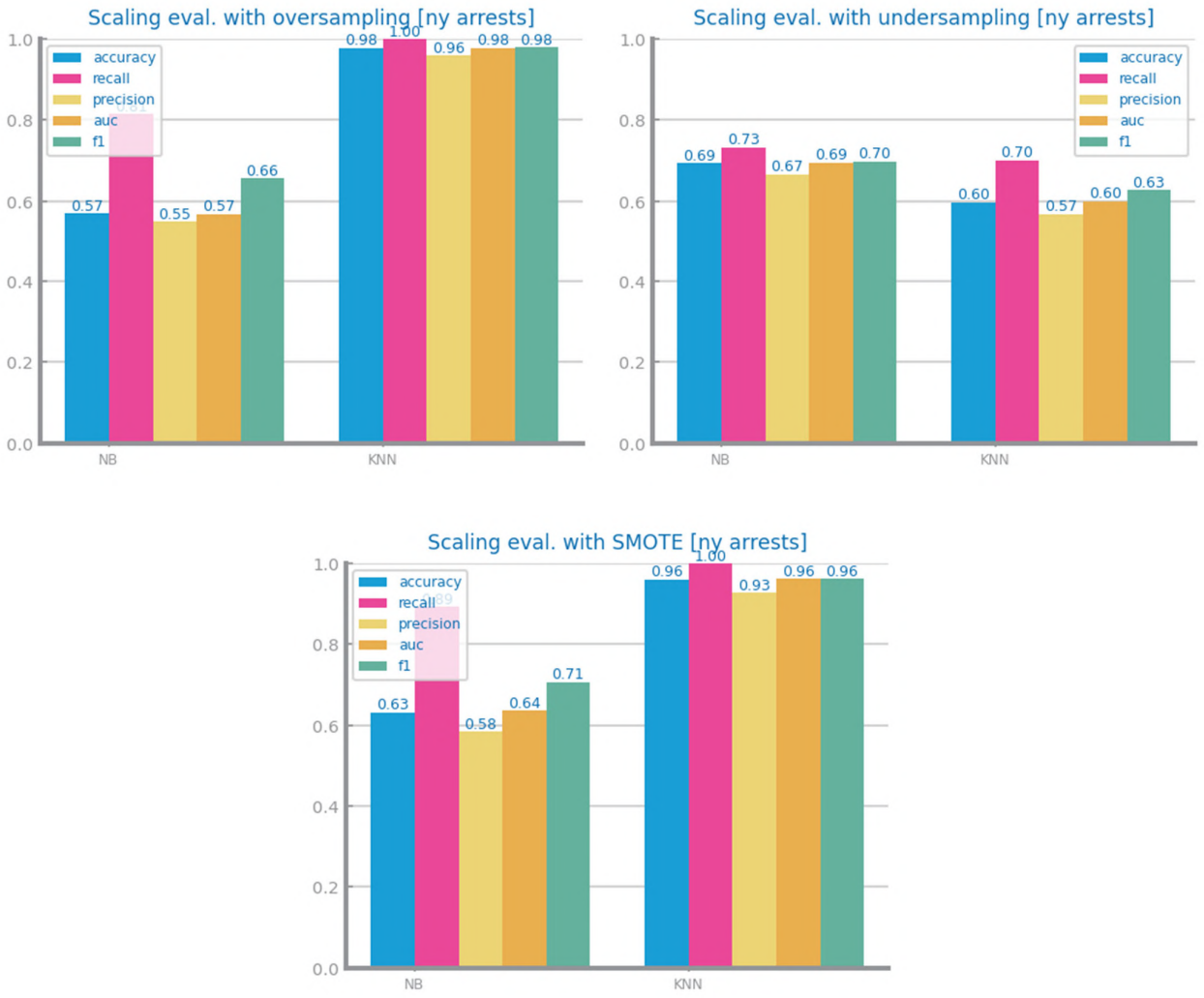
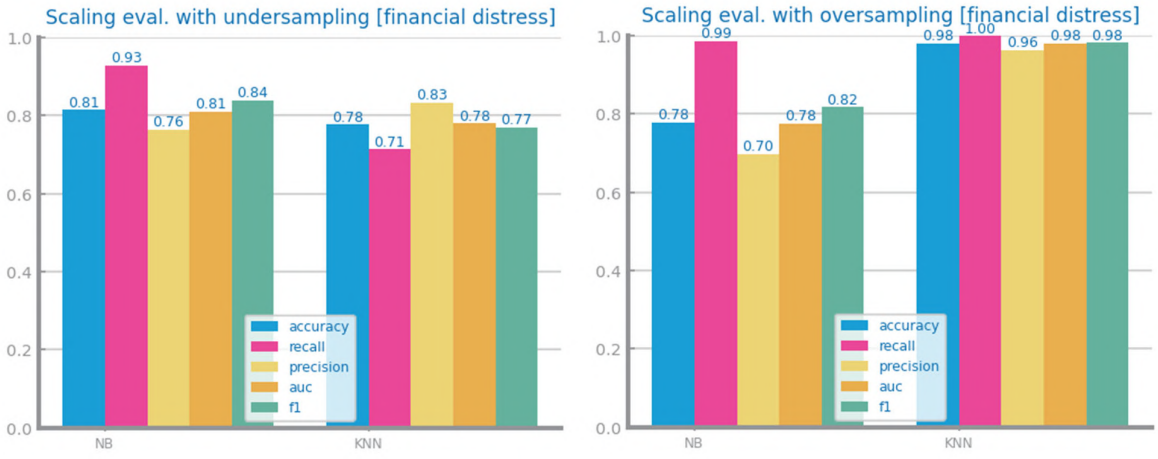


Figure 24 Balancing results with different approaches for dataset 1



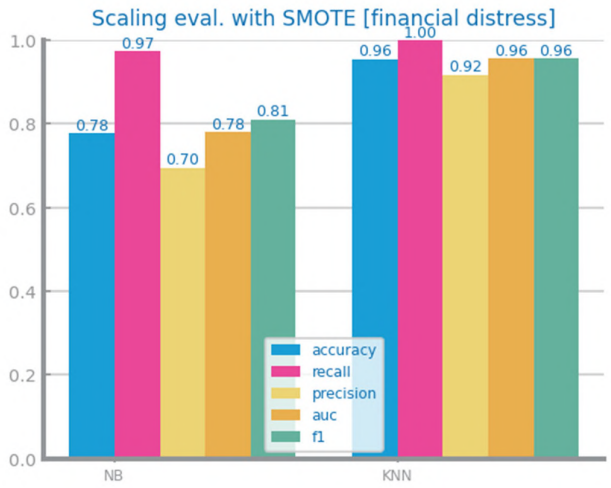
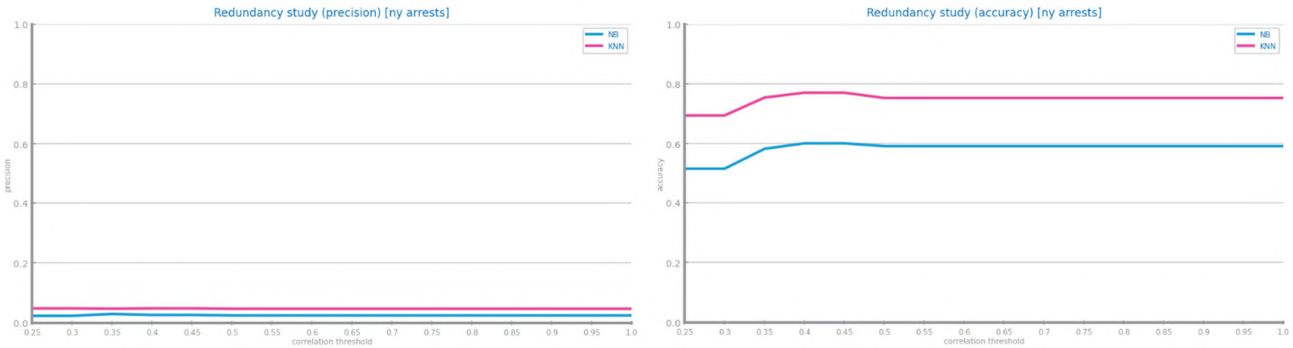


Figure 25 Balancing results with different approaches for dataset 2

Feature Selection

In Dataset 1, the variables PD_DESC, OFNS_DESC, Latitude, and Longitude were discarded as they were already encoded in the original variables PD_CD, KY_CD, X_COORD_CD, and Y_COORD_CD, respectively. The redundancy study using the metrics accuracy, precision and recall was inconclusive regarding the elimination of other redundant variables; similarly, the variance study with the same metrics was inconclusive regarding the removal of irrelevant variables.

In Dataset 2, the variable Financial Distress, with its high correlation (74.19%) with the target class, was removed because its data is used to obtain the target through a threshold-based classification (data leakage phenomenon). The redundancy study using the metrics accuracy, precision and recall led to the rejection of variables with correlations above a threshold of 0.85, resulting in the removal of 22 redundant variables. The variance study with the same metrics led to the rejection of variables variance below a threshold of 0.075, resulting in the removal of 55 irrelevant variables. The 9 most important features remained.



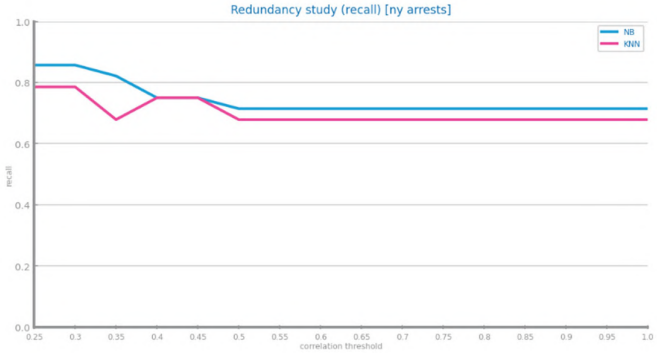
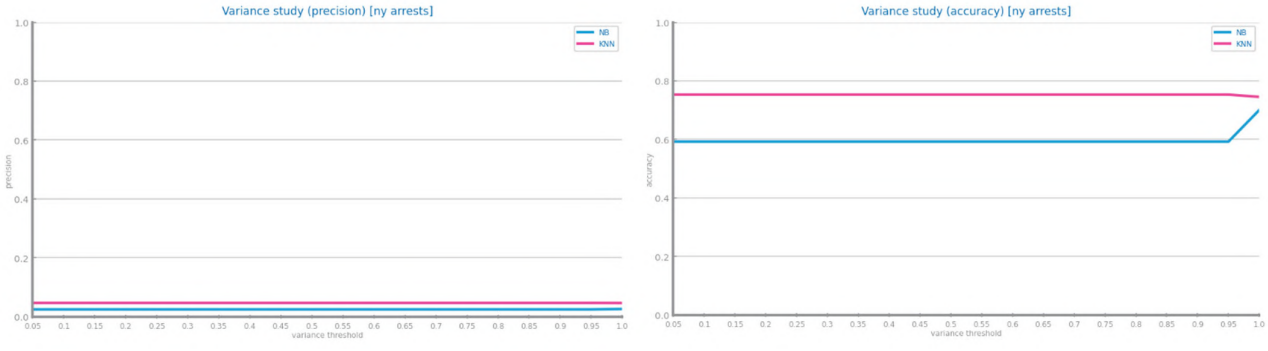


Figure 26 Feature selection of redundant variables results with different parameters for dataset 1

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



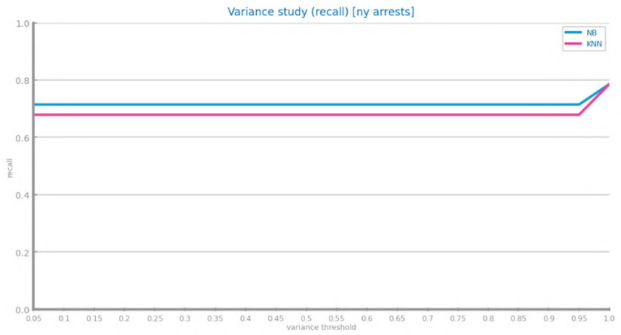


Figure 28 Feature selection of relevant variables results with different parameters for dataset 1 (variance study)

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

Feature Extraction (optional)

Shall contain all relevant information and charts respecting to feature extraction, in particular PCA. The different choices and their impact on the modelling results shall be presented and explained. **Shall not exceed 200 characters.**

Figure 30 Principal components analysis and feature extraction results for dataset 1

Figure 31 Principal components analysis and feature extraction results for dataset 2

Additional Feature Generation (if done)

Shall contain all relevant information and charts respecting to feature generation. The different choices and their impact on the modelling results shall be presented and explained. Shall summarise all variables generated and the formula used to derive them (in a table). **Shall not exceed 200 characters.**

Figure 32 Feature generation results for dataset 1

Figure 33 Feature generation results for dataset 2

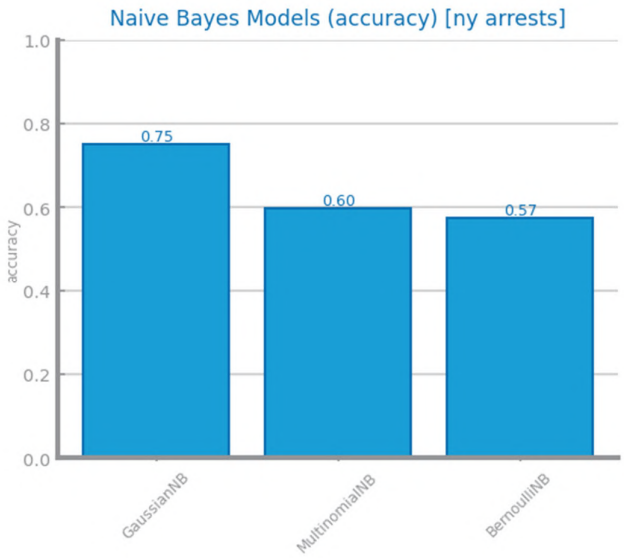
3 MODELS' EVALUATION

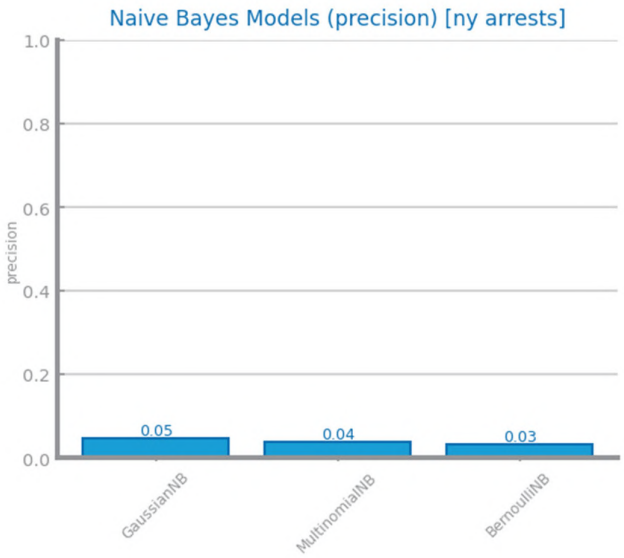
The analysis of both datasets revealed distinct model performance patterns and optimization challenges. Dataset 1 demonstrated moderate-high accuracy but struggled with precision due to class imbalance, while Dataset 2 exhibited similar accuracy with overfitting concerns in certain configurations. Techniques like KNN, Random Forests, Gradient Boosting, and Neural Networks were tuned through hyperparameters such as depth, learning rate, and k-values. The evaluation emphasized the balance between accuracy, precision, and recall while addressing overfitting and class imbalance issues.

Naïve Bayes

In Dataset 1, the model achieved middle-high accuracy and median recall, but low precision. This indicates that it successfully identifies most true positive cases (high recall) but makes many errors by classifying negative cases as positive (low precision). This means that while the model appears to perform well overall (high accuracy), it is not reliable when it comes to distinguishing between positive and negative instances, leading to a higher number of false positives.

Since Dataset 2 contains only a few thousand items, cross-validation (StratifiedKFold) with n_splits=10 was used. In Dataset 2, the model achieved middle-high accuracy and recall, but low precision. It indicates that while the model correctly identifies most of the positive cases (high recall), it also incorrectly labels many negative cases as positive (low precision). This means the model struggles with minimizing false positives.





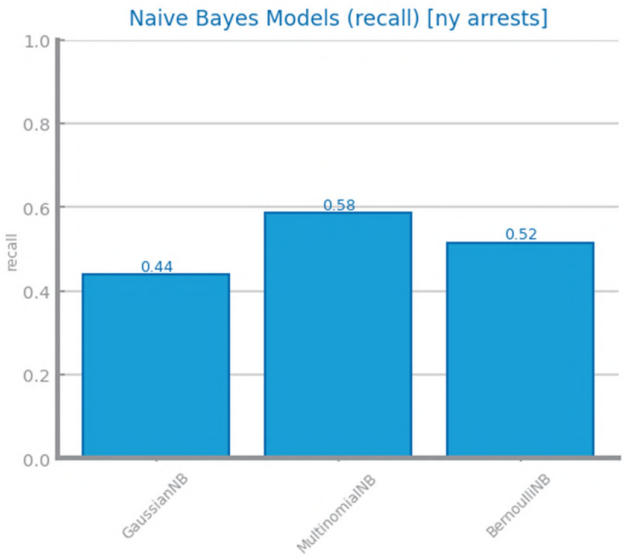


Figure 34 Naïve Bayes alternatives comparison for dataset 1

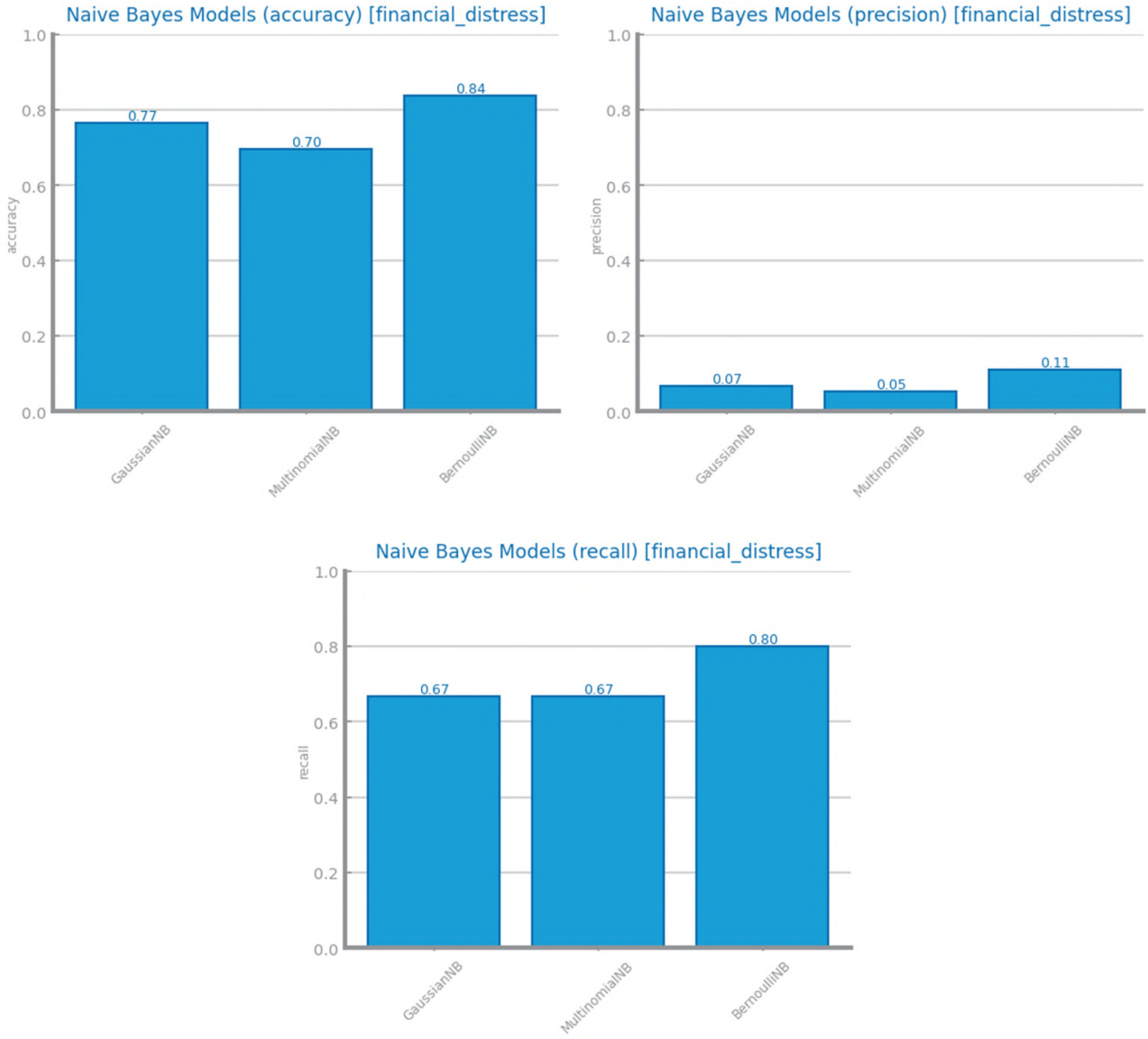


Figure 35 Naïve Bayes alternative comparison for dataset 2

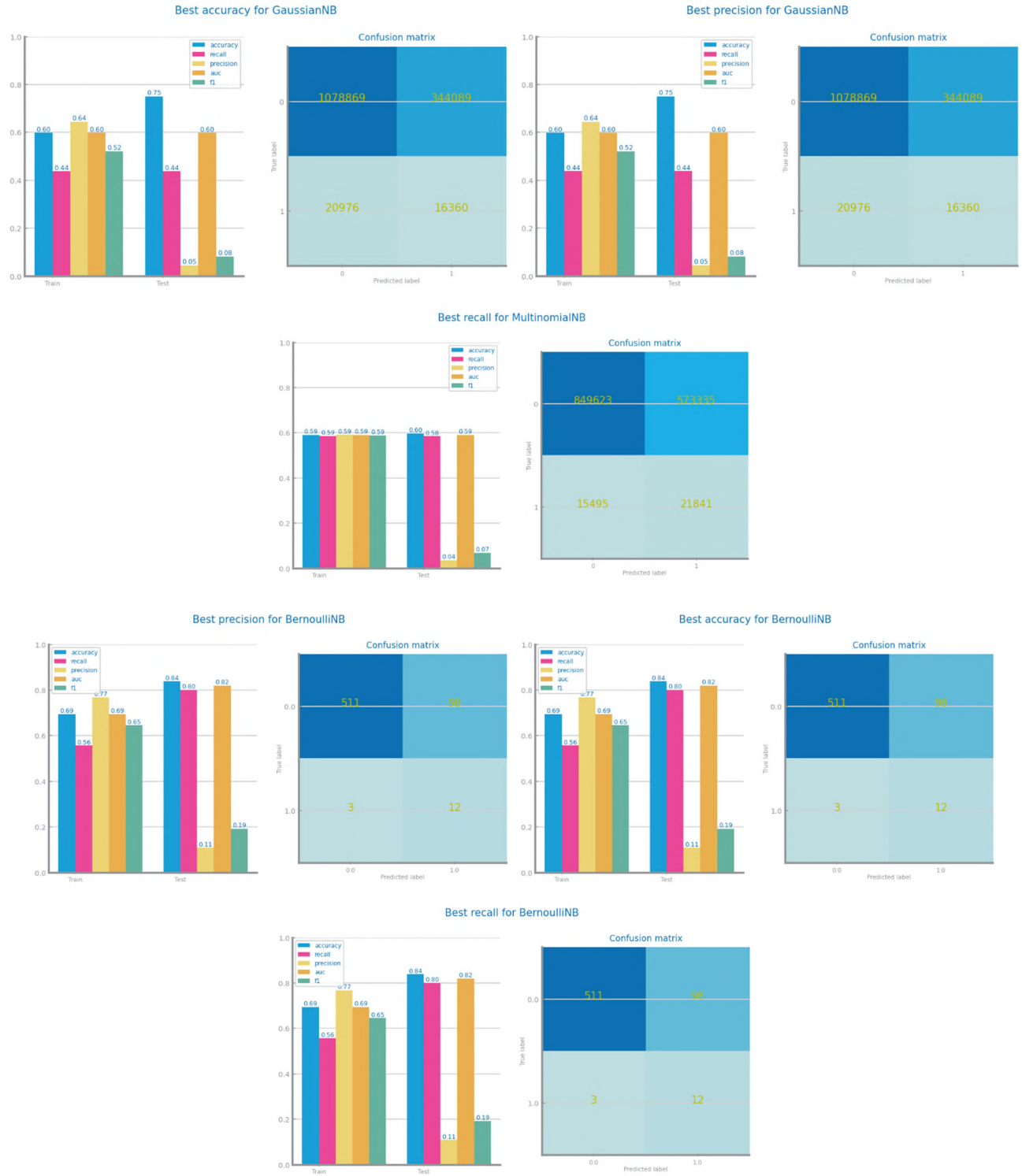


Figure 36 Naïve Bayes best model results for dataset 1 (above) and dataset 2 (below)

KNN

In Dataset 1, the K-Nearest Neighbors (KNN) model achieved its best performance when the number of neighbors (k) was set to 9, using Manhattan distance as the metric. This configuration resulted in both good accuracy and precision, indicating a balanced capability to correctly classify both positive and negative cases. Additionally, the choice of k=9 appeared to mitigate the risk of overfitting, as the accuracy levels on the training and test sets were closely aligned. This balance suggests that the model successfully generalized from the training data to unseen data, making k=9 an optimal value for this dataset.

In Dataset 2, the best results were obtained with k=1, also using Manhattan distance. While this configuration provided high accuracy and precision, it raised concerns about overfitting. A k value of 1 means that the model relies solely on the closest neighbor to make classifications. This approach can lead to an overly sensitive model, capturing noise and specific patterns in the training data rather than broader trends. Consequently, while the accuracy may be high, the model's generalization to new data could be compromised. Thus, while k=1 offers strong initial performance, it may not be ideal for a more robust, generalizable solution.

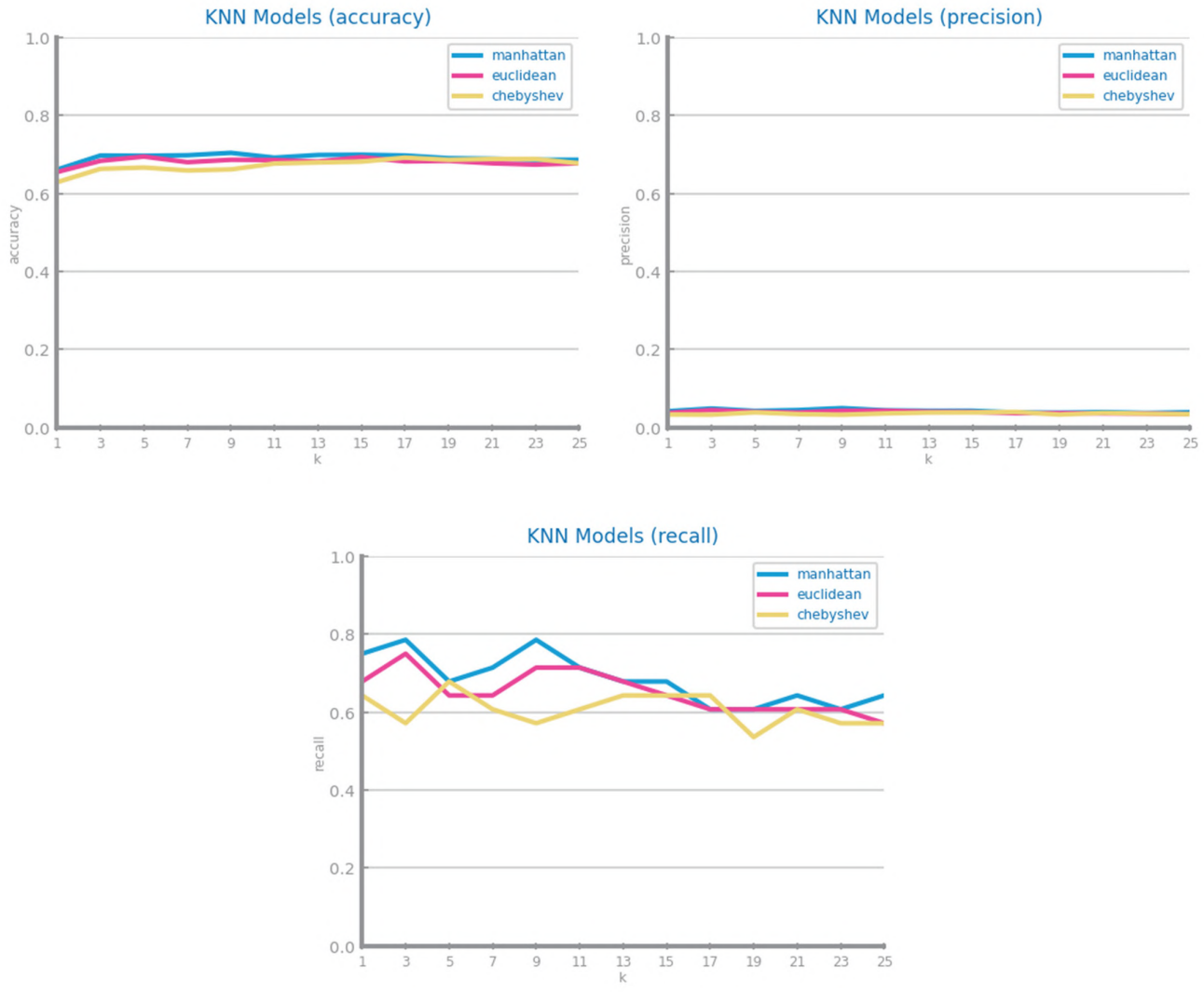


Figure 37 KNN different parameterizations comparison for dataset 1

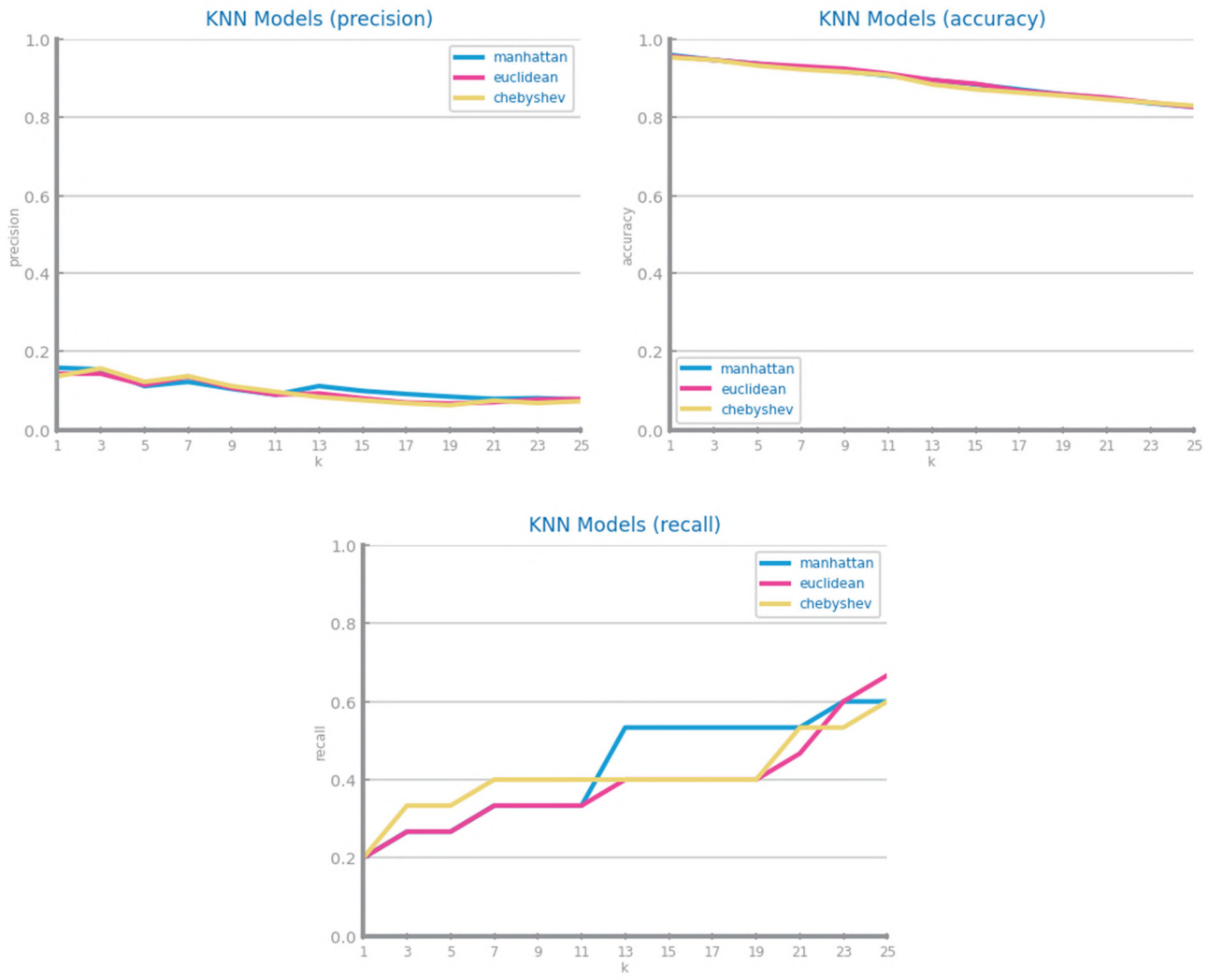


Figure 38 KNN different parameterizations comparison for dataset 2

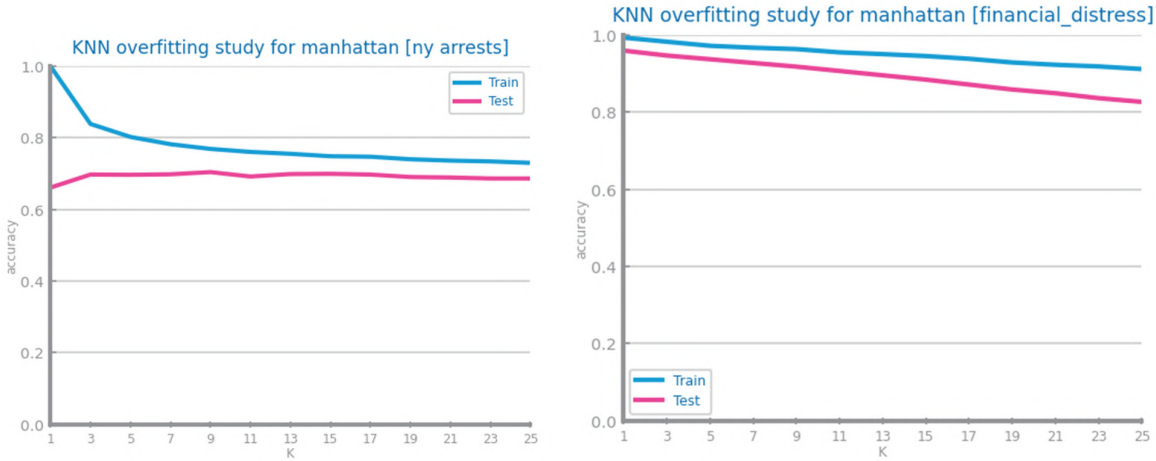


Figure 39 KNN overfitting analysis for dataset 1 (left) and dataset 2 (right)

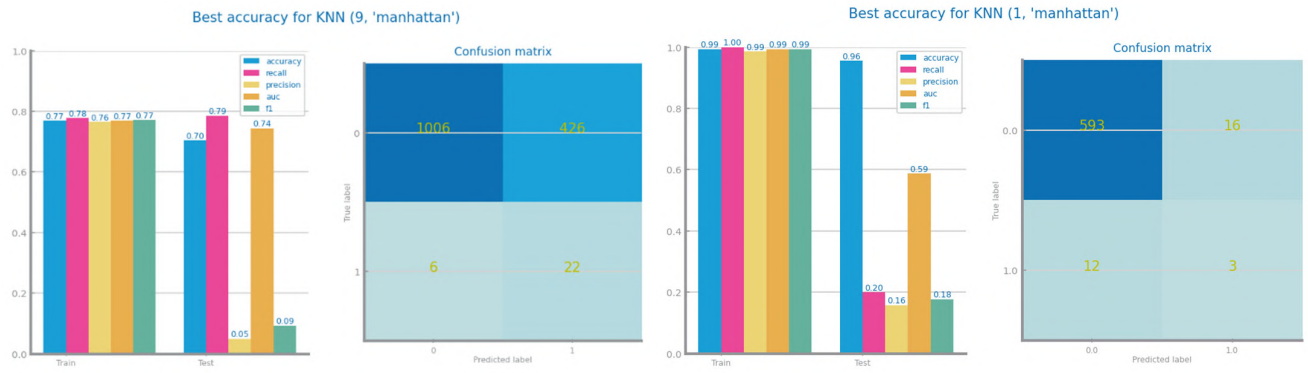


Figure 40 KNN best model results for dataset 1 (left) and dataset 2 (right)

Decision Trees

In Dataset 1, the model with a depth of 12 has a high risk of overfitting because the test set accuracy is higher than the training set accuracy, indicating the model may be memorizing details of the training data. This risk disappears starting from depth 17, when the model begins to generalize better.

In Dataset 2, the model with a depth of 6 does not face the risk of overfitting, simply because its complexity is low enough to avoid fitting too closely to the training data, maintaining good performance on both the training and test sets.

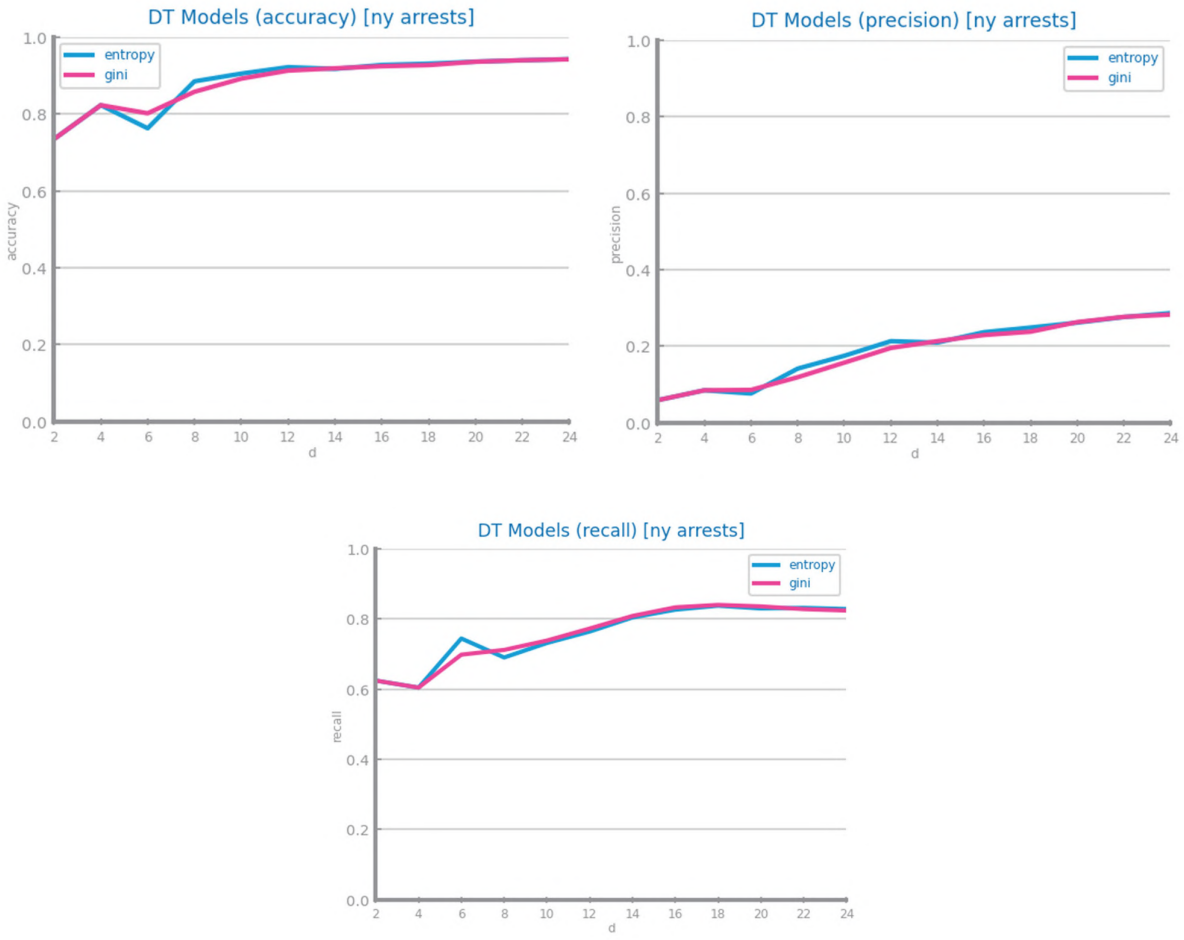


Figure 41 Decision Trees different parameterisations comparison for dataset 1

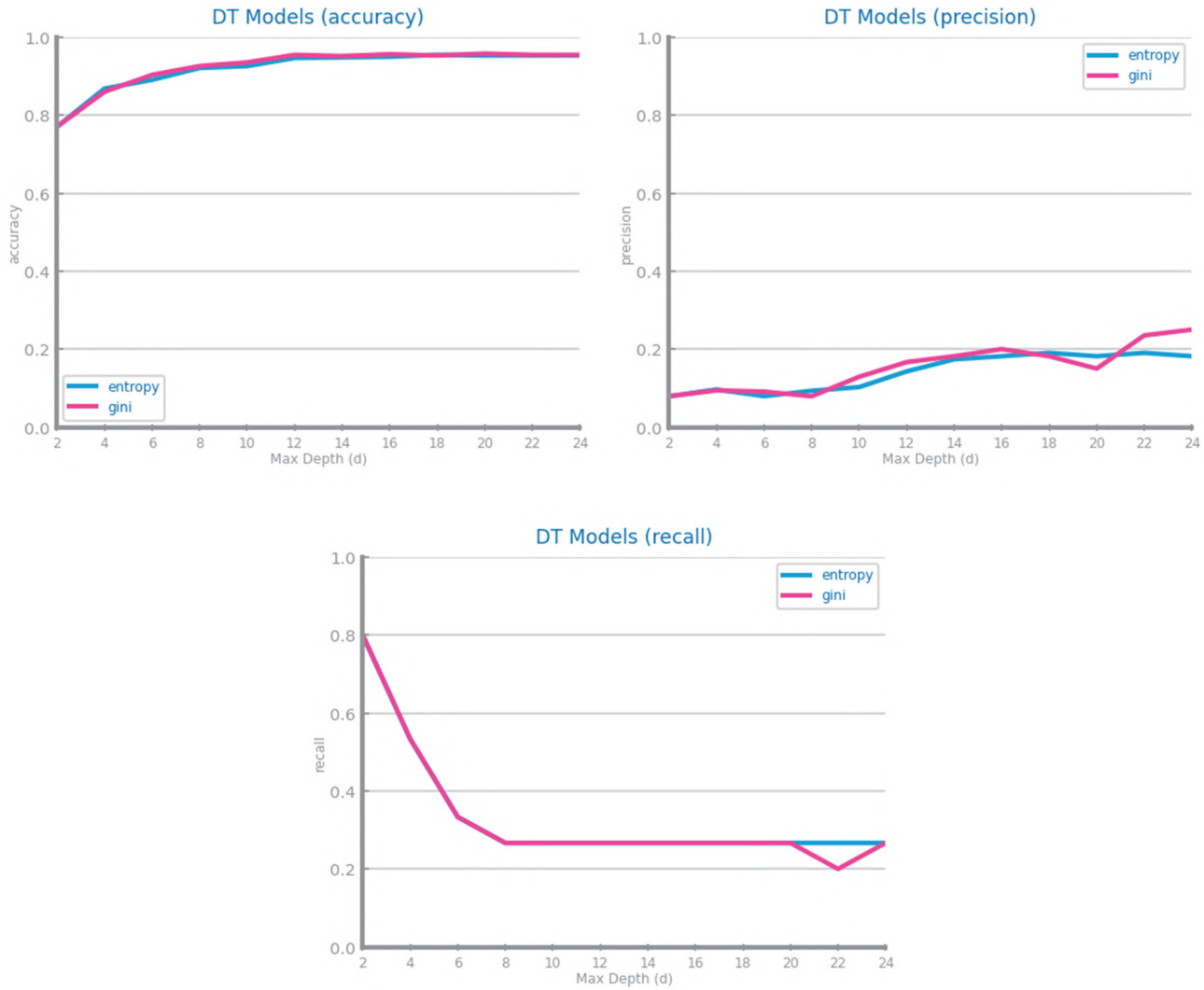


Figure 42 Decision Trees different parameterisations comparison for dataset 2

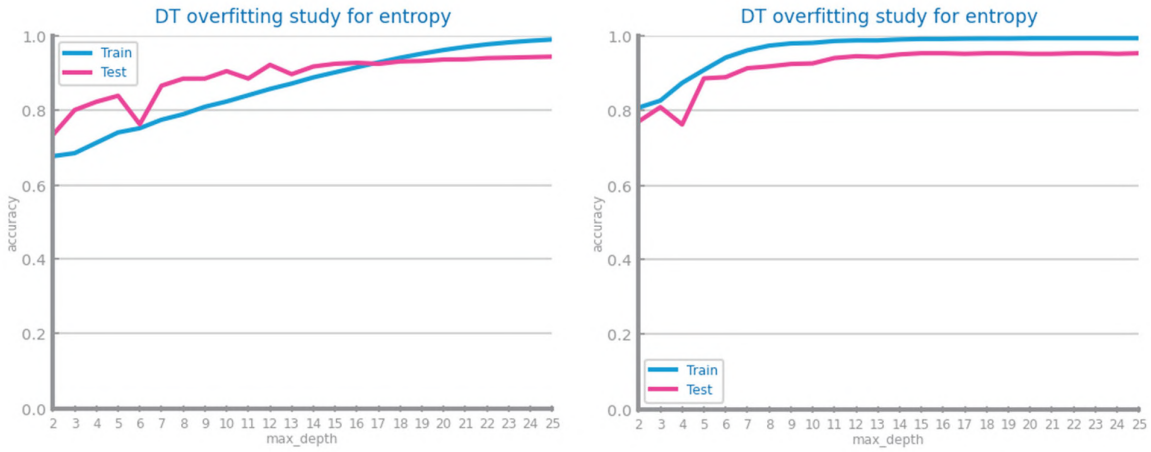


Figure 43 Decision Trees overfitting analysis for dataset 1 (left) and dataset 2 (right)

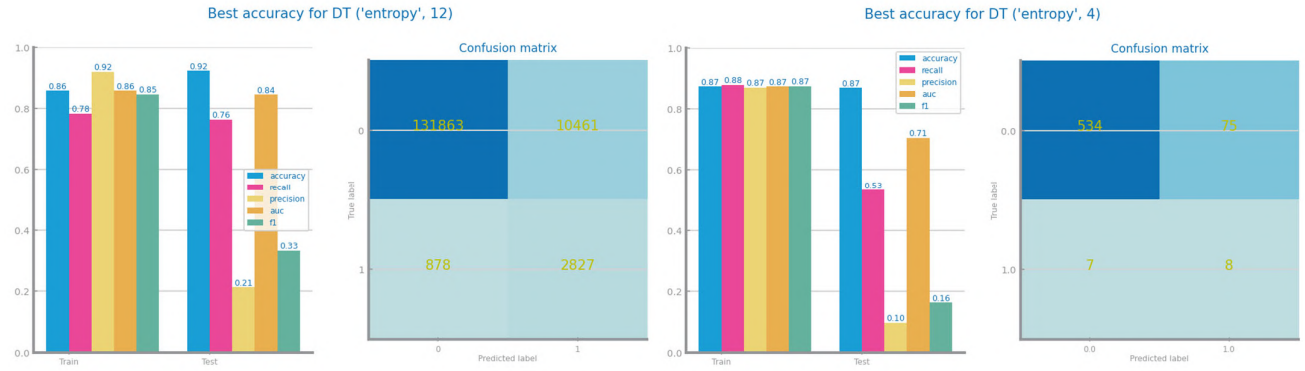


Figure 44 Decision trees best model results for dataset 1 (left) and dataset 2 (right)

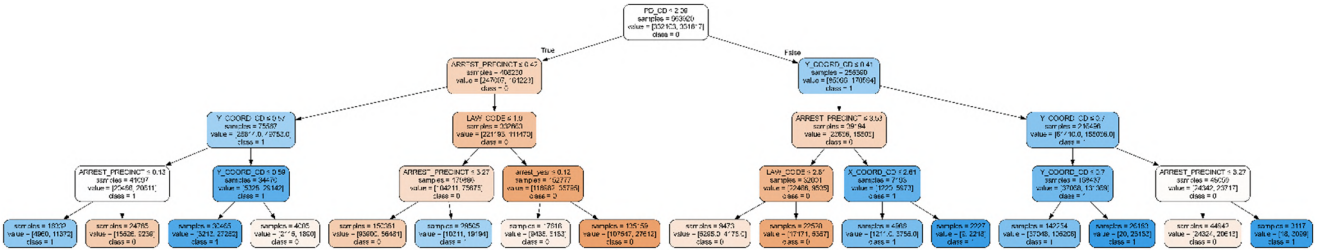


Figure 45 Best tree for dataset 1

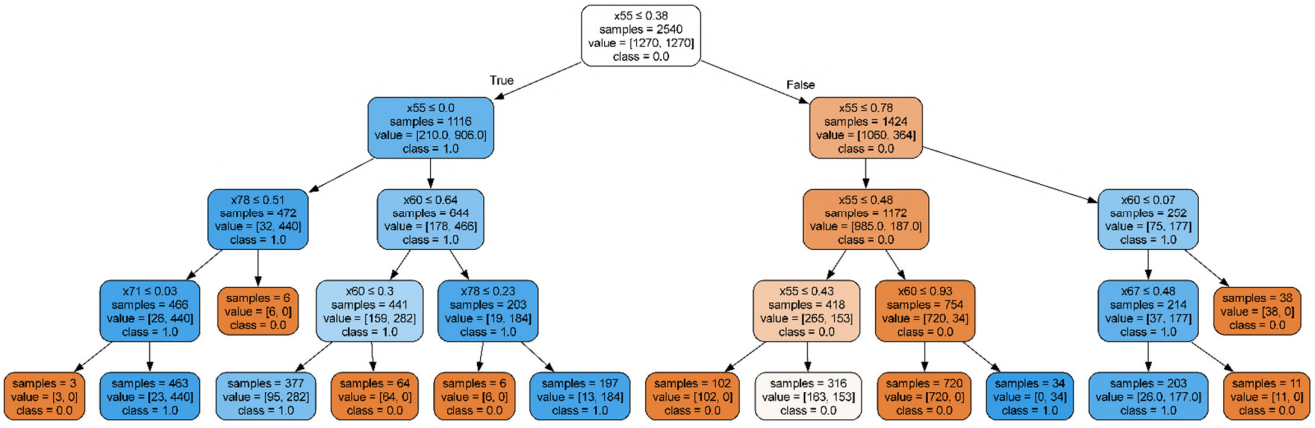


Figure 46 Best tree for dataset 2

Random Forests

In Dataset 1, the best metrics are obtained through the highest depth tested (7), although from around nr_estimators=500 there is a risk of overfitting. Therefore, one might opt for a smaller depth. Despite the high accuracy, the precision remains relatively low for the test set. The most important variables are Y_COORD_CD, ARREST_PRECINCT, and X_COORD_CD, which are thematically related to the class (NY and non-NY jurisdictions), as well as PD_CD, LAW_CODE, KY_CD, arrest_year, and ARREST_BORO (the latter also being geographical).

In Dataset 2, the optimal metrics are achieved at the maximum tested depth (7), with the overfitting analysis showing no evident signs of overfitting. The model demonstrates high accuracy but relatively low precision and recall when the Random Forest algorithm is applied to the test set. Low precision and recall can result from imbalanced data, noisy information, underfitting, or poor hyperparameter tuning. These can be addressed by balancing the data, cleaning the dataset, or refining the model and features. The most influential variables are x55 and x60.

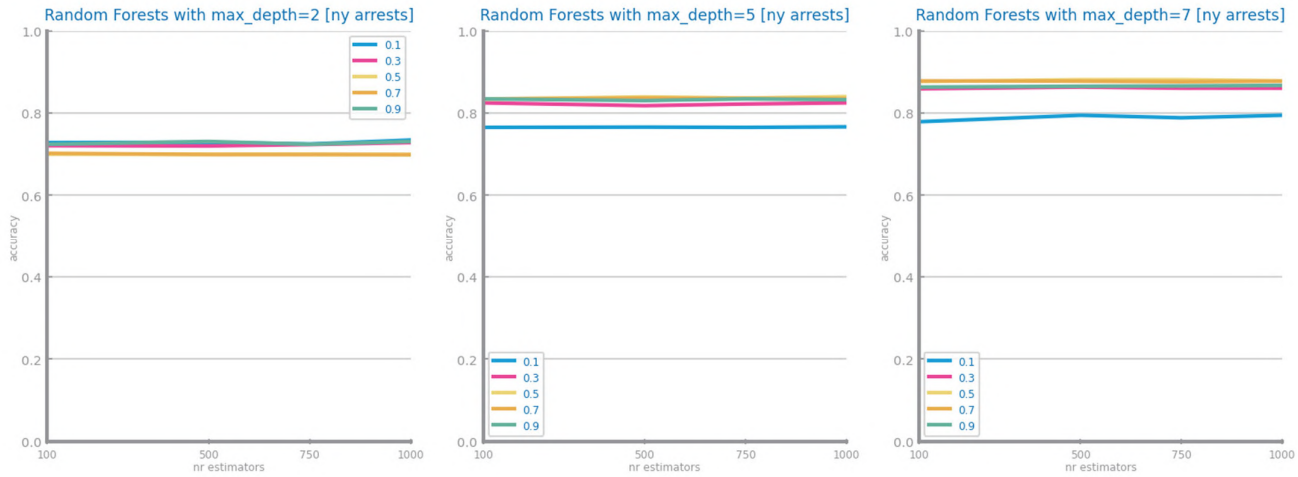


Figure 47 Random Forests different parameterizations comparison for dataset 1

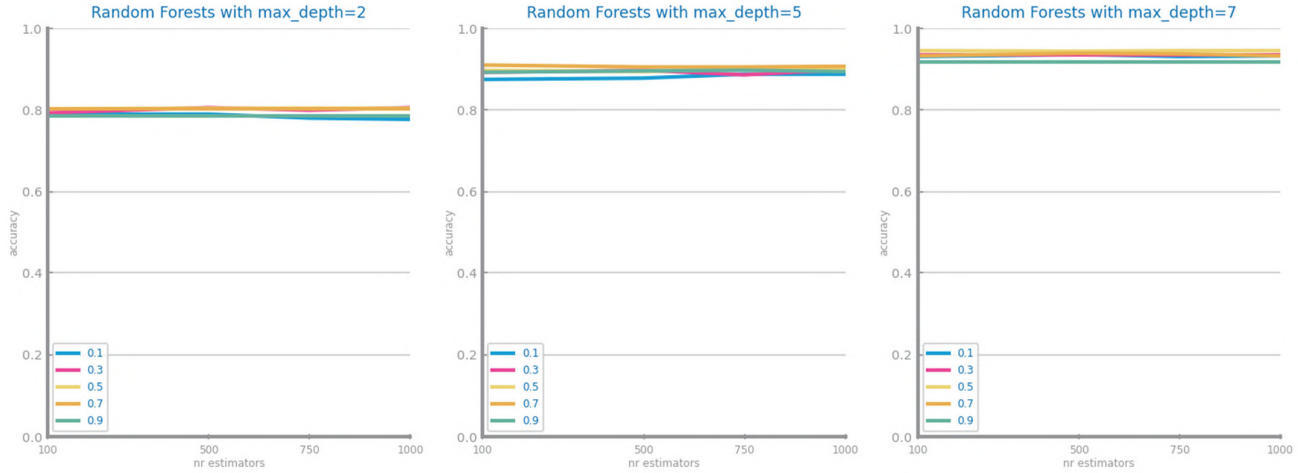


Figure 48 Random Forests different parameterizations comparison for dataset 2

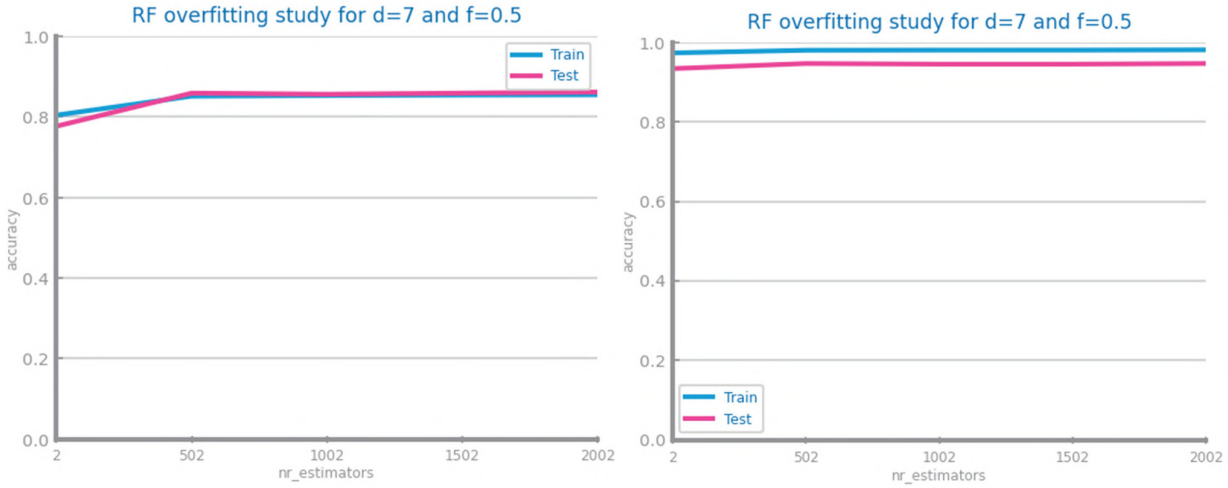


Figure 49 Random Forests overfitting analysis for dataset 1 (left) and dataset 2 (right)

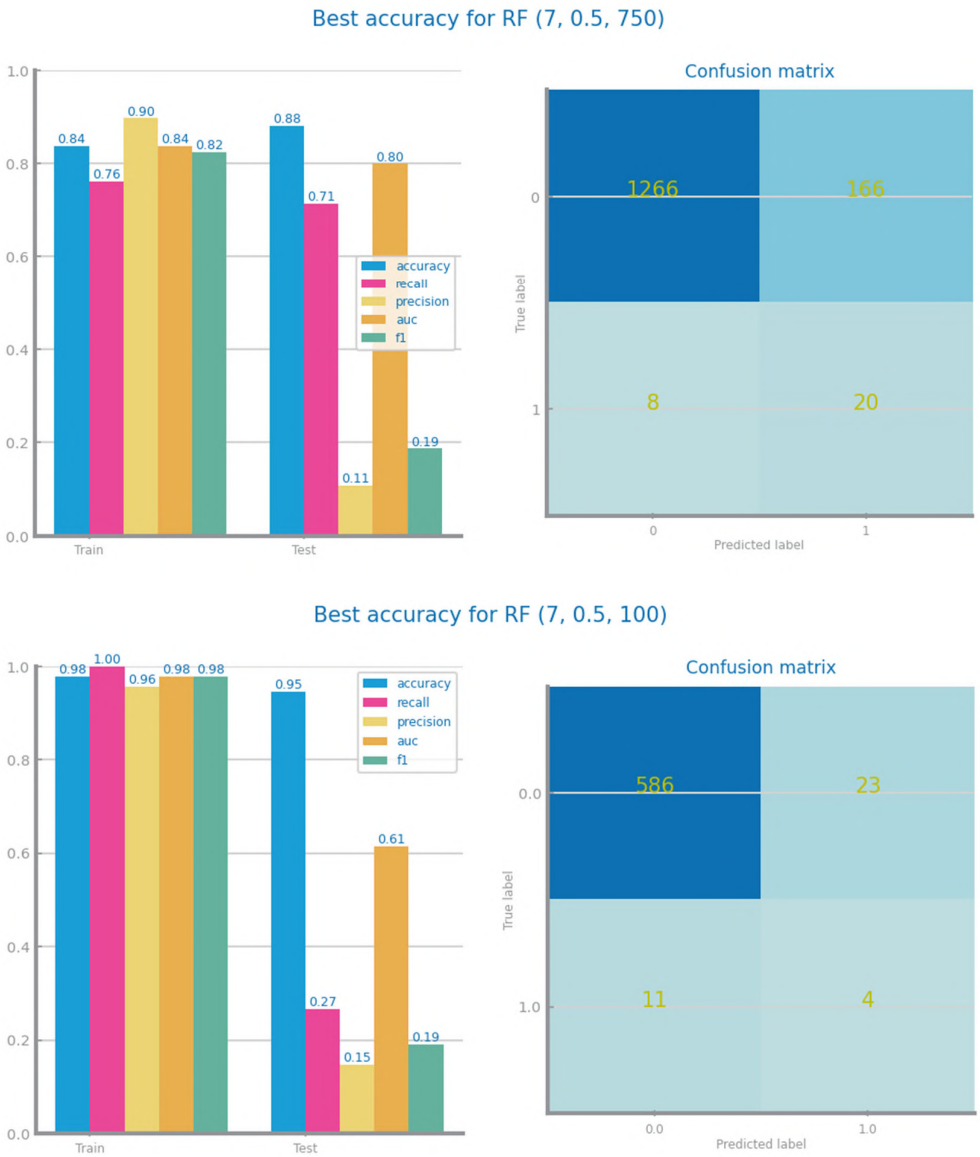


Figure 50 Random Forests best model results for dataset 1 (above) and dataset 2 (below)

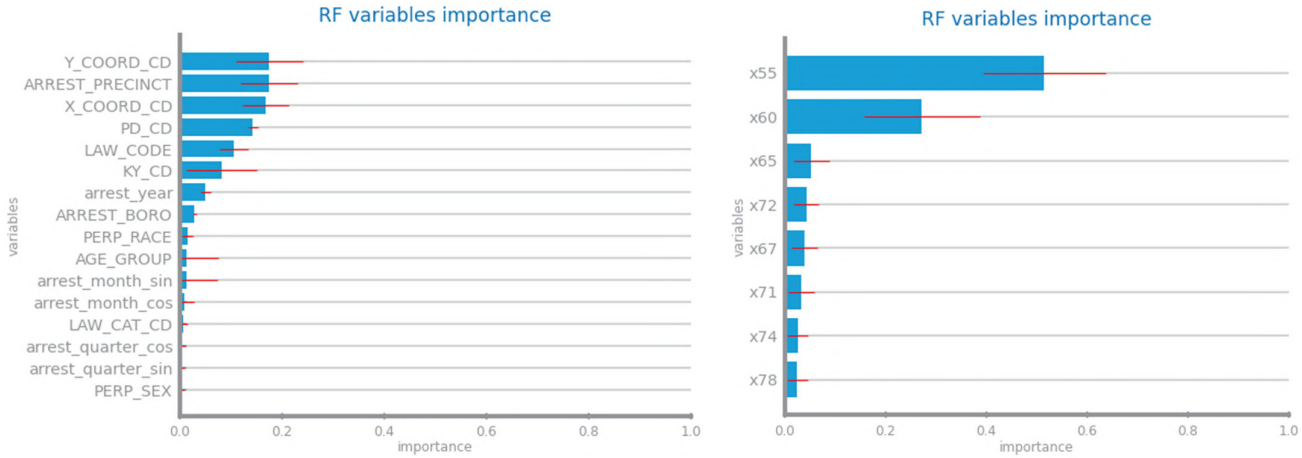


Figure 51 Random Forests variables importance for dataset 1 (left) and dataset 2 (right)

Gradient Boosting

In Dataset 1, the best metrics are obtained through the highest depth tested (7), although up to nr_estimators=500 there is a risk of overfitting. Despite the high accuracy, the precision remains relatively low for the test set. The most important variables are Y_COORD_CD, X_COORD_CD, and ARREST_PRECINCT, which are thematically related to the class (NY and non-NY jurisdictions), as well as PD_CD, LAW_CODE, arrest_year, and KY_CD.

In Dataset 2, the best metrics are achieved with a depth of 5, and the overfitting analysis shows no signs of overfitting. The model demonstrates high accuracy but relatively low precision in the gradient boosting analysis when applied to the test set. The top 5 most important variables for this dataset are X55 and X60.

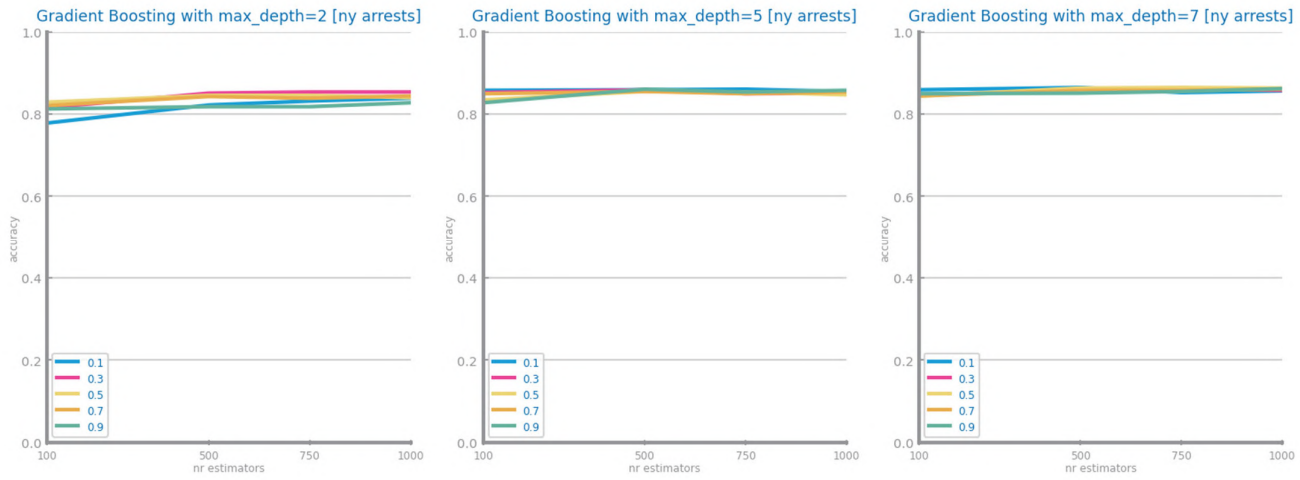


Figure 52 Gradient boosting different parameterizations comparison for dataset 1

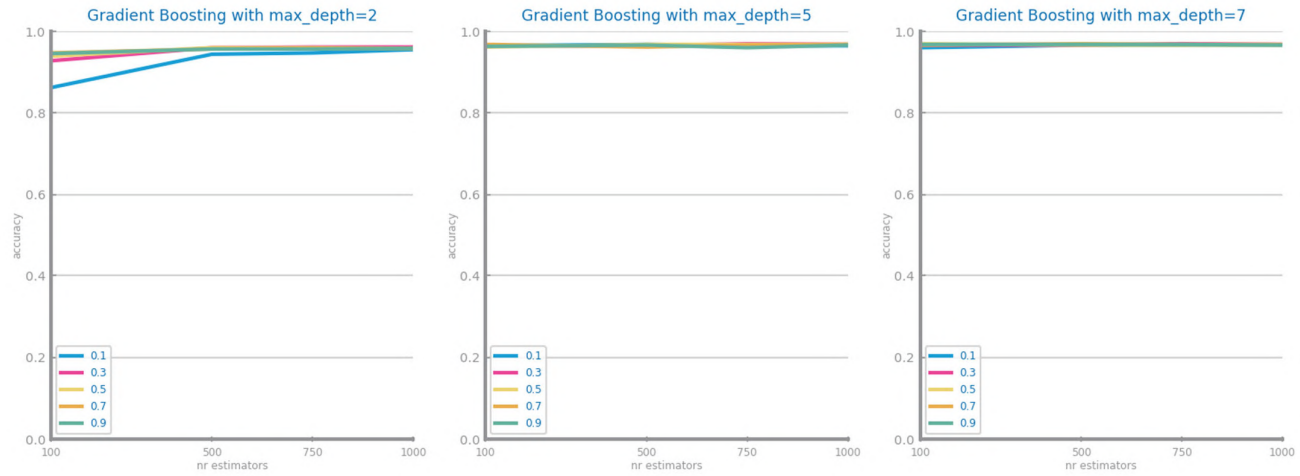


Figure 53 Gradient boosting different parameterizations comparison for dataset 2

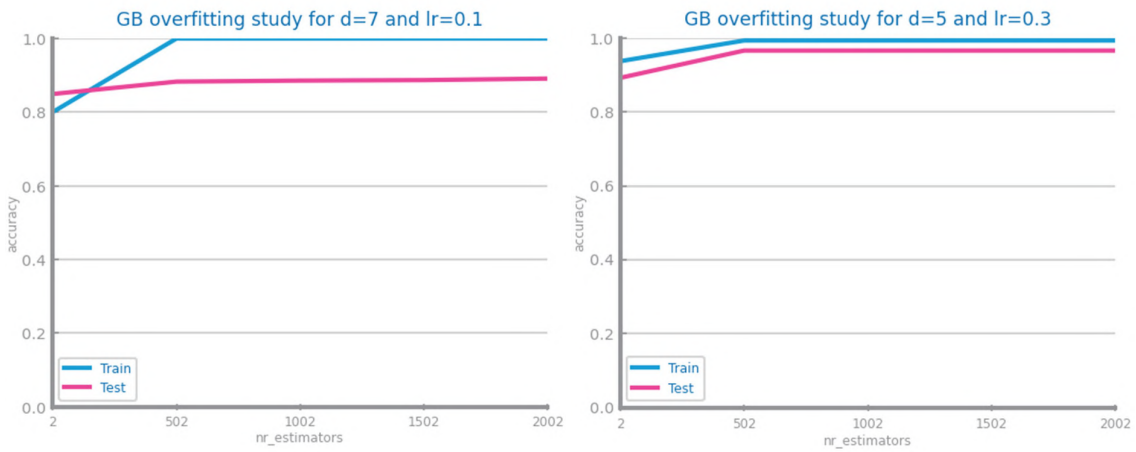


Figure 54 Gradient boosting overfitting analysis for dataset 1 (left) and dataset 2 (right)

Figure 55 Gradient boosting best model results for dataset 1 (above) and dataset 2 (below)

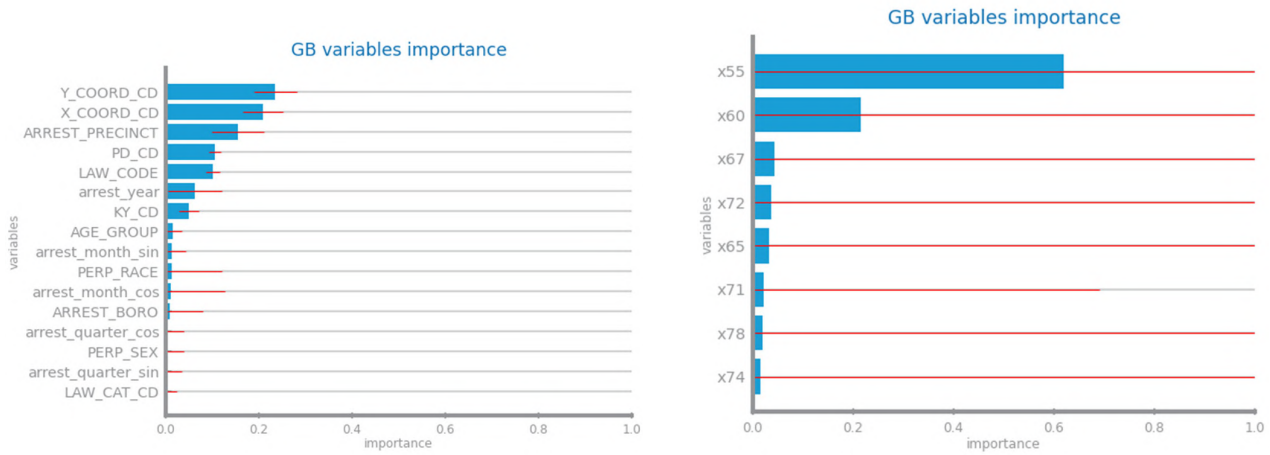


Figure 56 Gradient boosting variables importance for dataset 1 (left) and dataset 2 (right)

Multi-Layer Perceptrons

In Dataset 1, the best metrics are obtained with a constant learning rate of 0.5 and 3,000 iterations. In two intervals of nr_iterations, there is some risk of overfitting. The error curve decreases regularly, which means that the model converges well. The metrics, in general, are average and high, except for the precision on the test set, which may be caused by class imbalance in the test set or overfitting, which is very common in neural networks.

In Dataset 2, the best metrics are achieved with a constant learning rate of 0.05 and 3,500 iterations. While the algorithm shows good accuracy, its low recall and precision indicate that it is not well-suited for this dataset. Overfitting becomes apparent after 3,000 iterations, as the model appears overly specialized to the training data, with significant fluctuations and divergence from the validation performance. The loss curve further emphasizes this issue, exhibiting considerable instability, which means that the model struggles to converge.

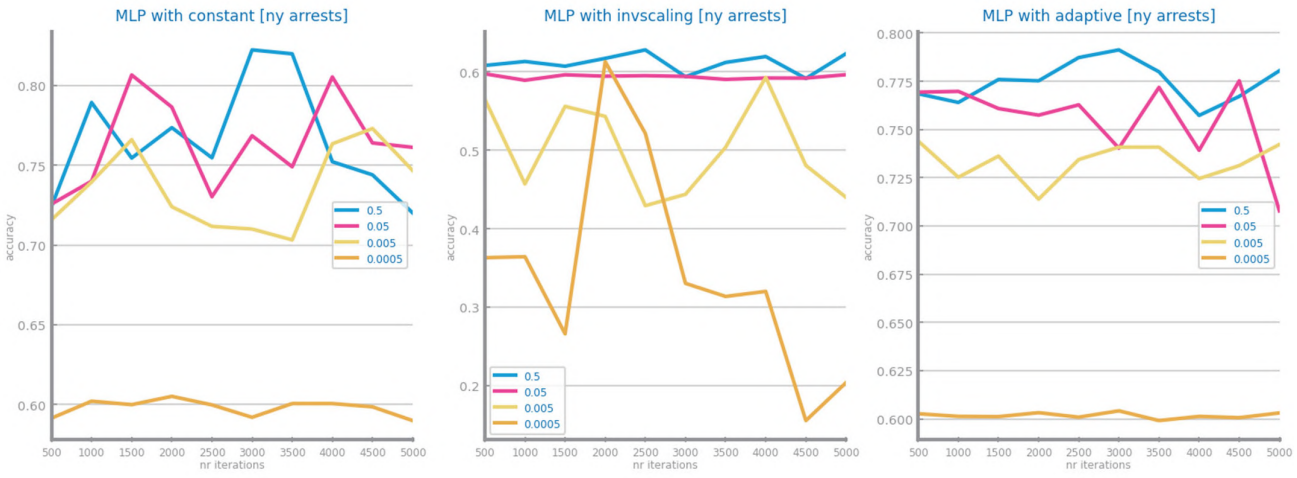


Figure 57 MLP different parameterizations comparison for dataset 1

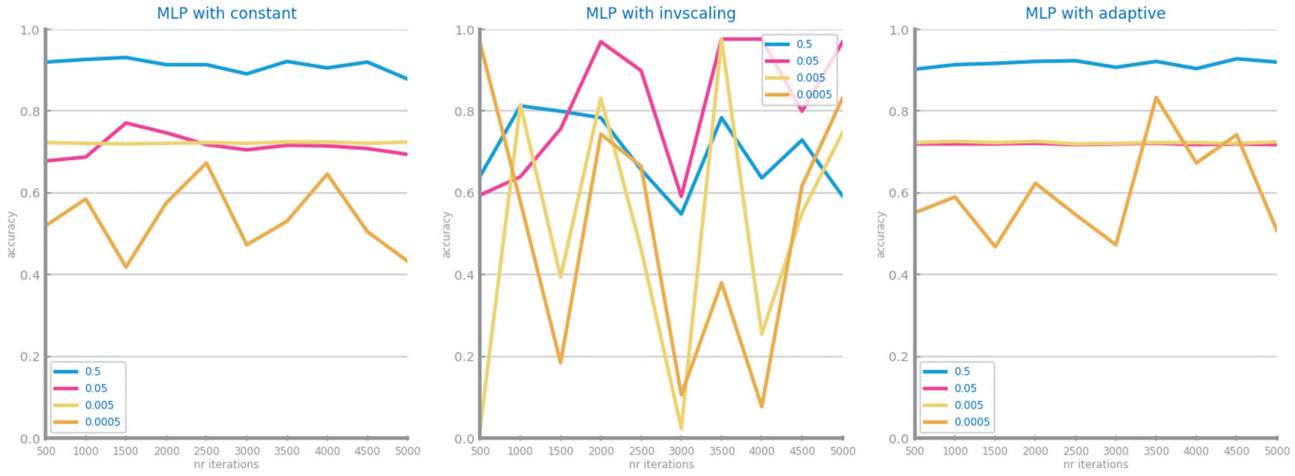


Figure 58 MLP different parameterizations comparison for dataset 2

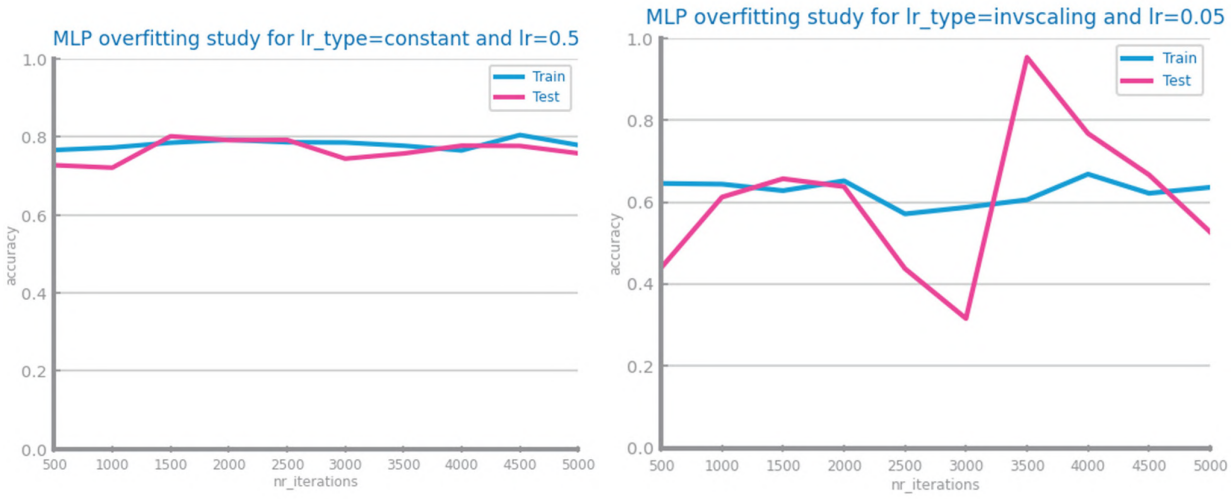


Figure 59 MLP overfitting analysis for dataset 1 (left) and dataset 2 (right)

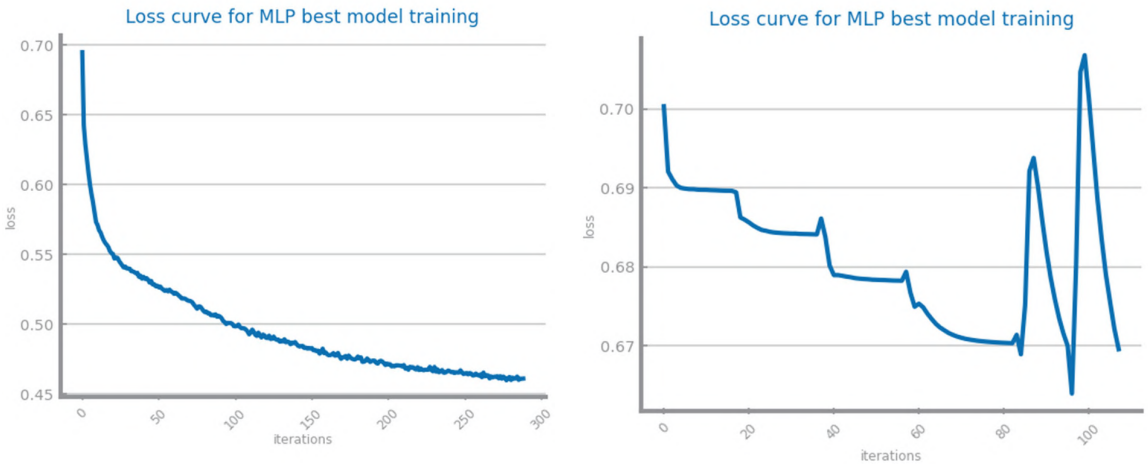


Figure 60 Loss curve analysis for dataset 1 (left) and dataset 2 (right)

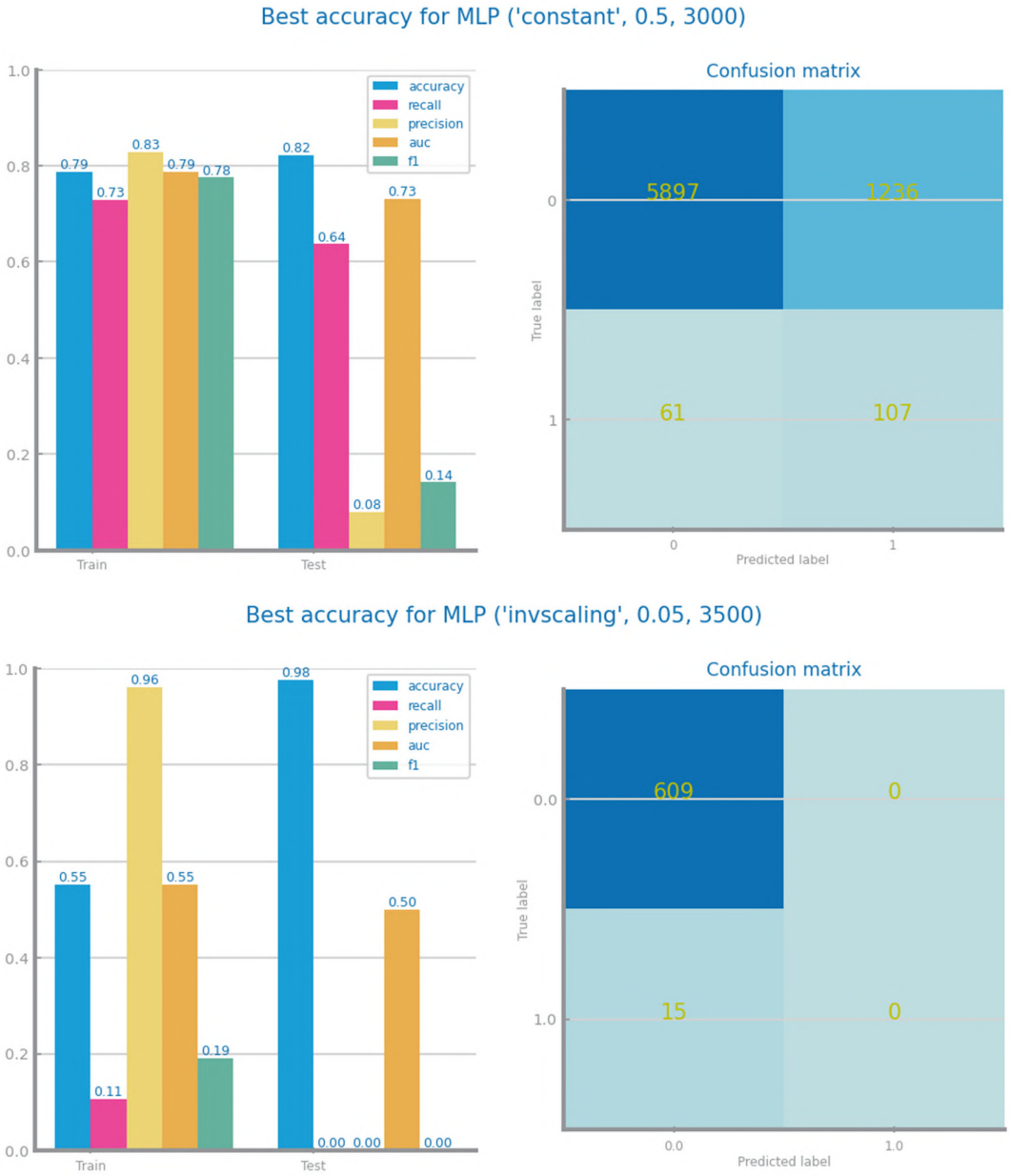


Figure 61 MLP best model results for dataset 1 (left) and dataset 2 (right)

4 CRITICAL ANALYSIS

Summary and Critical Assessment of Modeling Results

The **data preparation process** was pivotal in shaping model performance. Key tasks included encoding symbolic variables, scaling, handling missing data, and balancing class distributions. Dataset 1 benefited from Z-score scaling and oversampling to address imbalances, while Dataset 2 required MinMax scaling, outlier removal, and median imputation for better stability. Feature selection, particularly in Dataset 2, played a crucial role in reducing noise and computational complexity. These steps ensured that the datasets were well-suited for the modeling techniques applied.

Naïve Bayes achieved moderate accuracy and recall but consistently low precision in both datasets, indicating a tendency for false positives. **KNN**, with k=9 (Dataset 1) and k=1 (Dataset 2), showed good accuracy and precision; however, the lower k in Dataset 2 increased overfitting risk. **Decision Trees** demonstrated reliable generalization with depths of 17 (Dataset 1) and 6 (Dataset 2), avoiding overfitting while maintaining solid performance. **Random Forests** and **Gradient Boosting** excelled in accuracy and highlighted the importance of geographical and jurisdictional features in Dataset 1 (Y_COORD_CD, X_COORD_CD, ARREST_PRECINCT) and key variables in Dataset 2 (x55, x60). Despite their strengths, these ensemble methods exhibited relatively low precision, suggesting challenges with imbalanced data. **MLPs** achieved reasonable accuracy but were prone to overfitting, particularly in Dataset 2, as seen in unstable loss curves beyond 3,000 iterations.

Critical Assessment

Random Forests and **Gradient Boosting** were the best-performing models, consistently identifying influential variables and achieving high accuracy. However, their low precision, coupled with class imbalance and noise, limits their reliability for applications requiring precise positive predictions. Improvements in data balancing, feature engineering, and hyperparameter optimization are recommended to address these issues. While the models are promising, they require refinement to meet real-world standards. Visualizations could further enhance the understanding of performance patterns and feature importance.

5 DATA PROFILING

This study explores the impact of data granularity on time series analysis. We will examine how different time granularities affect pattern identification, trends, and correlations, along with assessing stationarity using statistical tests. The findings will help determine the most suitable methods for modeling and forecasting.

Data Dimensionality and Granularity

In Dataset 1, at the monthly granularity level, a cyclic pattern appears to occur, repeating annually, specifically with slight dips at the end of the year. As for the annual granularity, a peak is observed in the first half of the 2010s, followed by a gradual decline, reaching a minimum value around 2020. In Dataset 2, which is provided with yearly granularity, a steady increase in GDP is observed until 2010. Beyond 2010, the data exhibits greater volatility, with significant fluctuations. When aggregated to 5- and 10-year granularities, these details are lost, revealing only the overall upward trend. Therefore, the 5- and 10-year granularities are unsuitable for training.

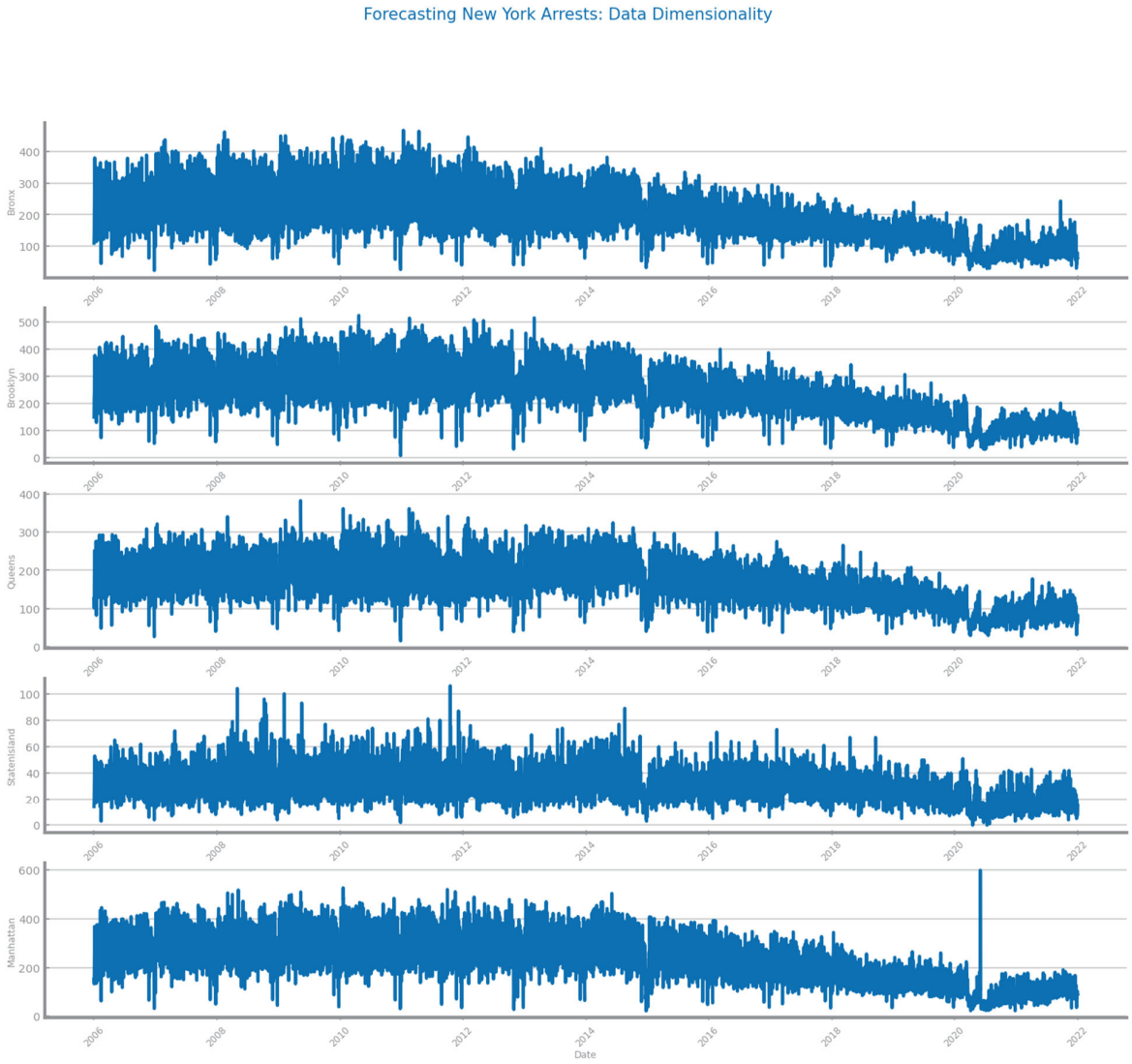


Figure 62 Original time series 1 (the most atomic detail)

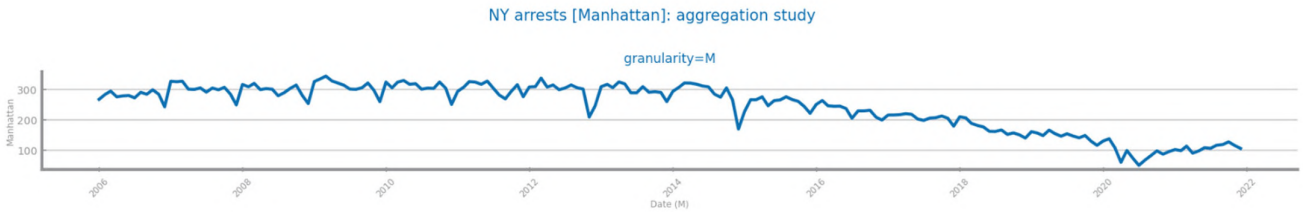


Figure 63 Time series 1 at the second chosen granularity

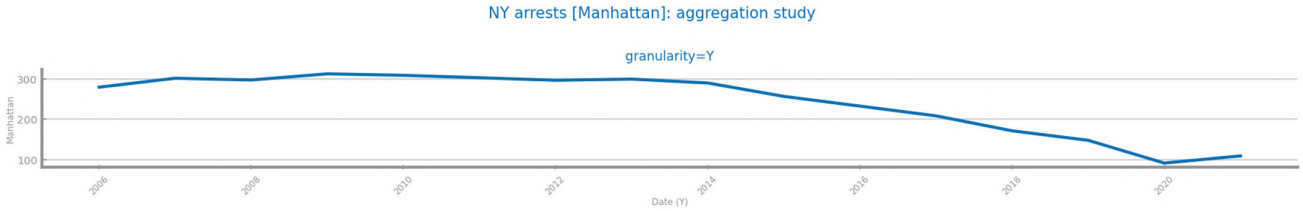


Figure 64 Time series 1 at the third chosen granularity

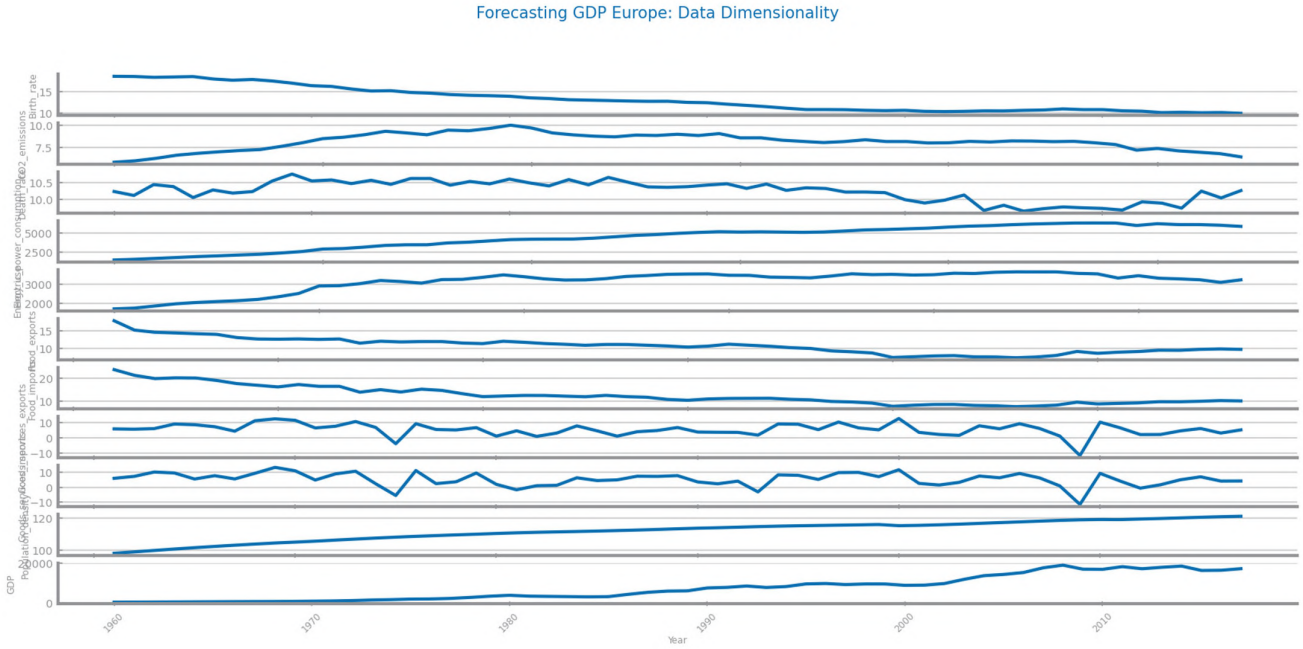


Figure 65 Original time series 2 (the most atomic detail)

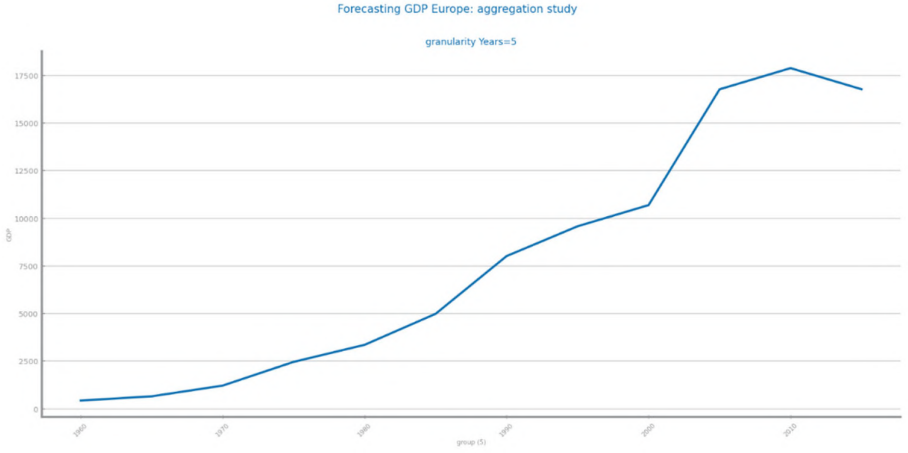


Figure 66 Time series 2 at the second chosen granularity

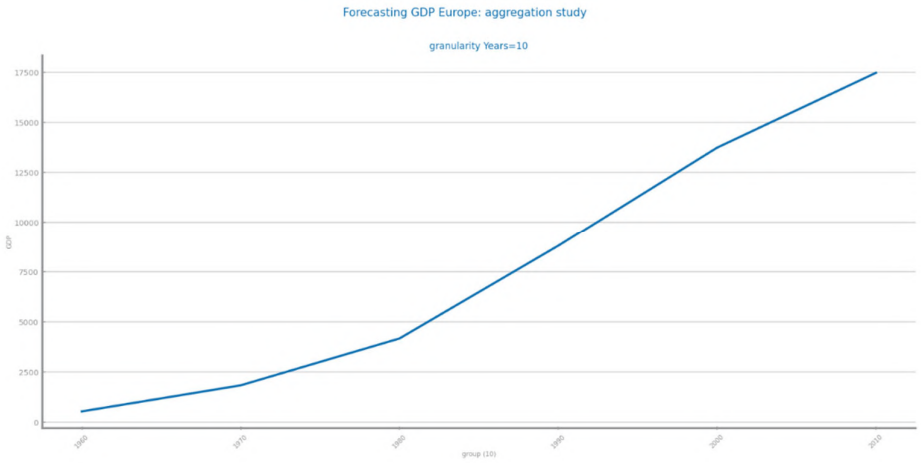


Figure 67 Time series 2 at the third chosen granularity

Data Distribution

In Dataset 1, boxplots were created for weekly, monthly, quarterly, and annual granularities, along with corresponding histograms. Additionally, graphs were generated for the monthly granularity with lags of 1, 10, and 20 units, as well as autocorrelation scatter plots using monthly data. Among these, lags 1 and 7 exhibited the strongest correlations.

For Dataset 2, boxplots and histograms were created for yearly, 5-year, and 10-year granularities. Graphs were also generated for the yearly granularity with lags of 1, 2, and 20 units, accompanied by autocorrelation scatter plots. In this case, the lag of 1 showed the strongest correlation.

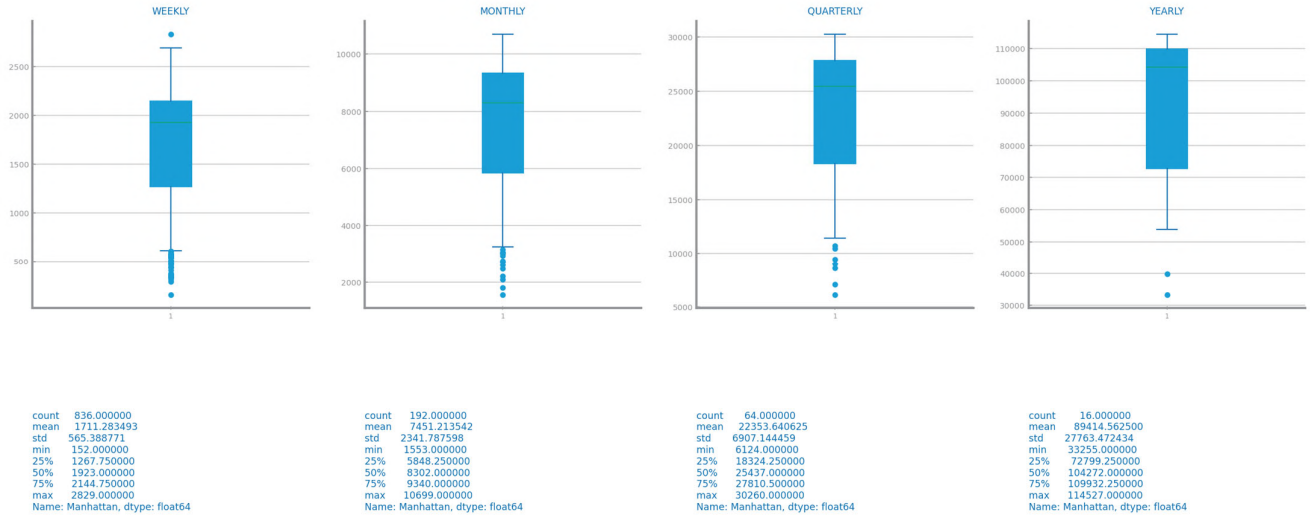


Figure 68 Boxplots for time series 1 at different granularities

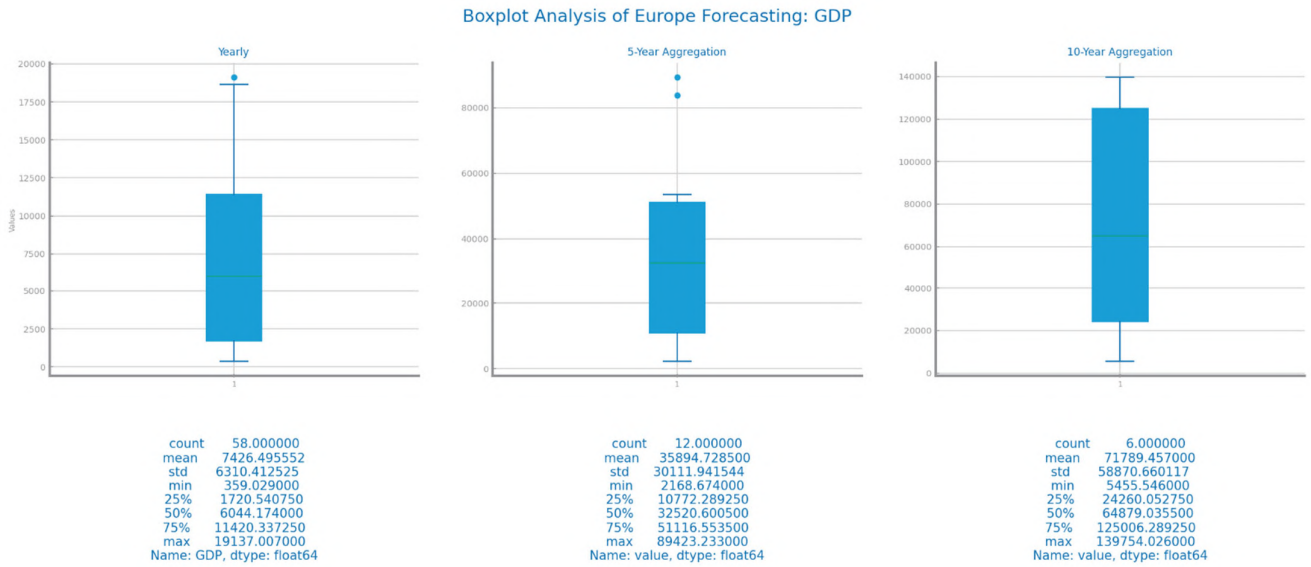


Figure 69 Boxplots for time series 2 at different granularities

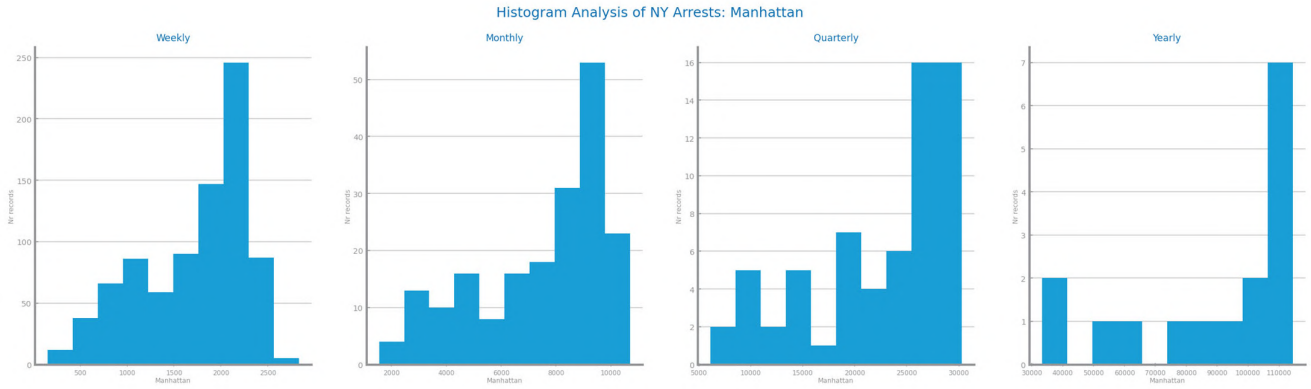


Figure 70 Histograms for time series 1 at different granularities

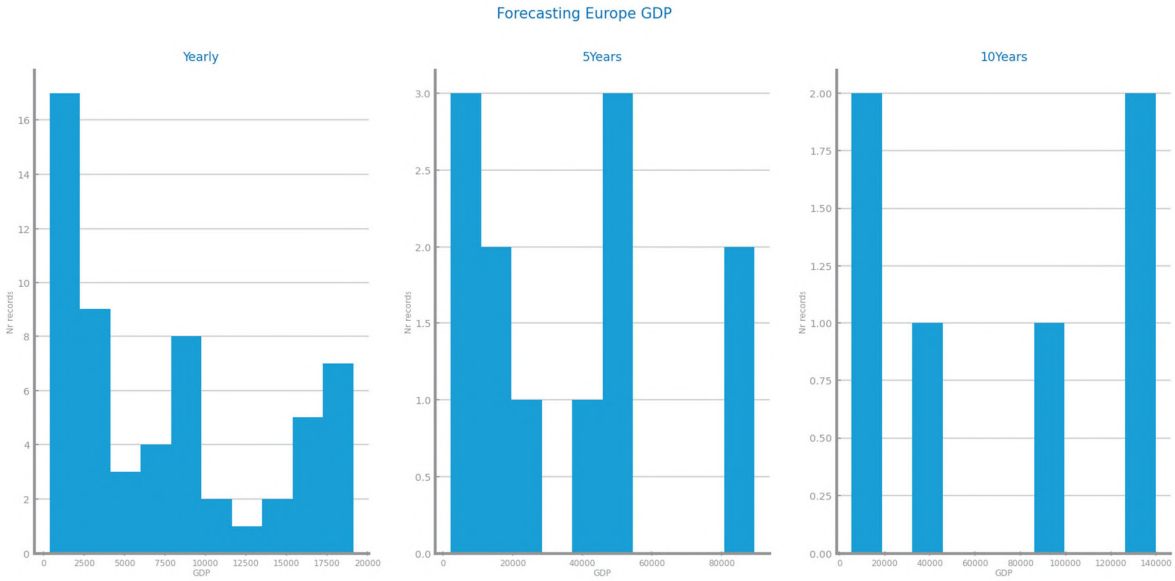


Figure 71 Histograms for time series 2 at different granularities

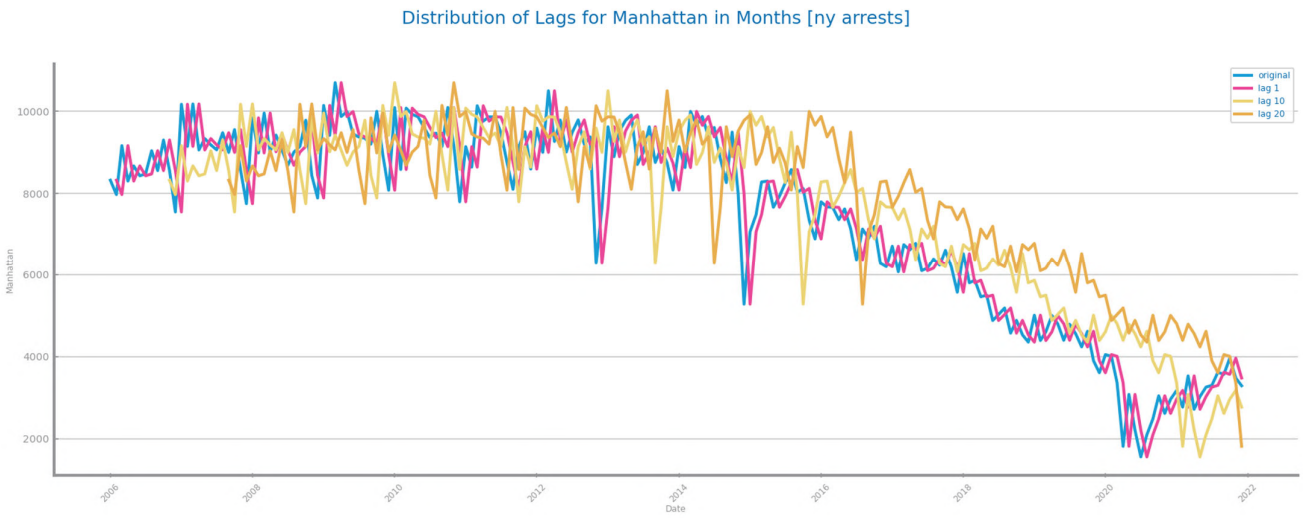


Figure 72 Autocorrelation lag-plots for original time series 1

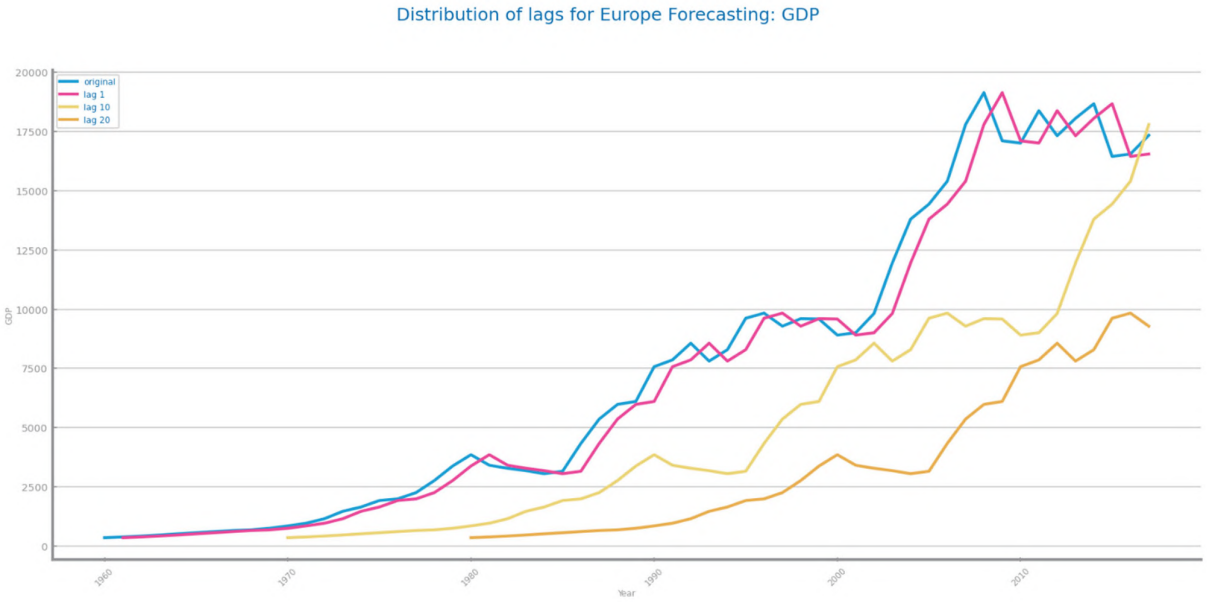


Figure 73 Autocorrelation lag-plots for original time series 2

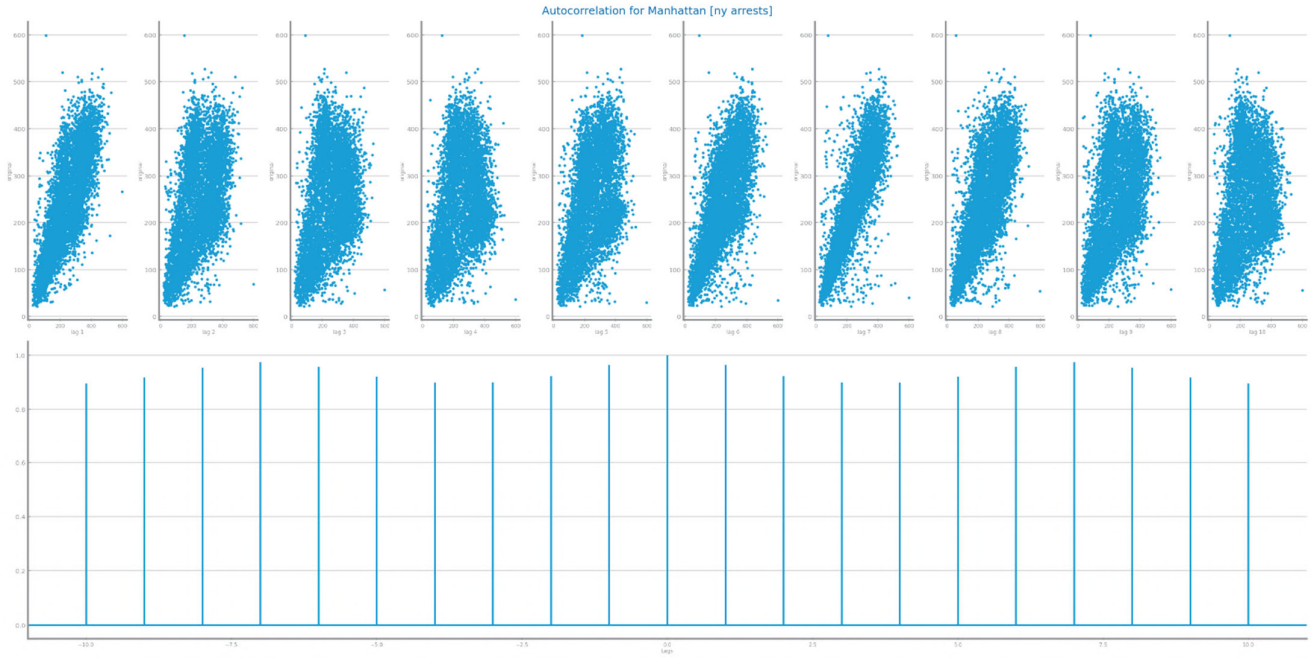


Figure 74 Autocorrelation correlogram for original time series 1

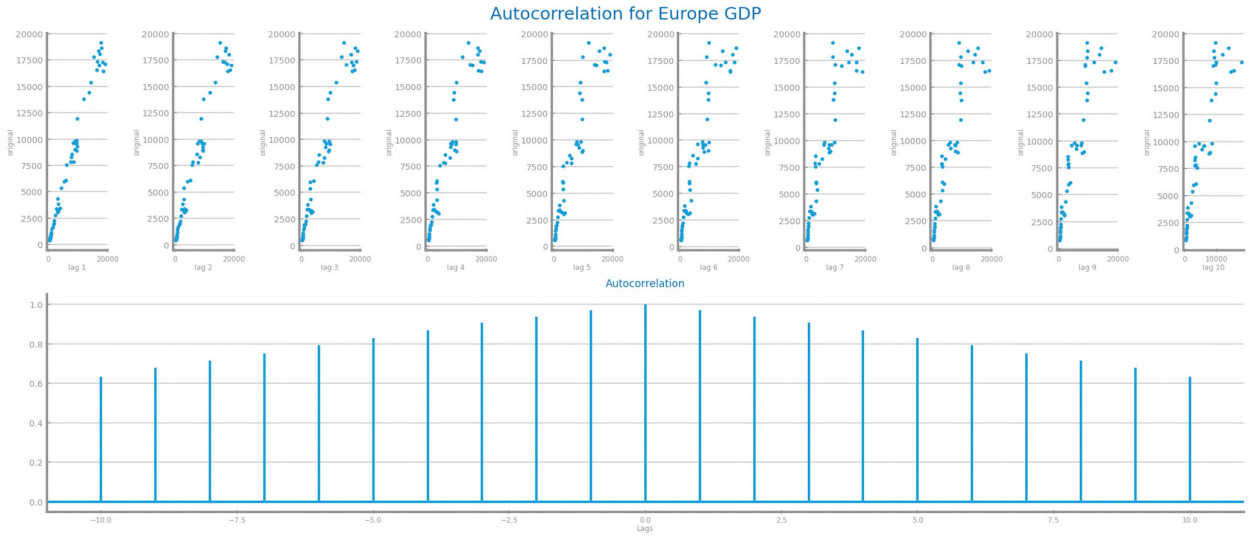


Figure 75 Autocorrelation correlogram for original time series 2

Data Stationarity

In Dataset 1, the component analysis revealed a distinct and heterogeneous trend, along with noticeable residuals. The moving average highlights significant variations, consistent with the identified trend. Furthermore, the p-value (0.338) exceeds 0.05, indicating that the series is non-stationary. The Dickey-Fuller test statistic of -1.887 confirms the non-stationarity of the series, regardless of the significance level (1%, 5%, or 10%).

In Dataset 2, the component analysis demonstrated steady GDP growth up to 2010, followed by fluctuations and instability. The p-value for this dataset (0.869) also exceeds 0.05, suggesting that the series is non-stationary. Additionally, the ADF statistic of -0.608 indicates non-stationarity across all significance levels (1%, 5%, and 10%).

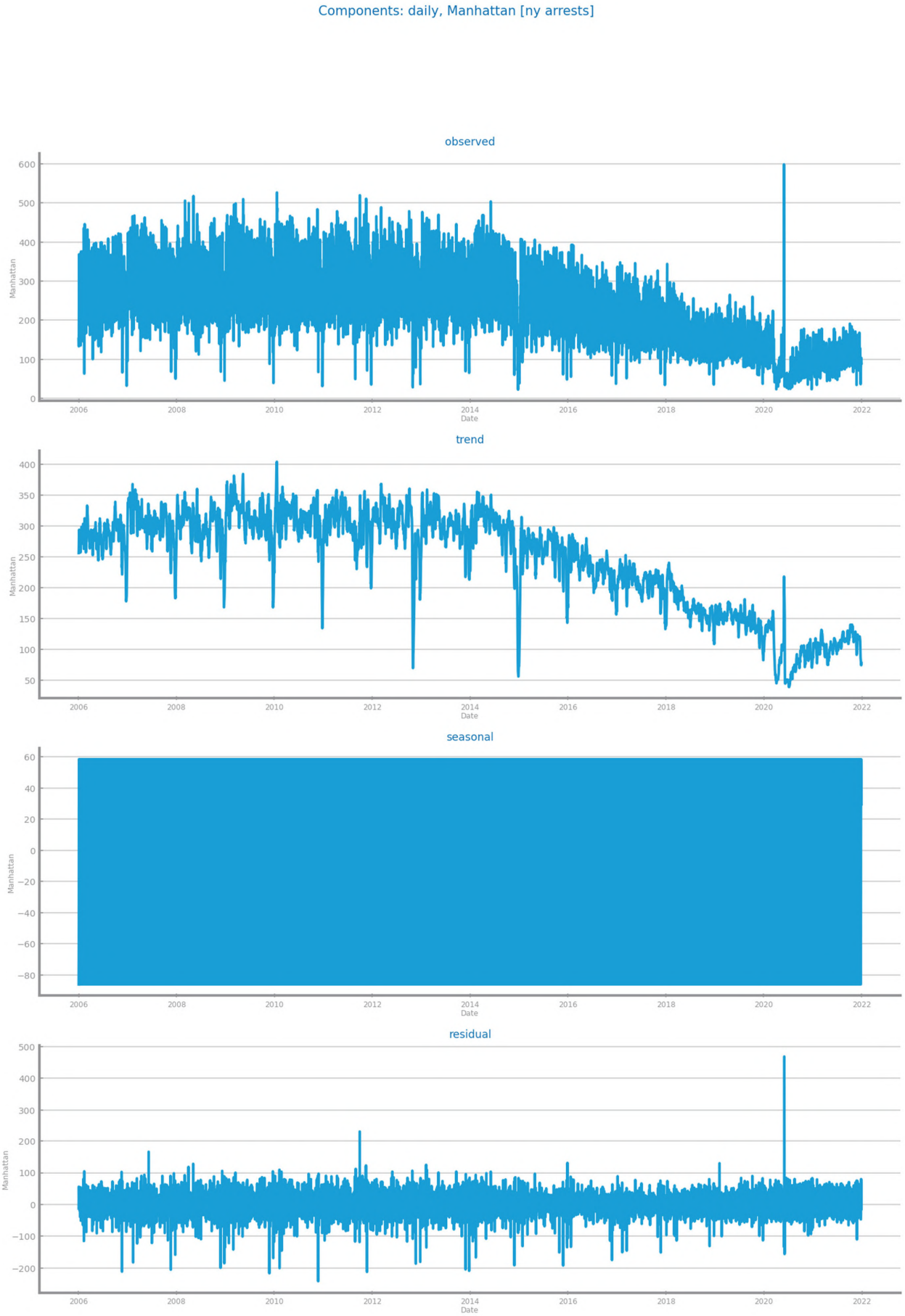


Figure 76 Components study for time series 1

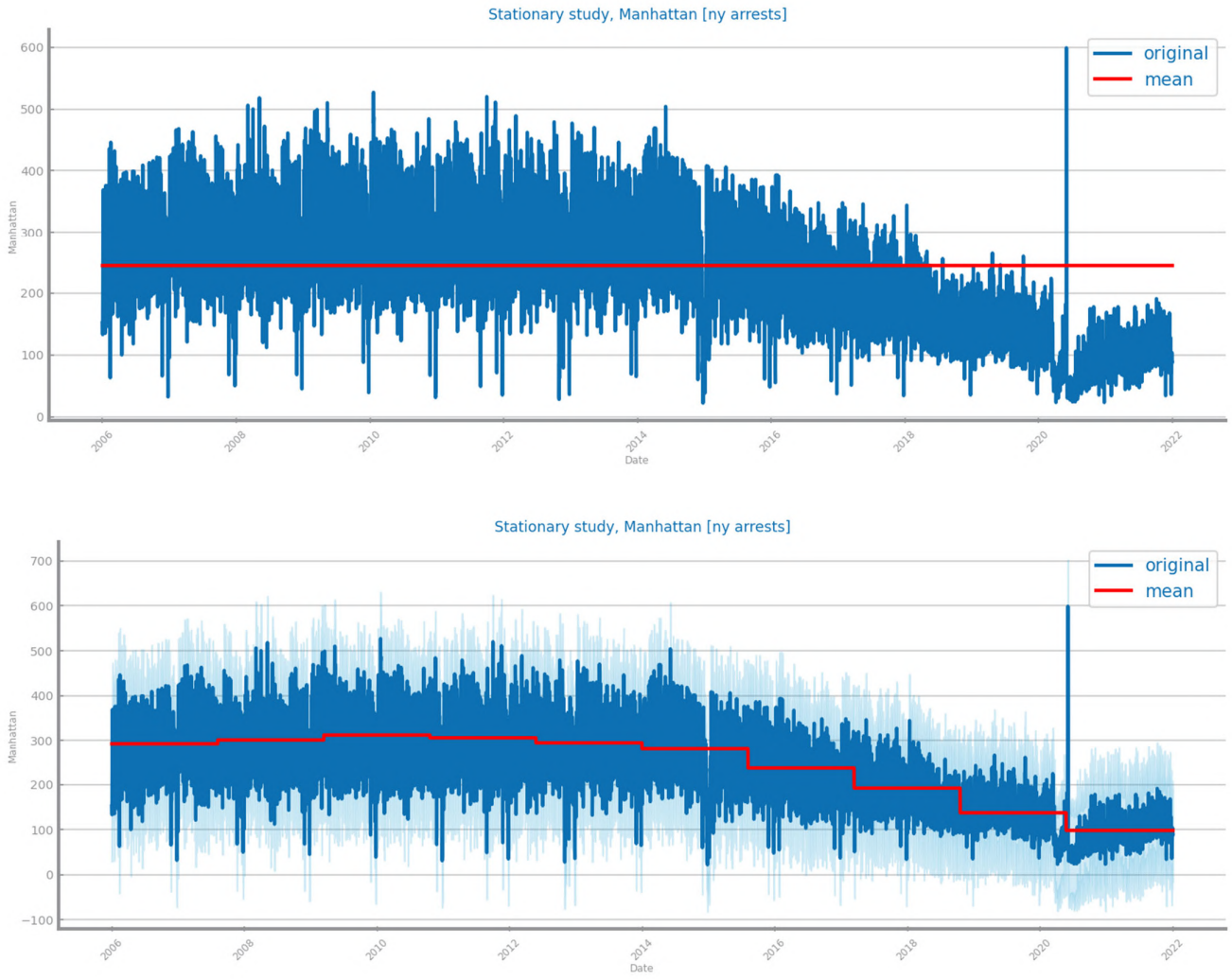


Figure 77 Stationarity study for time series 1

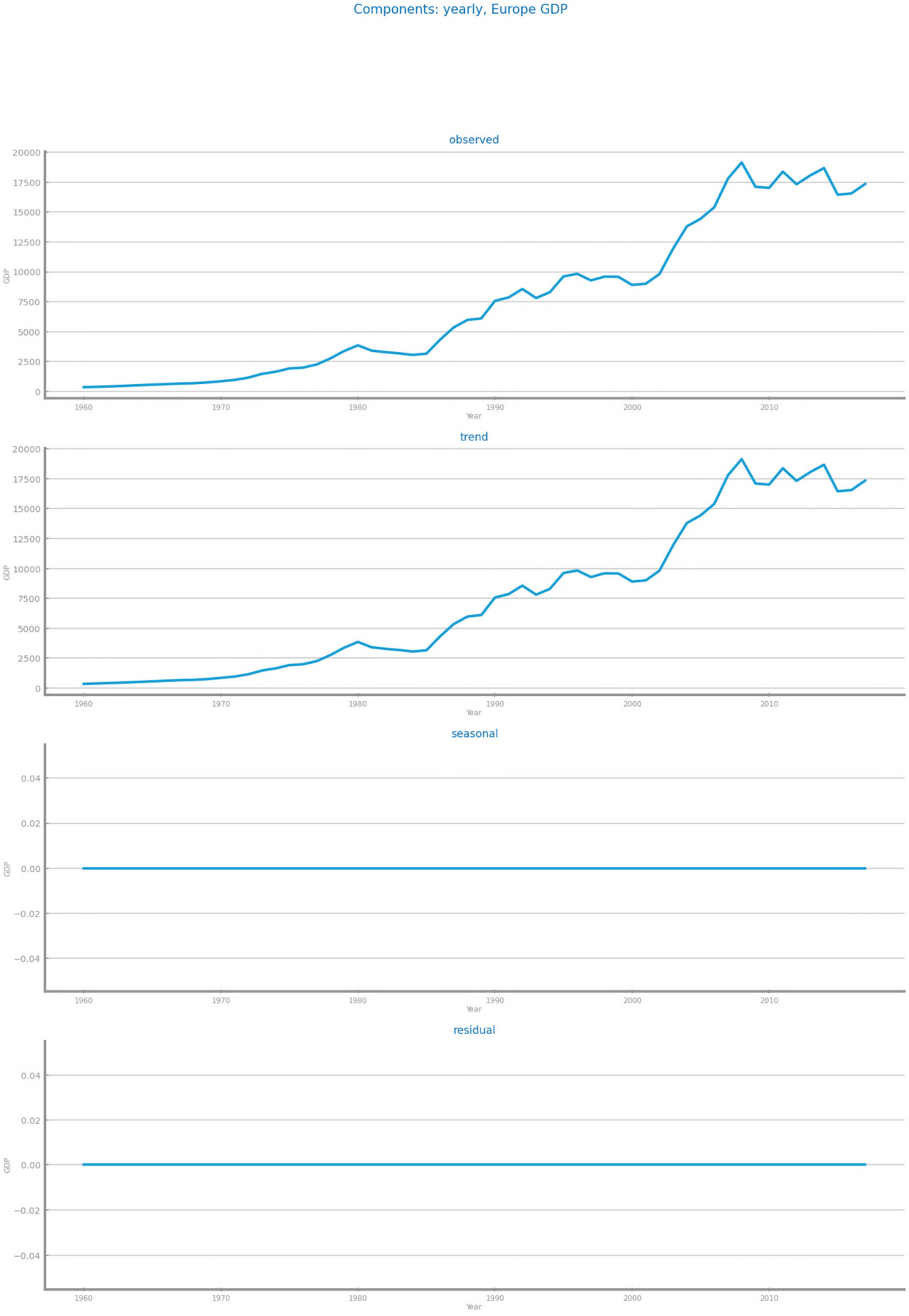


Figure 78 Components study for time series 2

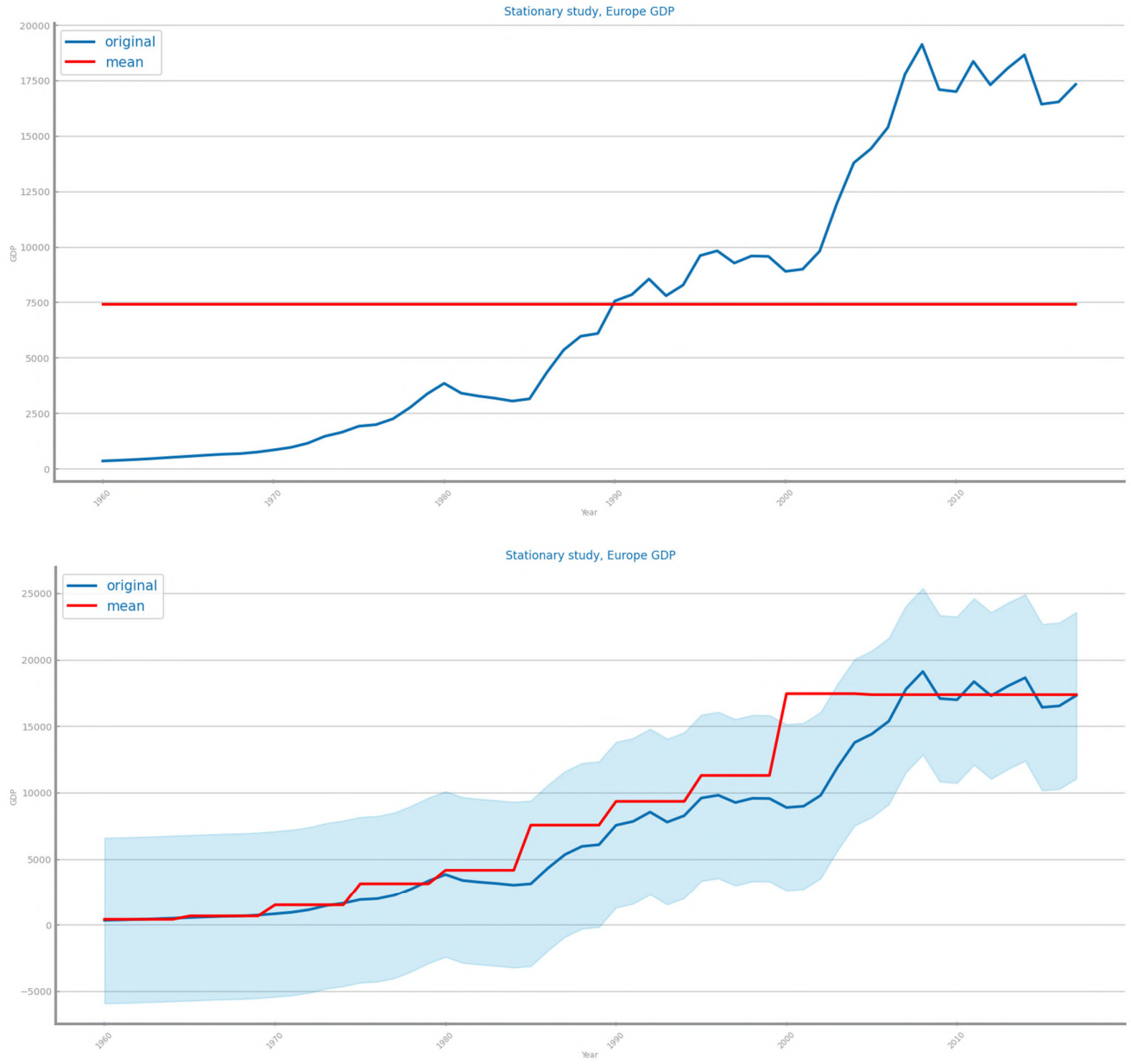


Figure 79 Stationarity study for time series 2

6 DATA TRANSFORMATION

This study addresses preprocessing operations applied to two datasets. Various aggregation methods will be considered, and smoothing will be applied to both datasets. Differentiation techniques will also be explored to handle trends and seasonality. Additionally, scaling methods will be applied to standardize the data, ensuring model stability and performance.

Aggregation

In Dataset 1, preference was given to the monthly granularity, as a quarterly aggregation would lose seasonal specificities within the year, and an annual aggregation over a relatively short 16-year period would eliminate valuable information. The aggregation was performed using the "sum" operation. In Dataset 2, preference was given to the 1-

year granularity (original), as a 5-year and a 10-year aggregation would lose important information. The aggregation was performed using the "mean" operation.

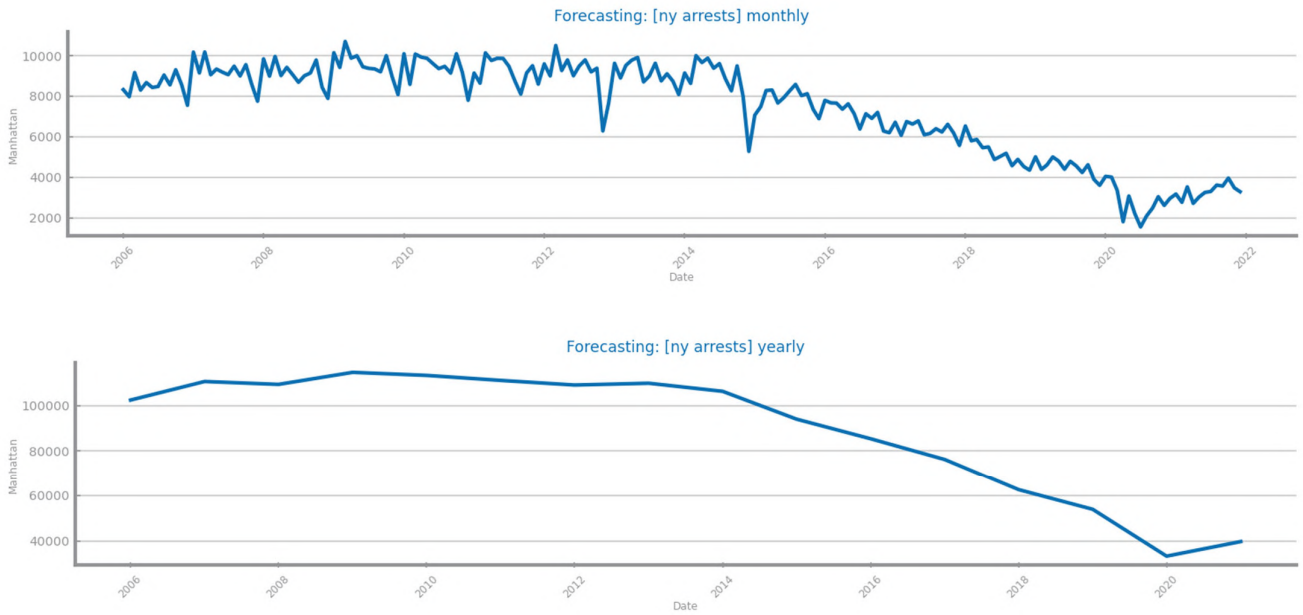


Figure 80 Forecasting plots after different aggregations on time series 1

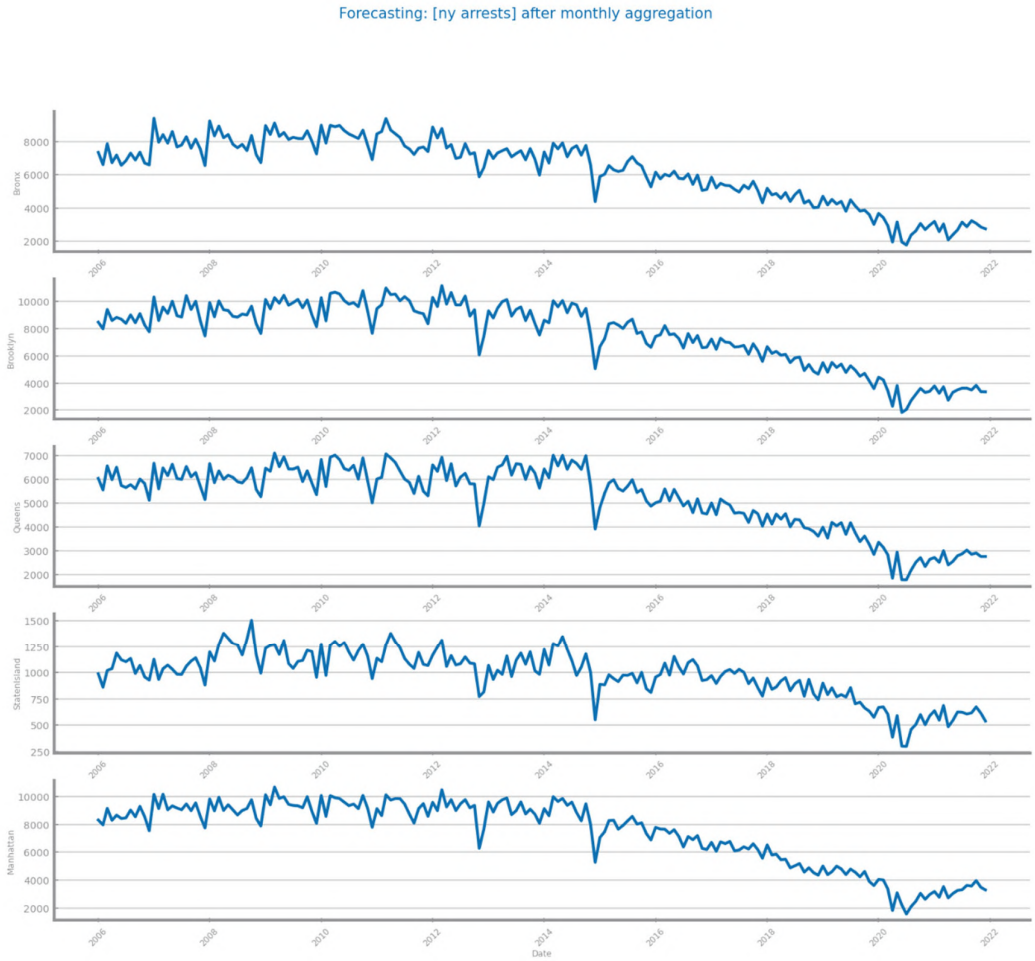


Figure 81 Forecasting results after different aggregations on time series 1

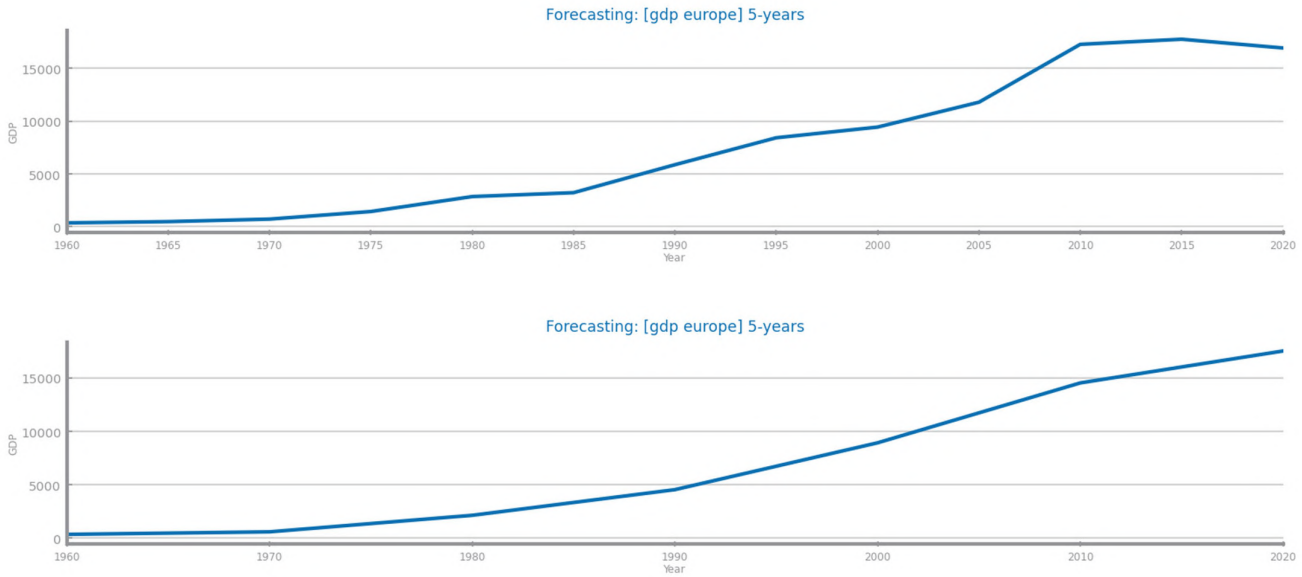


Figure 82 Forecasting plots after different aggregations on time series 2

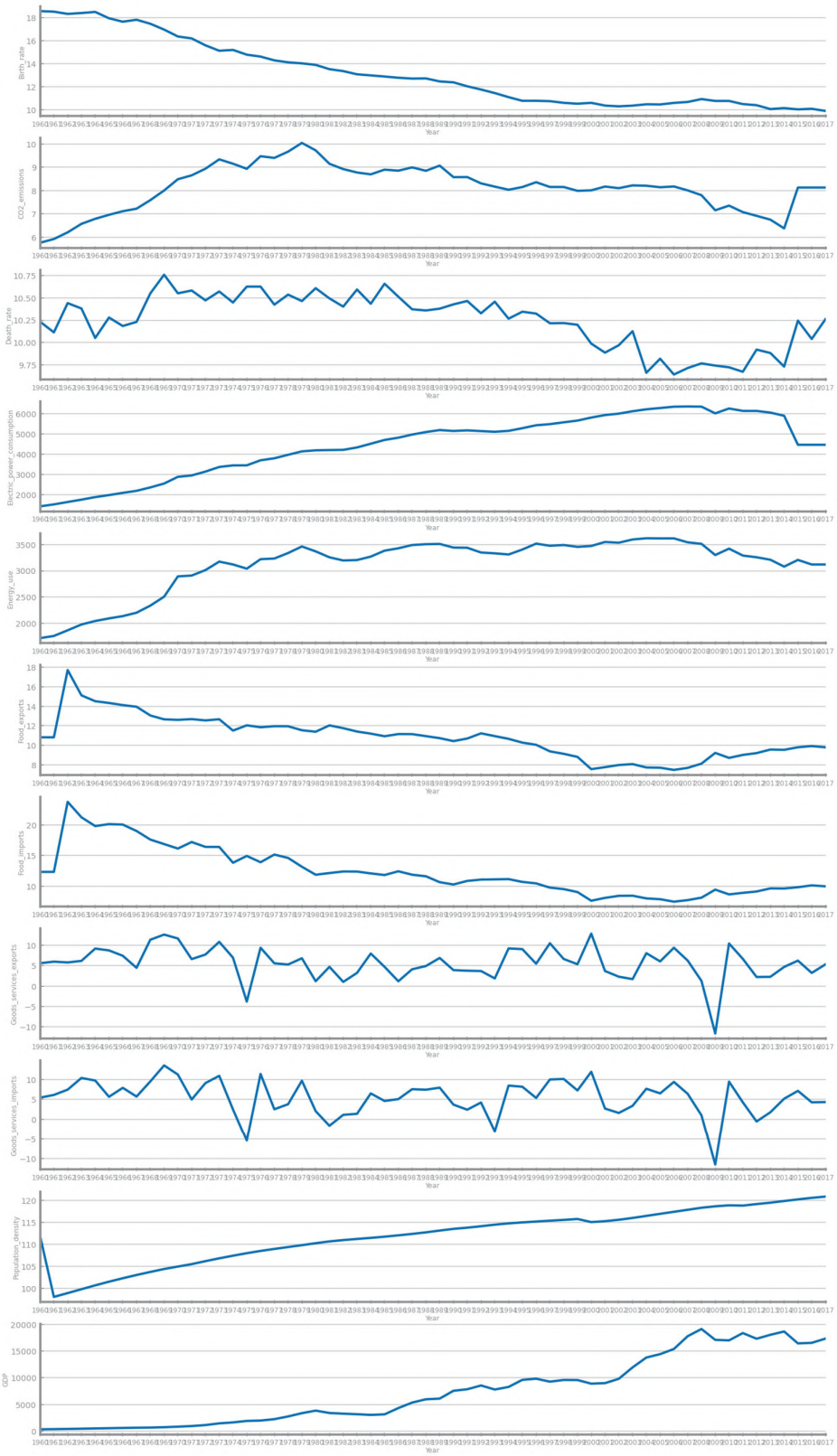


Figure 83 Forecasting results after different aggregations on time series 2

Smoothing

In both datasets, smoothing was applied using a window size of 2. This approach effectively smoothed the lines while preserving key patterns. Rows containing NaN values, generated during the smoothing process, were removed.

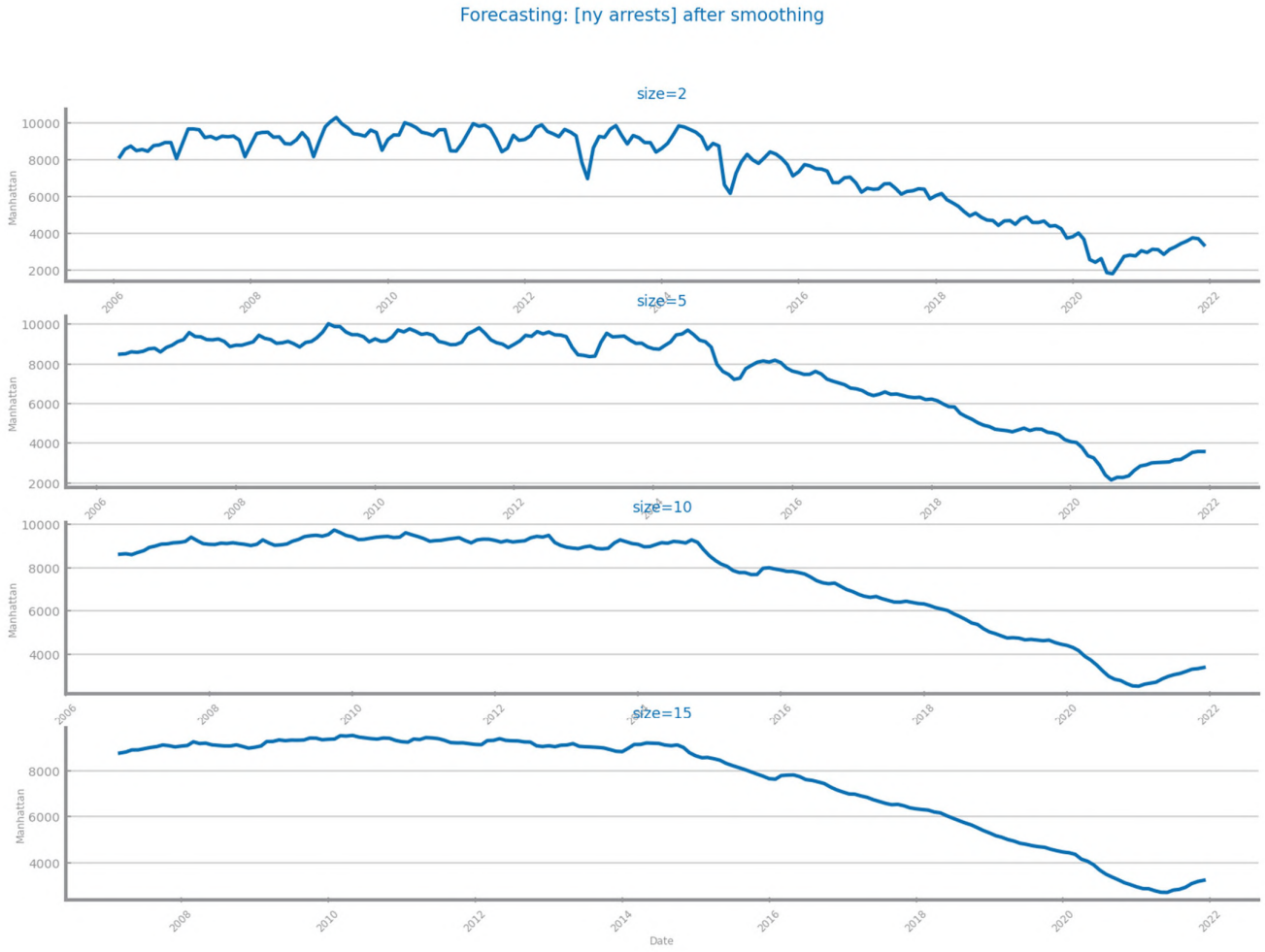


Figure 84 Forecasting plots after different smoothing parameterizations on time series 1

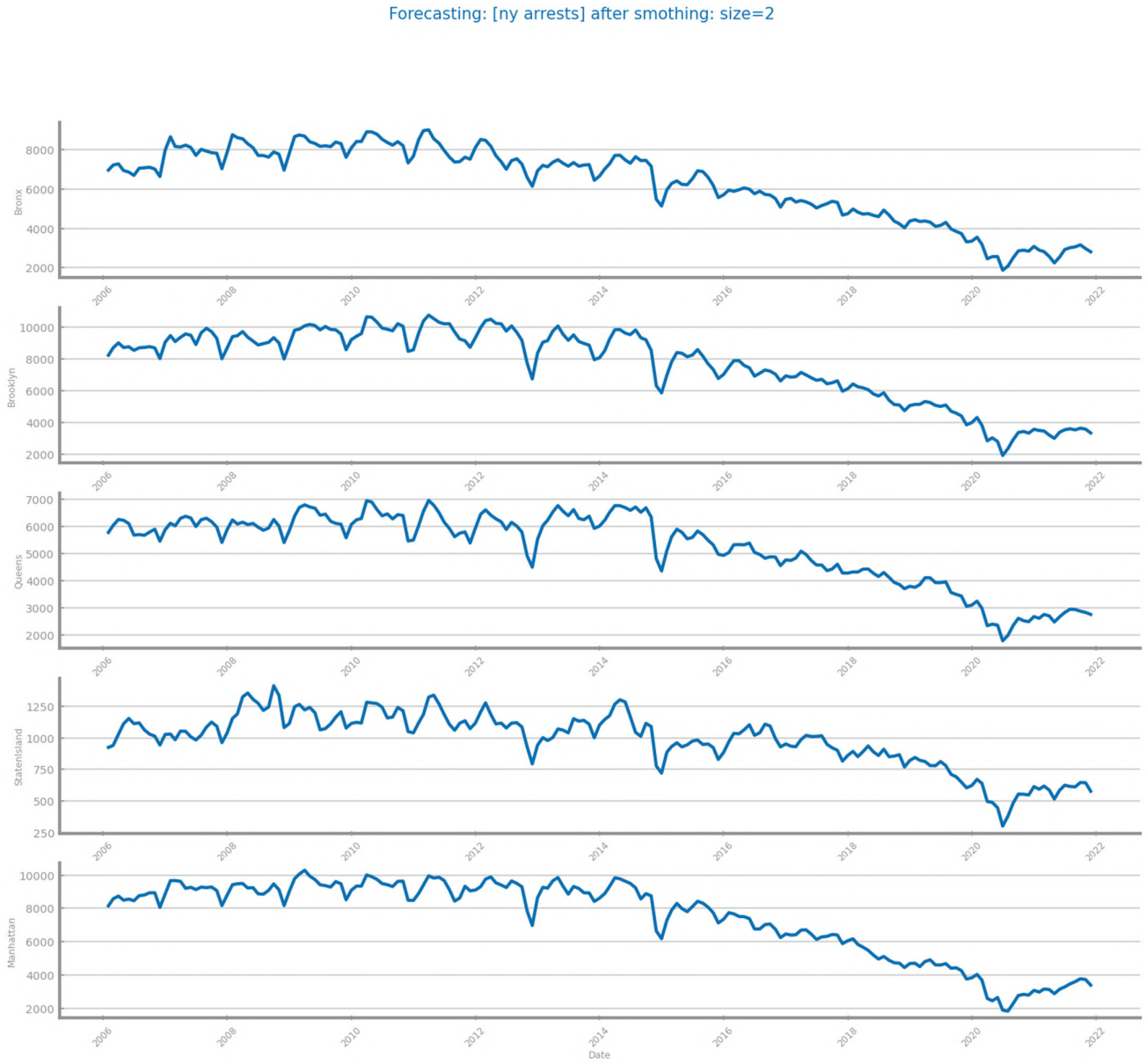


Figure 85 Forecasting results after different smoothing parameterizations on time series 1

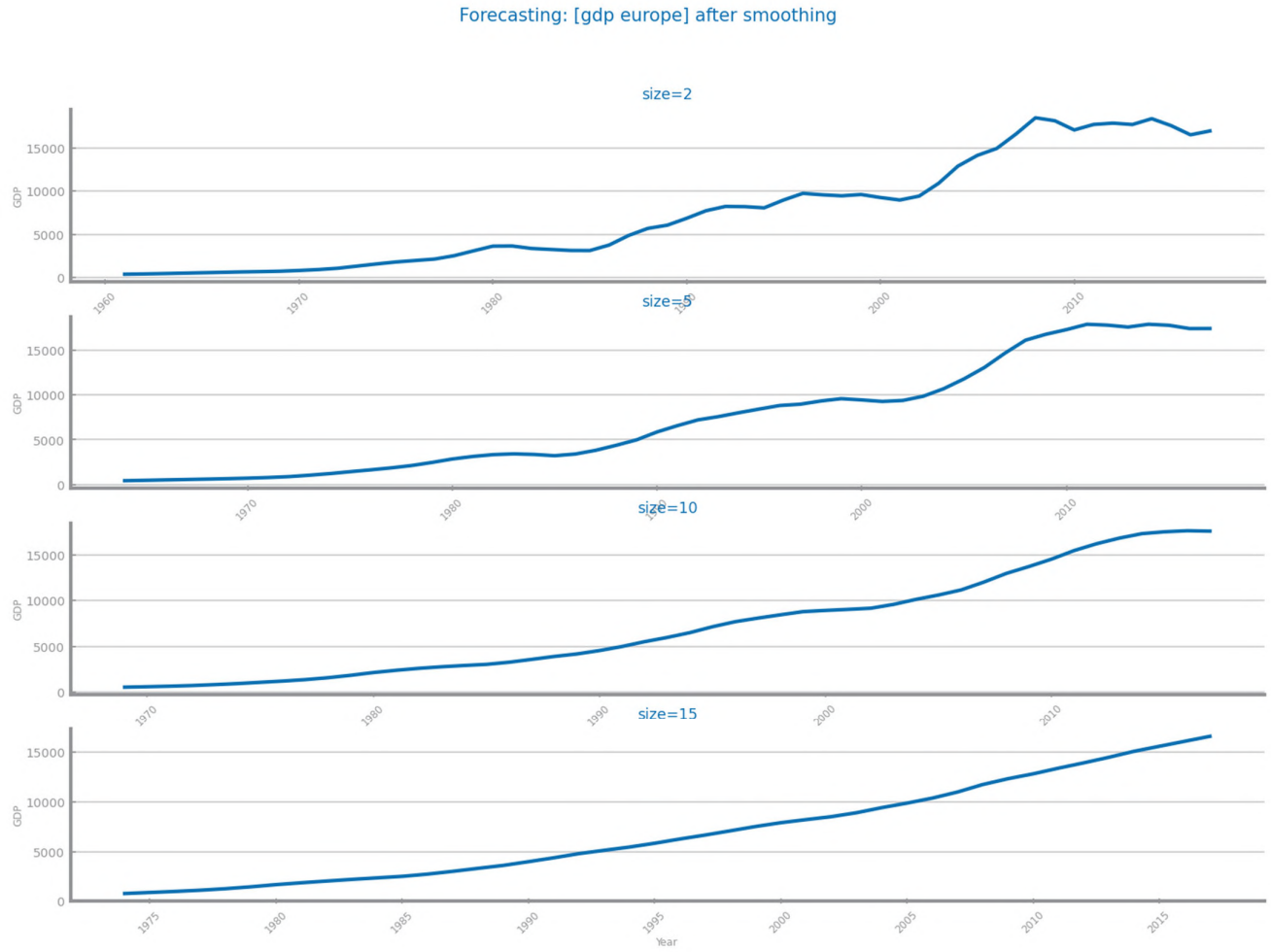


Figure 86 Forecasting plots after different smoothing parameterizations on time series 2

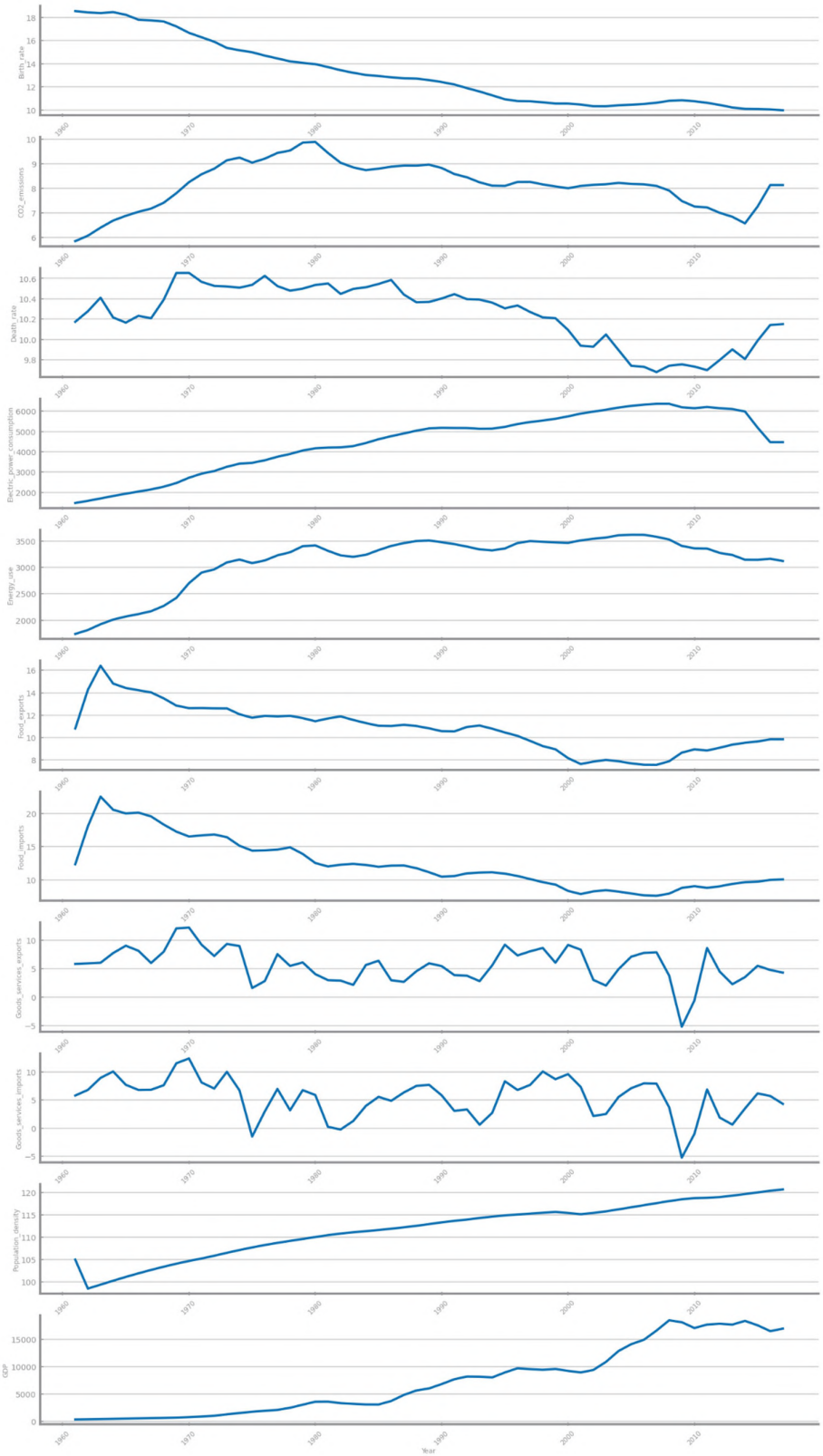


Figure 87 Forecasting results after different smoothing parameterizations on time series 2

Differentiation

In Dataset 1, one differentiation was applied to remove trends and seasonality, making the data stationary, which is crucial for effective modeling with algorithms like ARIMA or LSTMs. A second differentiation was tested but was discarded as it removed essential information needed for prediction, leading to higher forecasting errors.

In Dataset 2, two levels of differentiation were tested, but they did not provide any benefits. Instead, they introduced excessive noise, making the data less interpretable and unsuitable for modeling. As a result, it was decided not to apply any differentiation to Dataset 2.

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

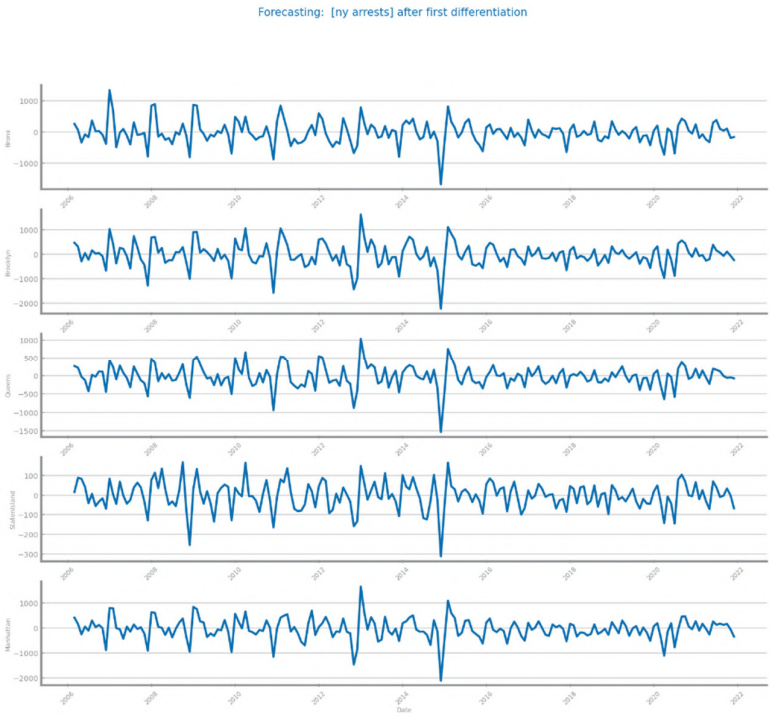


Figure 89 Forecasting results after first and second differentiation of time series 1

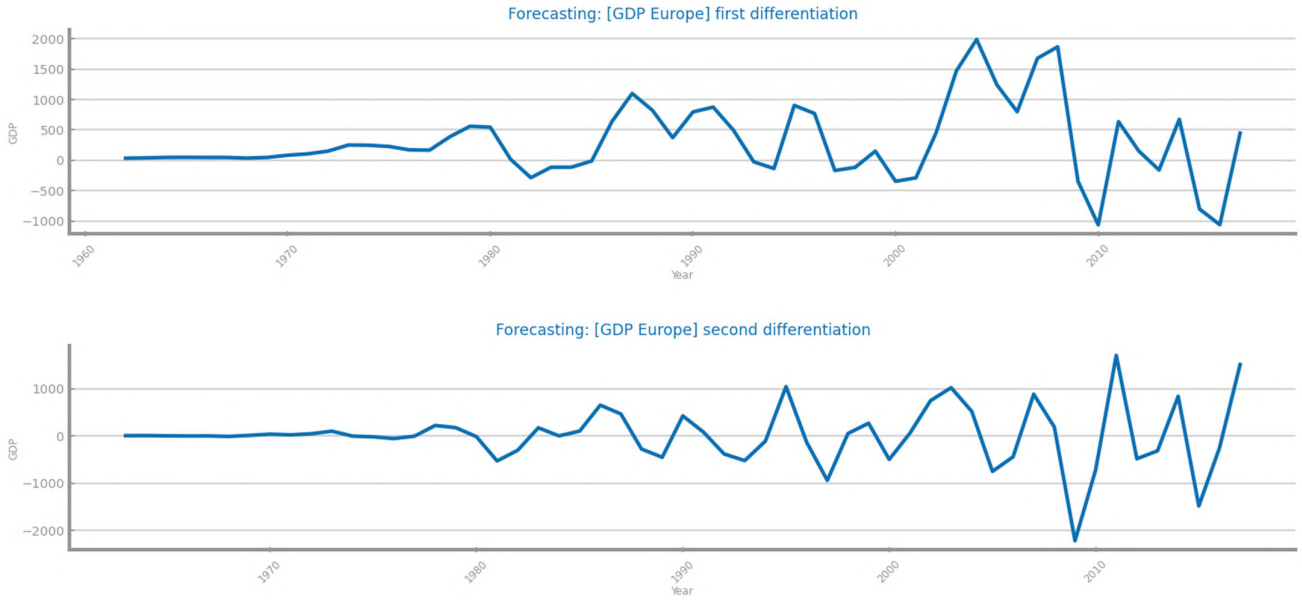


Figure 90 Forecasting plots after first and second differentiation of time series 2

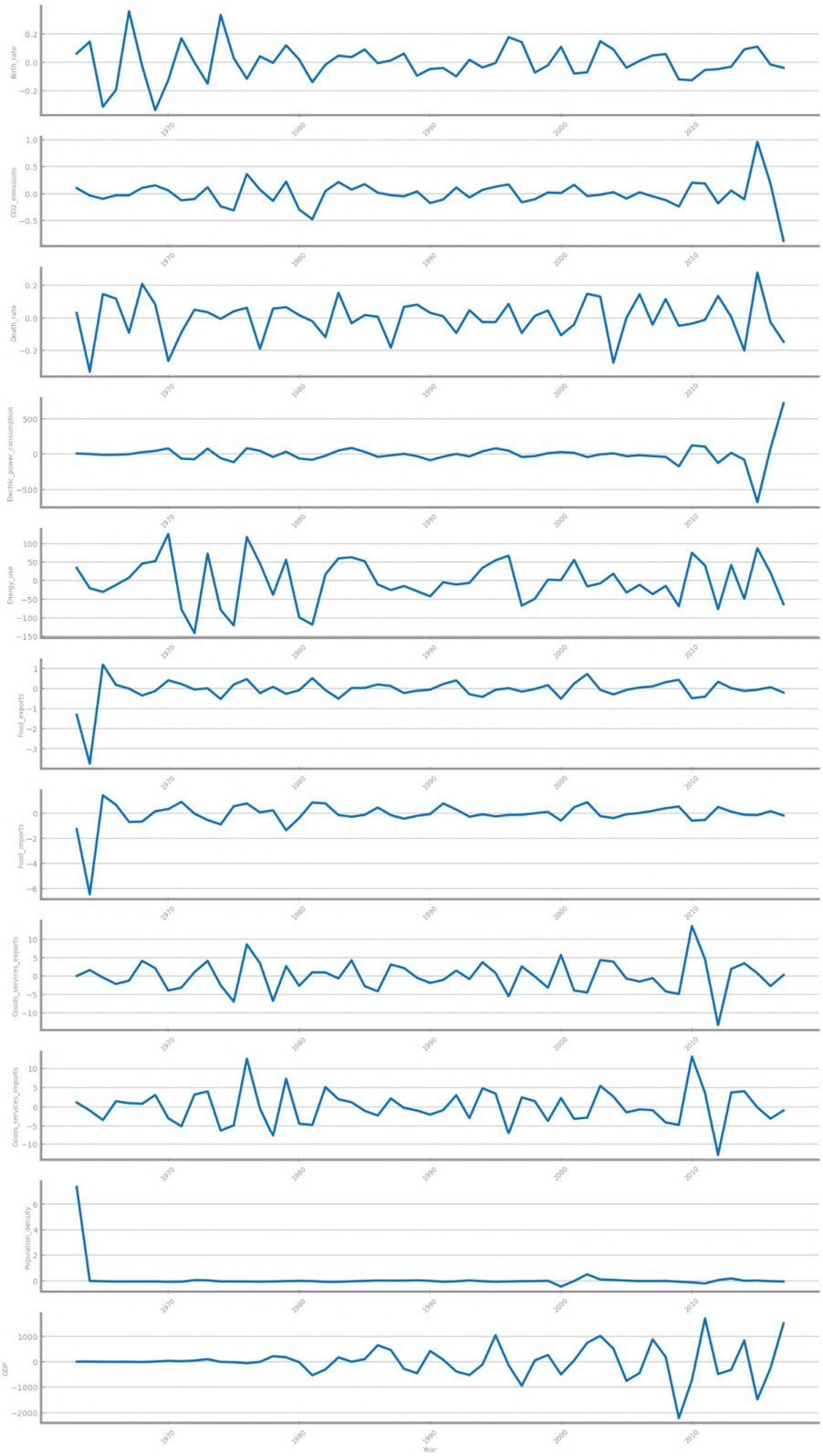


Figure 91 Forecasting results after first and second differentiation of time series 2

Other transformations (optional)

In Dataset 1 and 2, Min-Max Scaling was applied because LSTMs are sensitive to scale differences, as they use activation functions like sigmoid and tanh, which operate within limited ranges (in this case, [0,1]). Normalization improves training stability, preventing issues like exploding or vanishing gradients, and also accelerates model convergence.

Figure 92 Forecasting plots after applying other transformations over time series 1

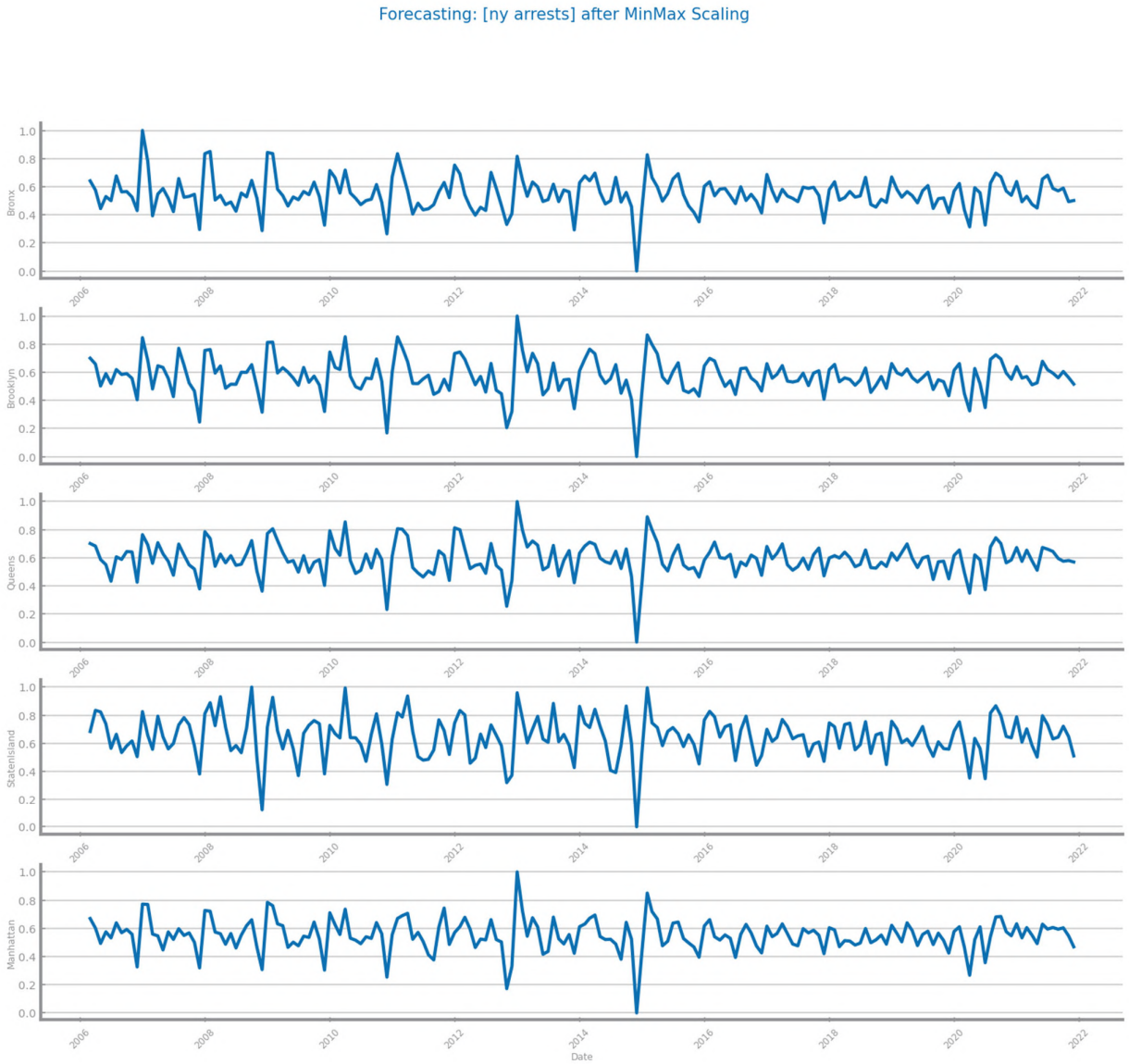


Figure 93 Forecasting results after applying other transformations over time series 1

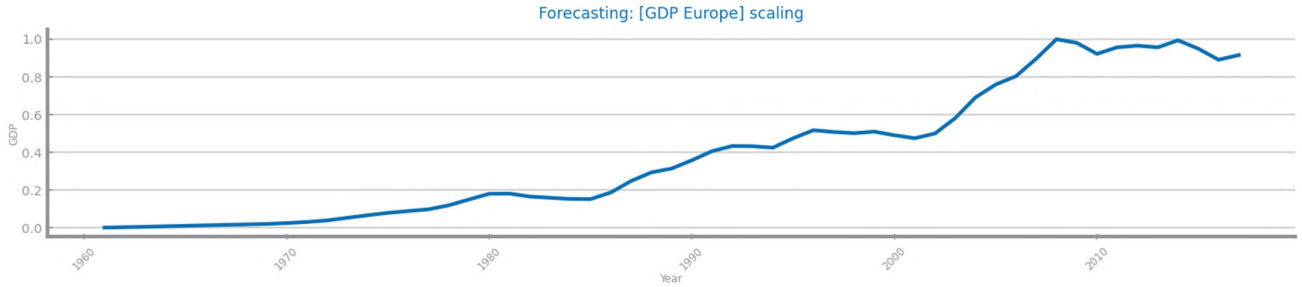


Figure 94 Forecasting plots after applying other transformations over time series 2

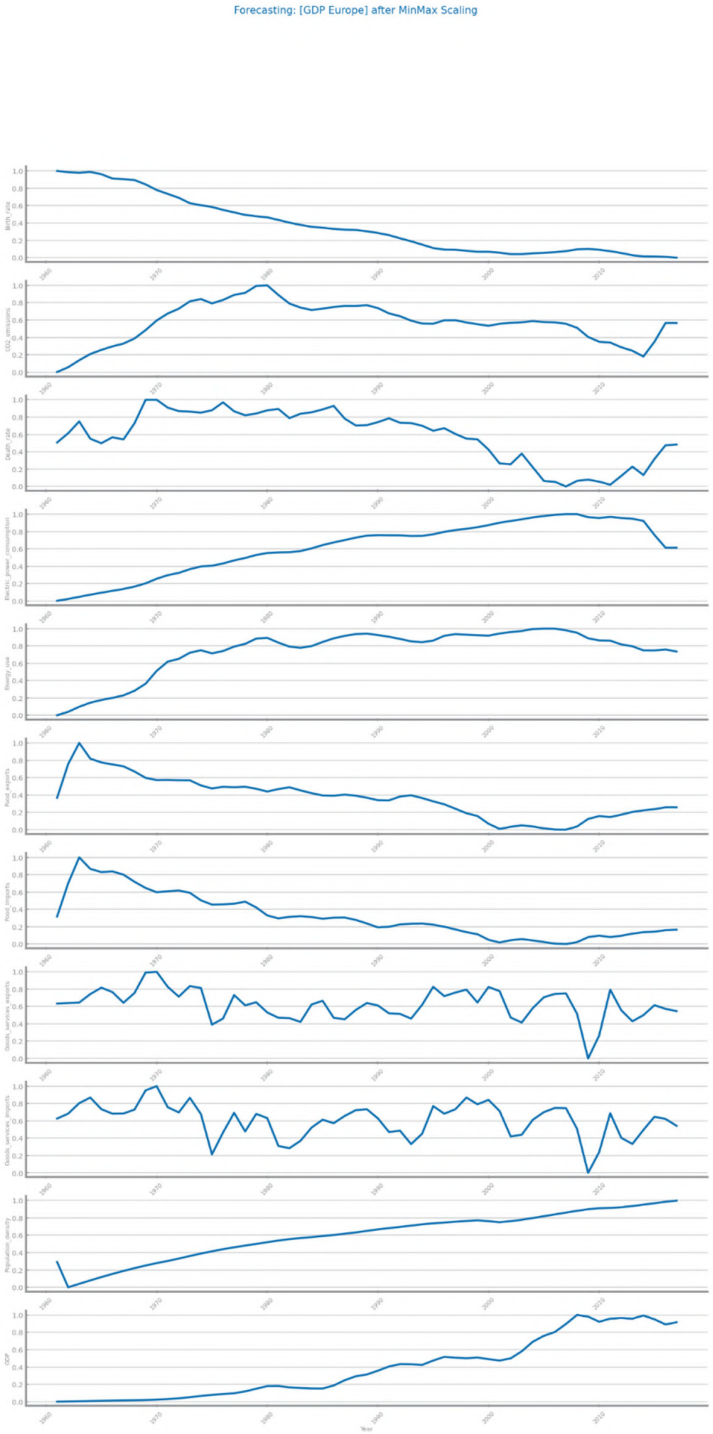


Figure 95 Forecasting results after applying other transformations over time series 2

7 MODELS' EVALUATION

This study evaluates several models for forecasting time series data from two datasets. These models include the Simple Average, Persistence, Rolling Mean, Exponential Smoothing, Linear Regression, ARIMA, and LSTMs. Various metrics, such as RMSE, MAE, MAPE, and R^2, will be used to assess the models' performance in capturing underlying patterns and making predictions. The aim is to identify which model predicts more accurately, minimizing errors in the process.

Simple Average Model

In Dataset 1, although the simple average model yields relatively small errors in absolute terms (RMSE and MAE), its overall performance is weak, particularly given the very low or negative R^2 values. This suggests that the model fails to capture important patterns in the time series. While the MAPE improves on the test set, the model remains suboptimal due to its poor R^2 performance.

In Dataset 2, the model fails to generalize, as evidenced by the high test errors, elevated MAPE values, and low or negative R^2. These results indicate overfitting and a lack of predictive capability for unseen data. The model is too simplistic to effectively capture the underlying GDP patterns.

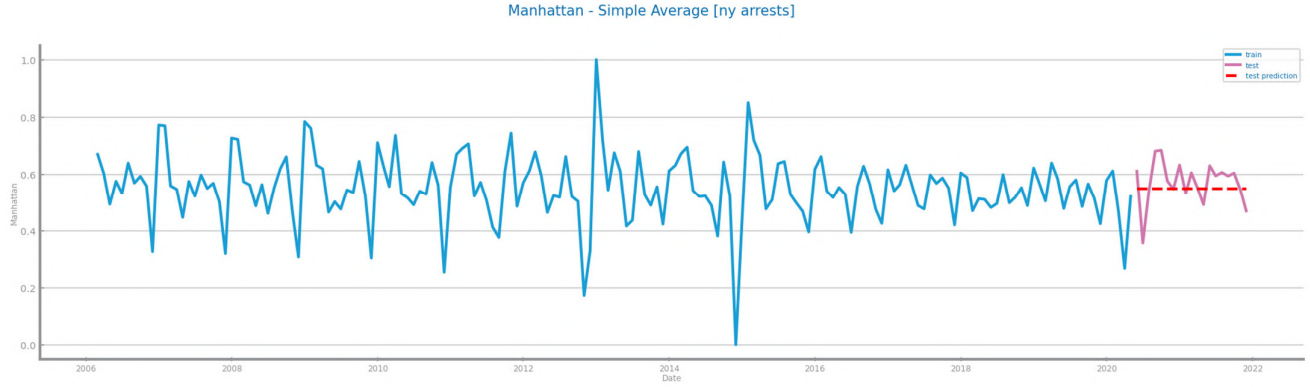


Figure 96 Forecasting plots obtained with Simple Average model over time series 1

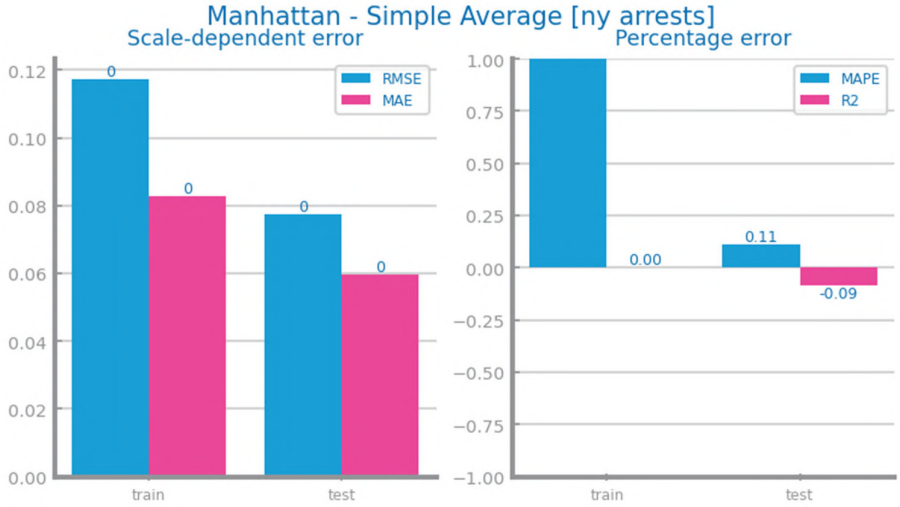


Figure 97 Forecasting results obtained with Simple Average model over time series 1

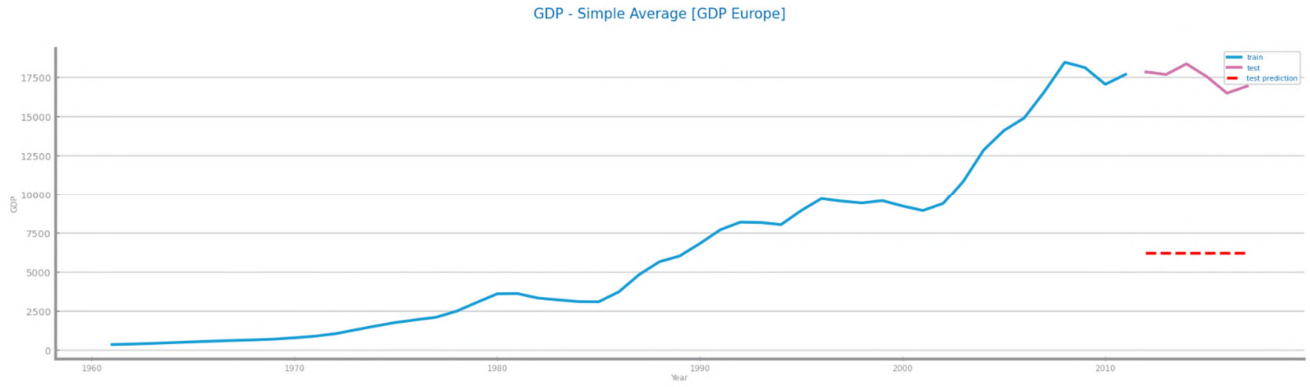


Figure 98 Forecasting plots obtained with Simple Average model over time series 2

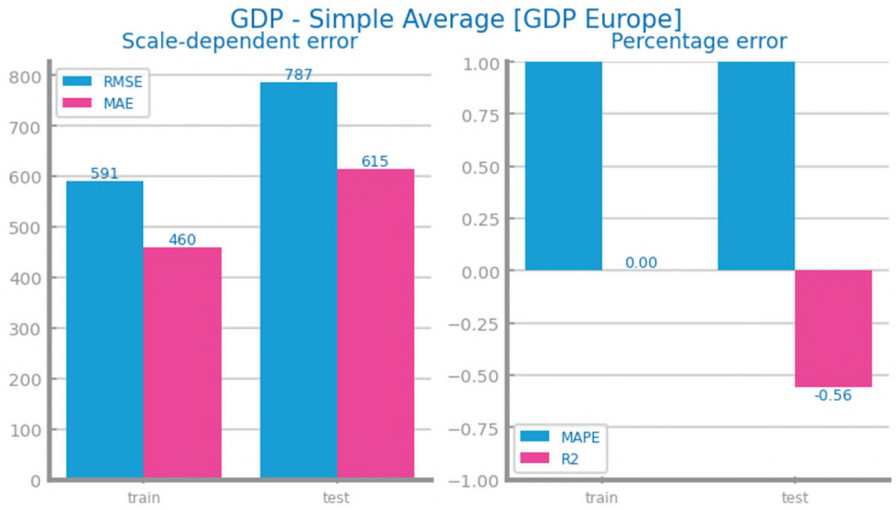


Figure 99 Forecasting results obtained with Simple Average model over time series 2

Persistence Model

In Dataset 1, the persistence models were obtained with the data before differentiation and scaling, as they allowed for better results. The optimist model performs well, with a high R^2 and reasonable errors. The forecasting is good, but the model may have some overfitting to the training data, as the errors on the training set are higher than on the test set. The realist model shows poor performance, with very high errors and negative R^2 values both on the training set and test set. The model fails to capture the patterns in the data, even showing some improvement on the test set.

In Dataset 2, both approaches (optimistic and realistic) showed poor performance in terms of R^2. Although the test set exhibited fewer errors (RMSE, MAE) than the train set and the R^2 of the optimistic model on the train set was good, all other indicators were poor (such as the MAPE of the train set and the R^2 of the test sets), with the R^2 of the realistic model on the train set being particularly bad.

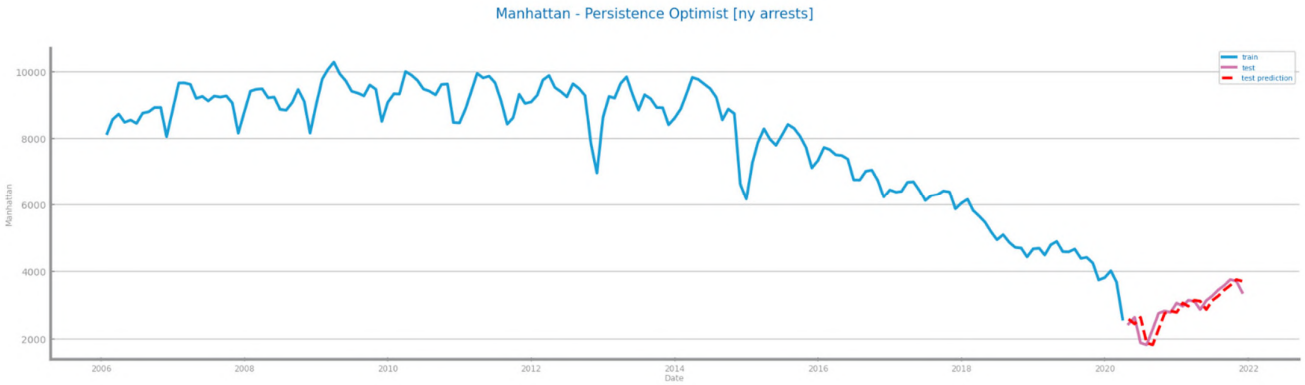


Figure 100 Forecasting plots obtained with Persistence model (long term) over time series 1

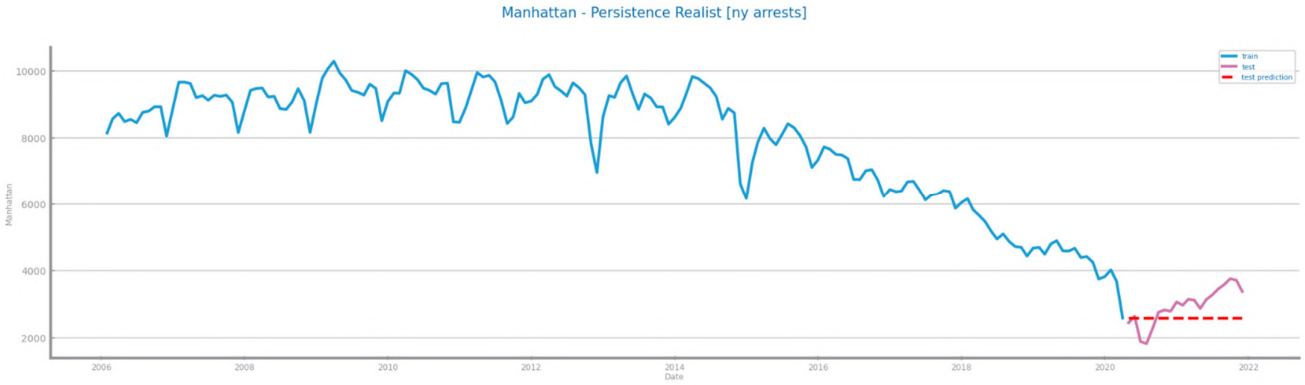
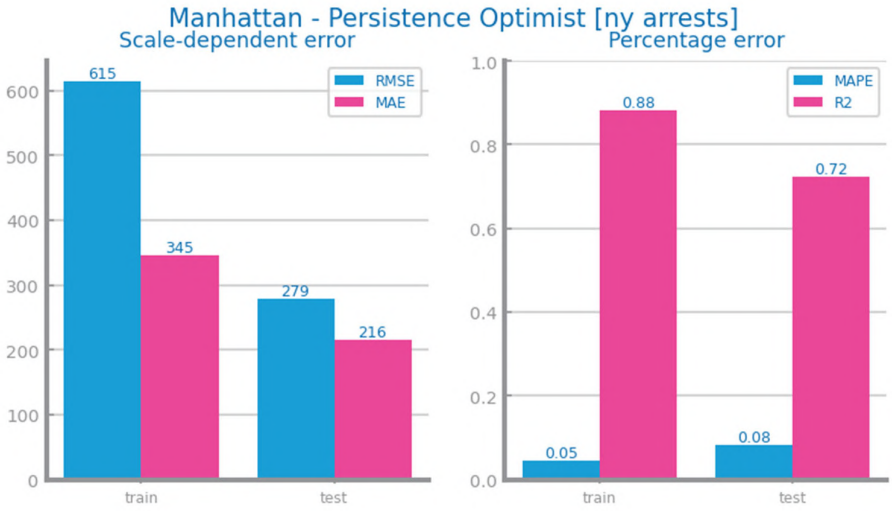


Figure 101 Forecasting plots obtained with Persistence model (next point) over time series 1



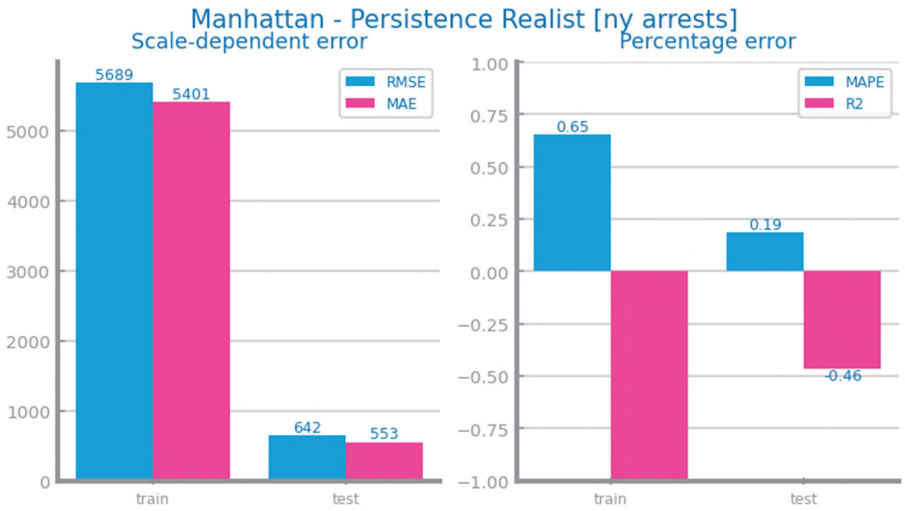


Figure 102 Forecasting results obtained with Persistence model in both situations over time series 1

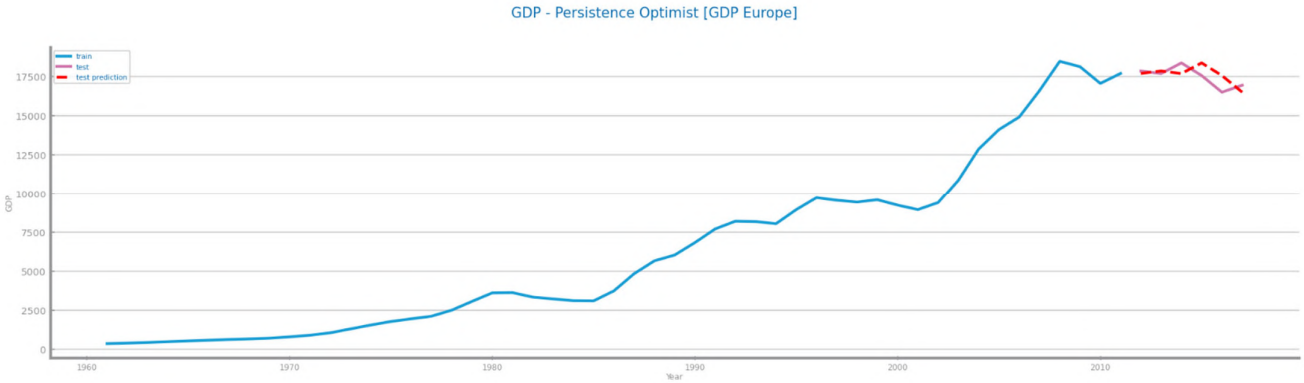


Figure 103 Forecasting plots obtained with Persistence model (long term) over time series 2

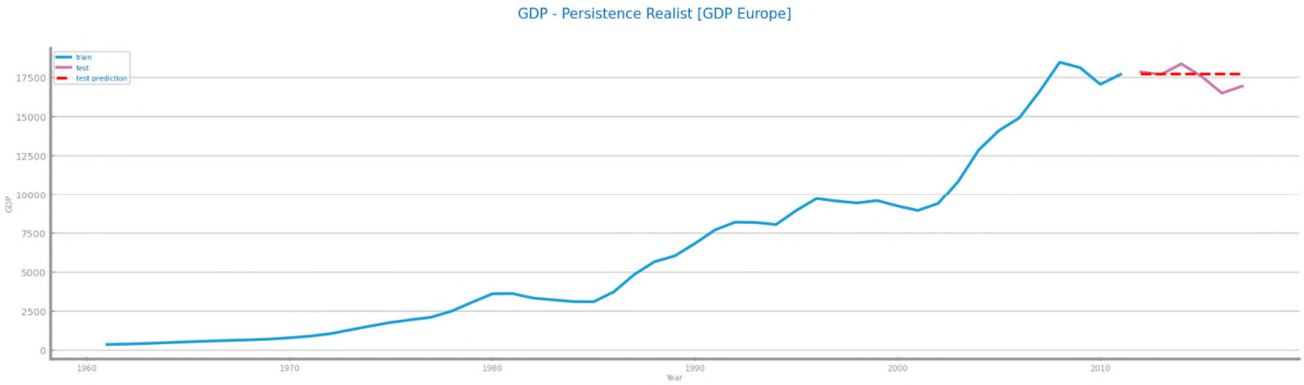


Figure 104 Forecasting plots obtained with Persistence model (next point) over time series 2

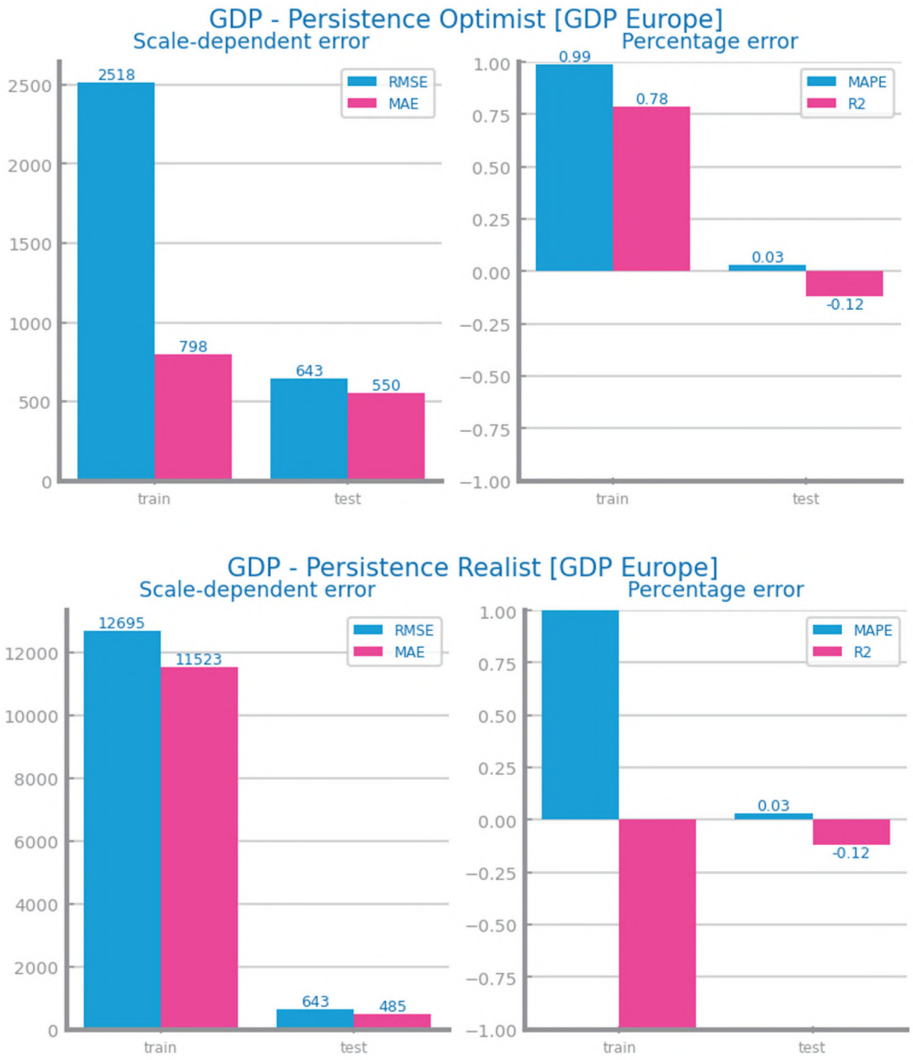


Figure 105 Forecasting results obtained with Persistence model in both situation over time series 2

Rolling Mean Model

In Dataset 1, the ideal win_size (170) is close to the total number of items (190), which causes the rolling mean to approach the global average of the series. This results in extreme smoothing, hiding local fluctuations and short-term patterns. While this method is suitable for stable series, it is inadequate for volatile series or those with seasonal patterns, as it does not capture recent variations. Although the rolling mean model has relatively small errors (low RMSE and MAE) and a reduced MAPE in the test set, the very low (and negative in the test set) R^2 indicates that the model is not effectively capturing the variability in the data.

In Dataset 2, the rolling mean algorithm is not ideal for predicting GDP. Even with the best win_size (5), we observe a high number of training errors compared to significantly lower errors in the test set. While the low MAPE for the test set indicates minimal prediction error, the R^2 value for the test set is close to zero, suggesting that the model is underperforming.

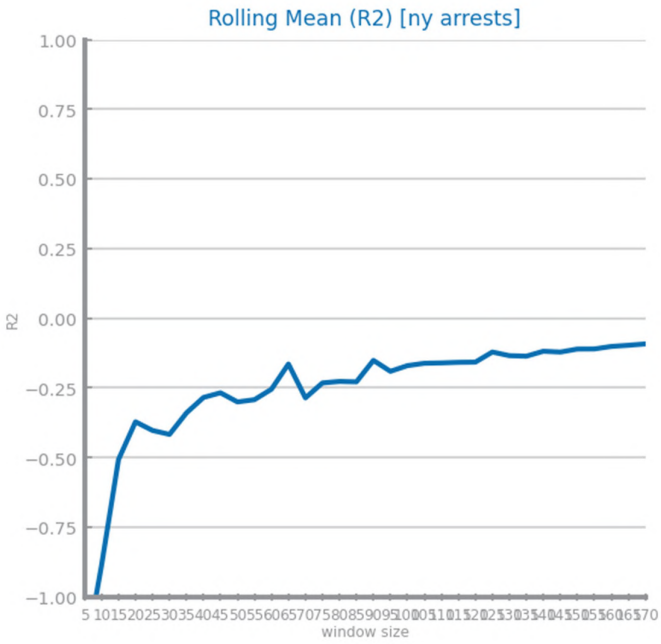


Figure 106 Forecasting study over different parameterizations of the Rolling Mean algorithm over time series 1

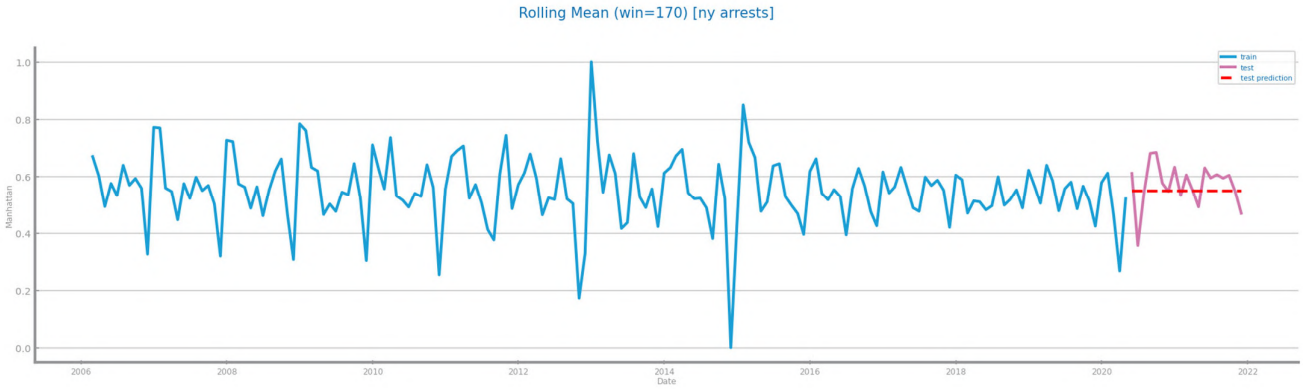


Figure 107 Forecasting plots obtained with the best parameterization of Rolling Mean algorithm, over time series 1

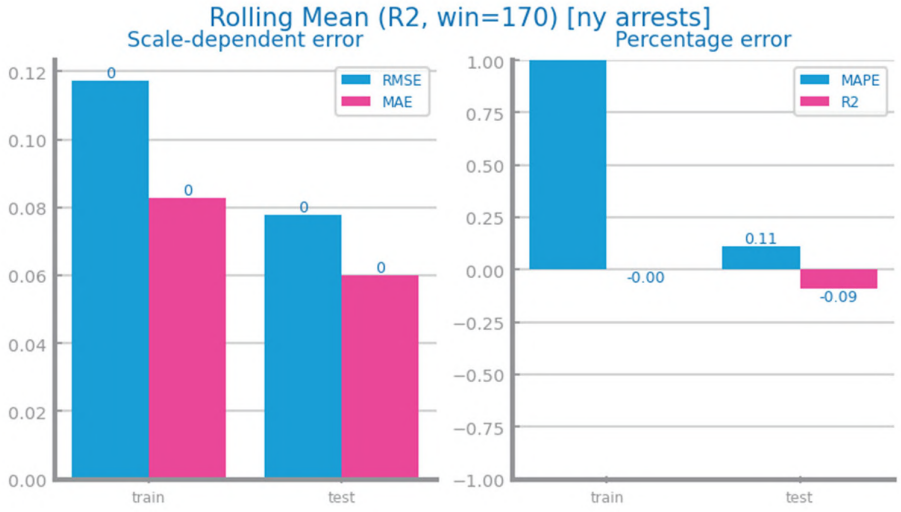


Figure 108 Forecasting results obtained with the best parameterization of Rolling Mean algorithm, over time series 1

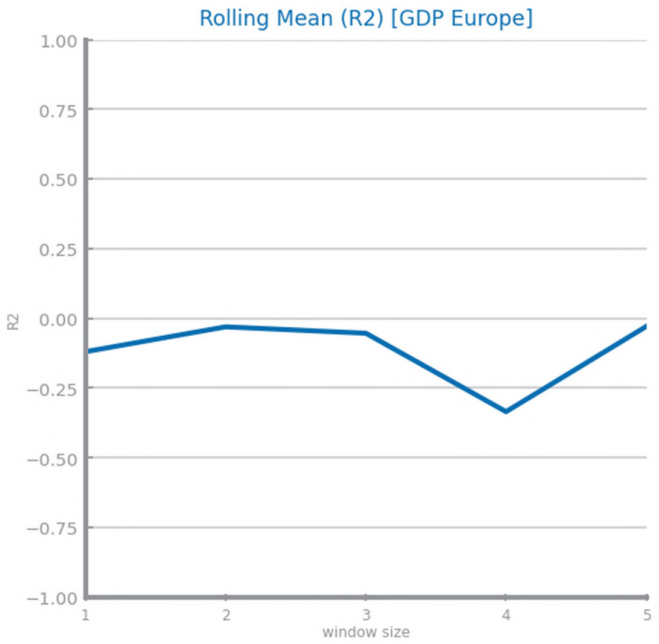


Figure 109 Forecasting study over different parameterizations of the Rolling Mean algorithm over time series 2

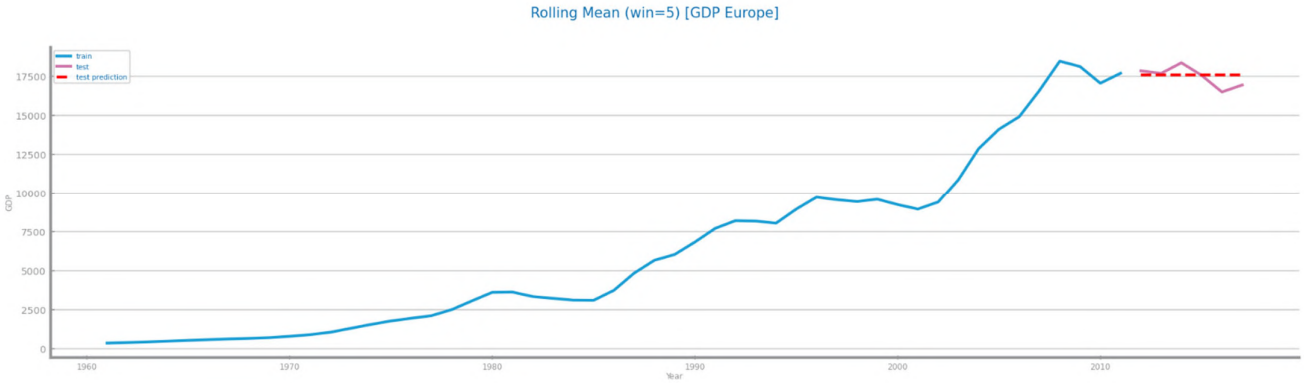


Figure 110 Forecasting plots obtained with the best parameterization of Rolling Mean algorithm, over time series 2

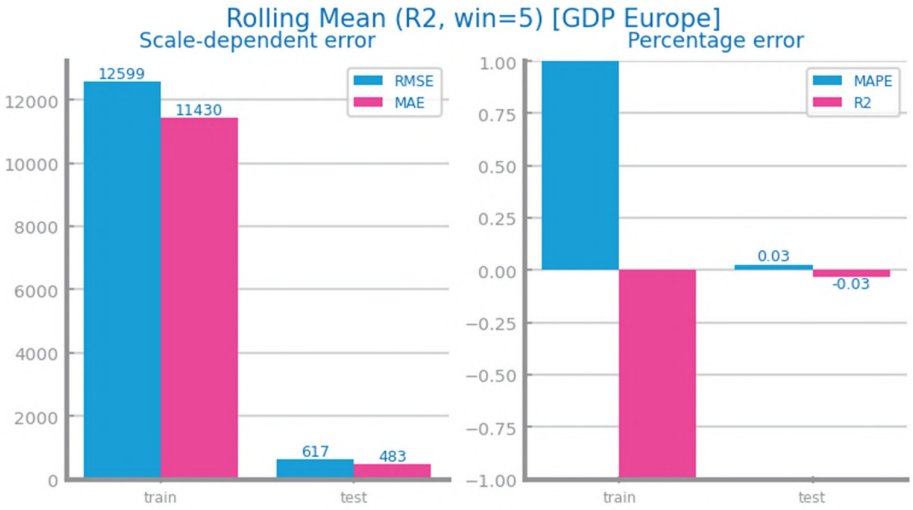


Figure 111 Forecasting results obtained with the best parameterization of Rolling Mean algorithm, over time series 2

Exponential Smoothing

In Dataset 1, the data was used without undergoing differentiation and scaling, as so it produced better results. The best alpha parameter was 0.7, which means that the model is highly sensitive to the most recent data, responding quickly to changes or fluctuations in the time series. The model performs very well, especially on the test set, with a very high R^2 (0.93), meaning it explains most of the variability in the data. The absolute error and RMSE are relatively low, but the MAE and MAPE on the test set are slightly higher, suggesting a slight loss of accuracy when generalizing to new data. Overall, the model is effective and has good forecasting capability for the time series, performing well both on the training and test sets.

In Dataset 2, the model with an alpha value of 0.5 demonstrated the best performance. It shows lower RMSE and MAE values for the test set compared to the train set. Additionally, the model performs well during training, as indicated by a very high R^2 value of 0.96, suggesting an excellent fit. However, the near-zero R^2 value for the test set indicates that while the model captures GDP trends effectively, it struggles with generalization to unseen data.

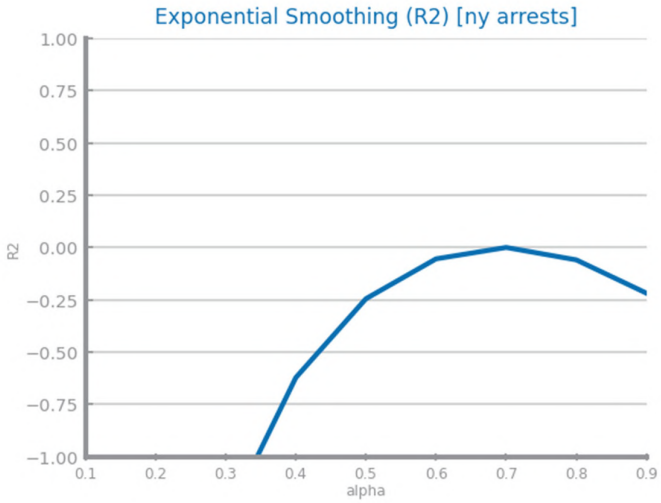


Figure 112 Forecasting study over different parameterizations of the Exponential Smoothing algorithm over time series 1

Figure 113 Forecasting plots obtained with the best parameterization of Exponential Smoothing algorithm, over time series 1

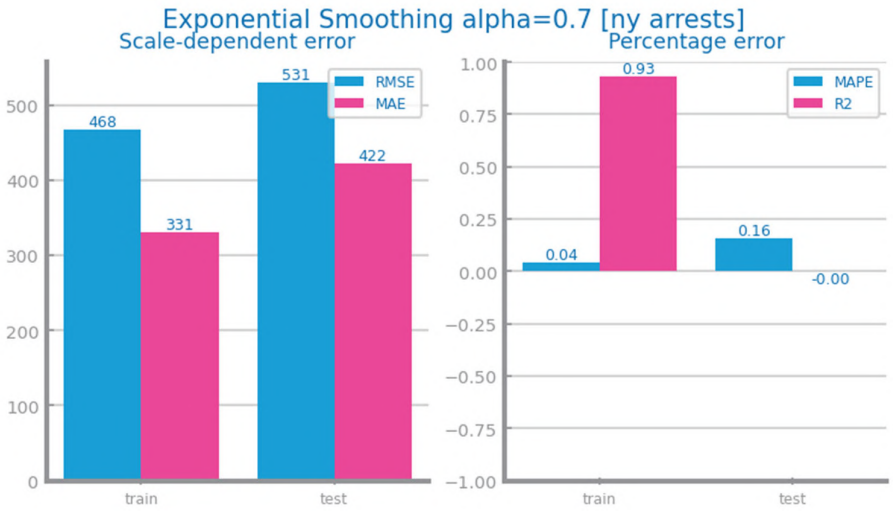


Figure 114 Forecasting results obtained with the best parameterization of Exponential Smoothing algorithm, over time series 1

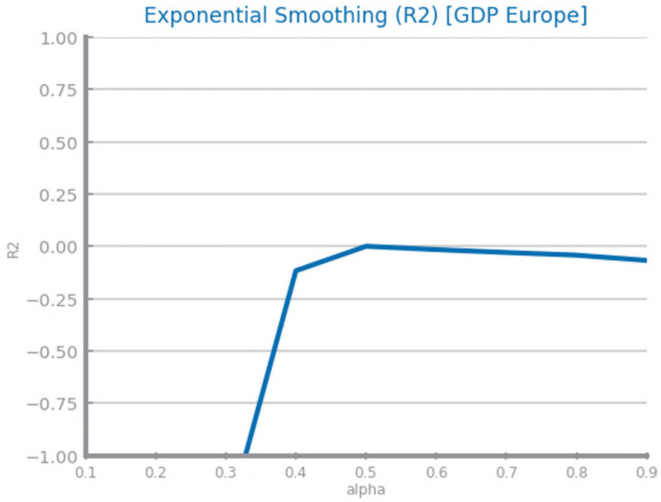


Figure 115 Forecasting study over different parameterizations of the Exponential Smoothing algorithm over time series 2

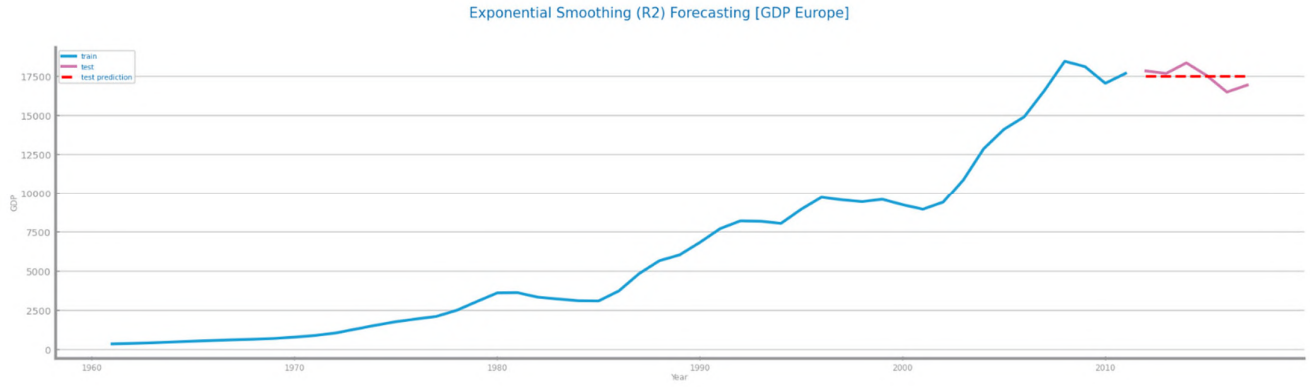


Figure 116 Forecasting plots obtained with the best parameterization of Exponential Smoothing, over time series 2

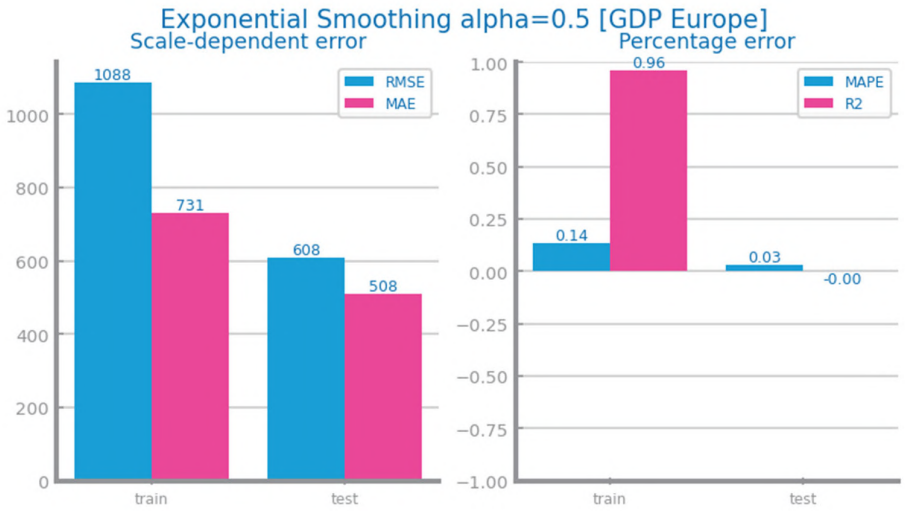


Figure 117 Forecasting results obtained with the best parameterization of Exponential Smoothing algorithm, over time series 2

Linear Regression

In Dataset 1, the linear regression model shows limited performance. The RMSE, MAE, and MAPE indicate that the model has relatively low errors for both training and testing, but the difference between them suggests that the model may not be adequately capturing the patterns of the time series. The R^2 value for the training set is close to zero, and the negative value for the test set indicates that the model is not explaining the variability of the data well, and in the case of the test set, it performs worse than simply using the mean. This is a clear sign that the model is not making good predictions.

In Dataset 2, linear regression shows poor performance. Both RMSE and MAE indicate substantial errors for both the training and testing data. While the R^2 value for the training set is strong (0.90), it is significantly disappointing for the test set (-1). This discrepancy suggests that the model is poorly suited for making accurate predictions with this dataset.

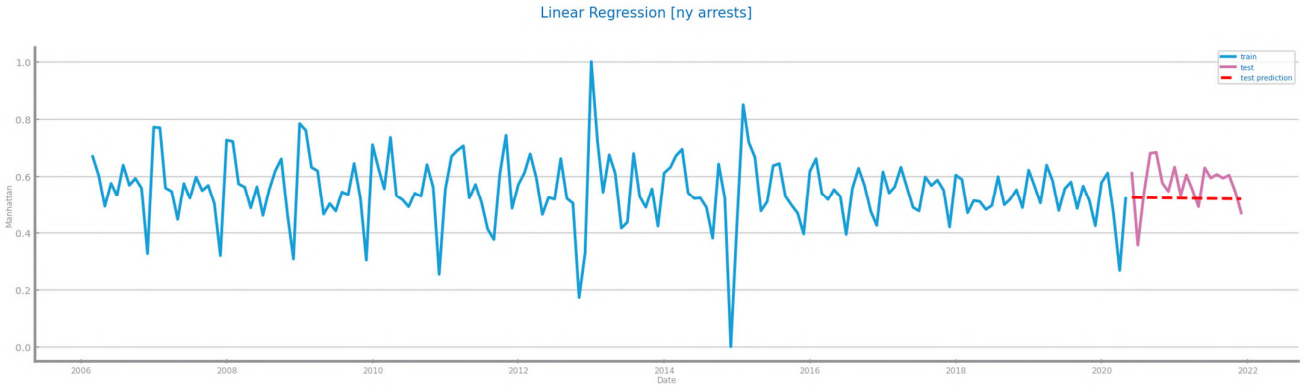


Figure 118 Forecasting plots obtained with Linear Regression model over time series 1

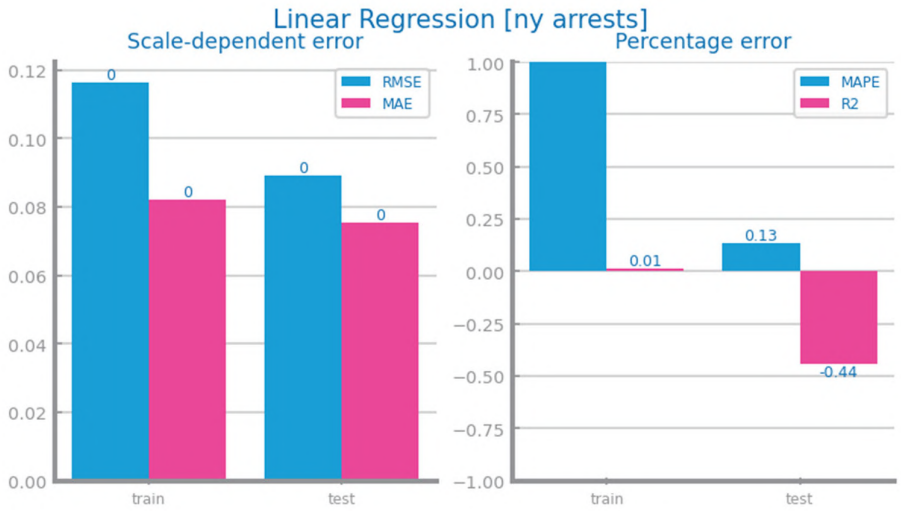


Figure 119 Forecasting results obtained with Linear Regression model over time series 1

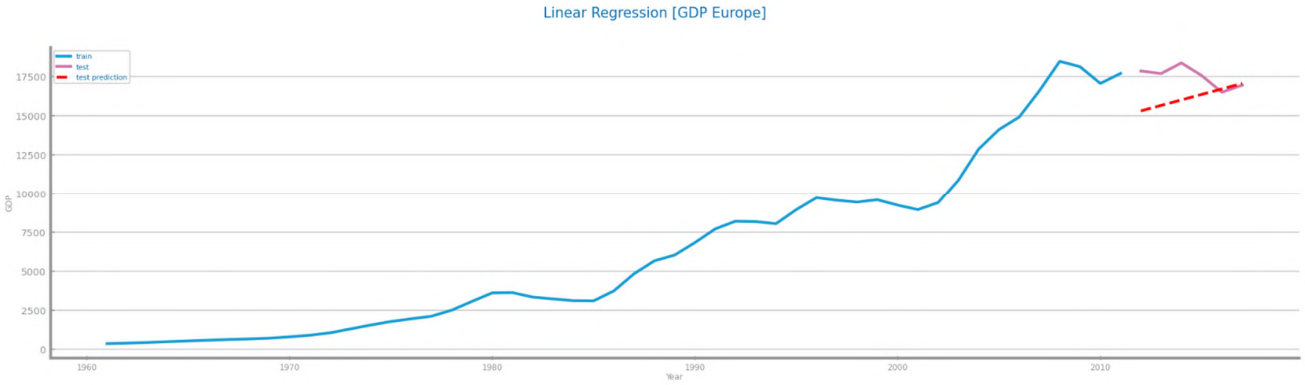


Figure 120 Forecasting plots obtained with Linear Regression model over time series 2

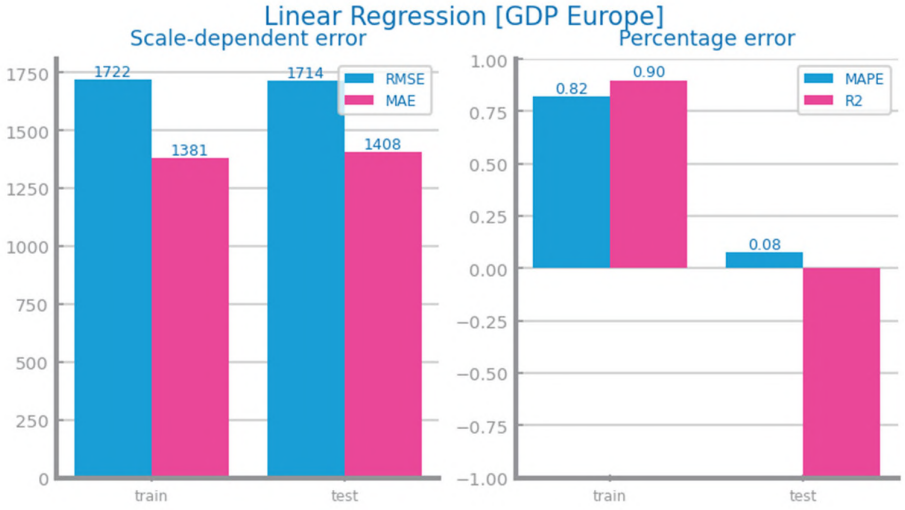


Figure 121 Forecasting results obtained with Linear Regression model over time series 2

ARIMA Model

In Dataset 1, for the ARIMA model the data was used without undergoing differentiation and scaling, as it produced better results. By univariate modelling, the model fits the training data *very well*, but its generalization to new, unseen data is weaker. The model appears to overfit the training set, as seen in the high R^2 for training and the higher RMSE and MAE for the test set. For the multivariate modelling, the model performs *excellently* on both the training and test sets, with high accuracy, low error, and strong explanatory power. While there is a slight drop in performance on the test set (evidenced by the higher RMSE, MAE, and lower R^2), the model still generalizes well and is very effective for this multivariate time series task.

In Dataset 2, the ARIMA univariate algorithm fits the training data very well, but its performance on unseen test data is highly disappointing. The high errors in both RMSE and MAE for the test data highlight the model's weakness in making accurate predictions. The ARIMA multivariate model shows good training performance, supported by a relatively high R^2 value of 0.72. However, the RMSE and MAE indicate that the model performed better on the training set than on the test set. In fact, the test results show a bad R^2 value of -1.00. Therefore, the ARIMA multivariate model is unsuitable for making accurate predictions with this dataset.

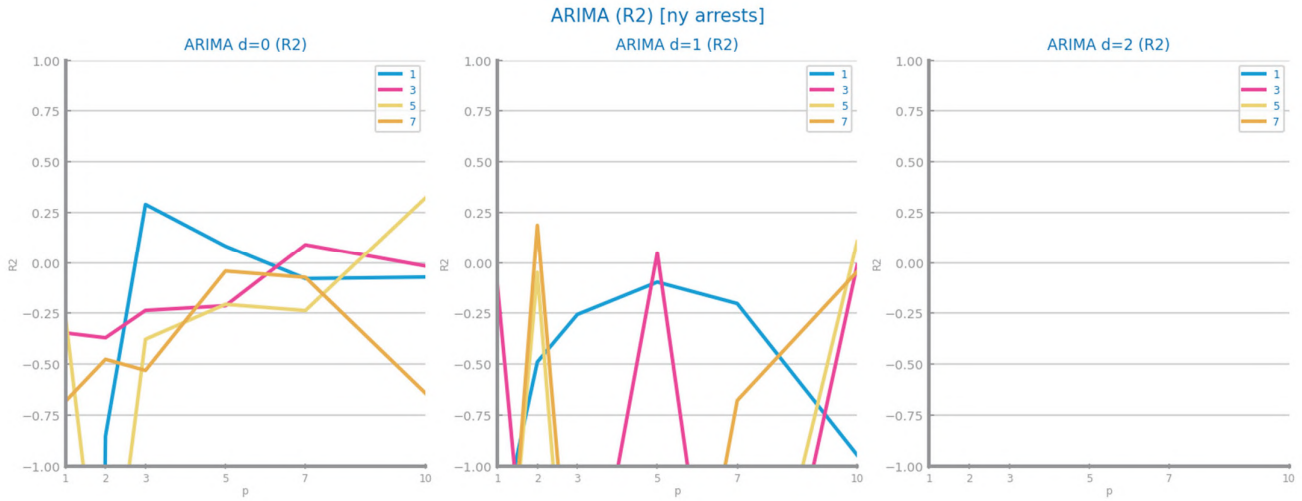


Figure 122 Forecasting study over different parameterizations of the ARIMA algorithm over time series 1, only with the target variable

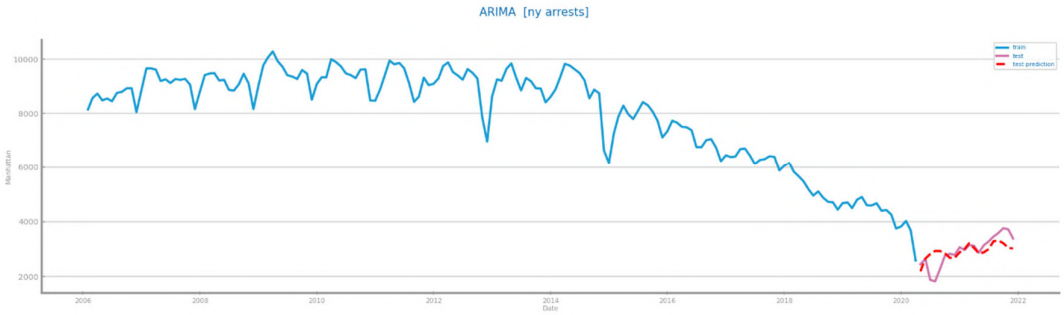


Figure 123 Forecasting plots obtained with the best parameterization of ARIMA algorithm, over time series 1, only with the target variable

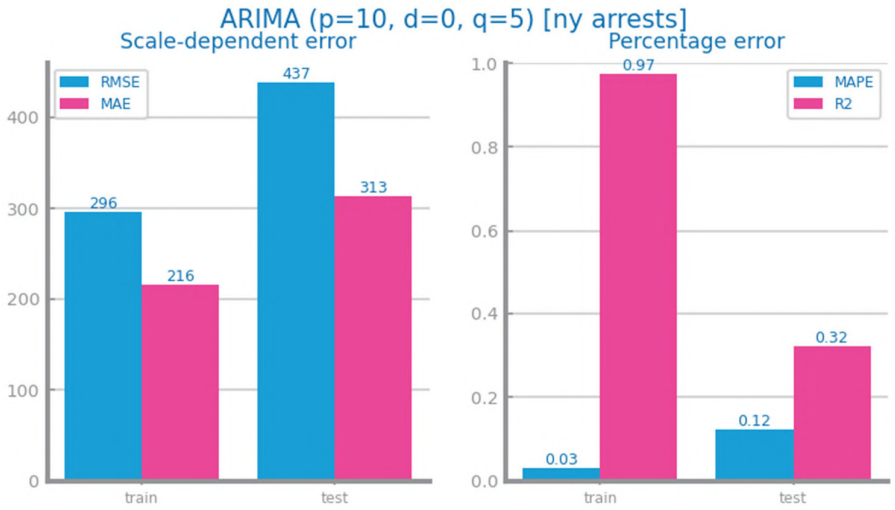


Figure 124 Forecasting results obtained with the best parameterization of ARIMA algorithm, over time series 1, only with the target variable

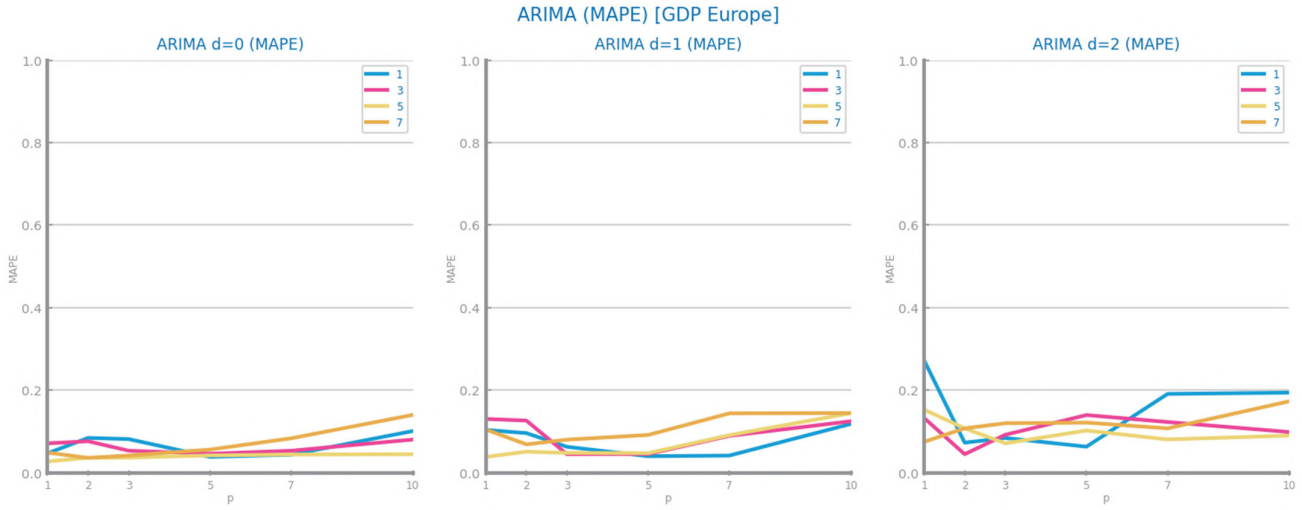


Figure 125 Forecasting study over different parameterizations of the ARIMA algorithm over time series 2, only with the target variable

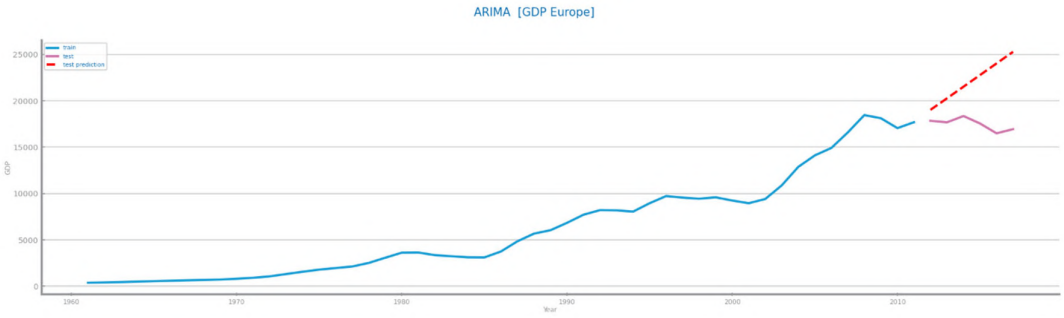


Figure 126 Forecasting plots obtained with the best parameterization of ARIMA algorithm, over time series 2, only with the target variable

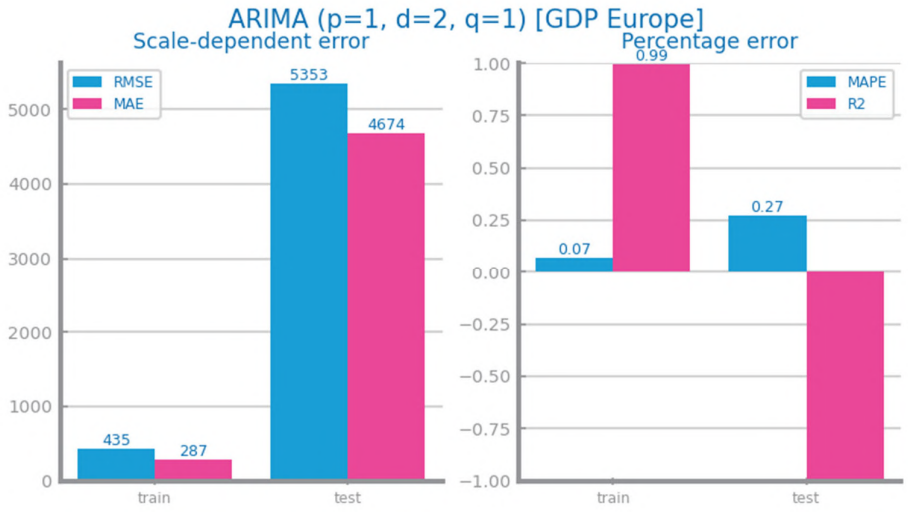


Figure 127 Forecasting results obtained with the best parameterization of ARIMA algorithm, over time series 2, only with the target variable

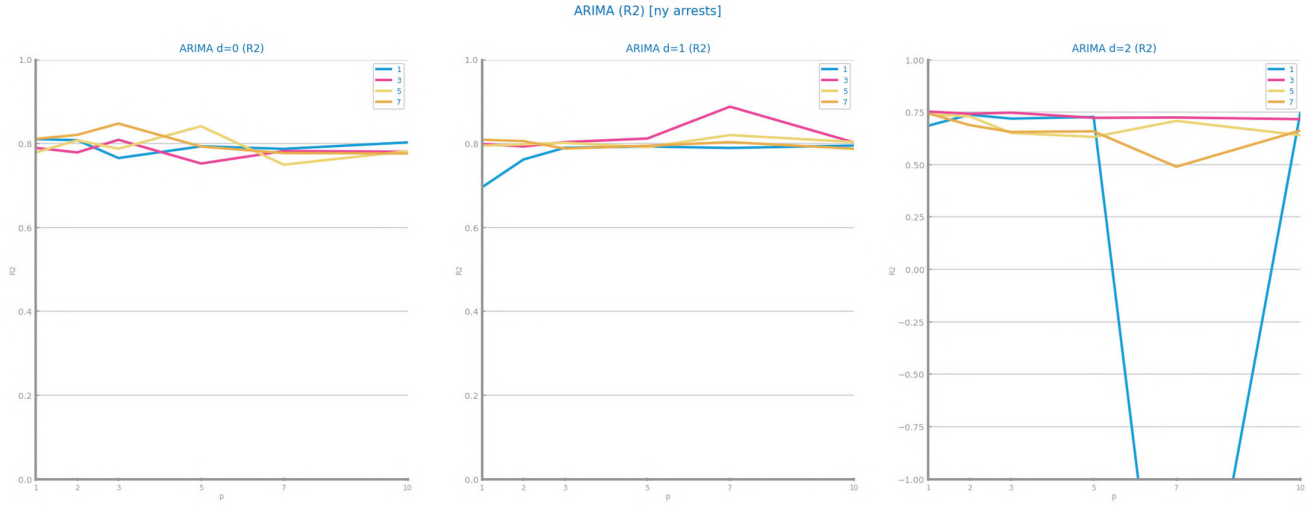


Figure 128 Forecasting study over different parameterizations of the ARIMA algorithm over time series 1, with multiple variables

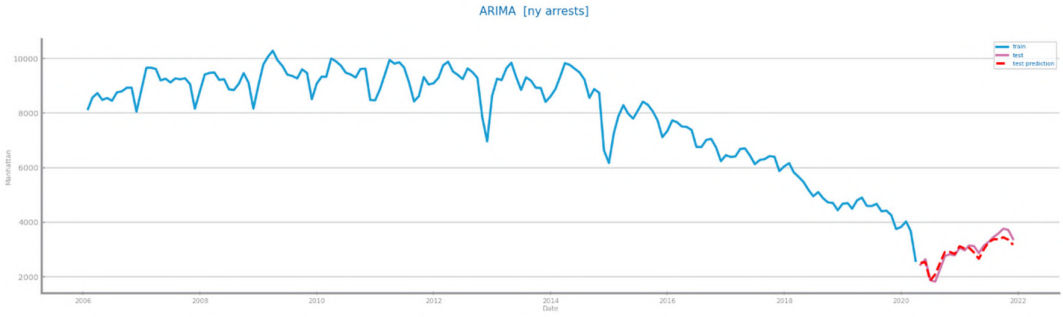


Figure 129 Forecasting plots obtained with the best parameterization of ARIMA algorithm, over time series 1, with multiple variables

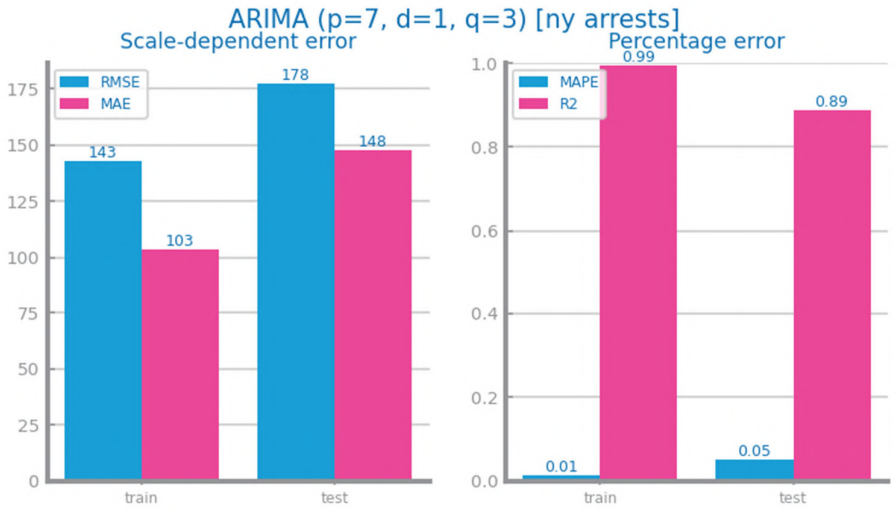


Figure 130 Forecasting results obtained with the best parameterization of ARIMA algorithm, over time series 1, with multiple variables

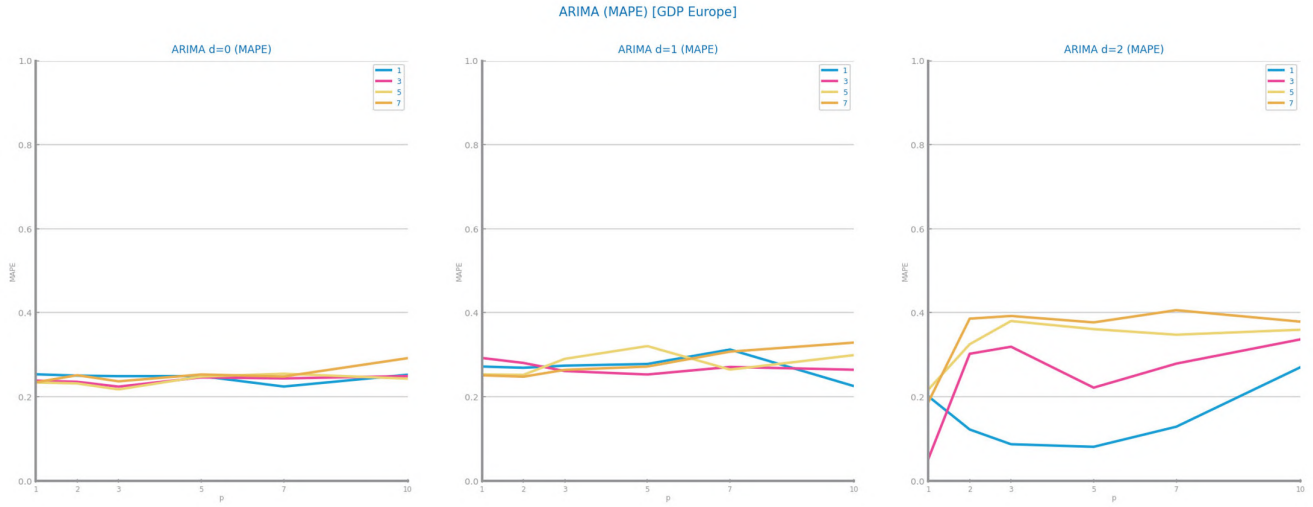


Figure 131 Forecasting study over different parameterizations of the ARIMA algorithm over time series 2, with multiple variables

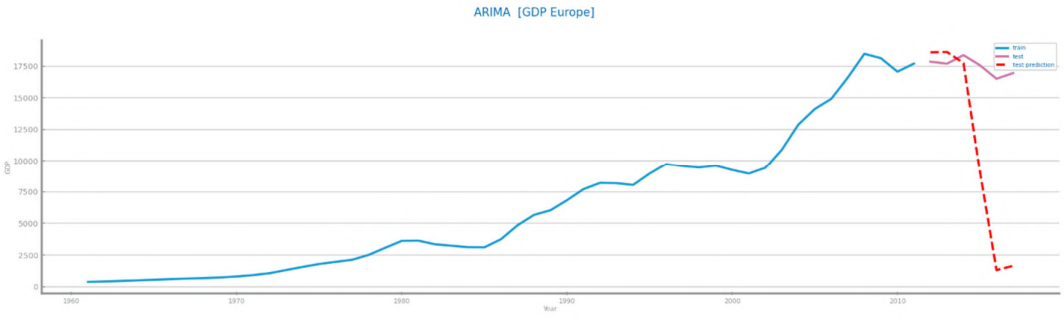


Figure 132 Forecasting plots obtained with the best parameterization of ARIMA algorithm, over time series 2, with multiple variables

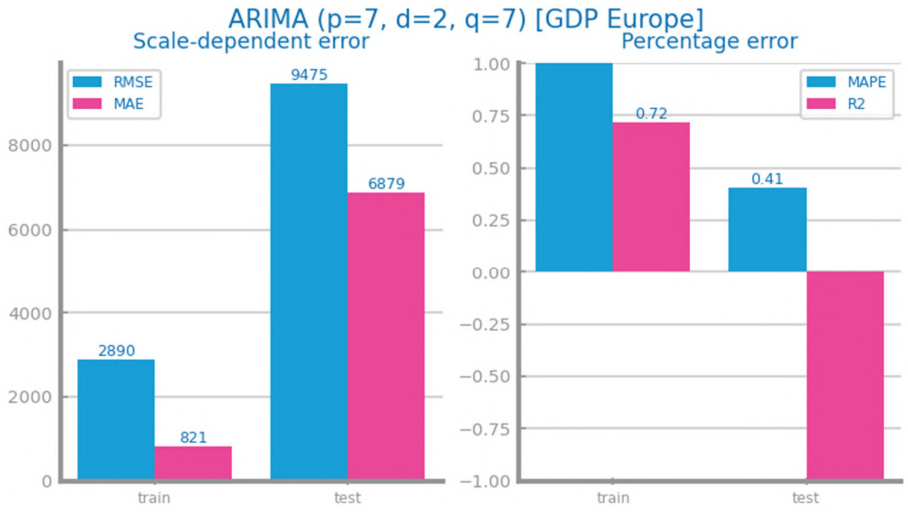


Figure 133 Forecasting results obtained with the best parameterization of ARIMA algorithm, over time series 2, with multiple variables

LSTMs Model

In Dataset 1, the LSTM univariate model is performing *reasonably well* on the test data in terms of error metrics (RMSE, MAE, MAPE), but it is not providing a good fit for the training data. The low R^2 values, especially the negative R^2 for the test set, indicate that the model is not capturing the underlying patterns in the data effectively. The model might be overfitting the training data and not generalizing well to unseen data. The LSTM multivariate model performs *relatively well* on the test data, with low RMSE, MAE, and MAPE, and a positive R^2, suggesting that it can make accurate predictions on unseen data. The model is also capturing some of the patterns in the multivariate data. The high MAPE for the training set suggests that the model might not be fitting the training data well in percentage terms, possibly due to overfitting.

In Dataset 2, the LSTM univariate model demonstrates moderate performance, with few errors observed in both the training and testing datasets. While the high R^2 value for the training set (1) indicates a perfect fit, the modest R^2 value for the test set suggests that the model is memorizing the data rather than learning underlying patterns. The LSTM multivariate model performs much worse, exhibiting a high error rate and a very negative R^2 value. Therefore, this model is not suitable for making predictions with this dataset.

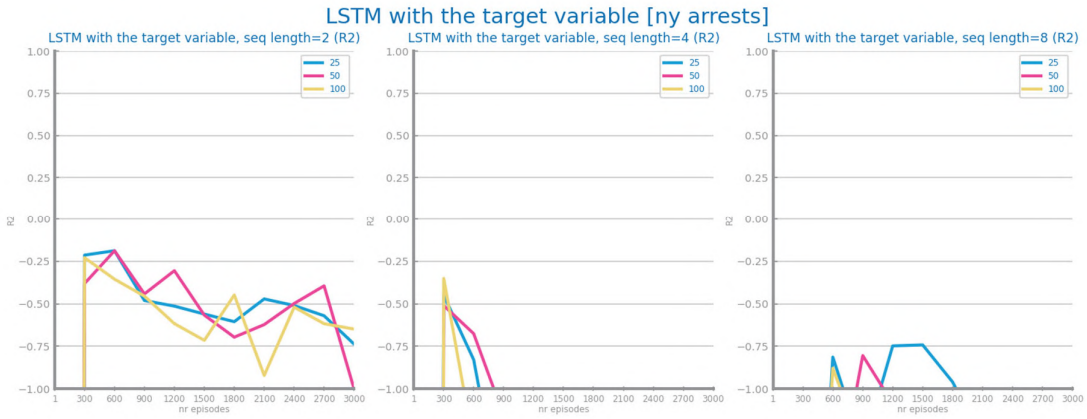


Figure 134 Forecasting study over different parameterizations of LSTMs over time series 1, only with the target variable

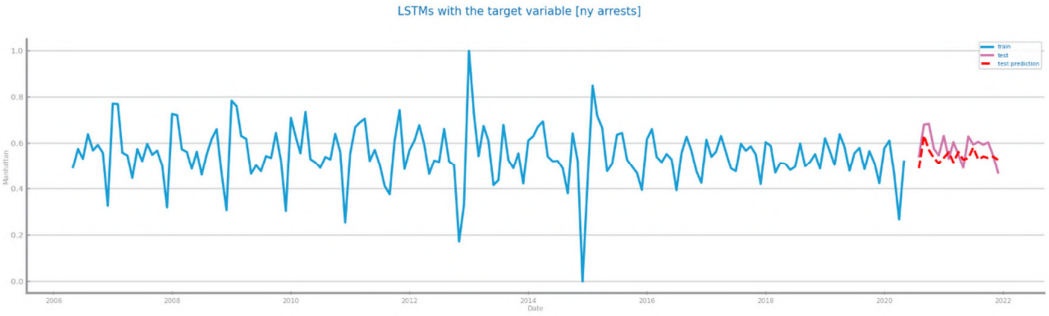


Figure 135 Forecasting plots obtained with the best parameterization of LSTMs over time series 1, only with the target variable

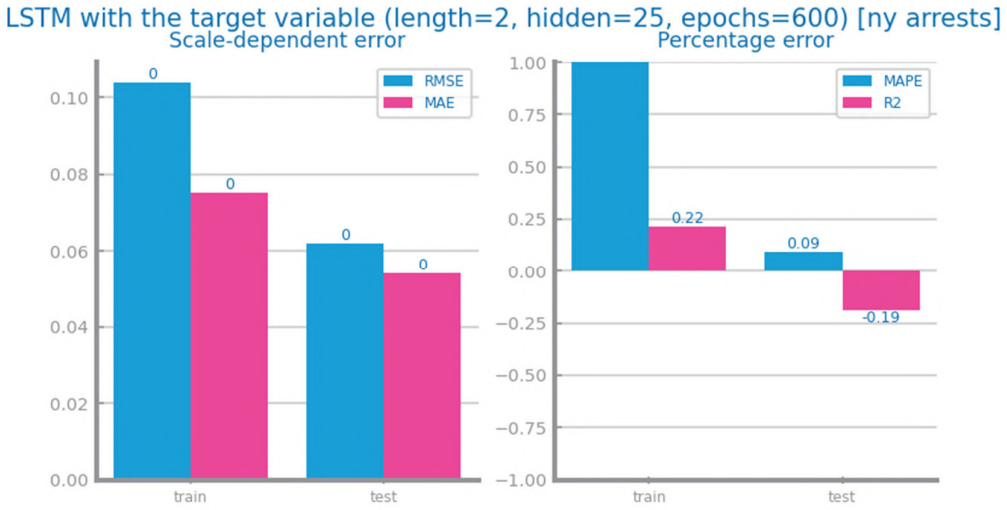


Figure 136 Forecasting results obtained with the best parameterization of LSTMs over time series 1, only with the target variable

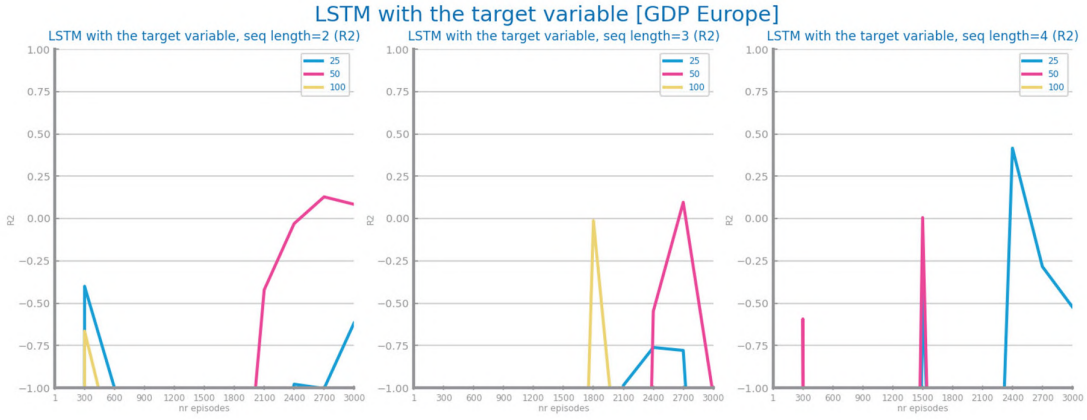


Figure 137 Forecasting study over different parameterizations of the LSTMs over time series 2, only with the target variable

Figure 138 Forecasting plots obtained with the best parameterization of LSTMs over time series 2, only with the target variable

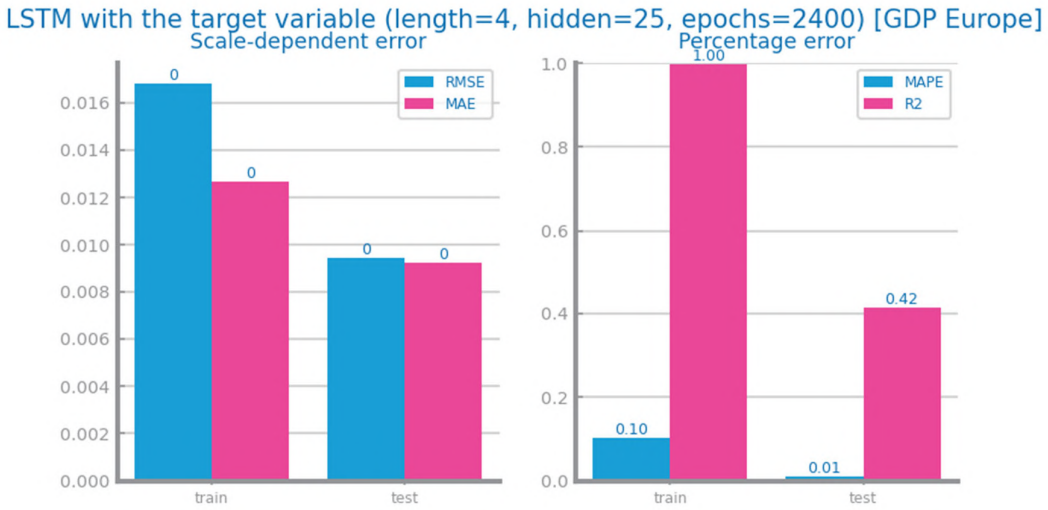


Figure 139 Forecasting results obtained with the best parameterization of LSTMs over time series 2, only with the target variable

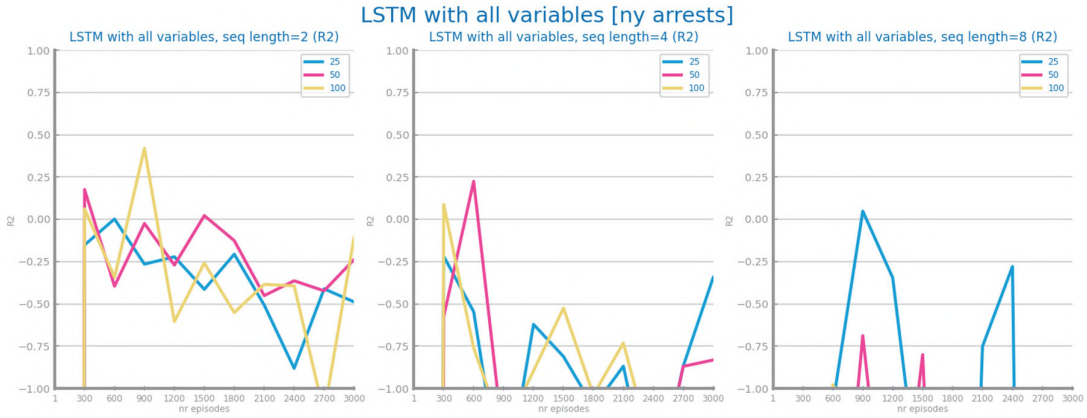


Figure 140 Forecasting study over different parameterizations of LSTMs over time series 1, with multiple variables

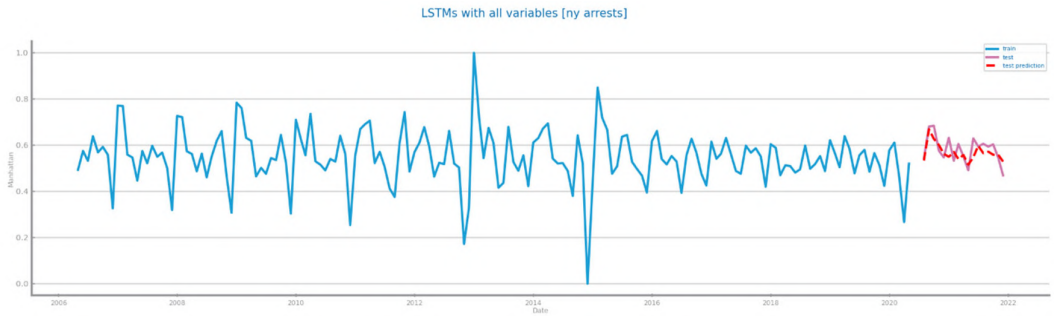


Figure 141 Forecasting plots obtained with the best parameterization of LSTMs over time series 1, with multiple variables

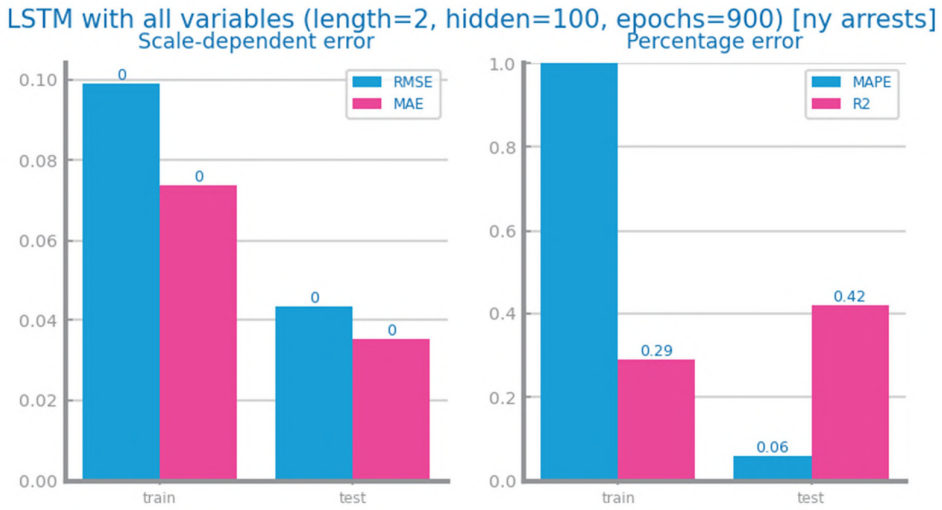


Figure 142 Forecasting results obtained with the best parameterization of LSTMs over time series 1, with multiple variables

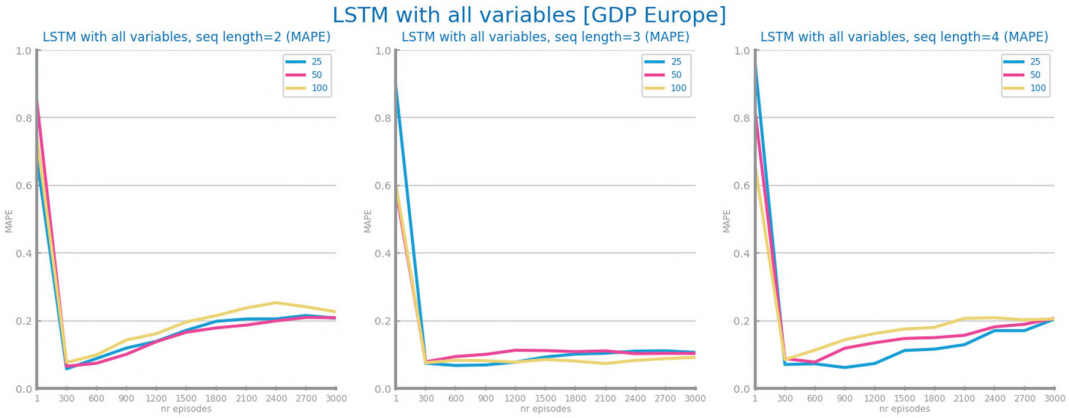


Figure 143 Forecasting study over different parameterizations of the LSTMs over time series 2, with multiple variables

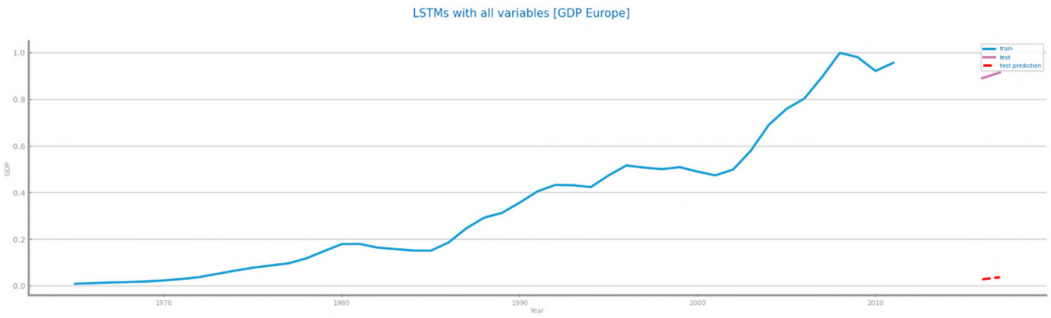


Figure 144 Forecasting plots obtained with the best parameterization of LSTMs over time series 2, with multiple variables

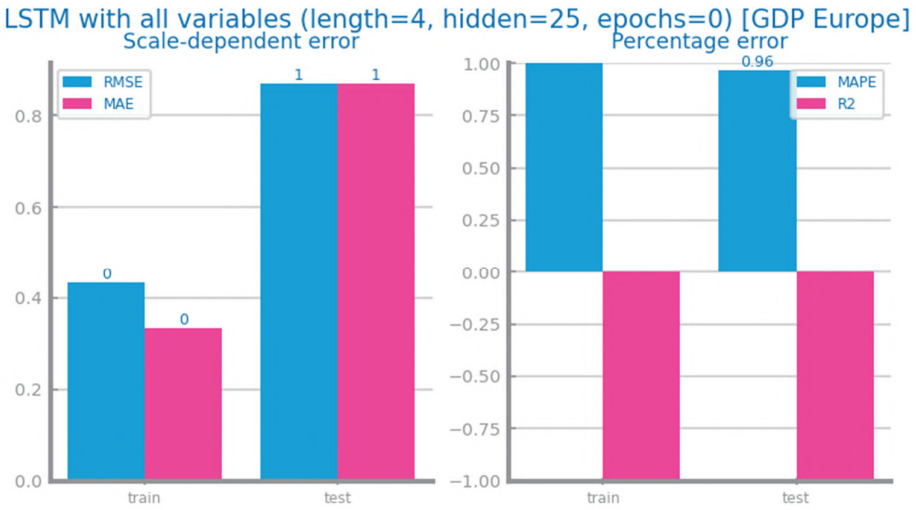


Figure 145 Forecasting results obtained with the best parameterization of LSTMs over time series 2, with multiple variables

8 CRITICAL ANALYSIS

Summary and Critical Assessment of Modeling Results

The **data preparation** process for both Dataset 1 and Dataset 2 involved several key steps to ensure the models' effectiveness. For Dataset 1, which had monthly granularity, adjustments were made to capture seasonal patterns, while Dataset 2 maintained yearly granularity to keep important fluctuations. Smoothing was applied using a window size of 2 to minimize noise, and differentiation was employed on Dataset 1 to remove trends, though it was not used on Dataset 2 due to its potential to introduce noise. Min-Max normalization was performed to stabilize the neural network model, namely the LSTM.

The models tested on Datasets 1 and 2 displayed mixed results. In Dataset 1, the **Simple Average** model showed small errors but poor R^2, indicating it missed key patterns. The **Persistence** model performed well with an optimistic approach but showed signs of overfitting. The **Rolling Mean** model, while having low errors, struggled to capture fluctuations due to excessive smoothing. **Exponential Smoothing** performed well, especially in Dataset 1, but faced challenges in Dataset 2, where it struggled with generalization. **Linear Regression** performed poorly in both datasets, with low R^2 values. **ARIMA** performed well on training data but struggled with test data, especially in univariate form. The **LSTM** model had issues with overfitting in Dataset 1, although the multivariate approach performed better. In Dataset 2, LSTM struggled with poor performance, particularly in multivariate forecasting.

Critical Assessment

The results reveal that many models overfit or fail to generalize, particularly when faced with new data. Simpler models like the **Simple Average** and **Persistence** models were unable to capture the complexities of the time series data, while more advanced models like **ARIMA** and **LSTM** also struggled with generalization. The granularity of the data and preprocessing choices, such as the lack of differentiation or scaling, played a key role in model performance. The **Rolling Mean** model, for example, could have failed to capture important variations due to excessive smoothing. Overall, while some models showed potential, there is a clear need for improved regularization techniques, hybrid models, and better handling of data fluctuations to improve predictive accuracy.