

Data Science Project

Team nr: 16	Student 1: Antero Morgado IST nr: 1119213 Student 2: David Ferreira IST nr: 1107077 Student 3: José Fernandes IST nr: 1103727 Student 4: Olha Buts IST nr: 1116276
--------------------	---

CLASSIFICATION

1 DATA PROFILING

May be used to describe any useful observation about the data, and that was used in the current project. An example is the use of any domain knowledge to process the data or evaluate the results. **Shall not exceed 200 characters.**

Data Dimensionality

Regarding the Missing Values analysis, the completeness metric for Dataset 1 was adjusted to treat 'UNKNOWN' occurrences as missing values.

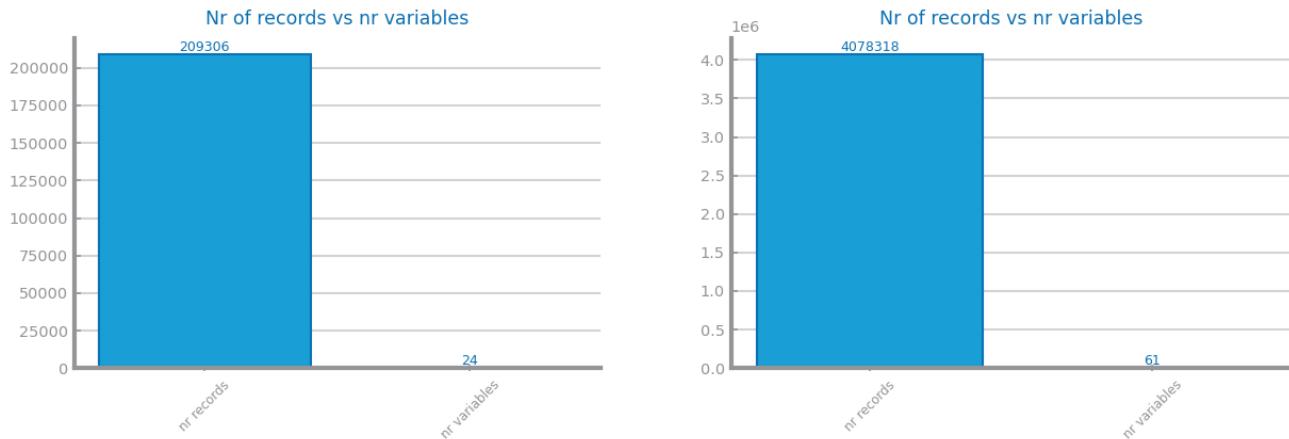


Figure 1: Nr Records x Nr variables for dataset 1 (left) and dataset 2 (right)

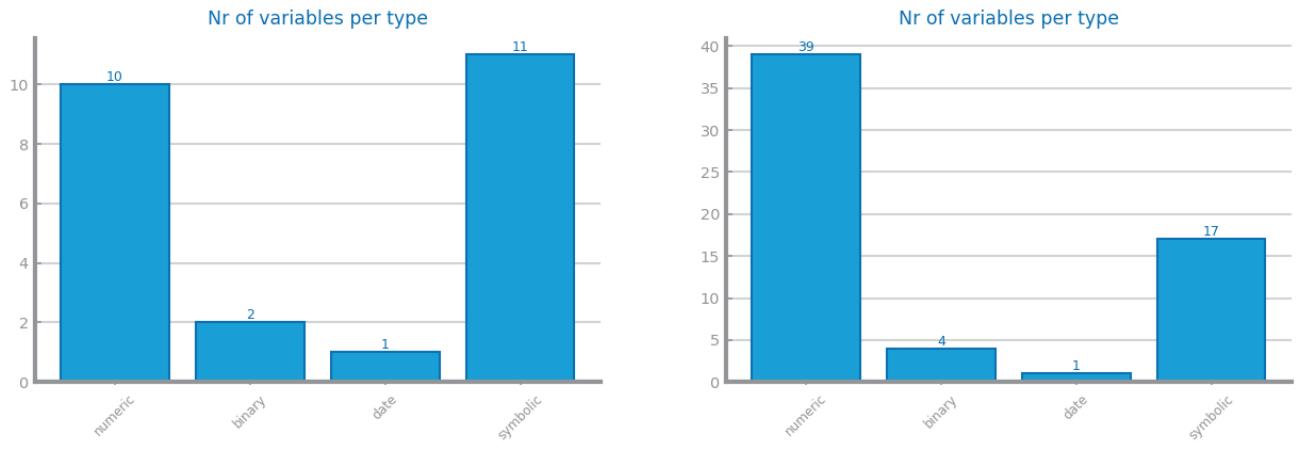


Figure 2: Nr variables per type for dataset 1 (left) and dataset 2 (right)

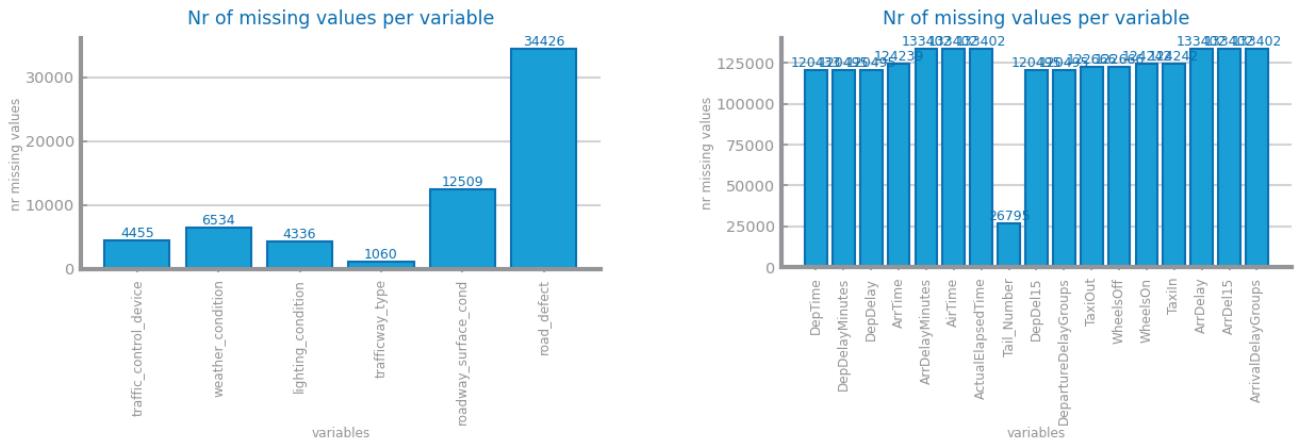


Figure 3: Nr missing values for dataset 1 (left) and dataset 2 (right)

Data Distribution

The distribution analysis reveals that the raw data is significantly dispersed, exhibiting high variance across several features. Certain variables show a clear lack of balance, with distributions heavily skewed toward specific classes or ranges and presence of evident outliers.

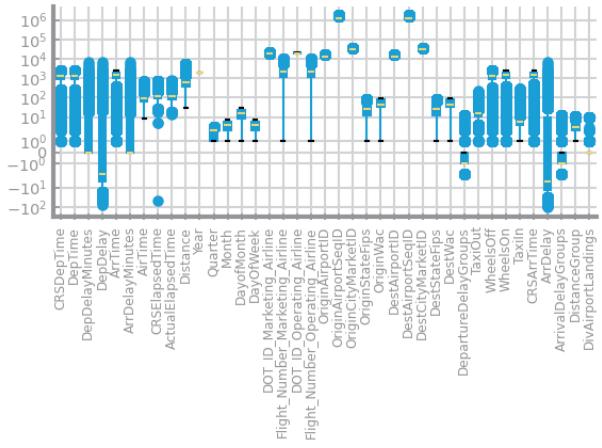
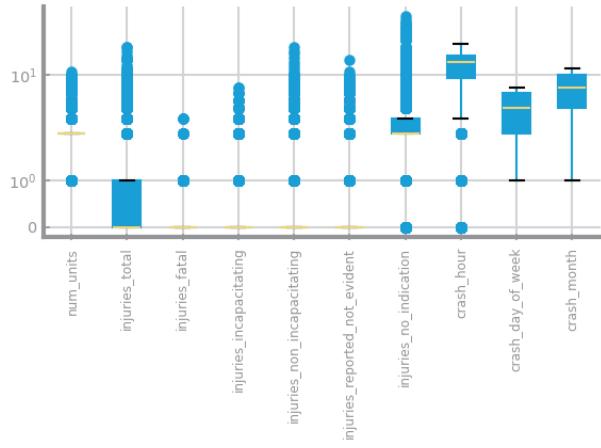


Figure 4: Global boxplots dataset 1 (left) and dataset 2 (right)

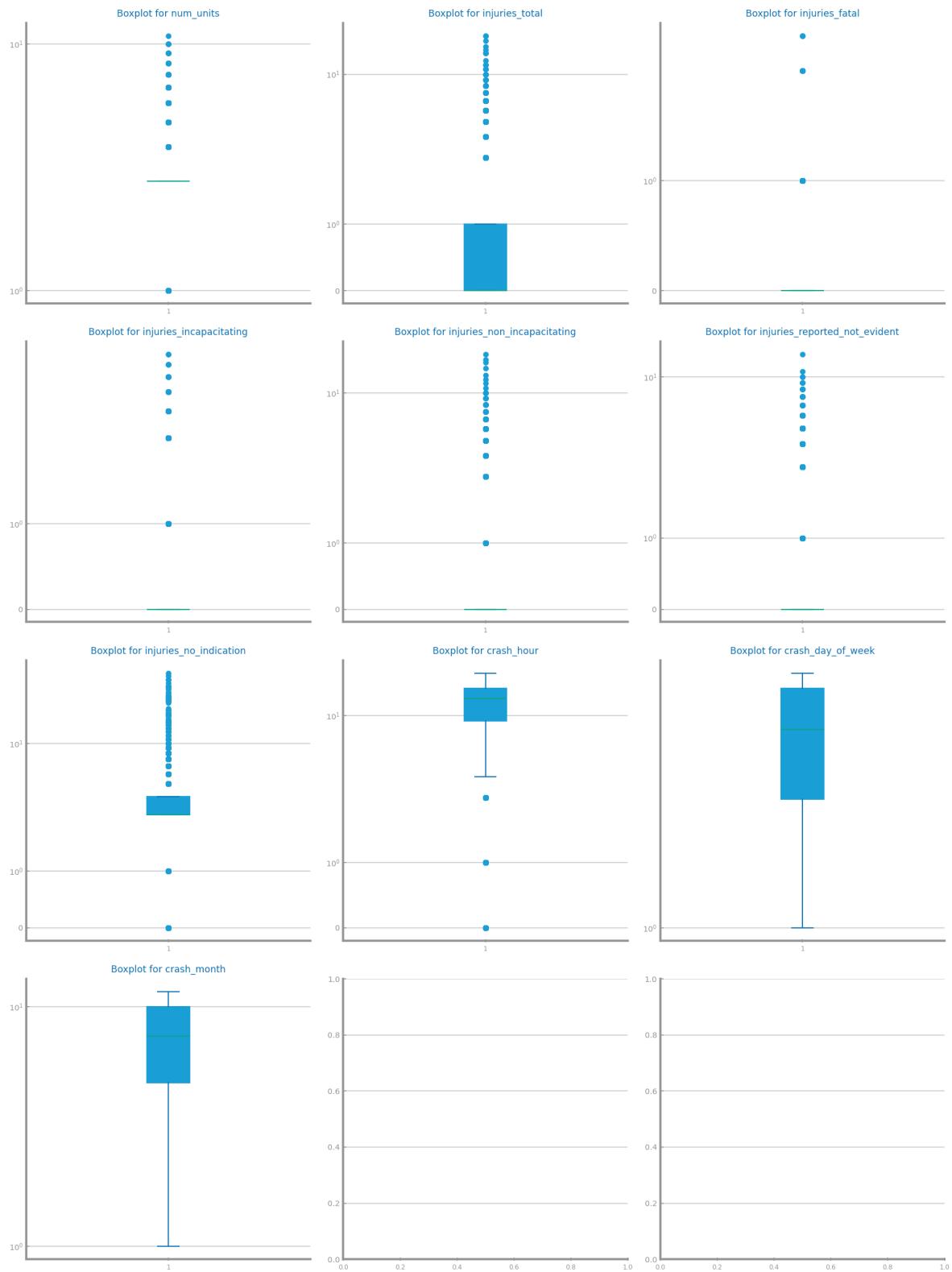


Figure 5: Single variables boxplots for dataset 1

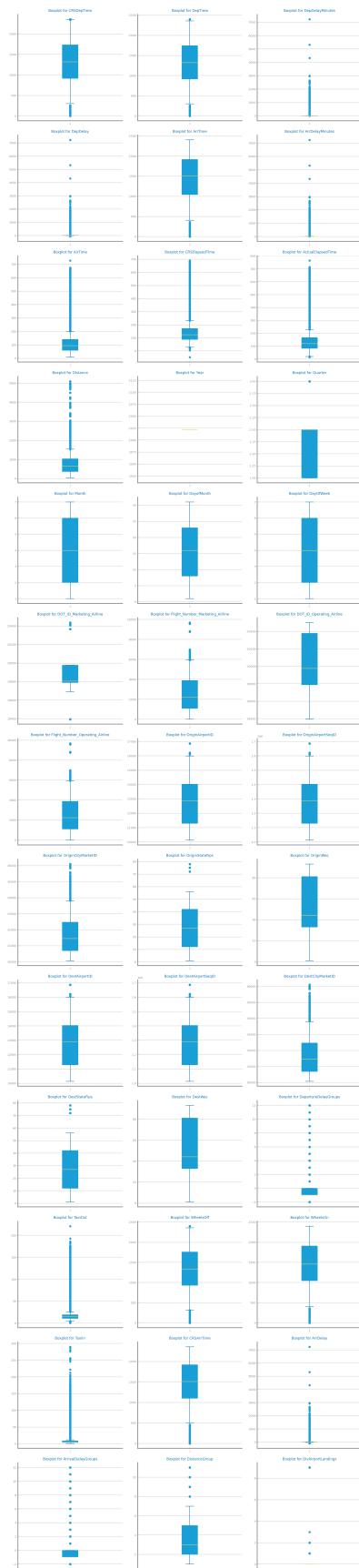


Figure 6: Single variables boxplots for dataset 2

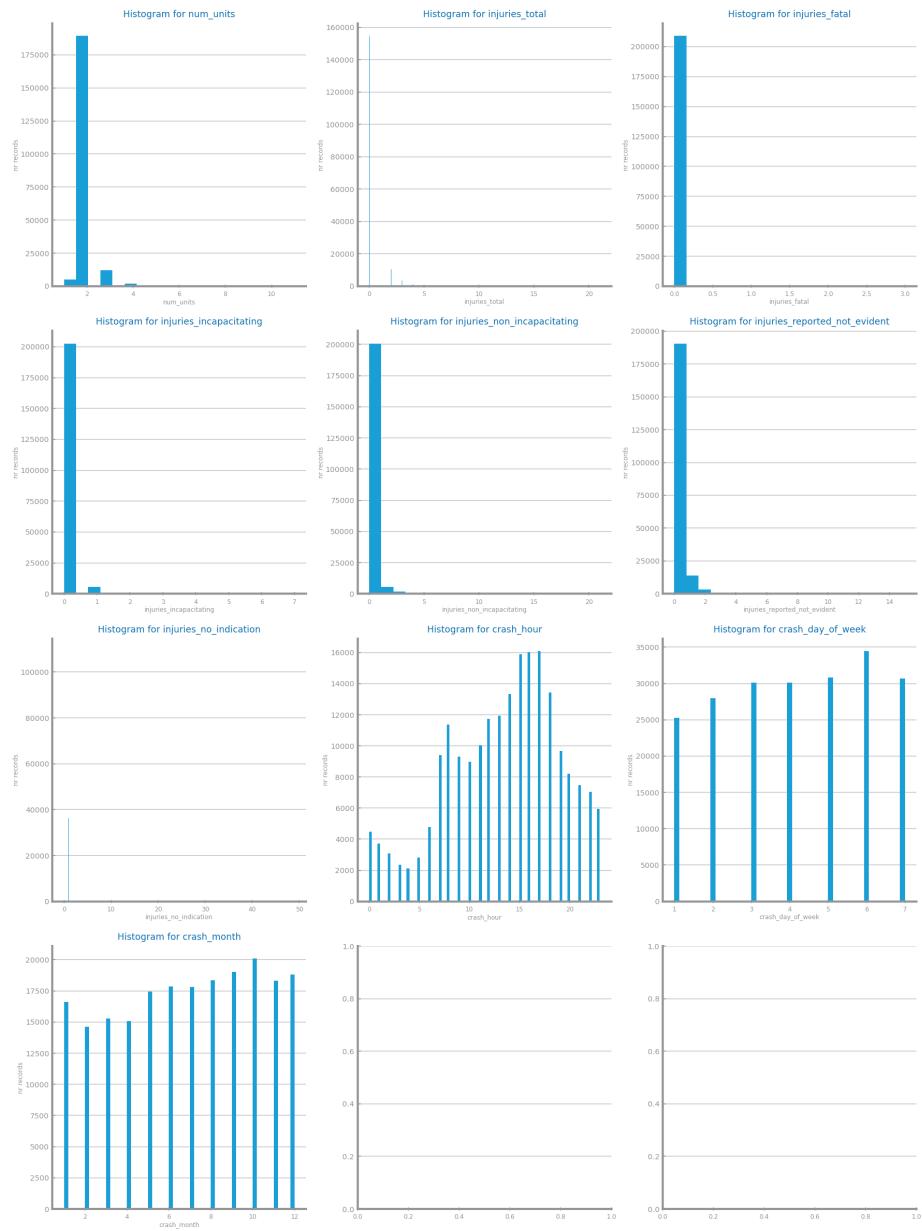


Figure 7: Histograms for dataset 1

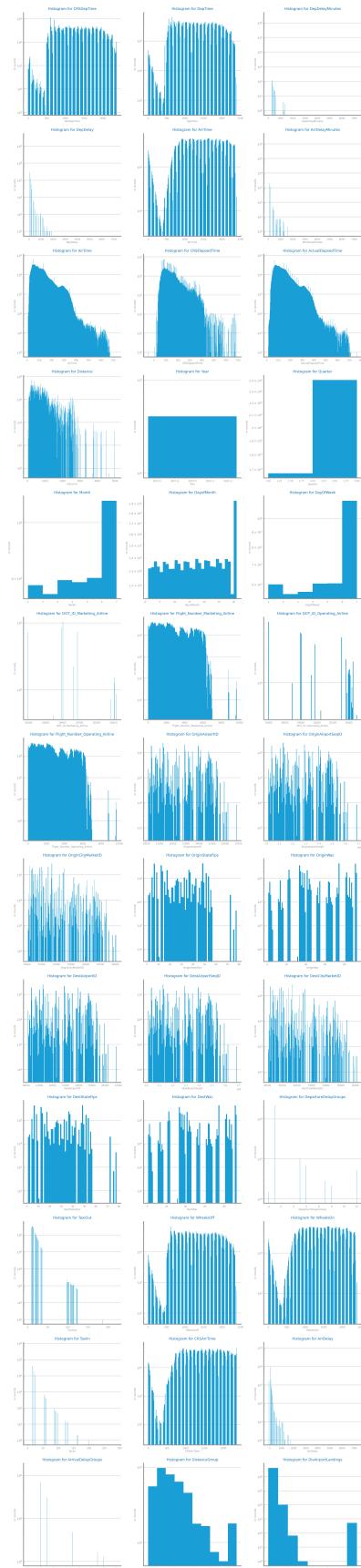


Figure 8: Histograms for dataset 2

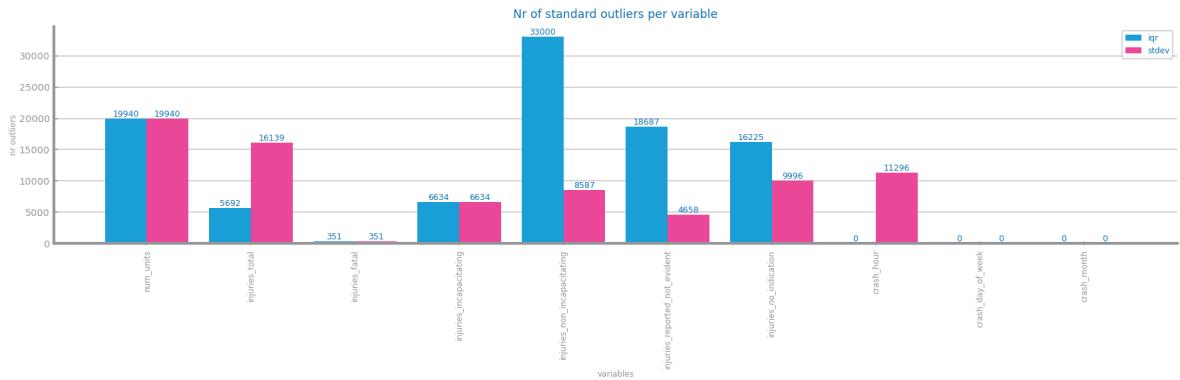


Figure 9: Outliers study dataset 1

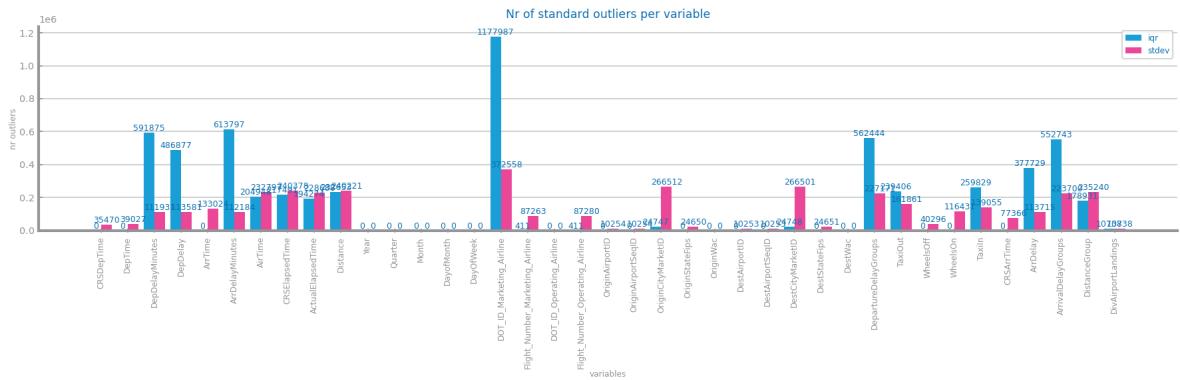


Figure 10: Outliers study dataset 2

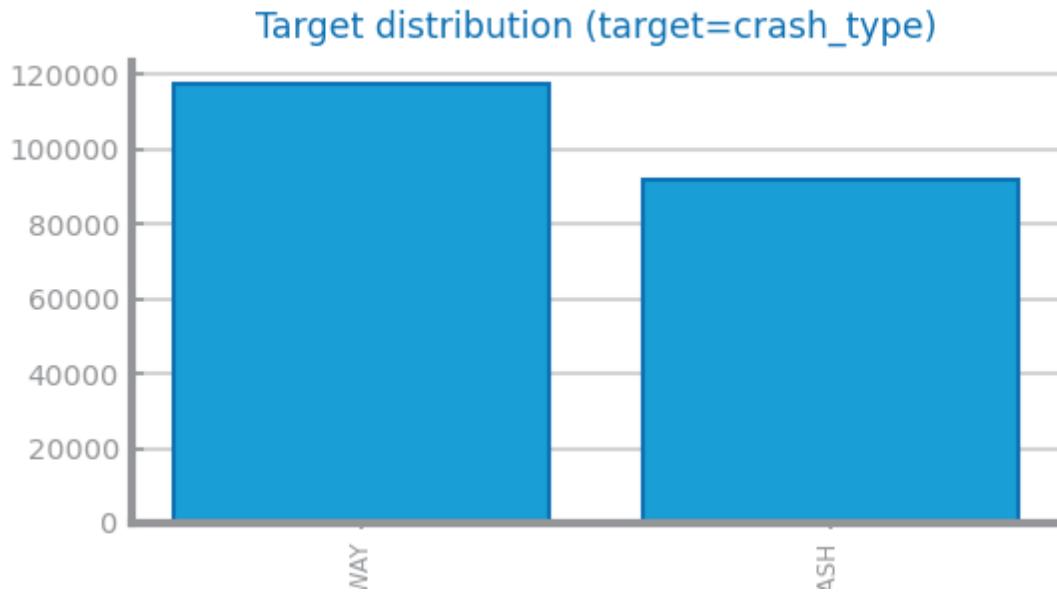


Figure 11: Class distribution for dataset 1

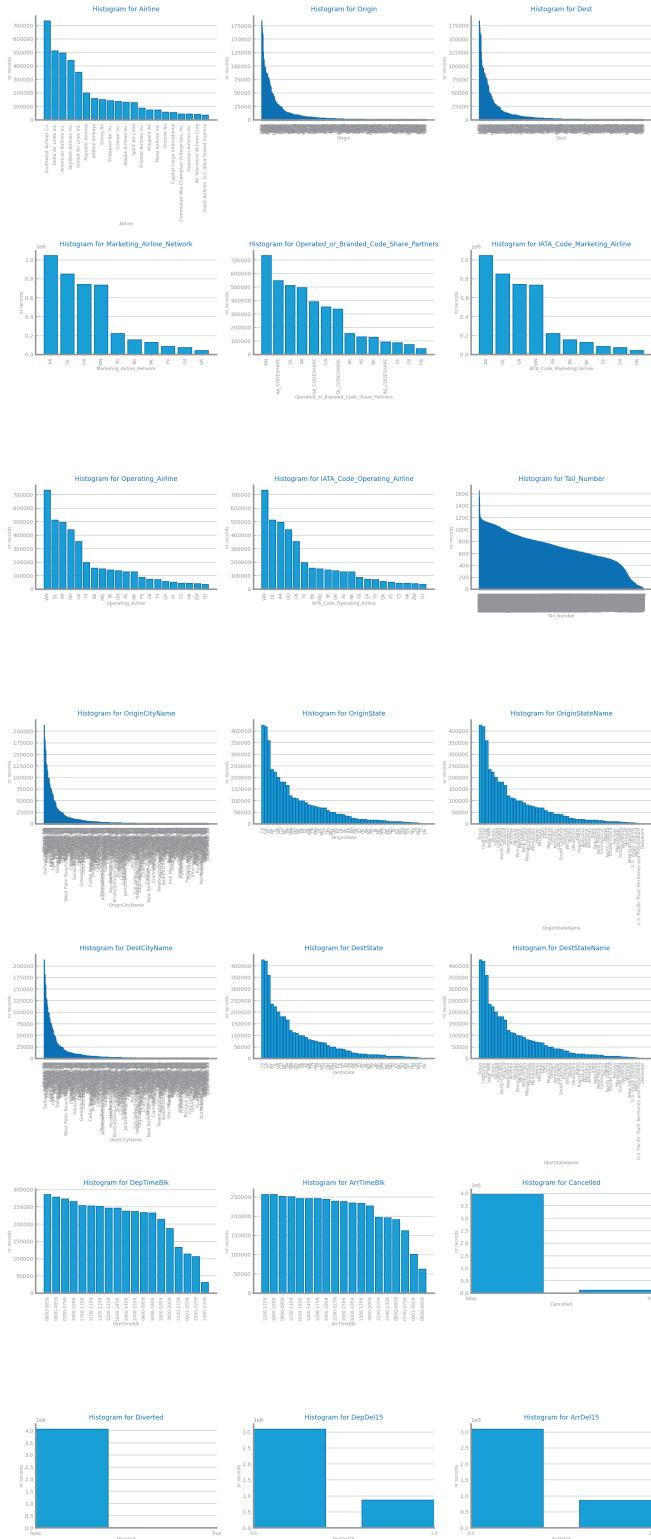


Figure 12: Class distribution for dataset 2

Data Granularity

The granularity analysis highlights significant variability across variables in both datasets. Categorical features such as weather, lighting conditions, airline, and airport present low to medium granularity, while temporal and numerical variables (hour, delays, distance) show higher granularity and dispersion. These differences suggest the need for appropriate grouping or discretization strategies to avoid sparsity and improve downstream analysis and modeling.

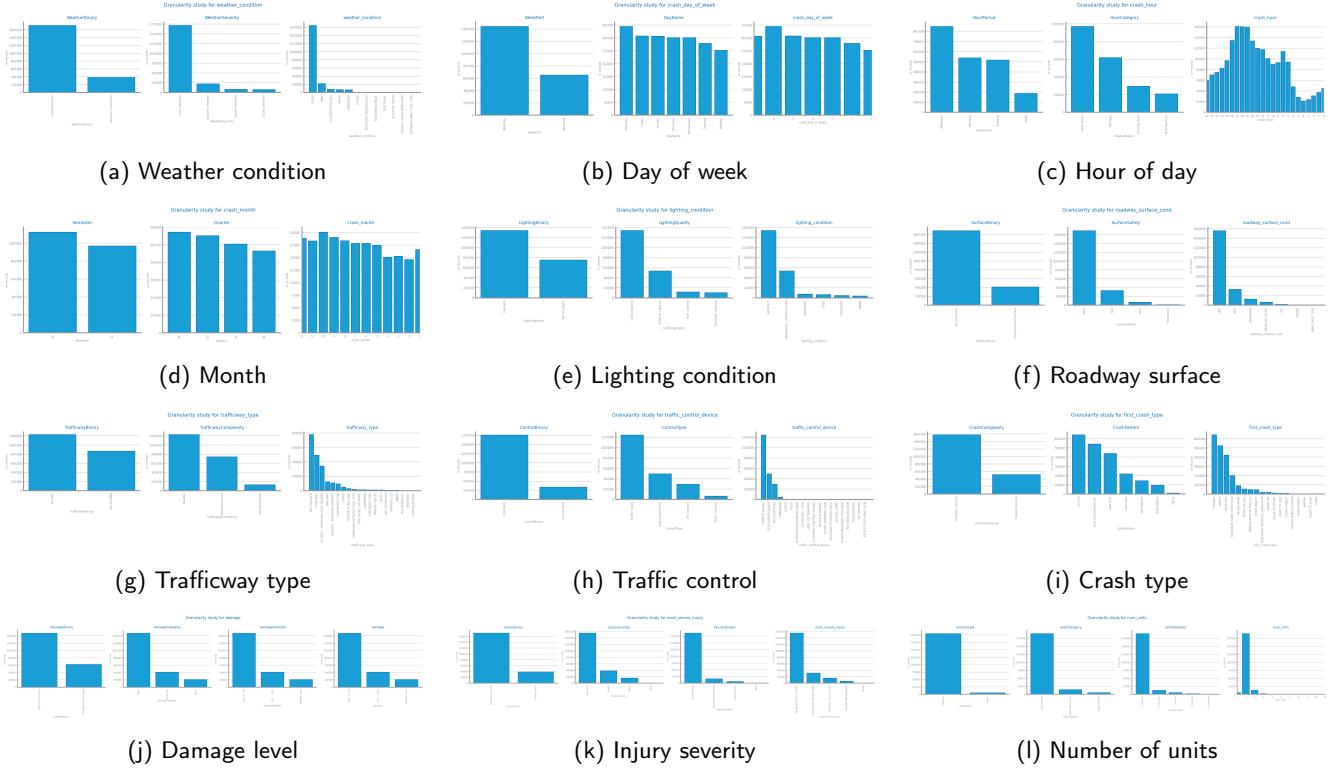


Figure 13: Granularity analysis for dataset 1

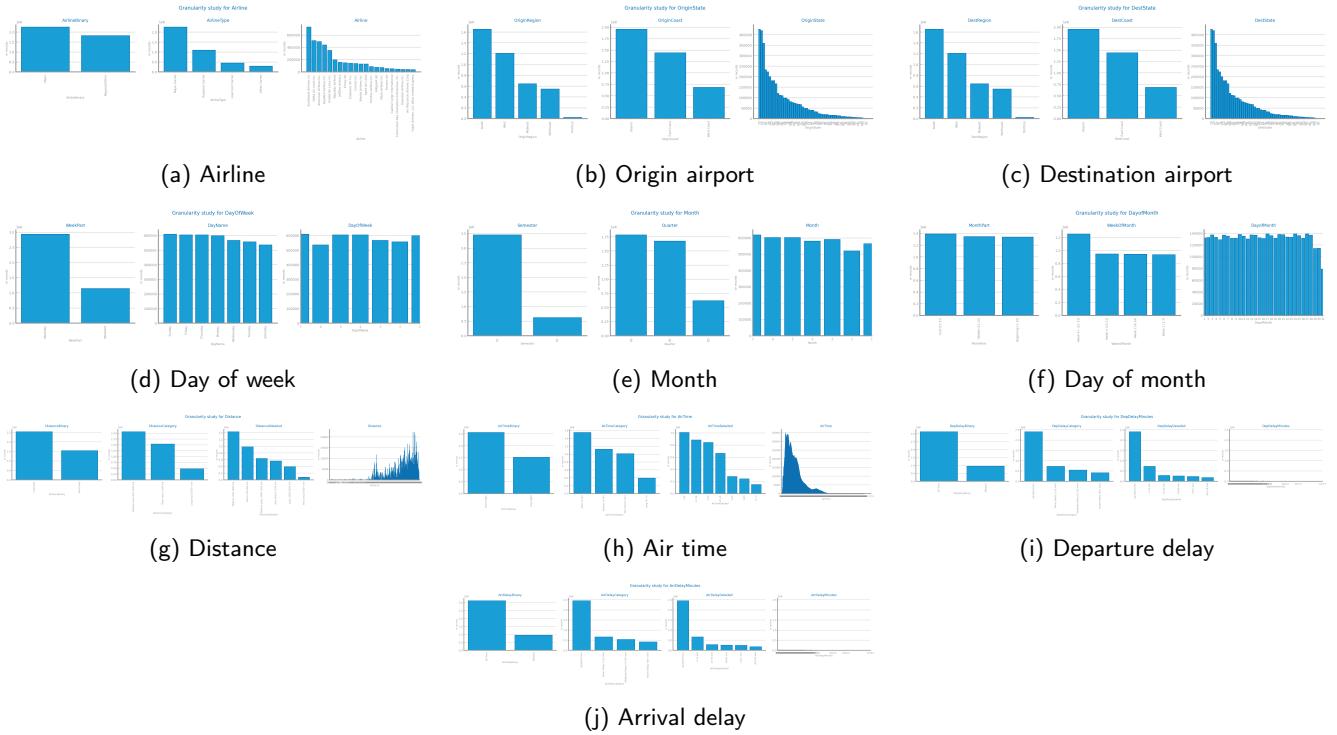


Figure 14: Granularity analysis for dataset 2

Data Sparsity

The sparsity analysis reveals uneven domain coverage across variables, with several categorical features dominated by a small number of frequent values and many rare ones. Correlation analysis shows limited strong dependencies between most variables, indicating sparse relationships in the feature space.



Figure 15: Sparsity analysis for dataset 1

Figure 16: Sparsity analysis for dataset 2 - [View on Google Drive](#)

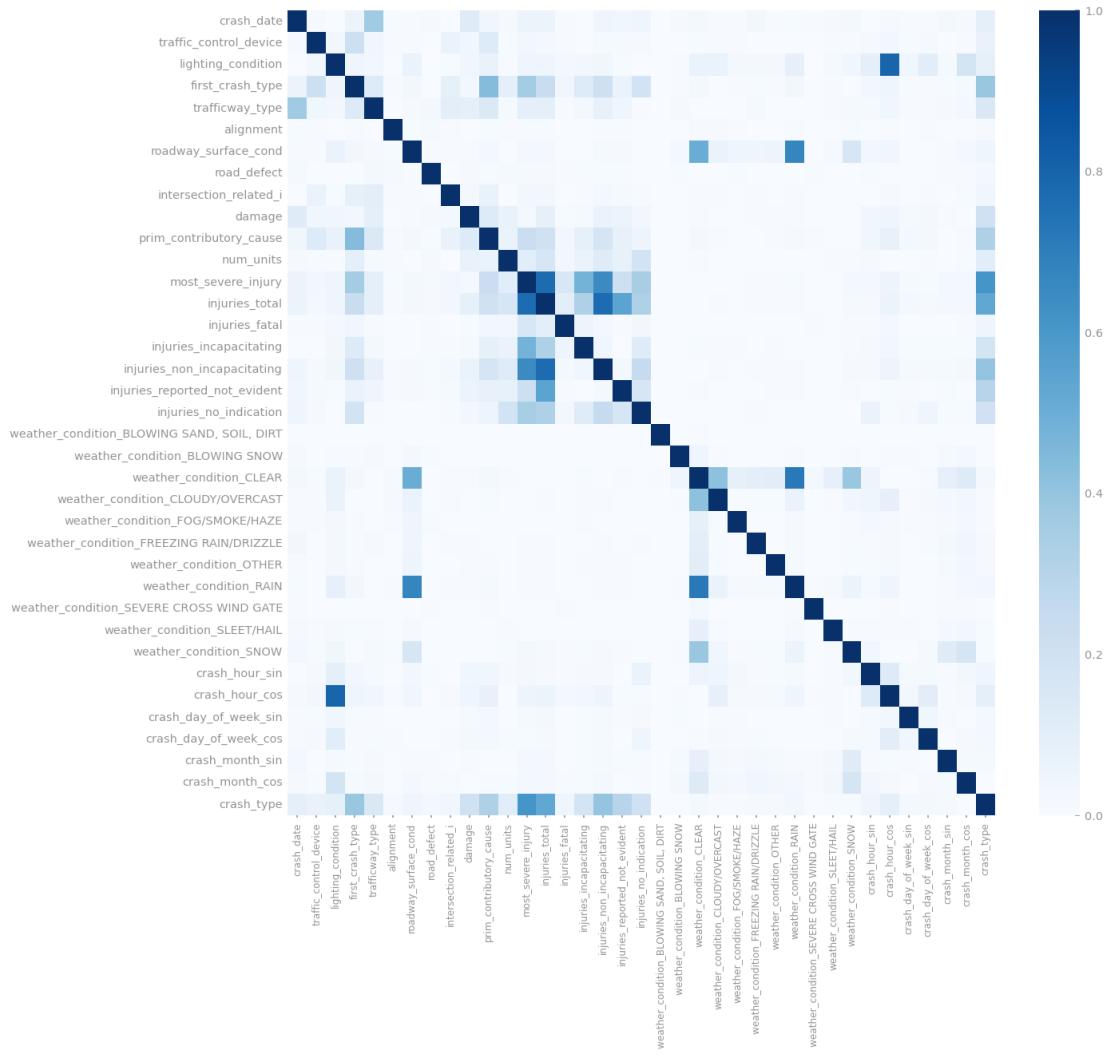


Figure 17: Correlation analysis for dataset 1

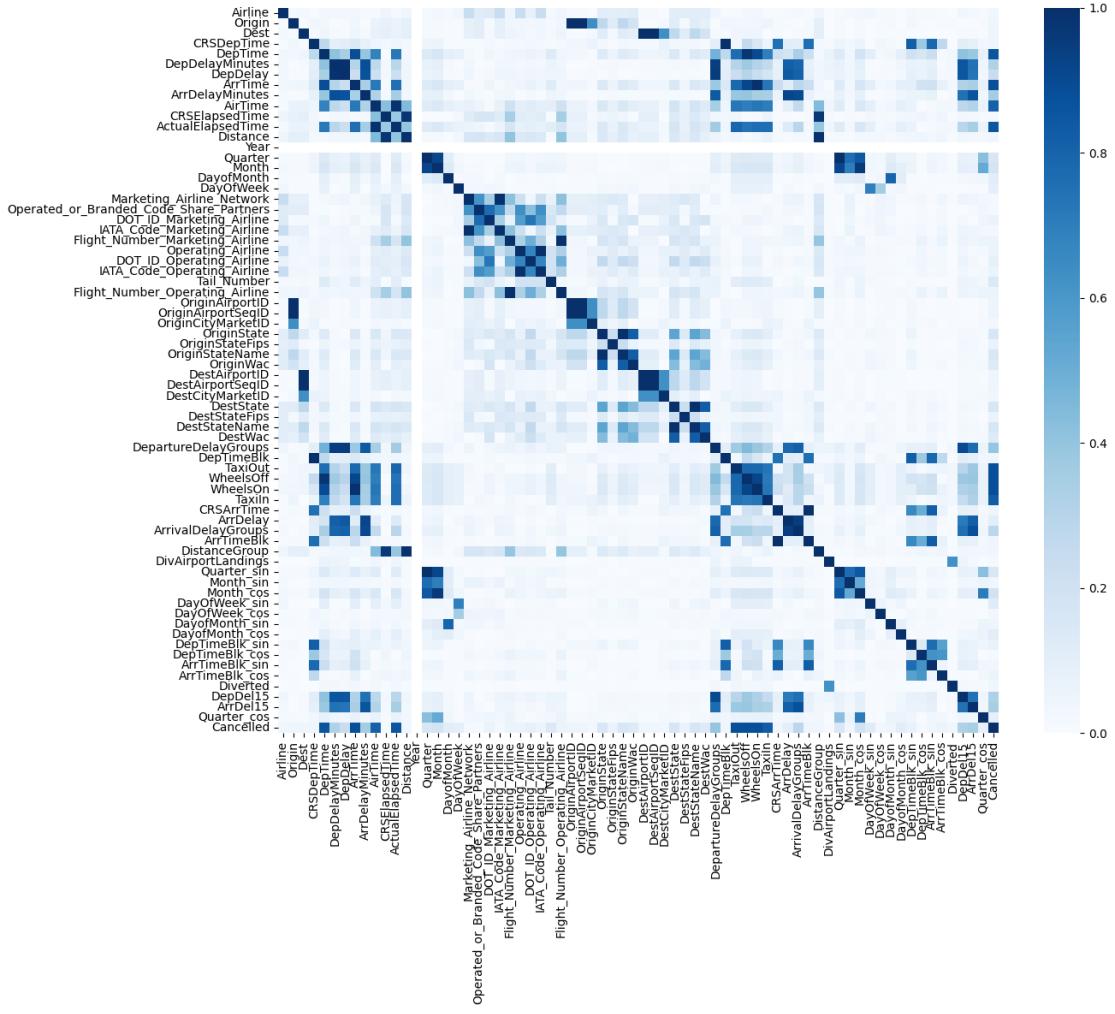


Figure 18: Correlation analysis for dataset 2

2 DATA PREPARATION

Variables Encoding

In Dataset 2, airline categories were ordinal-encoded according to market relevance. Time-block fields were converted to ordinal values (0–18). Cyclic variables (e.g., Quarter) were transformed using sin/cos encoding and the original columns removed. Numeric fields were kept unchanged, and redundant or leakage-prone columns had been removed earlier.

Missing Value Imputation

For the dataset 1, dropping records was compared with mode imputation. Imputation yielded higher performance , improving KNN accuracy from 0.51 (drop) to 0.54 (impute). Thus, mode imputation was selected to prevent data loss.

In Dataset 2, after removing leakage-prone columns, no missing values were found in the remaining variables. Therefore, imputation was not required and no method (filling or dropping) had any effect on the dataset.

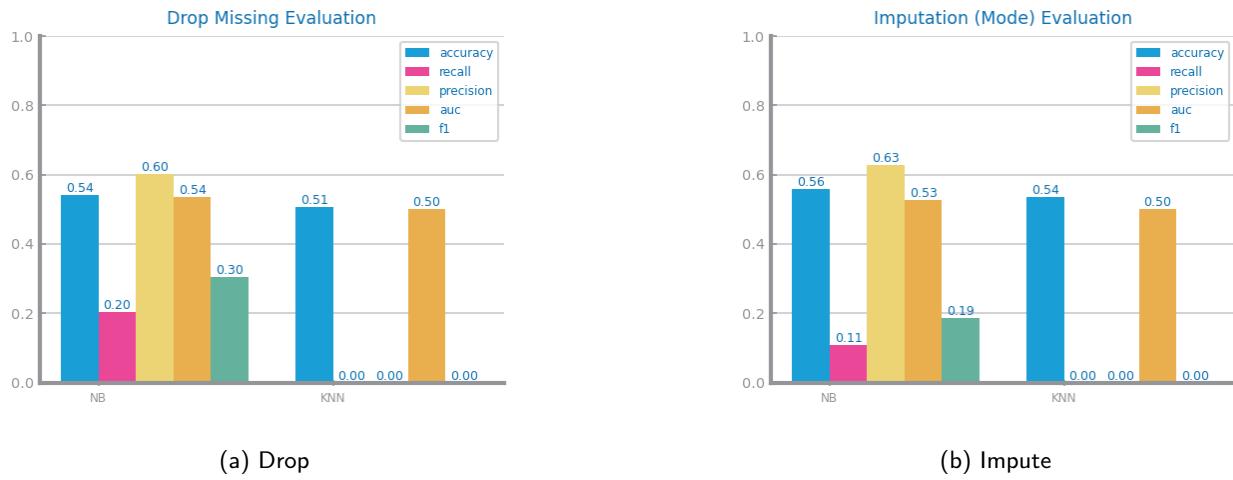


Figure 19: Missing values imputation results with different approaches for dataset 1

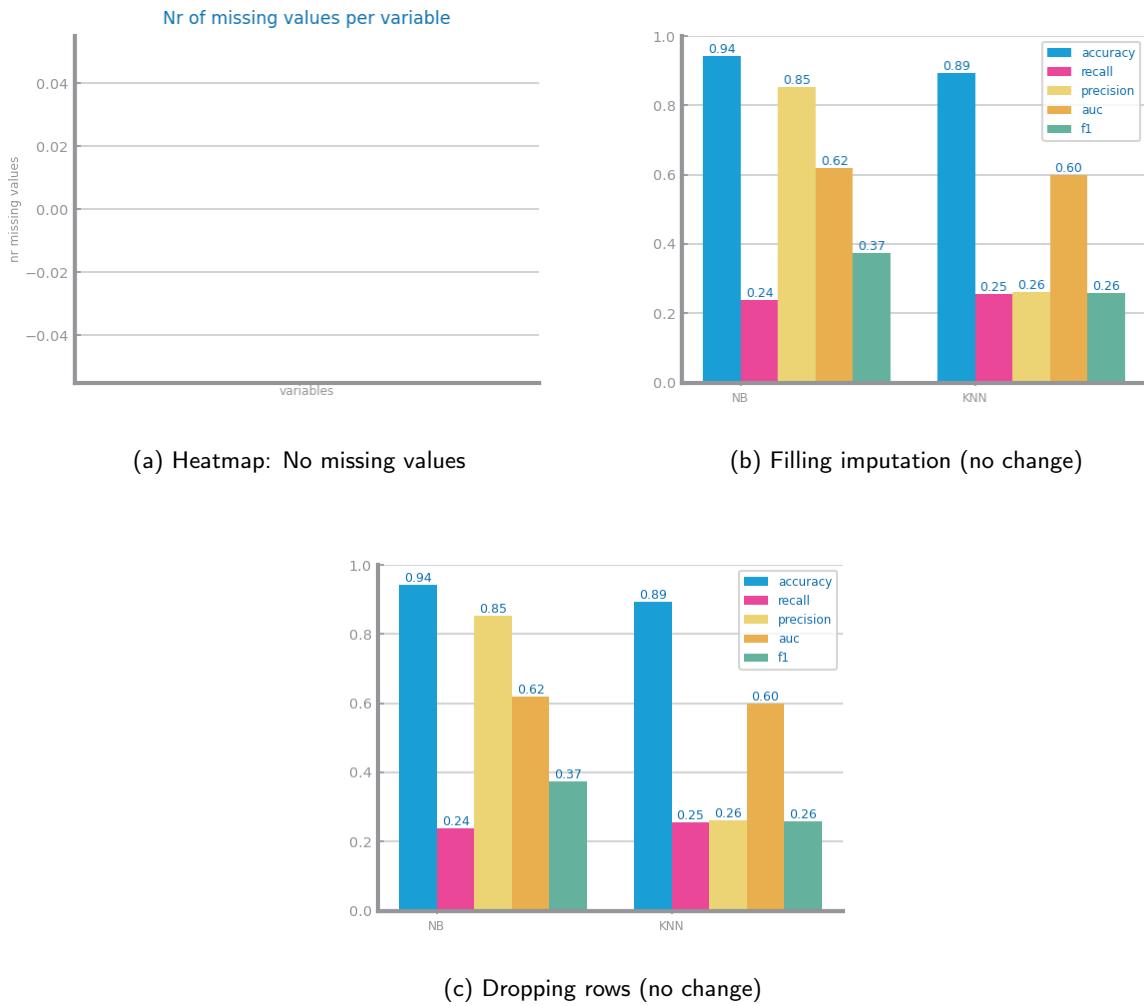


Figure 20: Missing values analysis and imputation attempts for dataset 2 (no missings detected)

Outliers Treatment

For Dataset 1, two approaches were tested: "Keep Outliers" and "Remove Outliers (Z-Score and Isolation Forest)". "Keep Outliers" yielded better accuracy (NB: 0.56, KNN: 0.54) compared to removal strategies (Z-Score: 0.55, Isolation Forest: 0.55). Thus, no outliers were removed.

For Dataset 2, outliers in CRSElapsedTime, Distance and airline codes were treated by dropping, median replacement, and truncation. Dropping was chosen as it delivered the best results (NB acc. 0.94, KNN 0.89)

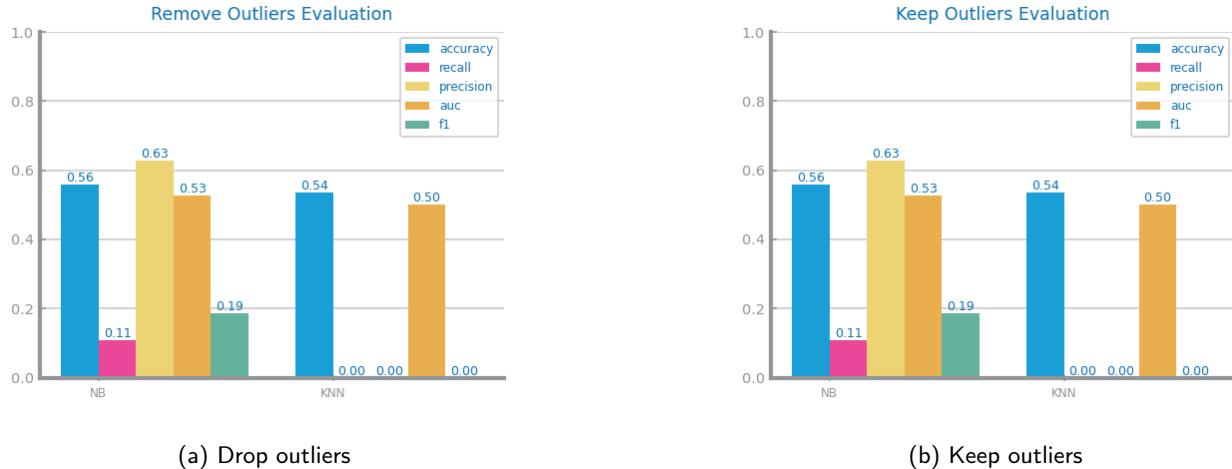
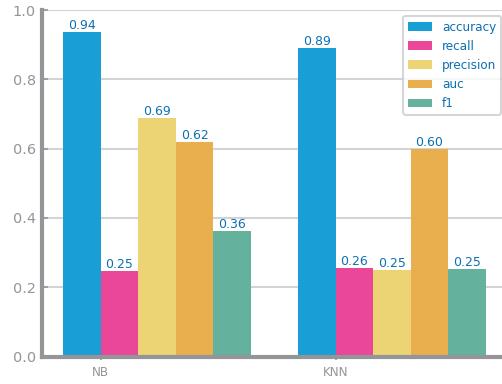
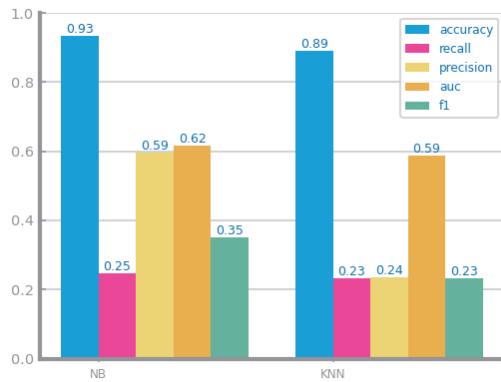


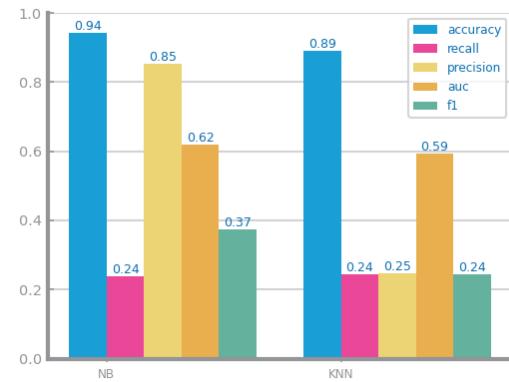
Figure 21: Outliers imputation results with different approaches for dataset 1



(a) Drop outliers (selected)



(b) Replace with median



(c) Truncate

Figure 22: Impact of different outlier treatments on NB and KNN performance for dataset 2

Scaling

For Dataset 1 StandardScaler outperformed MinMaxScaler as it improved KNN auc to 0.68 compared to 0.67. Flights: MinMaxScaler outperformed ABSScaler with the best performance - NB acc. 0.94, KNN 0.89.

The Dataset 2 numeric variables were scaled using ABSScaler, MinMaxScaler. MinMaxScaler was selected as it yielded the best model performance (NB acc. 0.94, KNN 0.89).

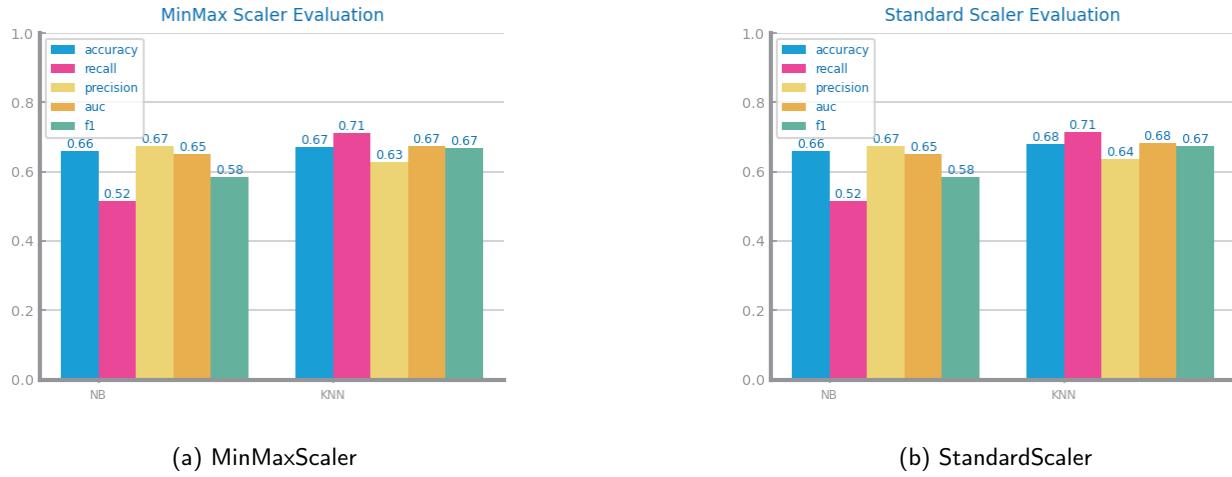


Figure 23: Scaling results with different approaches for dataset 1

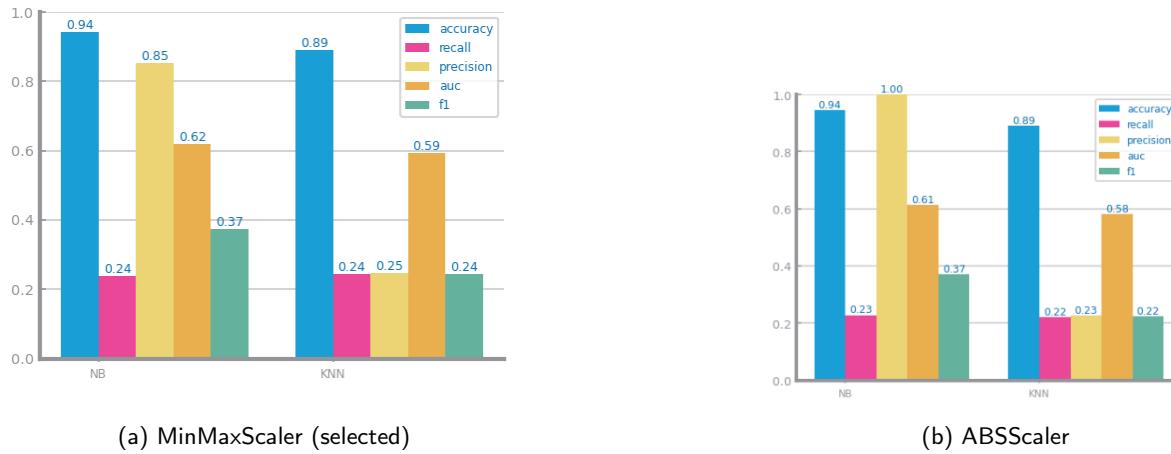


Figure 24: Scaling results with different approaches (NB and KNN performance) for dataset 2

Balancing

For Dataset 1, Random Undersampling and SMOTE were evaluated. Undersampling was selected (KNN acc 0.67) as it achieved a slightly higher recall (0.78) compared to SMOTE (recall 0.77).

The Dataset 2 was balanced using Random Undersampling, Random Oversampling, and SMOTE. Random Undersampling was chosen as it provided the best model performance (NB acc. 0.62, KNN 0.73).

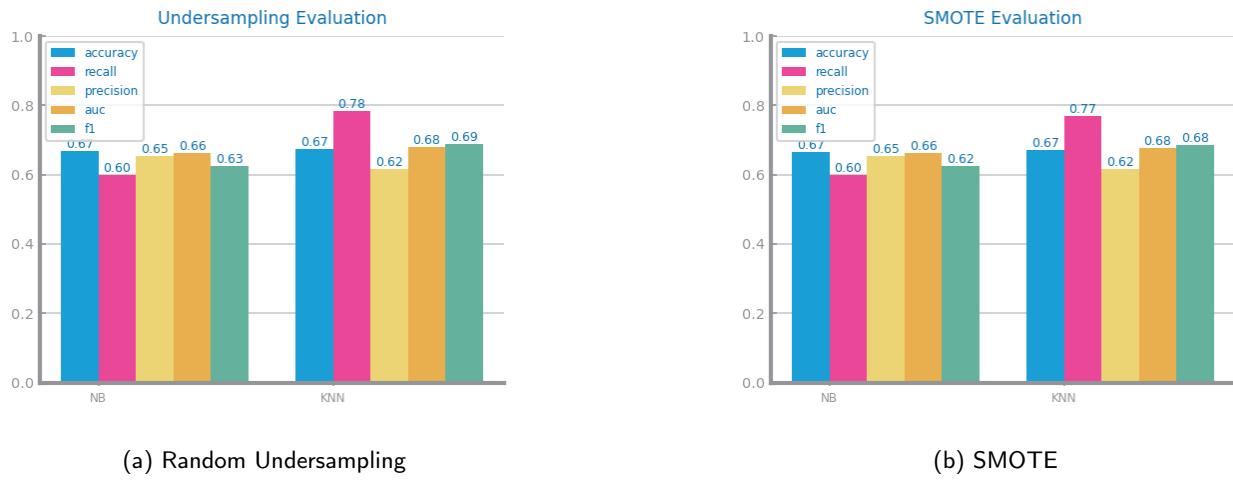


Figure 25: Balancing results with different approaches for dataset 1

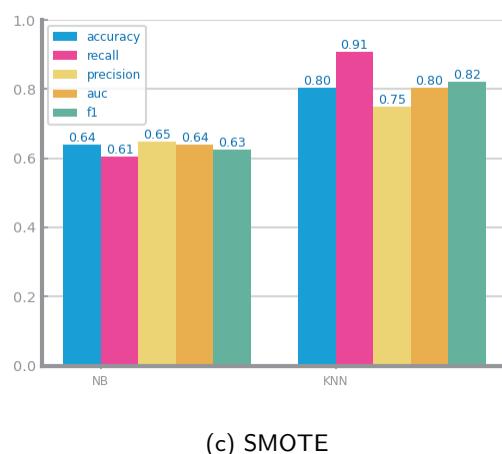
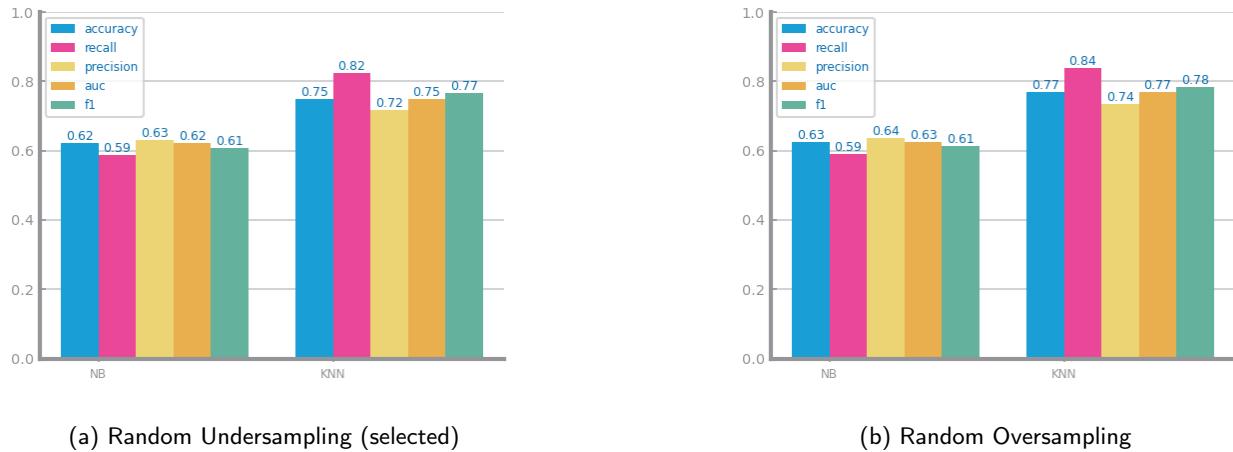


Figure 26: Balancing results with different approaches (NB and KNN performance) for dataset 2

Feature Selection

For Dataset 1, the Variance filter (threshold 0.8) retained all features due to standardization. The Redundancy filter (correlation > 0.75) was selected as it effectively reduced dimensionality, achieving the best trade-off with KNN Recall 0.78 and Accuracy 0.67.

For Dataset 2, feature selection was performed using redundant and relevant variable analysis. Redundant variables were identified via correlation analysis and removed. Relevant variables were selected based on variance thresholds. The final feature set improved model performance (NB acc. 0.62, KNN 0.68).

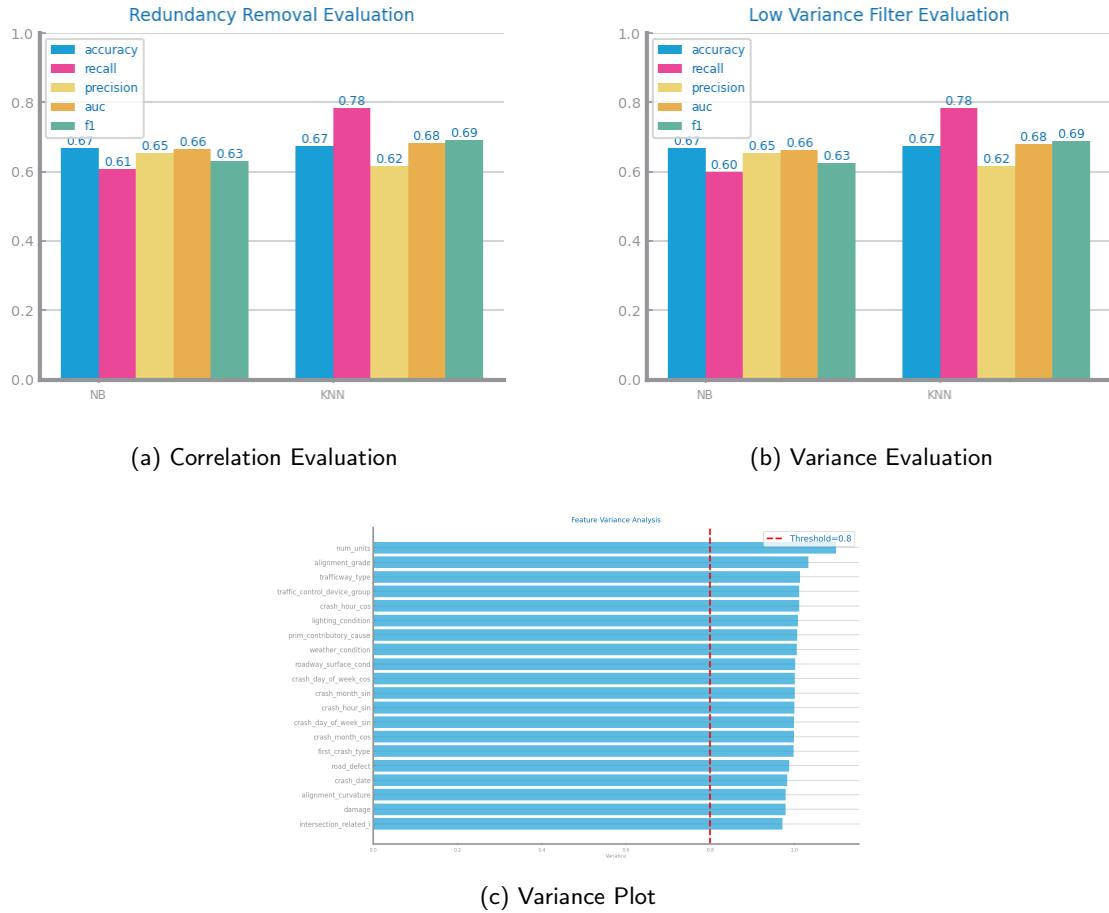
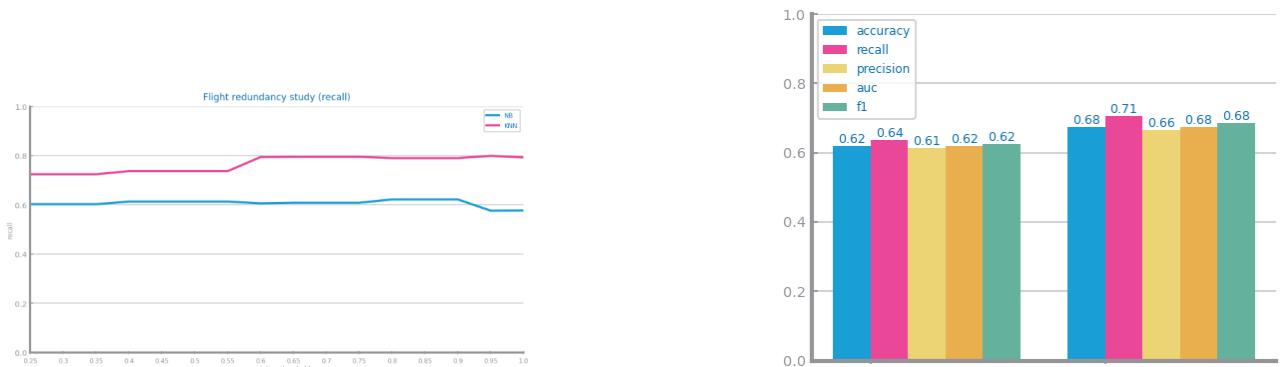


Figure 27: Feature selection of redundant variables results with different parameters for dataset 1

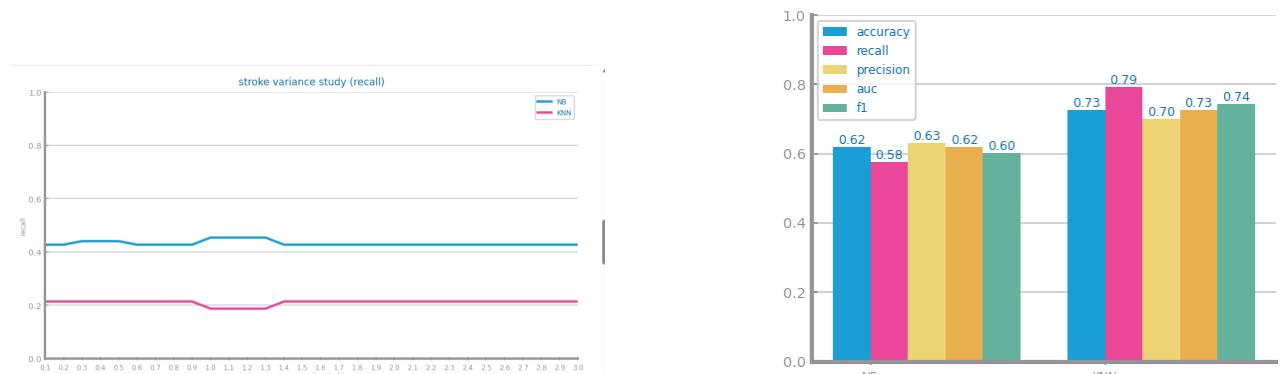


(a) Redundant variables removal (correlation matrix/threshold)

(b) Relevant variables selection (variance threshold study)

Figure 28: Feature selection results for redundant and low-variance variables in dataset 2

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



(a) Redundant variables removal (correlation matrix/threshold)

(b) Relevant variables selection (variance threshold study)

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

3 MODELS' EVALUATION

Shall be used to point out any important decision taken during the training, including training strategy and evaluation measures used. **Shall not exceed 500 characters.**

Naïve Bayes

Shall be used to present the results achieved with each one of Naïve Bayes implementations, comparing and proposing explanations for them. If any of the implementations is not used, a justification for it shall be presented. Shall be used to present the evaluation of the best model achieved. **Shall not exceed 300 characters.**

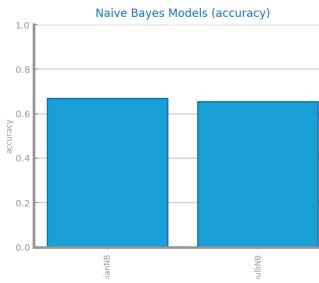


Figure 31: Naïve Bayes alternatives comparison for dataset 1

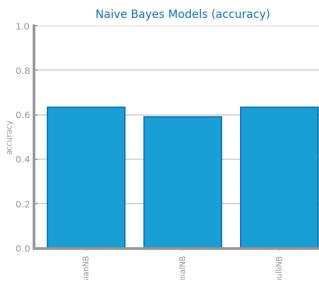


Figure 32: Naïve Bayes alternative comparison for dataset 2

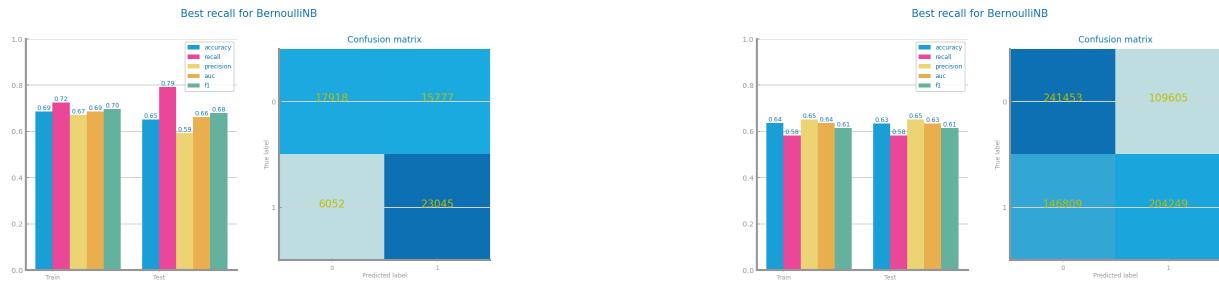


Figure 33: Naïve Bayes best model results for dataset 1 (left) and dataset 2 (right)

KNN

Shall be used to present the results achieved through different similarity measures and KNN parameterisations. The results shall be compared and explanations for them shall be presented. The justification for the chosen similarity measures shall be presented. Shall be used to address the overfitting phenomenon, studying the conditions under which models face it. Shall be used to present the evaluation of the best model achieved. **Shall not exceed 500 characters.**

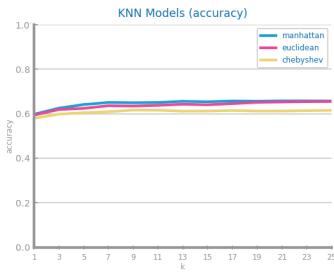


Figure 34: KNN different parameterisations comparison for dataset 1

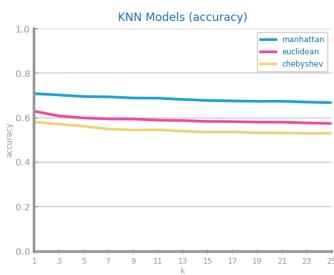


Figure 35: KNN different parameterisations comparison for dataset 2

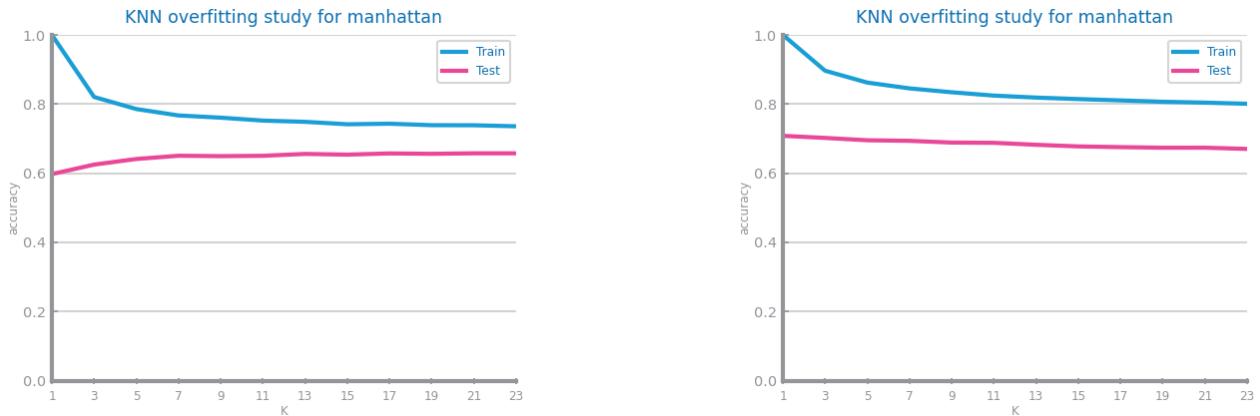


Figure 36: KNN overfitting analysis for dataset 1 (left) and dataset 2 (right)

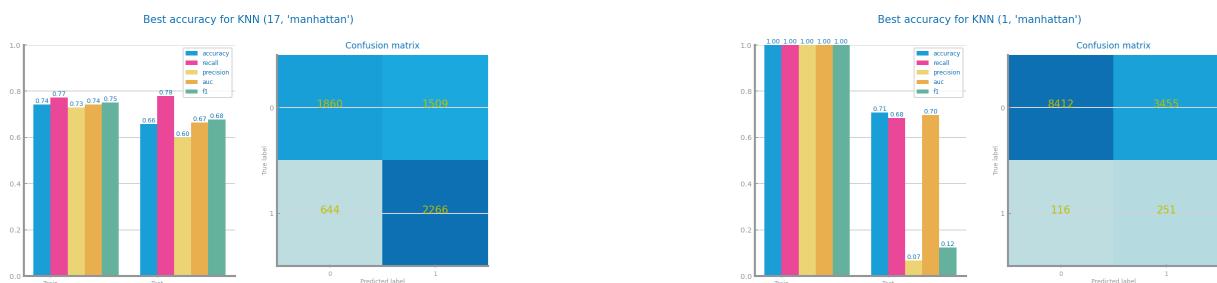


Figure 37: KNN best model results for dataset 1 (left) and dataset 2 (right)

Logistic Regression

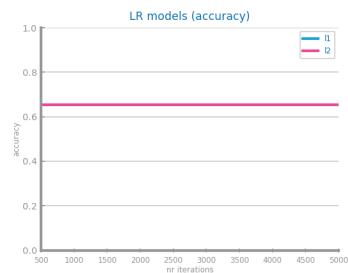


Figure 38: Logistic Regression different parameterisations comparison for dataset 1

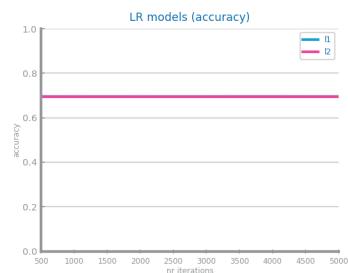


Figure 39: Logistic Regression different parameterisations comparison for dataset 2

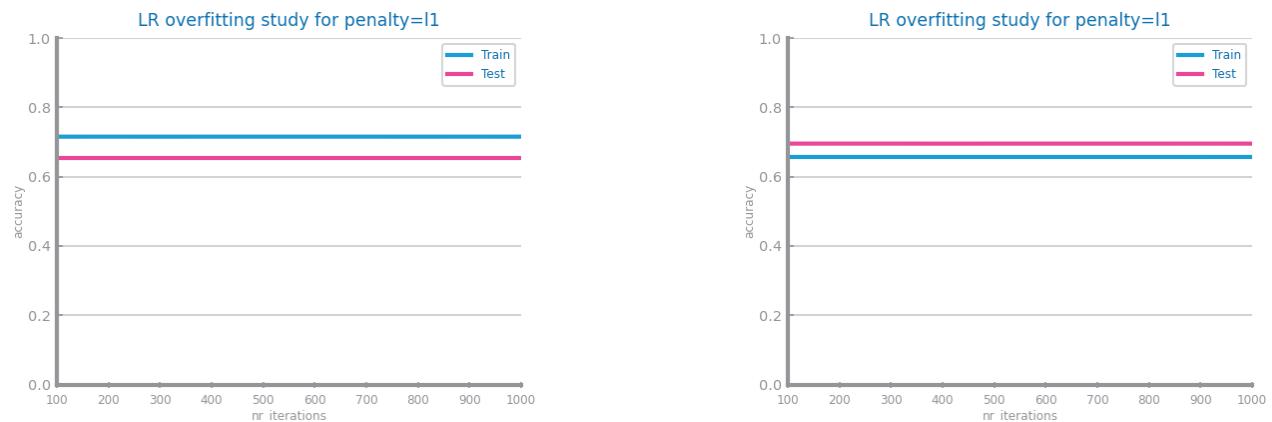


Figure 40: Logistic Regression overfitting analysis for dataset 1 (left) and dataset 2 (right)



Figure 41: Logistic Regression best model results for dataset 1 (left) and dataset 2 (right)

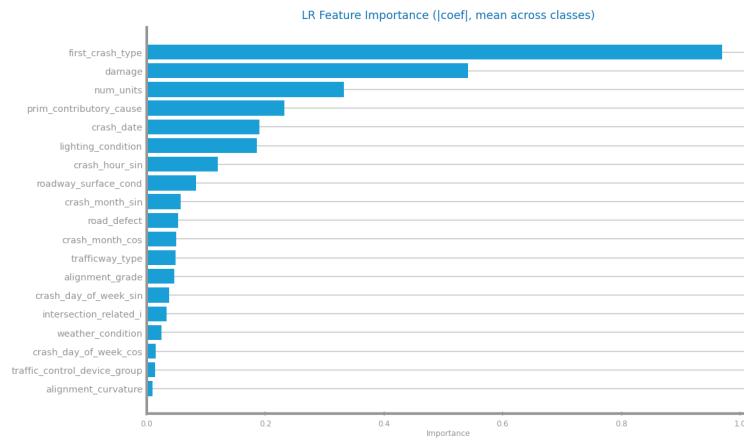


Figure 42: Logistic Regression feature importance for dataset 1

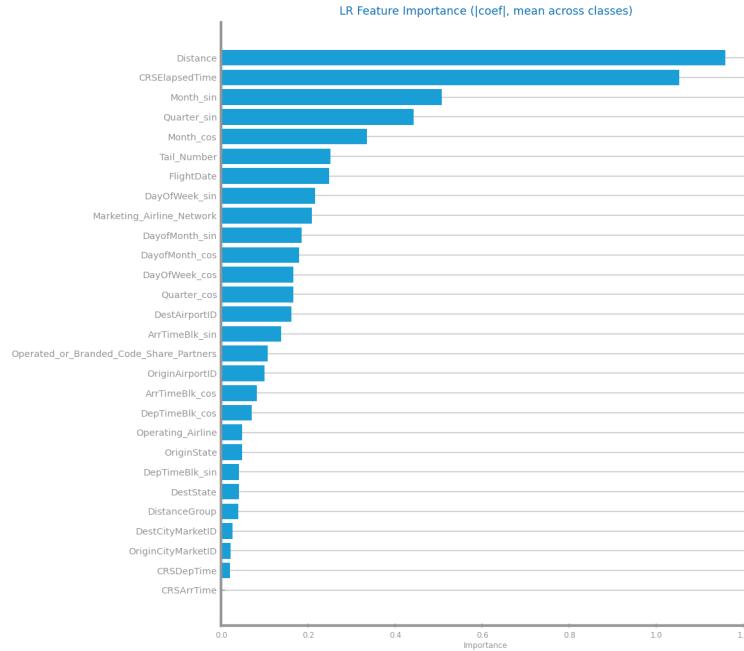


Figure 43: Logistic Regression feature importance for dataset 2

Decision Trees

Decision trees behave differently across the two datasets. In Dataset 1, accuracy peaks early (depth 6-8, 0.75) then slightly declines, showing mild overfitting. The best model (entropy, depth 6) reaches 0.74 accuracy with balanced precision and recall; key features include *crash_type*, *damage*, *dry_cause*, *num_units*, and *ash_date*. In Dataset 2, entropy slightly outperforms gini, although deep trees overfit. The best model (gini, depth 16) achieves 0.81 test accuracy.

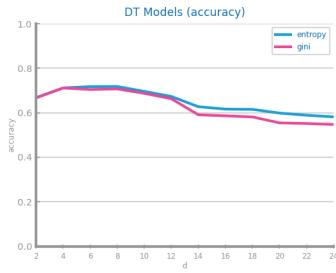


Figure 44: Decision Trees different parameterisations comparison for dataset 1

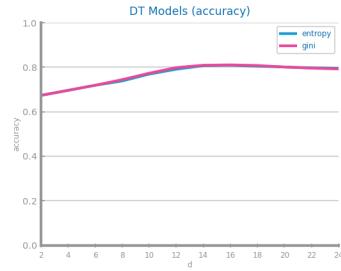


Figure 45: Decision Trees different parameterisations comparison for dataset 2

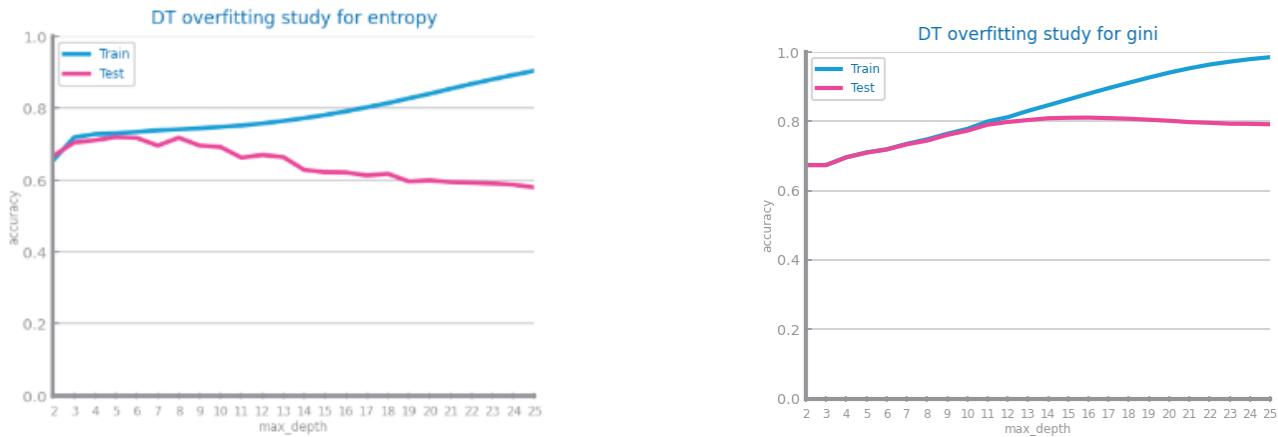


Figure 46: Decision Trees overfitting analysis for dataset 1 (left) and dataset 2 (right)

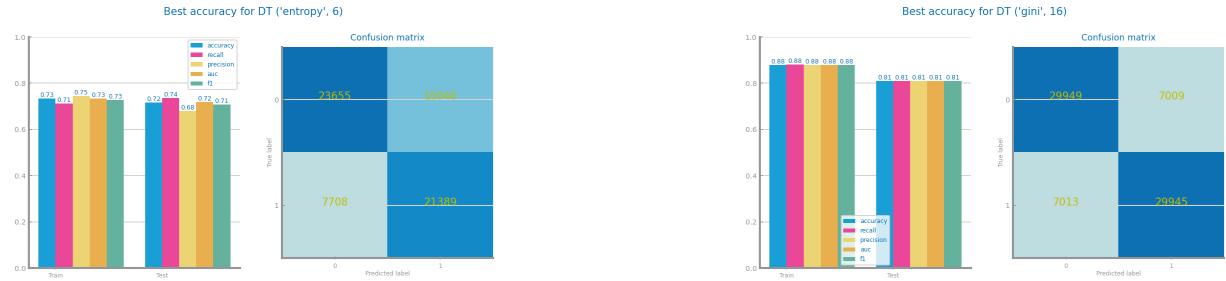


Figure 47: Decision trees best model results for dataset 1 (left) and dataset 2 (right)

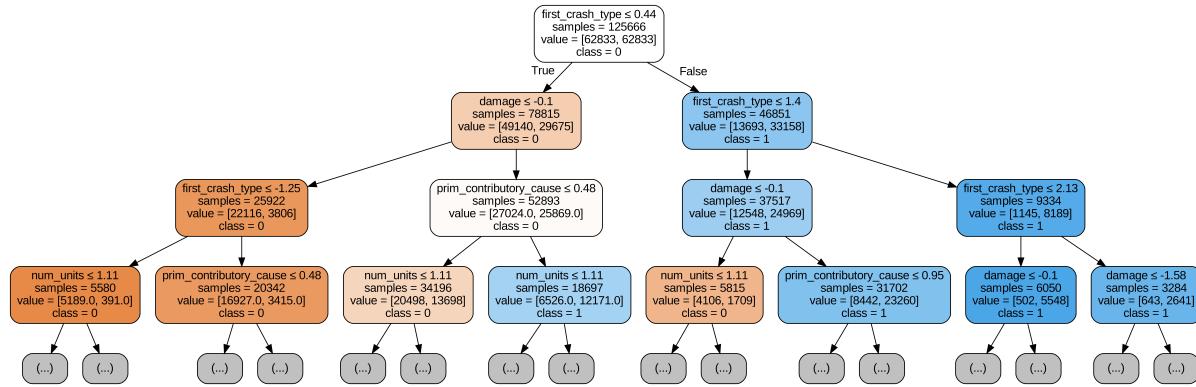


Figure 48: Best tree for dataset 1

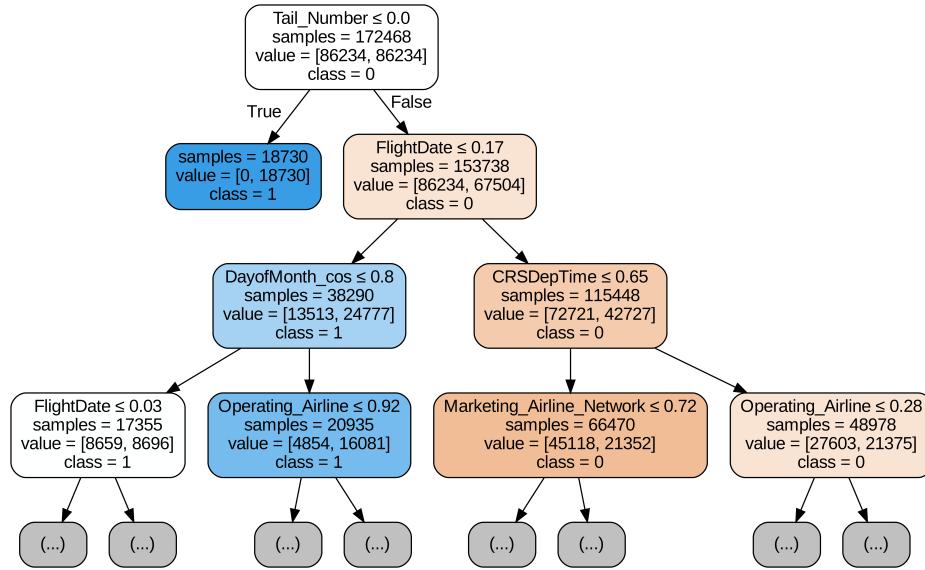


Figure 49: Best tree for dataset 2

Random Forests

Random forests improve steadily in both datasets. In Dataset 1, accuracy increases quickly and stabilizes. The best model (depth 5, feature fraction 0.5, 750 trees) reaches 0.72 test accuracy with balanced metrics. In Dataset 2, performance

remains stable across configurations. The best configuration (max depth 7, feature fraction 0.7, 500 trees) achieves 0.75 test accuracy, with strong precision (0.86) but lower recall (0.60).

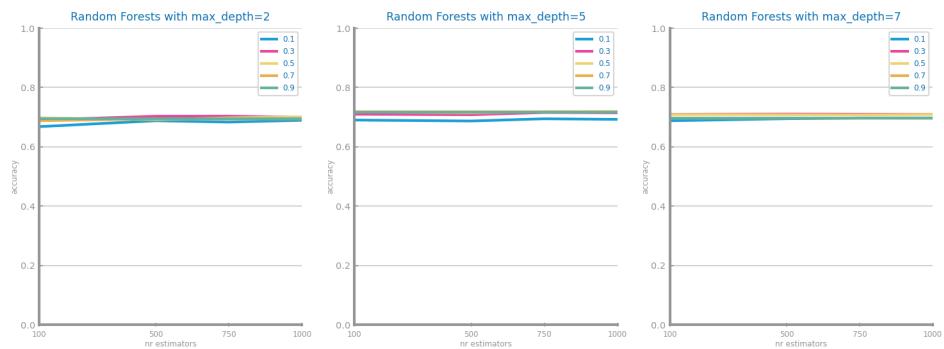


Figure 50: Random Forests different parameterisations comparison for dataset 1

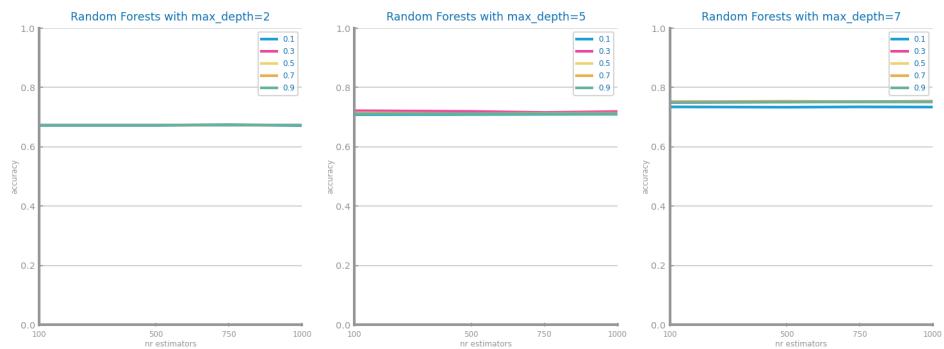


Figure 51: Random Forests different parameterisations comparison for dataset 2

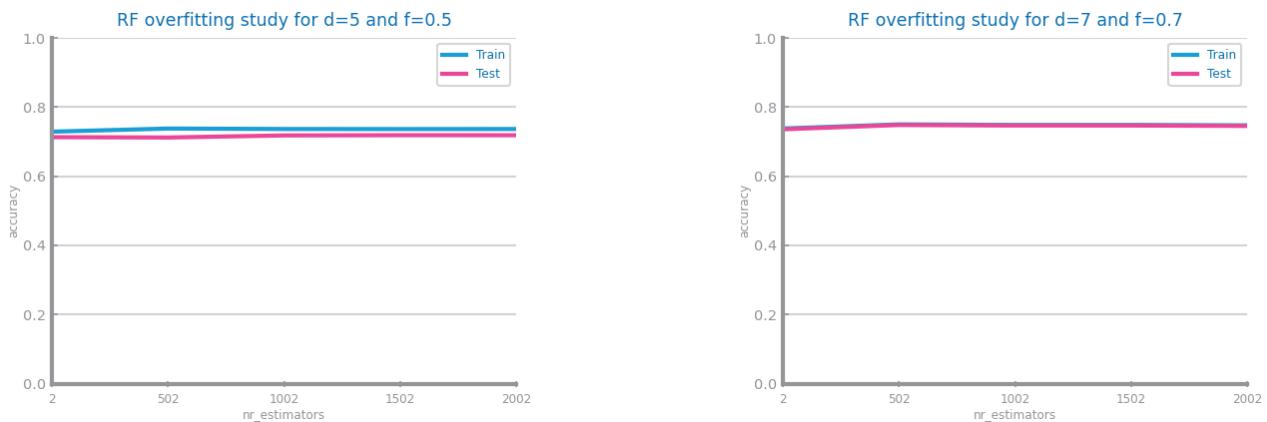


Figure 52: Random Forests overfitting analysis for dataset 1 (left) and dataset 2 (right)

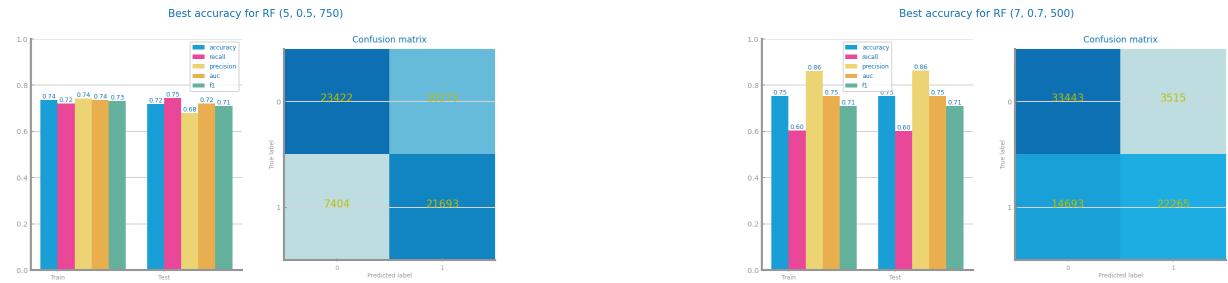


Figure 53: Random Forests best model results for dataset 1 (left) and dataset 2 (right)

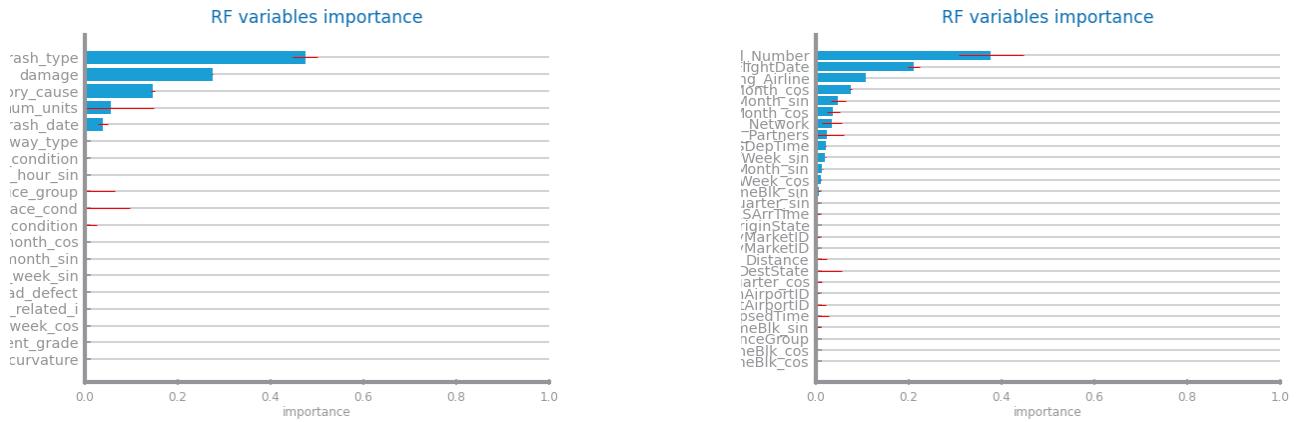


Figure 54: Random Forests variables importance for dataset 1 (left) and dataset 2 (right)

Gradient Boosting

Shall be used to present the results achieved through different parameterisations for the train of gradient boosting. The results shall be compared and explanations for them shall be presented. Shall be used to address the overfitting phenomenon, studying the conditions under which models face it. Shall be used to present the evaluation of the best model achieved. May be used to present the most important variables in the model. **Shall not exceed 500 characters.**

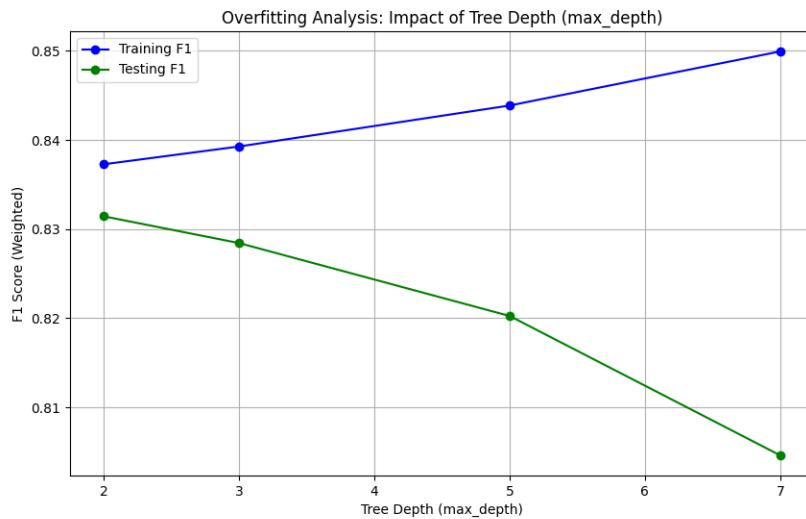


Figure 55: Gradient boosting different parameterisations comparison for dataset 1

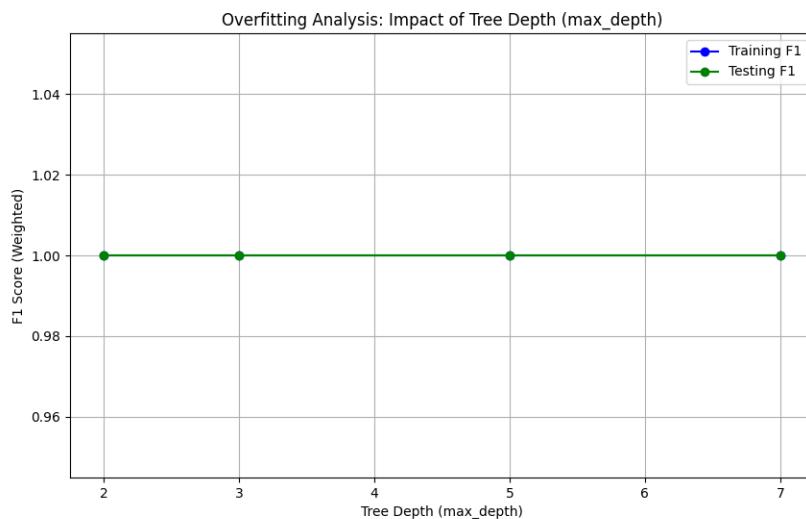


Figure 56: Gradient boosting different parameterisations comparison for dataset 2

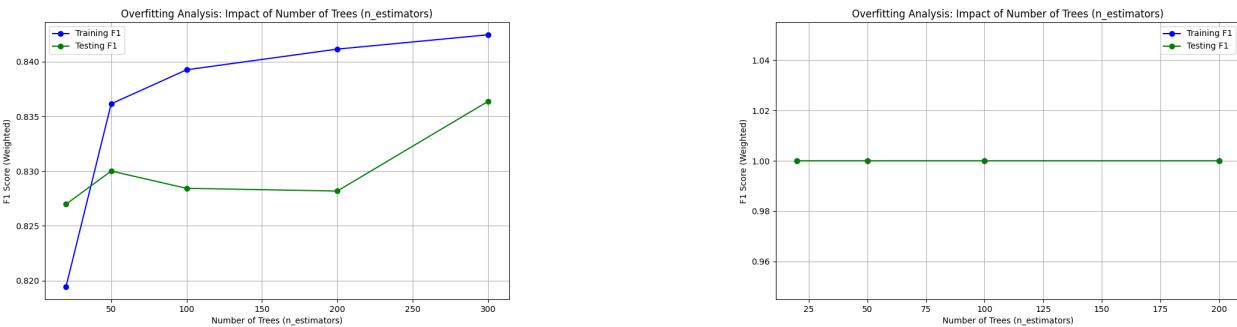


Figure 57: Gradient boosting overfitting analysis for dataset 1 (left) and dataset 2 (right)

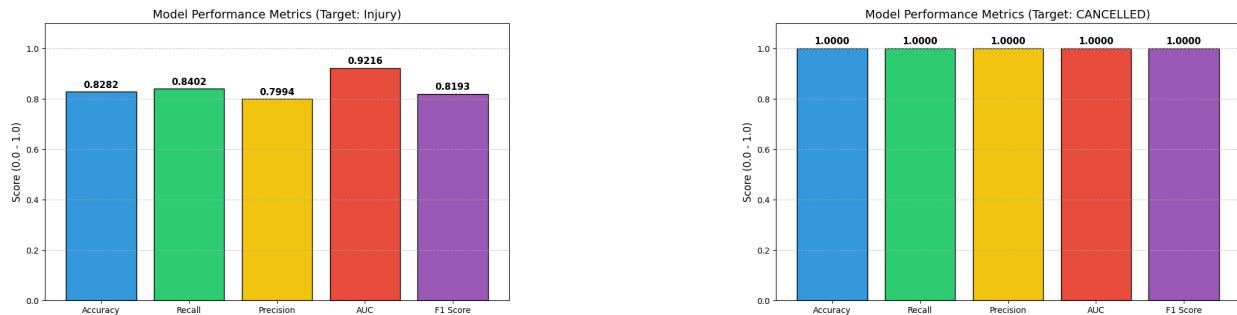


Figure 58: Gradient boosting best model results for dataset 1 (left) and dataset 2 (right)

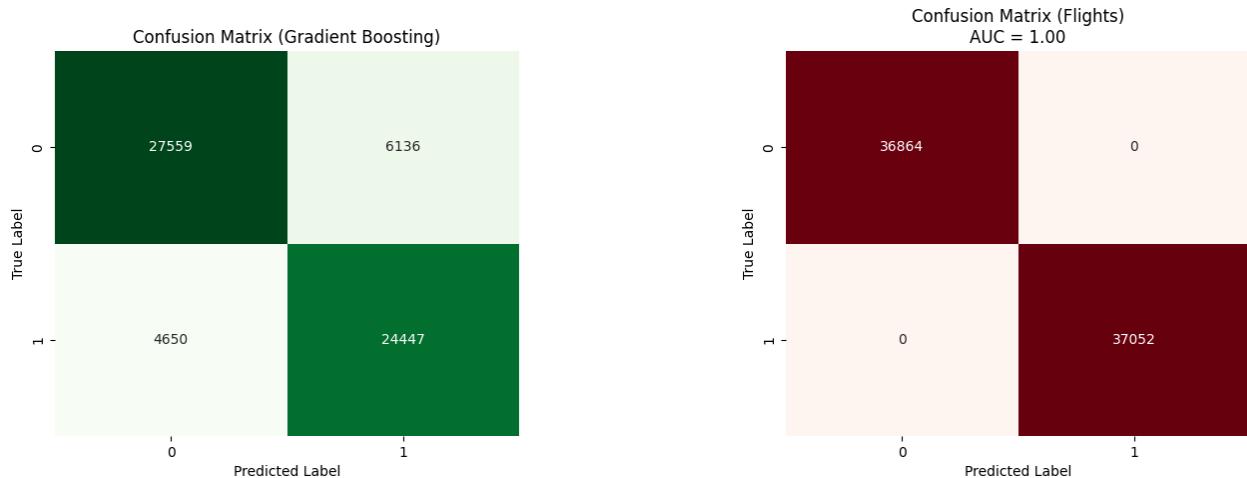


Figure 59: Gradient boosting variables importance for dataset 1 (left) and dataset 2 (right)

Multi-Layer Perceptrons

Shall be used to present the results achieved through different parameterisations for the train of MLPs. The results shall be compared and explanations for them shall be presented. Shall be used to address the overfitting phenomenon, studying

the conditions under which models face it. Shall be used to present the evaluation of the best model achieved. **Shall not exceed 500 characters.**

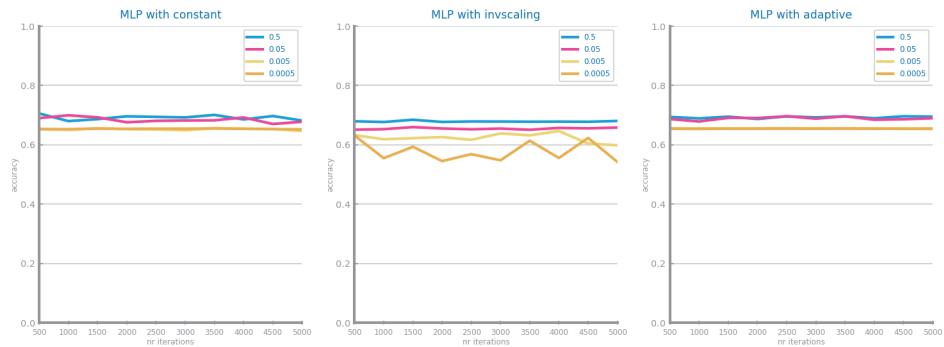


Figure 60: MLP different parameterisations comparison for dataset 1

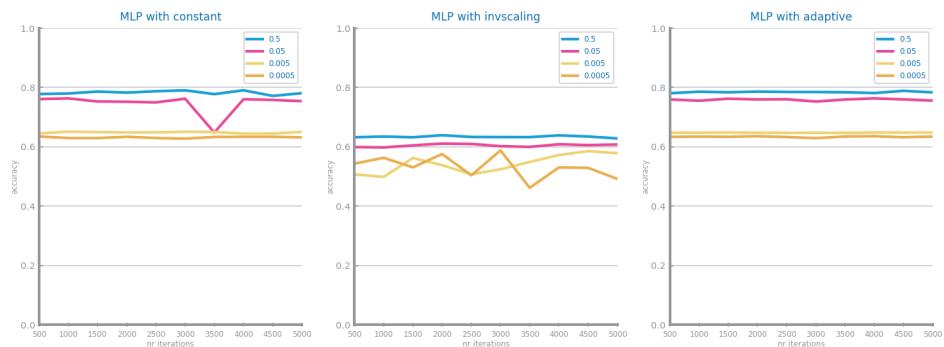


Figure 61: MLP different parameterisations comparison for dataset 2

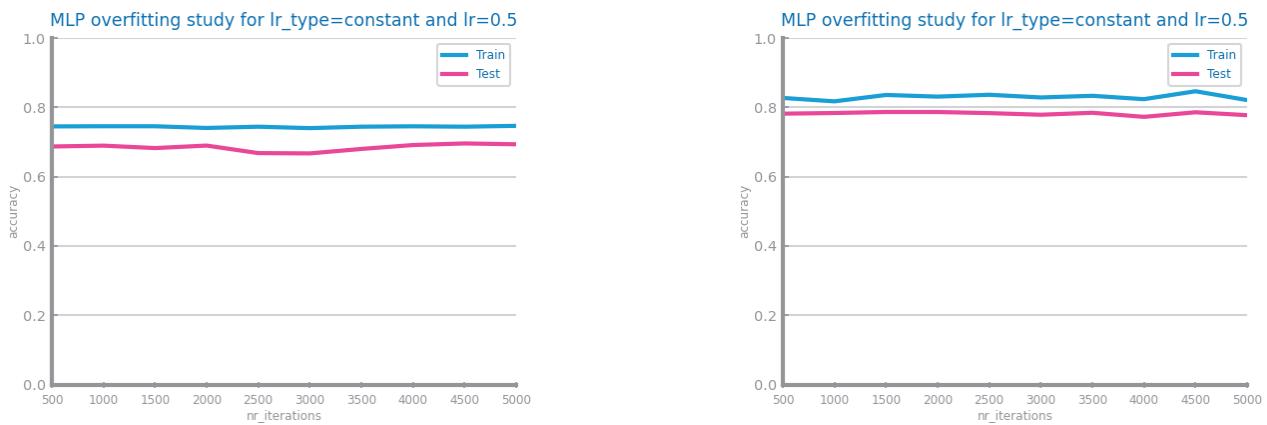


Figure 62: MLP overfitting analysis for dataset 1 (left) and dataset 2 (right)

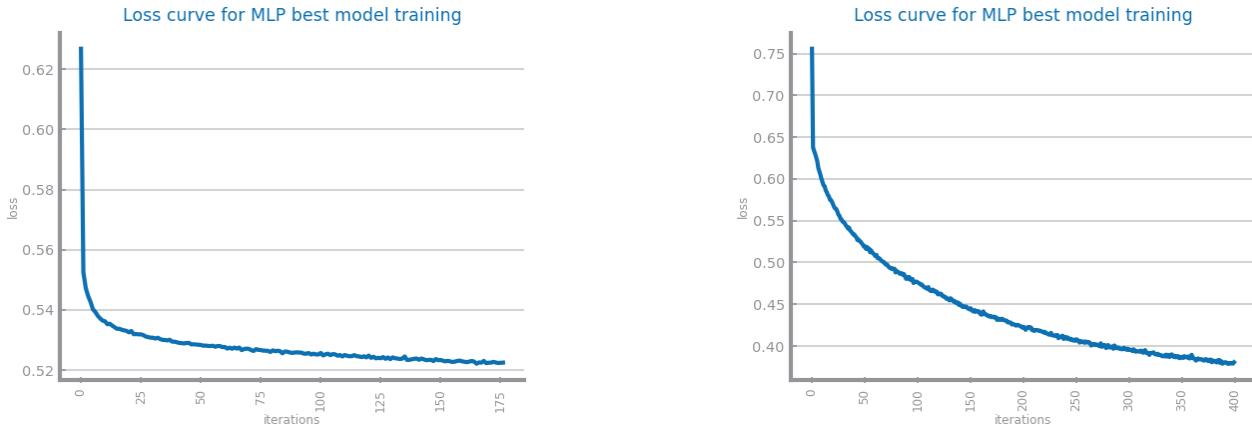


Figure 63: Loss curve analysis for dataset 1 (left) and dataset 2 (right)

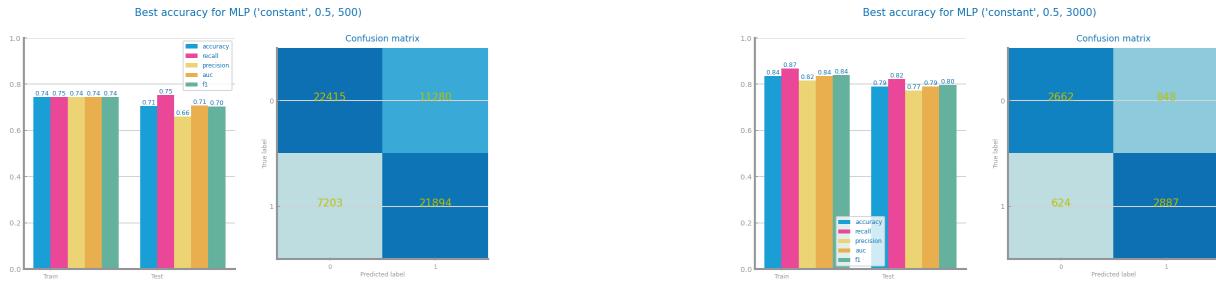


Figure 64: MLP best model results for dataset 1 (left) and dataset 2 (right)

4 CRITICAL ANALYSIS

Shall be used to present a summary of the results achieved with the different modelling techniques, and the impact of the different preparation tasks on their performance. A cross-analysis of the different models may also be presented, identifying the most relevant variables common to all of them (when possible) and the relation among the patterns identified within the different classifiers. A critical assessment of the best models shall be presented, clearly stating if the models seem to be good enough for the problem at hand. Additional charts may be presented here. Shall not exceed 2000 characters.

TIME SERIES ANALYSIS

5 DATA PROFILING

Data Dimensionality and Granularity

The time series was analyzed at three temporal granularities: 15-minute intervals (most atomic), daily aggregation, and weekly aggregation. The most granular level captures short-term variability and peaks, while daily and weekly granularities

smooth fluctuations and highlight medium- and long-term trends.

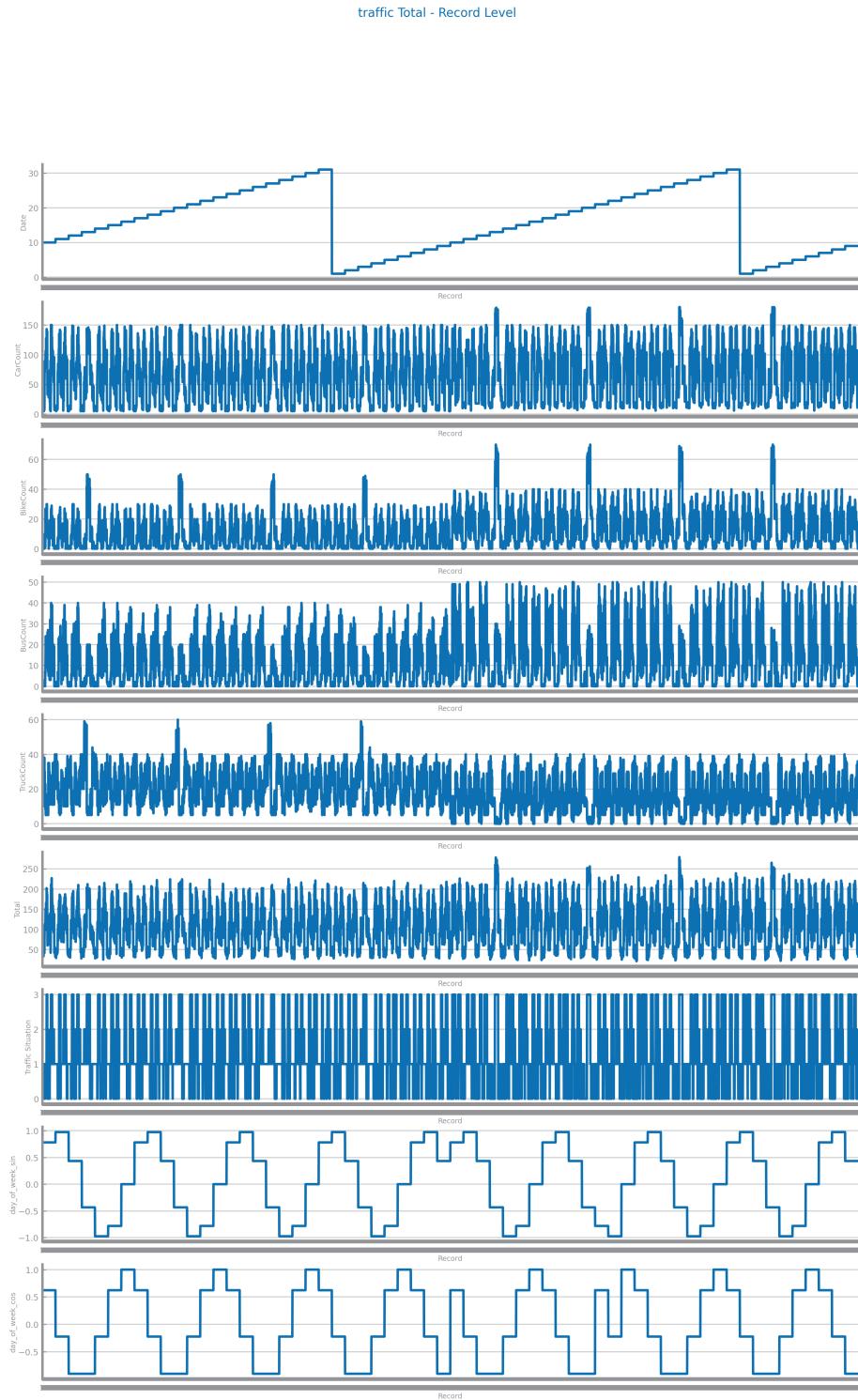


Figure 65: Time series 1 at the most granular detail

traffic Total - Daily

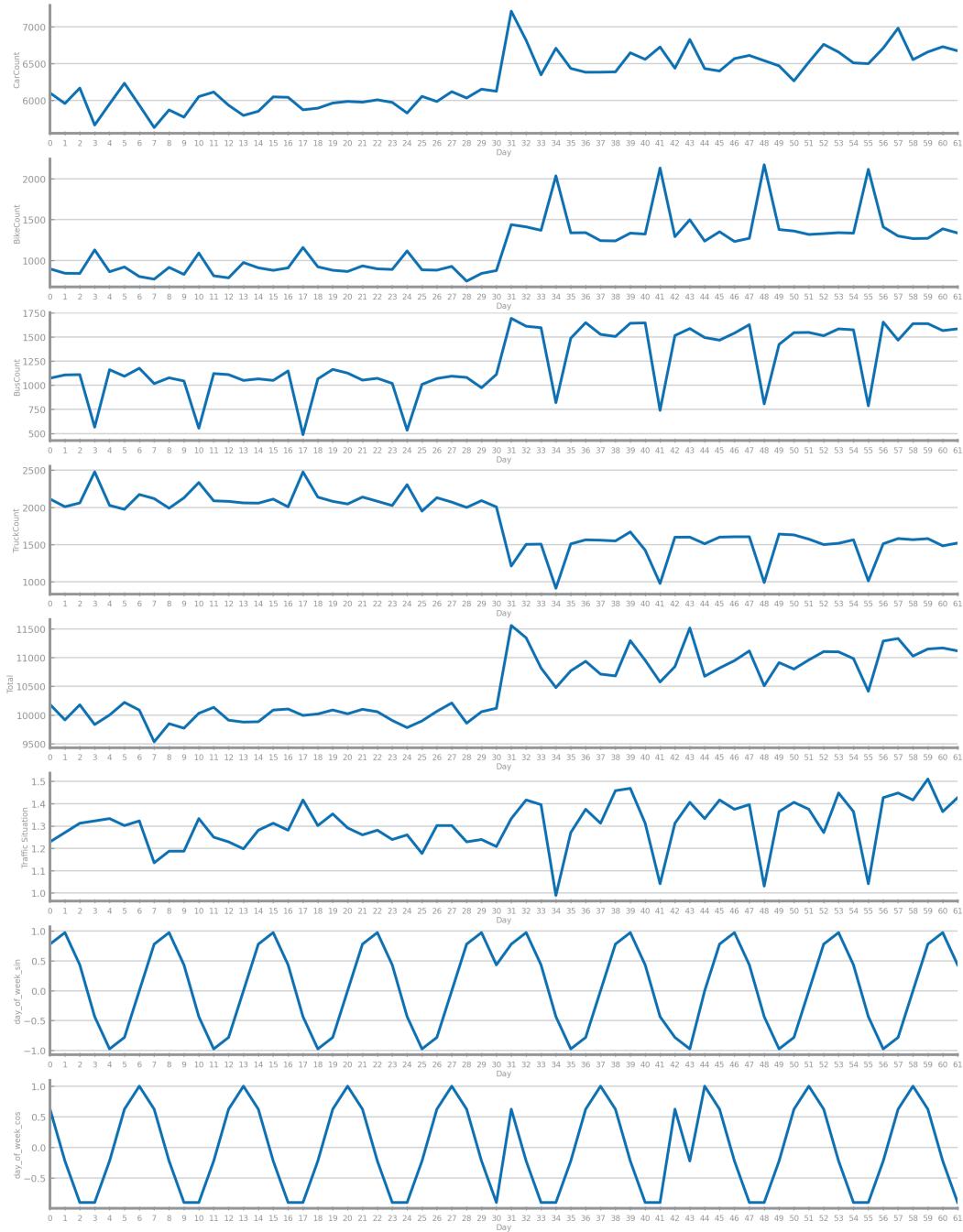


Figure 66: Time series 1 at the second chosen granularity

traffic Total - Weekly

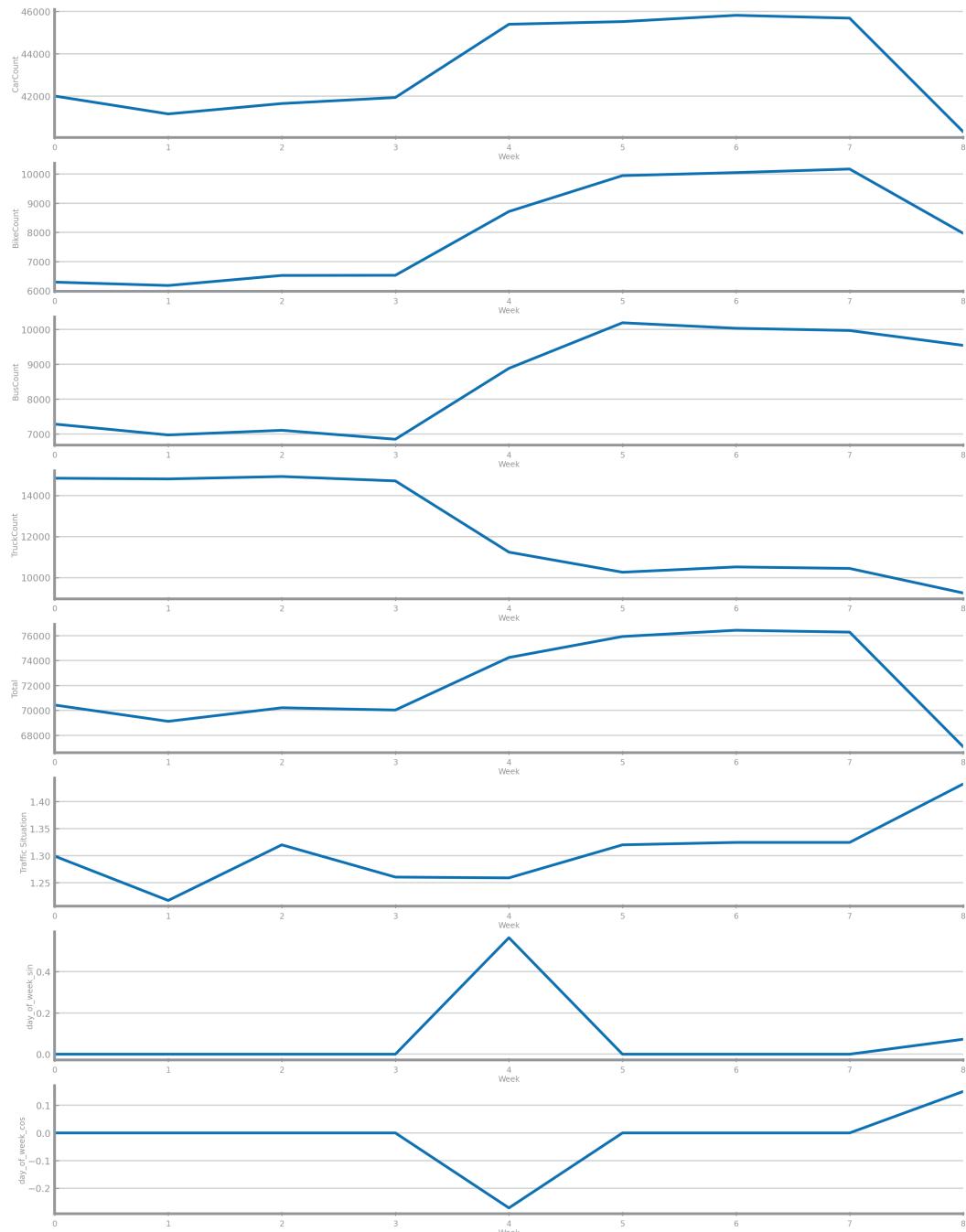


Figure 67: Time series 1 at the third chosen granularity

Data Distribution

The distribution analysis across granularities shows that aggregation reduces variability and outliers. The 15-minute series presents higher dispersion and skewness, while hourly and daily series exhibit smoother distributions.

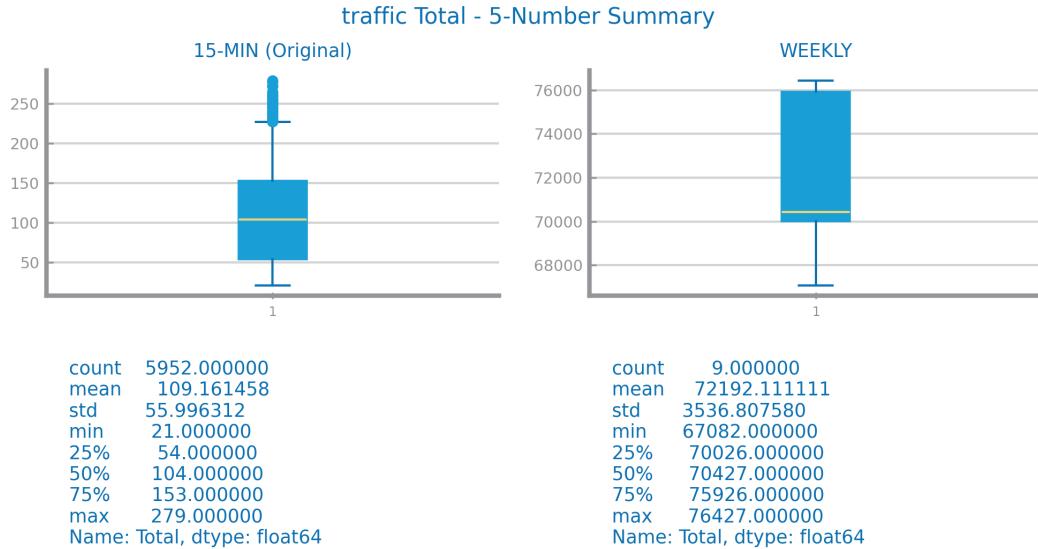


Figure 68: Boxplot(s) for time series 1

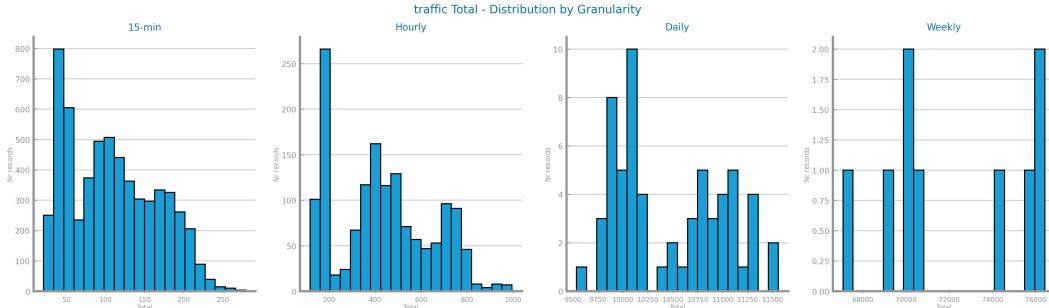


Figure 69: Histogram(s) for time series 1

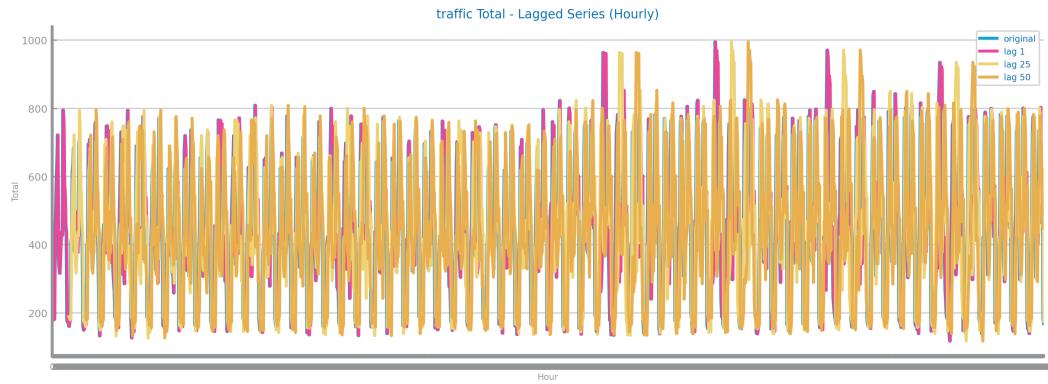


Figure 70: Autocorrelation lag-plots for original time series 1

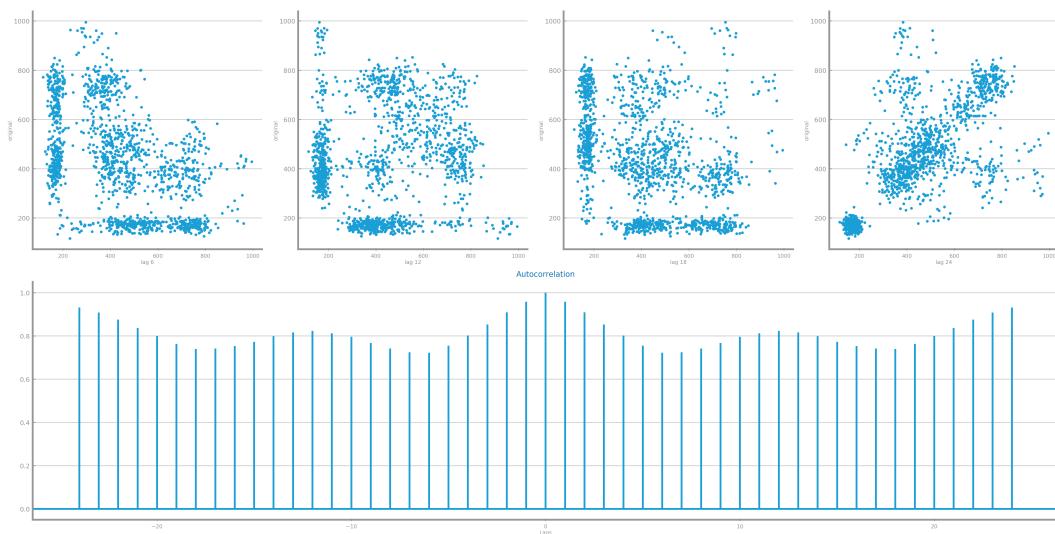


Figure 71: Autocorrelation correlogram for original time series 1

Data Stationarity

Stationarity analysis shows that the series is stationary at finer granularities but loses stationarity when aggregated daily. Augmented Dickey-Fuller Test confirm that trend components become more pronounced at coarser granularities, indicating the need for detrending or differencing when modeling aggregated series.

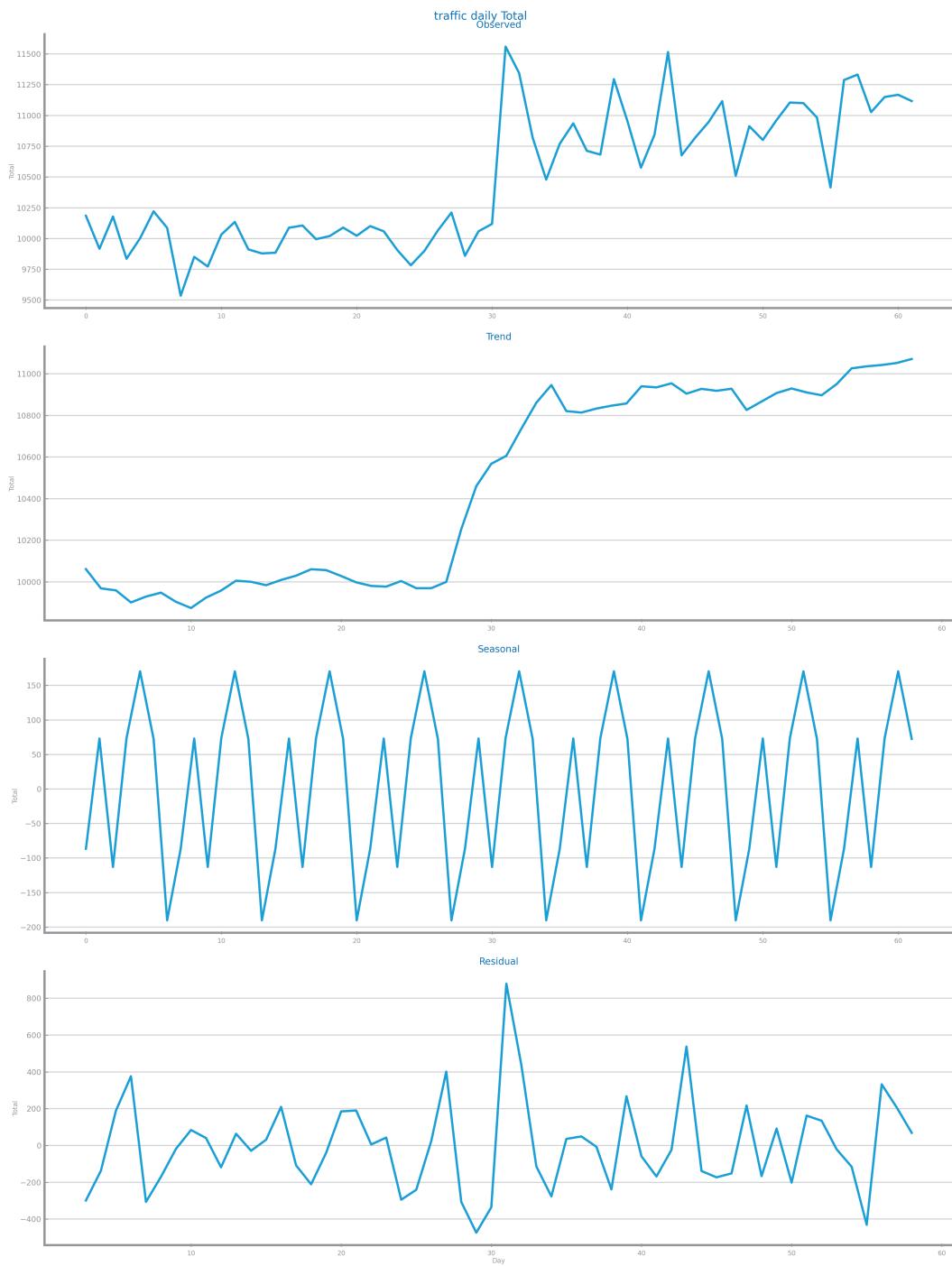


Figure 72: Components study for time series 1

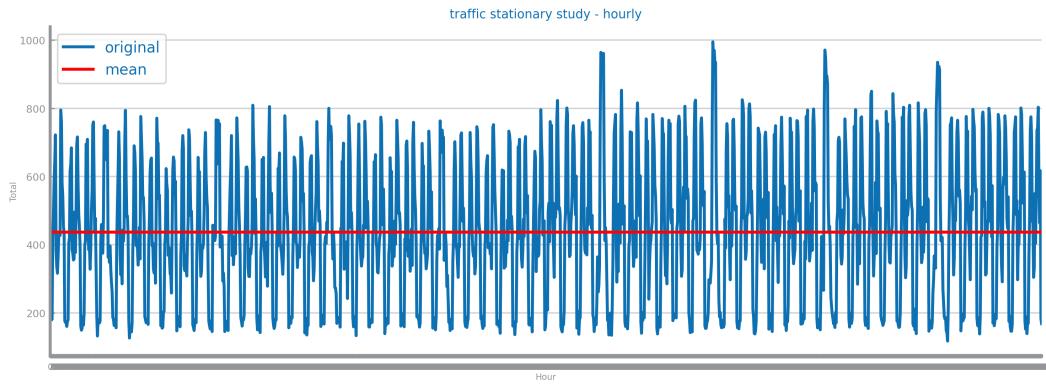


Figure 73: Stationarity study for time series 1

Augmented Dickey-Fuller Test Results:

Original (15-min):

- ADF Statistic: -14.441
- p-value: 0.000
- Critical Values: 1%: -3.431, 5%: -2.862, 10%: -2.567
- **The series IS stationary**

Hourly:

- ADF Statistic: -8.903
- p-value: 0.000
- Critical Values: 1%: -3.435, 5%: -2.864, 10%: -2.568
- **The series IS stationary**

Daily:

- ADF Statistic: -0.826
- p-value: 0.811
- Critical Values: 1%: -3.548, 5%: -2.913, 10%: -2.594
- **The series IS NOT stationary**

6 DATA TRANSFORMATION

Aggregation

We tested 30-minute, hourly, daily, and weekly aggregations to see how each level reduces noise and how much detail is lost. The plots (see Figures 75, 76, and 77) reveal that no aggregation keeps high-frequency noise, while daily and weekly

levels smooth the data too much. On the test set, linear regression produced constant predictions with R^2 close to 0, signaling a poor fit to the data. Meanwhile, the optimistic persistence model performed better, and the 30-minute level gave the best R^2 (0.67), so we chose it.

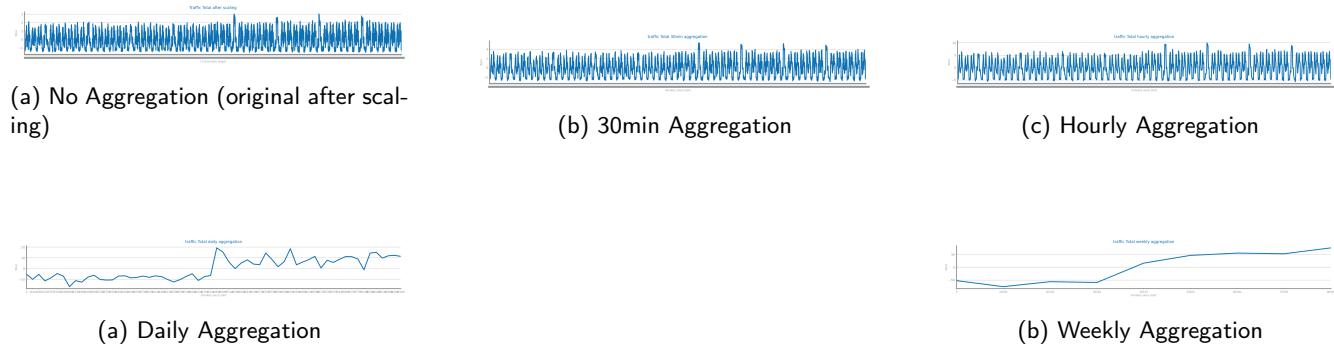


Figure 75: Time series plots after different levels of aggregation



Figure 76: Forecasting plots for Linear Regression and Persistence Optim after different aggregations

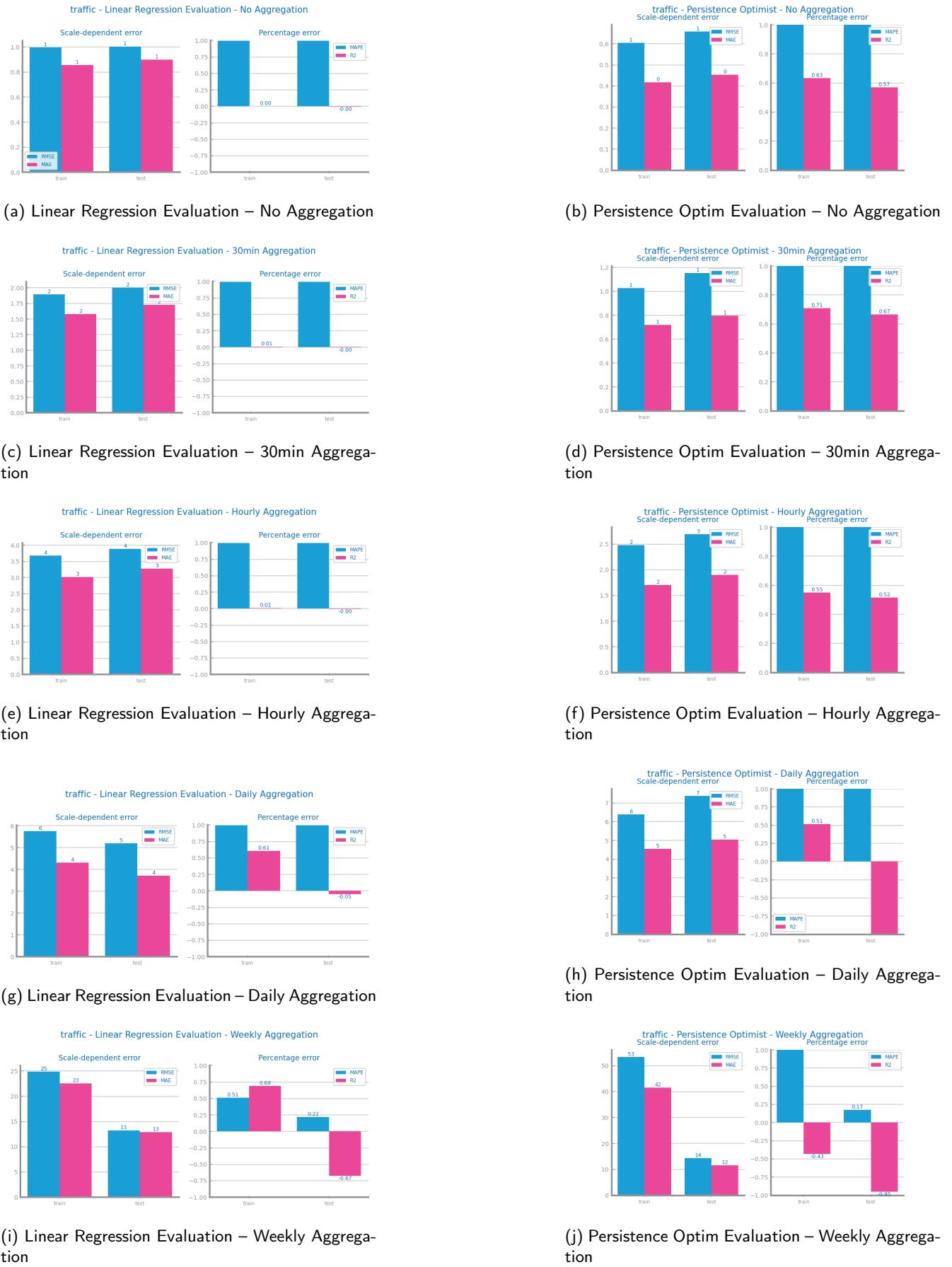
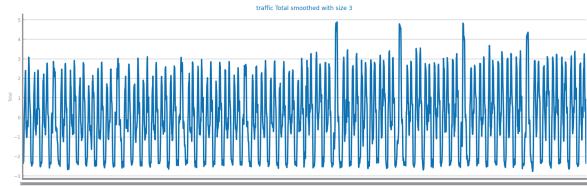


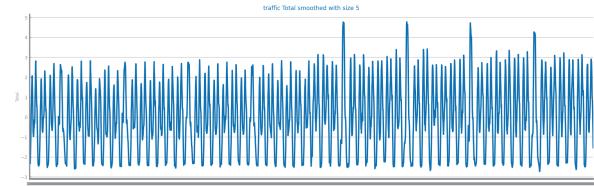
Figure 77: Evaluation results for Linear Regression and Persistence Optim after different aggregations

Smoothing

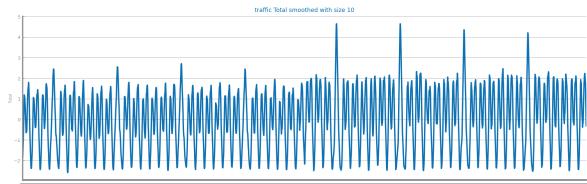
We tested moving-average smoothing with window sizes of 3, 5, 10, and 15 to see how noise reduction affects forecasting without distorting the data. Smaller windows kept more noise, while larger ones over-smoothed. The window of 15 worked best, giving RMSE 0.3, MAE 0.25, and R^2 0.96 in the optimistic persistence model. Linear regression stayed flat with low R^2 , so we chose size 15.



(a) Smoothing Size 3



(b) Smoothing Size 5



(c) Smoothing Size 10



(d) Smoothing Size 15

Figure 78: Time series plots after applying moving average smoothing with different window sizes

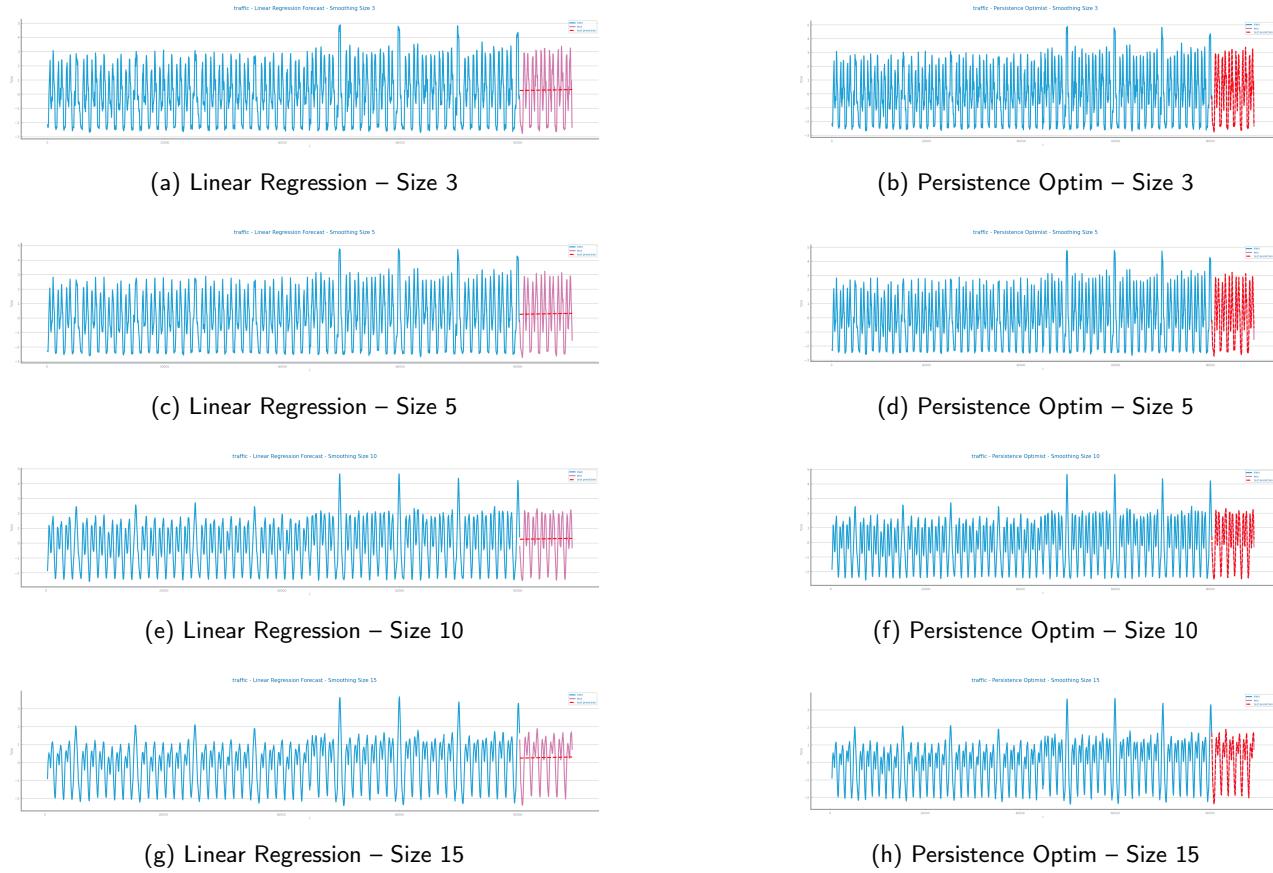


Figure 79: Forecasting plots for Linear Regression and Persistence Optim after different smoothing window sizes



Figure 80: Evaluation results for Linear Regression and Persistence Optimist after different smoothing window sizes

Differentiation

To handle possible non-stationarity, we tested no differencing, first-order, and second-order differencing. No differencing kept the trend, first-order removed linear trends and centered the data, and second-order added noise without helping. The

optimistic persistence model performed best with no differencing, reaching $R^2 = 0.95$. Linear regression improved only slightly, so we chose no differencing.

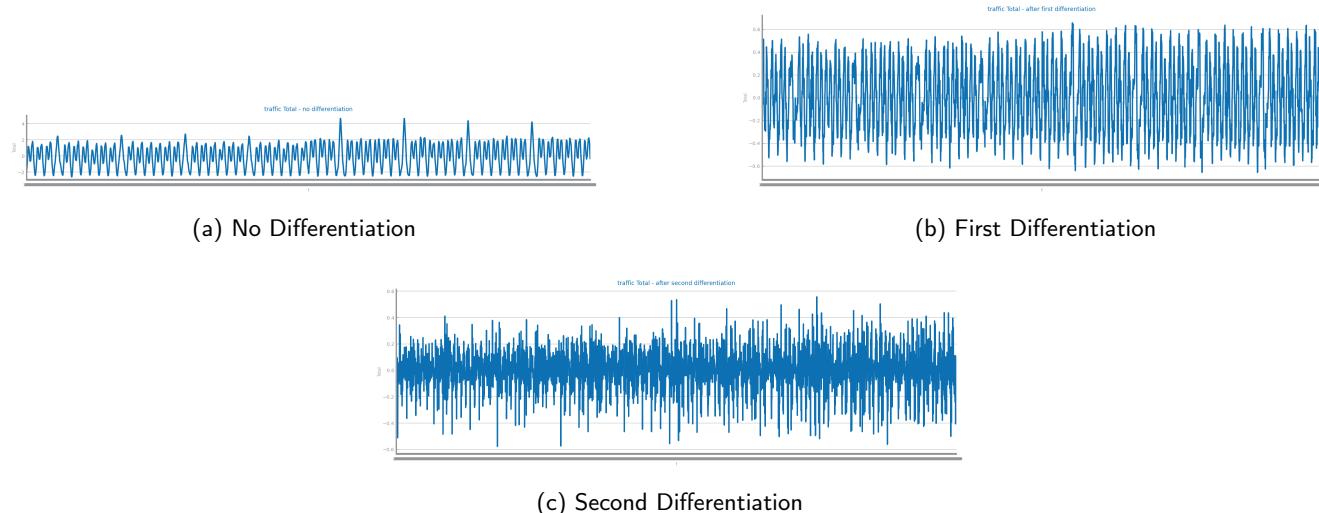


Figure 81: Time series plots without and after applying first, and second differentiation

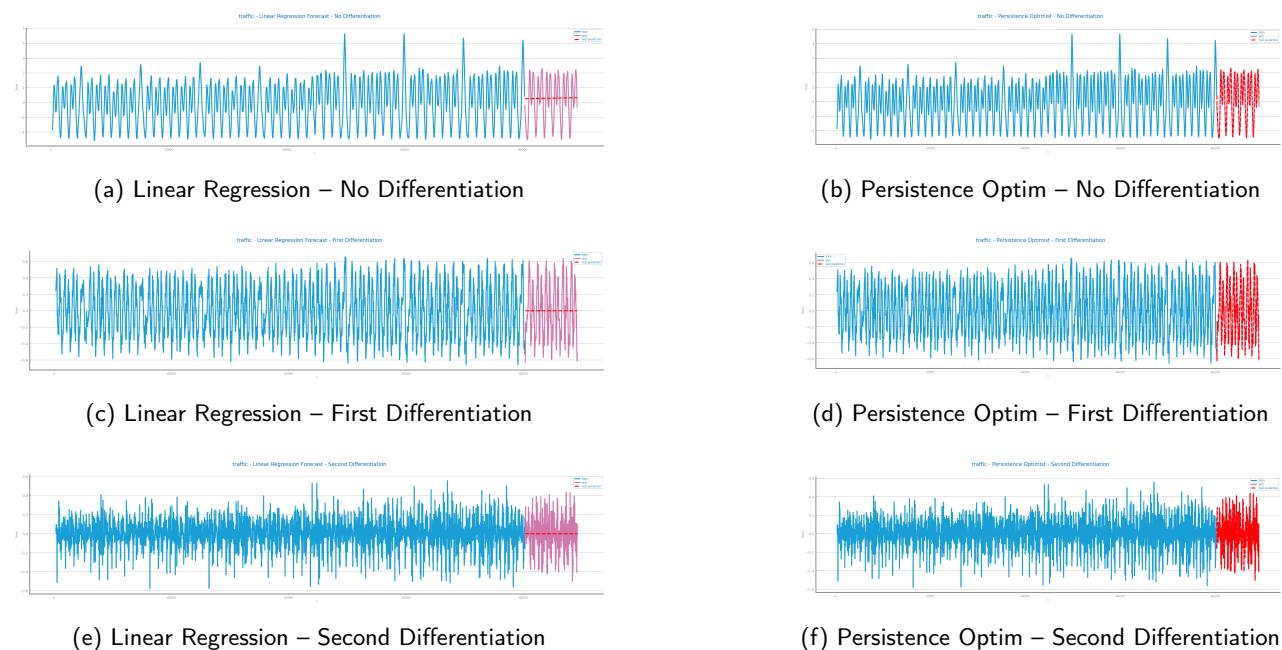


Figure 82: Forecasting plots for Linear Regression and Persistence Optim after different levels of differentiation



(a) Linear Regression Evaluation – No Differentiation



(b) Persistence Optimist Evaluation – No Differentiation



(c) Linear Regression Evaluation – First Differentiation



(d) Persistence Optimist Evaluation – First Differentiation



(e) Linear Regression Evaluation – Second Differentiation



(f) Persistence Optimist Evaluation – Second Differentiation

Figure 83: Evaluation results for Linear Regression and Persistence Optimist after different levels of differentiation

Scaling

We used StandardScaler to normalize the data, converting it to a mean of 0 and a standard deviation of 1. Before scaling, the values ranged from about 25 to 250 with high variance; after scaling, they fell roughly between -1 and 3. This improved model stability, and sped up convergence as shown in the before-and-after plots. We did not test other scaling methods because StandardScaler solved the scale issues without adding complexity. The final preparation pipeline for the traffic dataset is: 30-minute aggregation, smoothing with window size 15, no differencing, and scaling with StandardScaler.

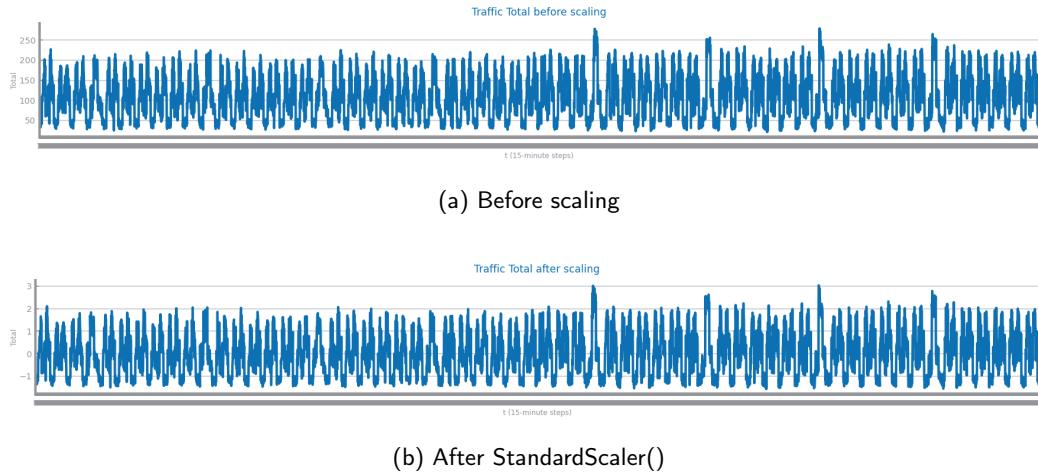


Figure 84: Effect of StandardScaler on the original 15-minute time series

7 MODELS’ EVALUATION

Exponential Smoothing Model

We applied simple exponential smoothing and tuned alpha from 0.1 to 0.9. The hyperparameter study (Figure 85) shows R^2 starts near 0.0 at alpha = 0.1 and drops sharply to -1.0 by alpha = 0.3, indicating the model overreacts to recent changes.

The forecast (Figure 86) follows the overall trend but smooths out sharp variations. The evaluation (Figure 87) shows poor test performance: RMSE ≈ 0.24 , MAE ≈ 0.20 , MAPE $\approx 100\%$, $R^2 \approx -0.01$, revealing weak accuracy and poor generalization.

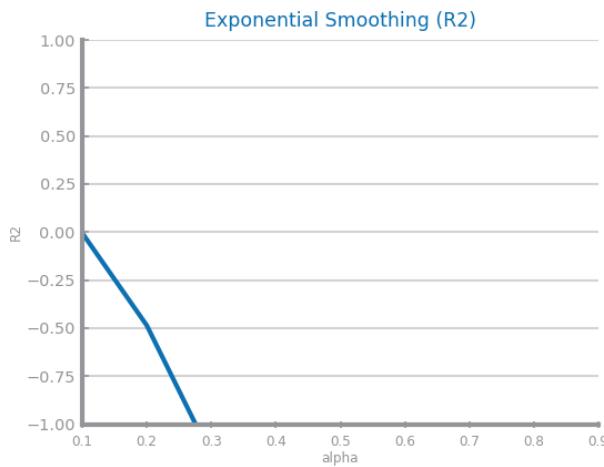


Figure 85: Hyperparameter study: R^2 as a function of alpha for Exponential Smoothing

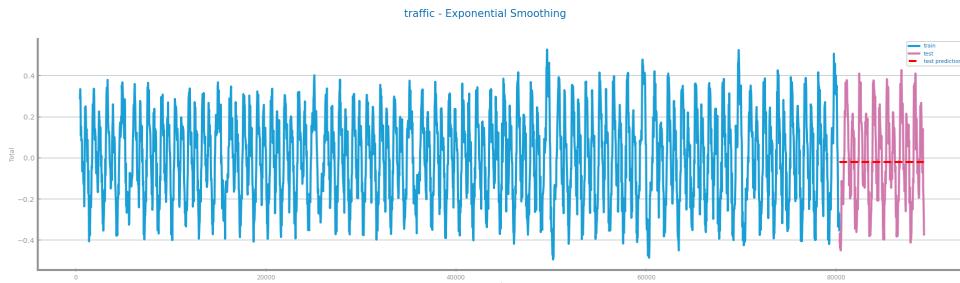


Figure 86: Forecasting plots obtained with the best Exponential Smoothing model (predictions in red vs actual test data in pink)

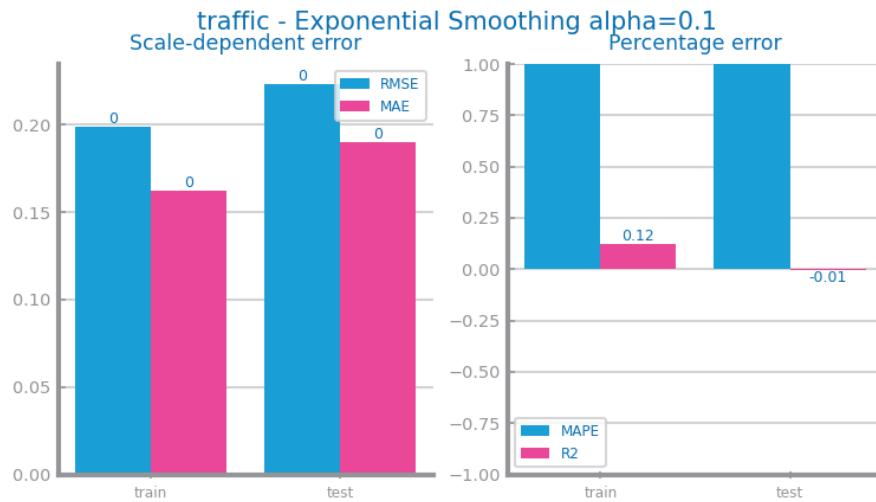


Figure 87: Performance metrics of the best Exponential Smoothing model (RMSE, MAE, MAPE, R^2)

Multi-layer Perceptrons Model

For the MLP model, we explored various hidden layer architectures, including single layers with 50 or 100 neurons, and multi-layers such as (50, 50) and (100, 50), trained with the Adam optimizer and MSE loss. The best setup was (50, 50), reaching test $R^2 \approx 0.86$ after ~ 800 epochs with stable training and no overfitting. The forecasting plot (Figure 89) closely matched the test data, capturing nonlinear patterns better than simpler models. Performance metrics (Figure 90) show RMSE ≈ 0.25 , MAE ≈ 0.20 , MAPE $\approx 18\%$, and $R^2 \approx 0.86$, confirming solid medium-term performance despite higher computational cost.

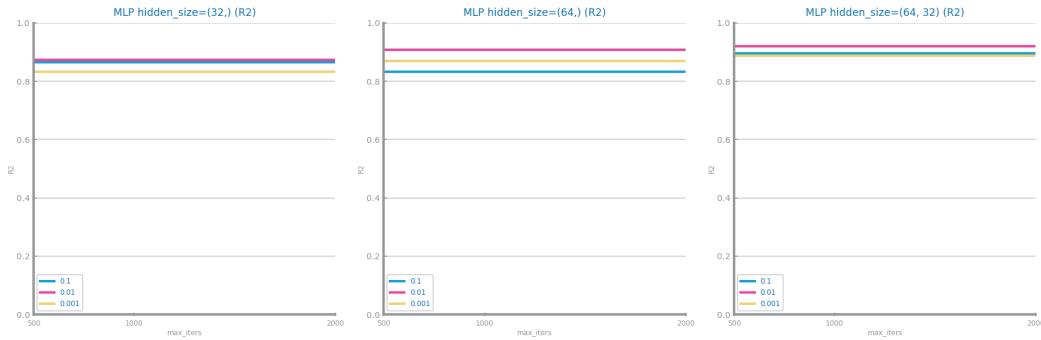


Figure 88: Hyperparameter study: R^2 convergence for different MLP hidden layer configurations

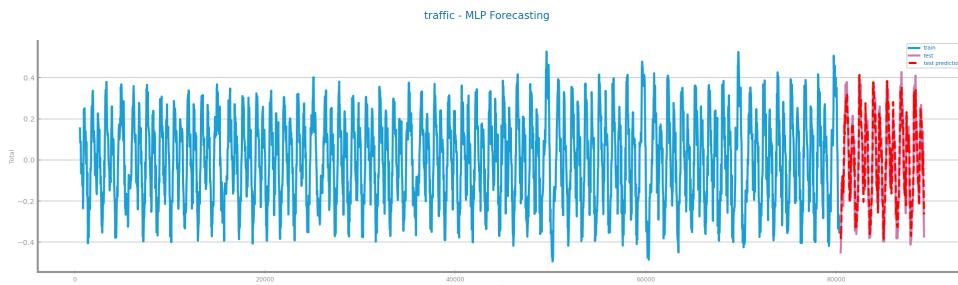


Figure 89: Forecasting plots obtained with the best MLP model (predictions in red vs actual test data in pink)

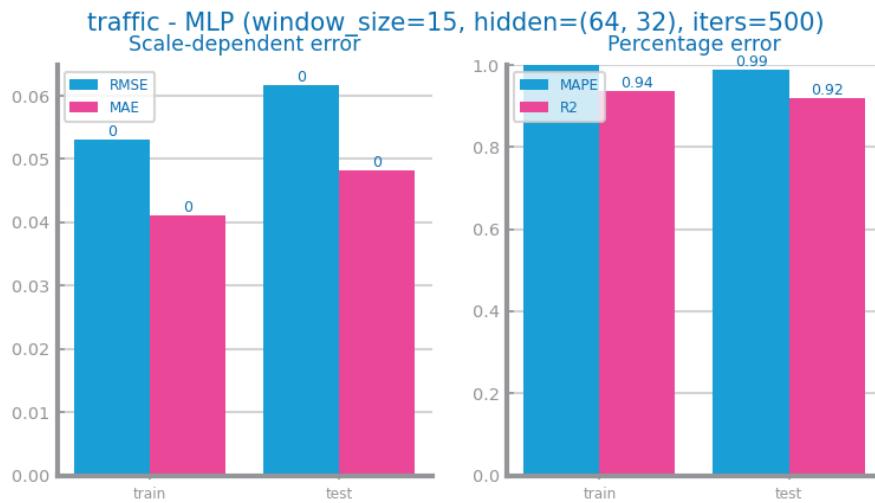


Figure 90: Performance metrics of the best MLP model (RMSE, MAE, MAPE, R^2)

ARIMA Model

We ran a grid search for ARIMA with p and q from 0 to 8 and d from 0 to 2. The study (Figure 91) highlights the optimal univariate configuration as $p=5$, $d=1$, $q=7$. ARIMA fit training data well ($R^2 \approx 0.85$) but dropped on test ($R^2 \approx 0.08$), with RMSE rising from 0.08 to 0.21 (Figure 93). VAR with $\text{lag}=4$ showed more stable training ($R^2 \approx 0.82$) but also weak

test performance ($R^2 \approx 0.06$) (Figure 96). Both models captured trends but failed on sharp changes, with high MAPE (100%) and limited generalization. (Figure 92)

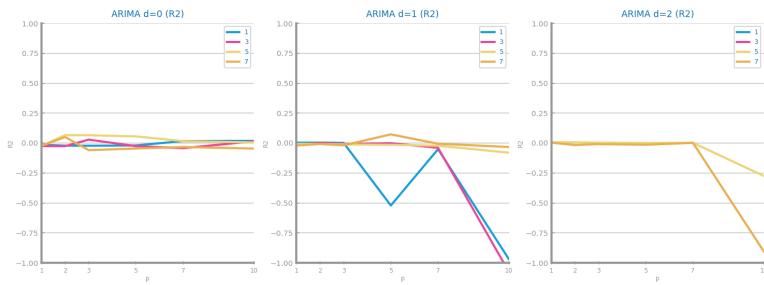


Figure 91: Hyperparameter study: best ARIMA configuration ($p=5, d=1, q=7$) – univariate

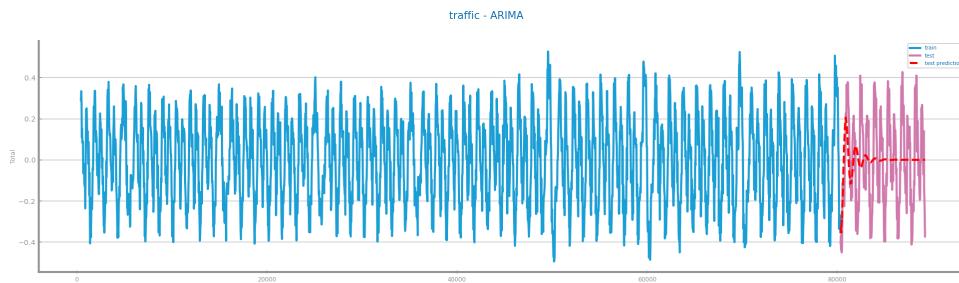


Figure 92: Forecasting plots obtained with the best ARIMA model (predictions in red vs actual test data in pink) – univariate



Figure 93: Performance metrics of the best ARIMA model (RMSE, MAE, MAPE, R²) – univariate

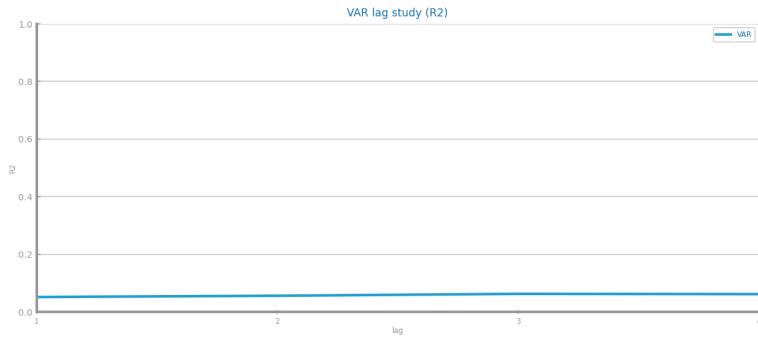


Figure 94: Hyperparameter study: best VAR configuration (lag=4) – multivariate



Figure 95: Forecasting plots obtained with the best VAR model (predictions in red vs actual test data in pink) – multivariate

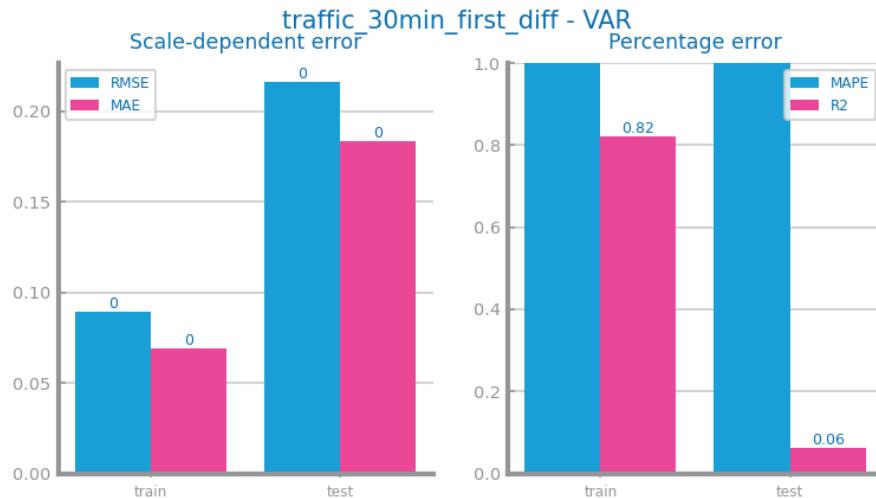


Figure 96: Performance metrics of the best VAR model (RMSE, MAE, MAPE, R²) – multivariate

LSTMs Model

We tuned LSTM with sequence lengths 3-6, hidden units 20-50, and up to 2500 epochs using Adam and MSE. The study (Figure 97) identifies the best univariate setup was seq=4, hidden=25, epochs=2100, reaching test R² 0.91. The

forecasting plot (Figure 98) closely matched test data, capturing trends and irregularities. Metrics (Figure 99) report test RMSE 0.20, MAE 0.15, MAPE 13%, R² 0.84. In the multivariate case, similar tuning led to stable R² 0.87 and better generalization across variables.

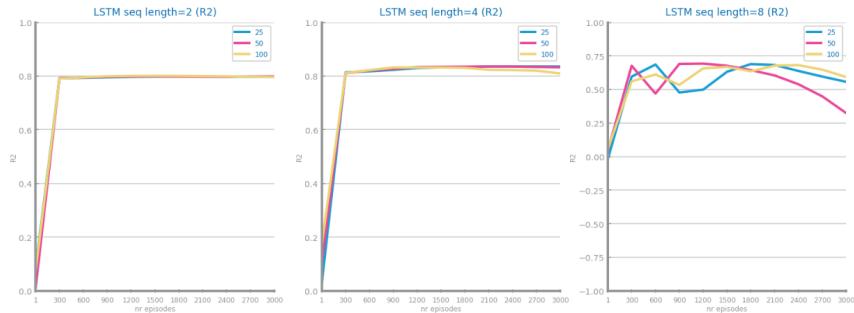


Figure 97: Hyperparameter study: best LSTM configuration (sequence length=4, hidden=25, epochs=2100) – univariate

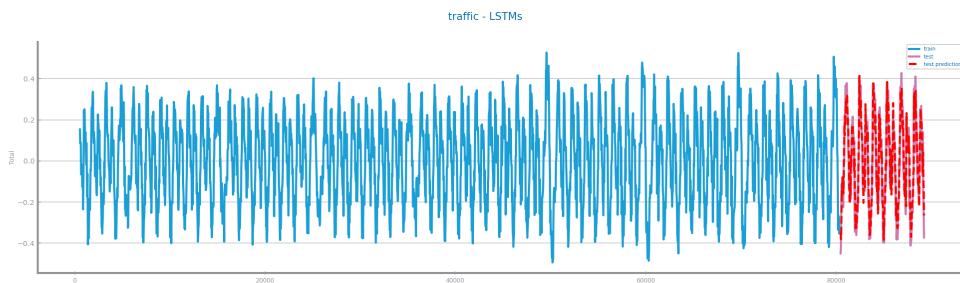


Figure 98: Forecasting plots obtained with the best LSTM model (predictions in red vs actual test data in pink) – univariate

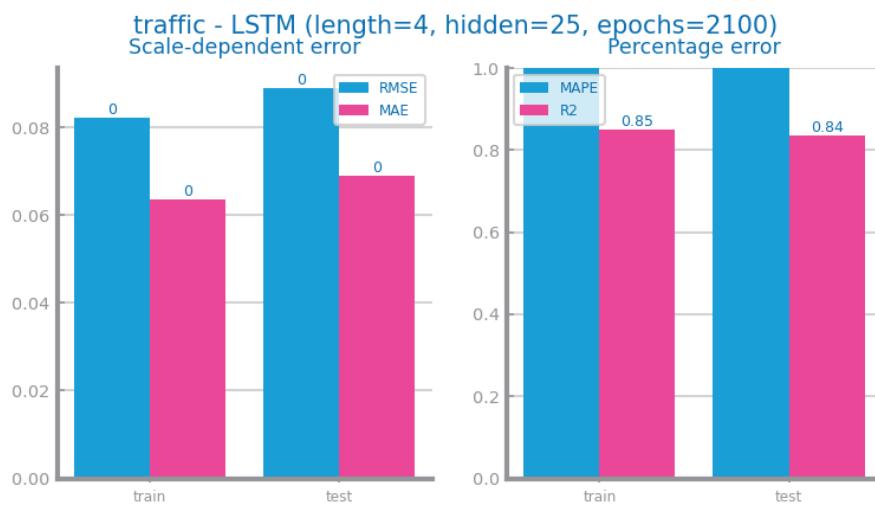


Figure 99: Performance metrics of the best LSTM model (RMSE, MAE, MAPE, R²) – univariate

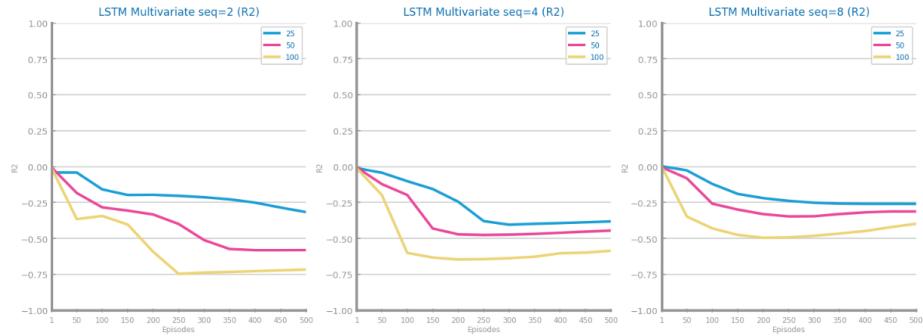


Figure 100: Hyperparameter study: best LSTM configuration – multivariate

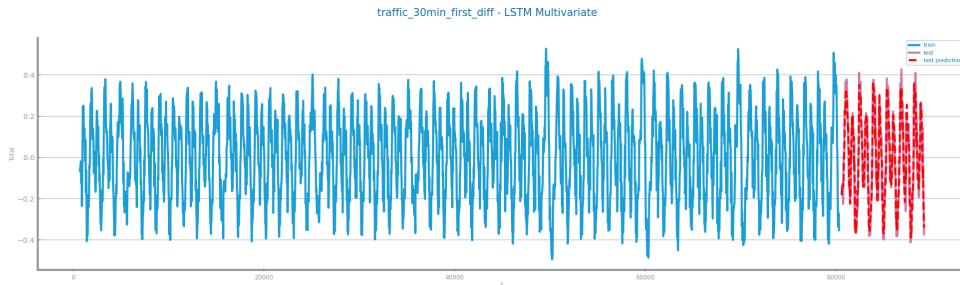


Figure 101: Forecasting plots obtained with the best LSTM model (predictions in red vs actual test data in pink) – multivariate

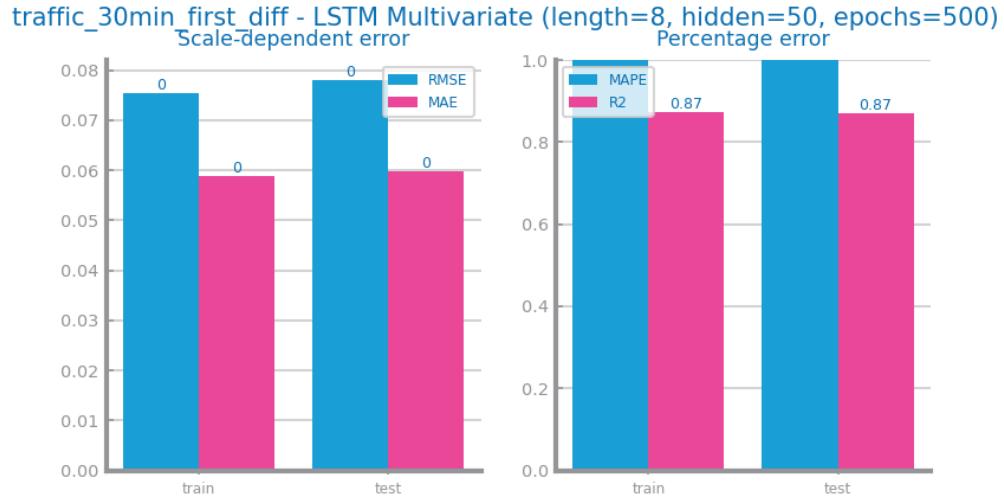


Figure 102: Performance metrics of the best LSTM model (RMSE, MAE, MAPE, R^2) – multivariate

8 CRITICAL ANALYSIS

The forecasting models showed distinct performance differences on the processed traffic data. Exponential smoothing performed the worst, with test R^2 near zero, very high MAPE, and poor ability to follow variability, making it unsuitable

for this task. ARIMA and VAR fit the training data well but generalized poorly, with low test R^2 (≈ 0.08 and 0.06). Both models captured overall trends but failed on sharp fluctuations due to their linear assumptions and limited flexibility. MLP delivered stronger results, reaching $R^2 \approx 0.86$ and capturing nonlinear patterns more effectively, though at a higher computational cost. LSTM clearly outperformed all other models. The univariate version reached $R^2 \approx 0.84$, while the multivariate version achieved ≈ 0.87 , with lower RMSE and MAPE. LSTM handled sequential dependencies, irregularities, and local variations better than statistical models and even better than MLP, making it the most reliable option for medium-term traffic forecasting. Aggregating the data into 30-minute intervals reduced high-frequency noise and exposed clearer temporal patterns. Smoothing with a window of 15 further stabilized the series without removing essential structure. Avoiding differencing prevented unnecessary noise amplification, since the aggregated and smoothed series was already close to stationary. Scaling was essential for neural models, ensuring stable gradients and faster convergence. Together, these steps transformed noisy raw data into a predictable signal that models could learn effectively.