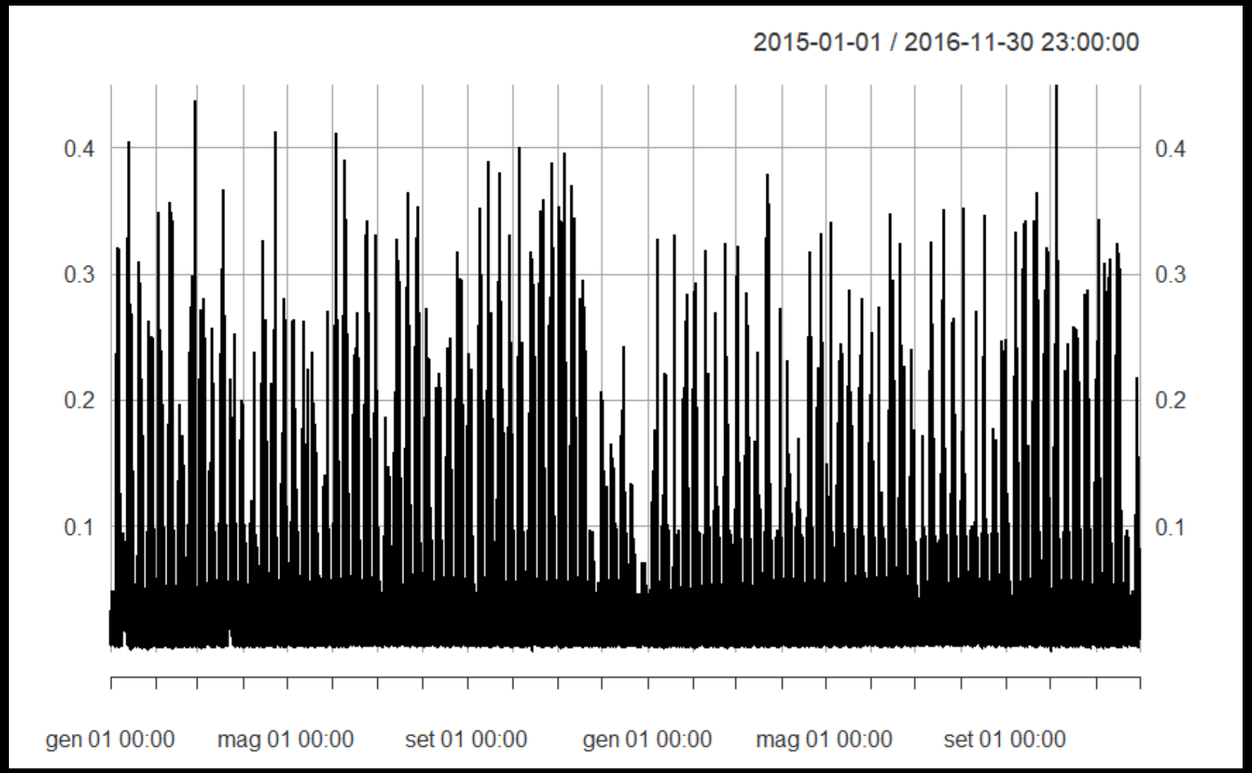
# TIME SERIES PROJECT

Andrea D’Amicis 869008

# CONTENTS

# INTRODUCTION




MEAN: 0.05
SD: 0.05
p50: 0.02

- Serie of **hourly** traffic congestion indicator for a freeway in a large U.S. city.
- Data from 2015-01-01 to 2016-11-30 (16800 known values)
- Predict the hourly values of last month from 2016-12-01 to 2016-12-31 (744 values)

# NA AND ZERO VALUES

## ZEROS

107 instances

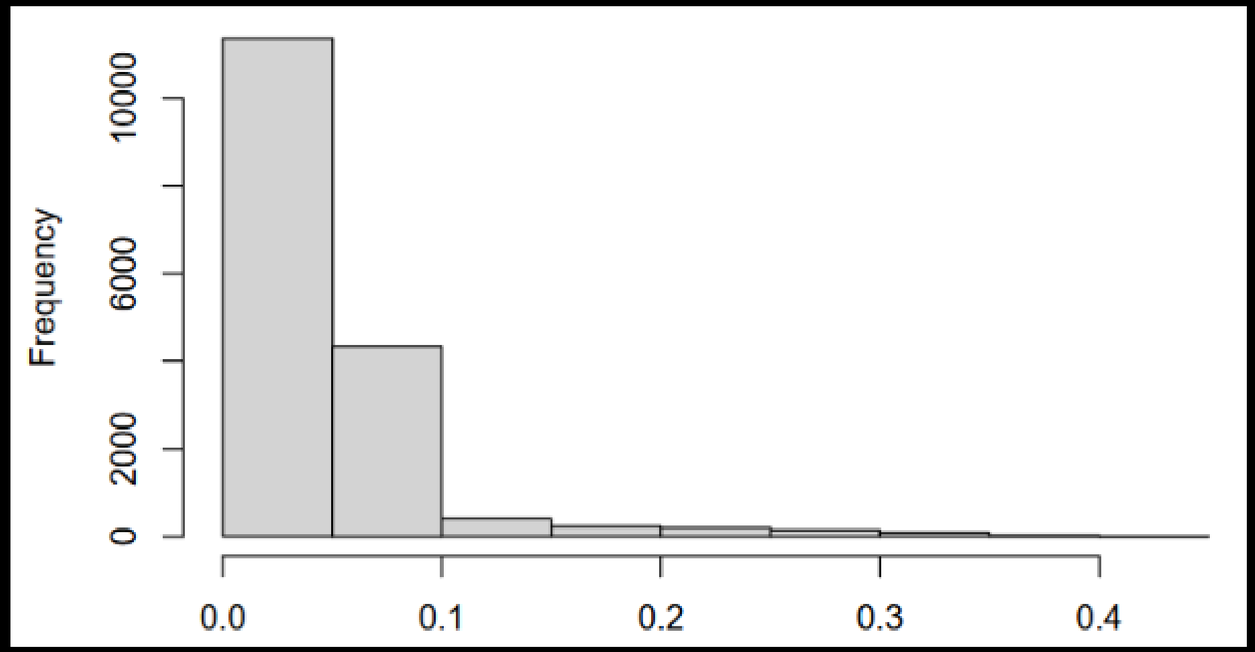
## NA VALUES

No presence

## SOLUTION

Replace values using a **moving median** method in 24-hours window (consisting of 12 hours before and 12 hours after) with the objective of smooths the serie

# OUTLIERS




- The analysis of frequency distribution reveal that majority of data range from 0 to 0.1
- Very low-frequency class are likely to be considered outliers

- Used 99th percentile to identify anomalies → Discovered 168 instances
- Replace these values with the median

# SPLITTING TIME SERIE

01 The hourly time series was divided into 24 daily time series with each series containing observations recorded at the **same hour of the day across all days**

02 Adopted to facilitate a more detailed analysis of seasonality across multiple levels, such as hourly, weekly and annual patterns.

03 Allows the behavior of each specific hour to be studied independently

01 Relationship between the means and standard deviations was analyzed to discover stationarity

→ Suggesting that every time serie is not stationary

02 To ensure consistency over time applying a variance-stabilizing transformation

→ Box-Cox transformation was applied to each of the 24 daily time series

# STATIONARITY AND BOX-COX

# BOX-COX AND LAMBDAS

- Optimal lambda value was calculated for each series
- The diversity of optimal lambda values highlights the different characteristics, indicating that each time series has distinct statistical properties.
- Table with a corresponding transformations related to lambdas is provided

| $\lambda$ | Transformation |
|---|---|
| $-2$ | $\frac{1}{x^2}$ |
| $-1$ | $\frac{1}{x}$ |
| $-0.5$ | $\frac{1}{\sqrt{x}}$ |
| $0$ | $\log(x)$ |
| $0.5$ | $\sqrt{x}$ |
| $1$ | $x$ |
| $2$ | $x^2$ |

| Optimal Lambdas | | | | | |
|---|---|---|---|---|---|
| -0.99995568 | -0.99995223 | -0.99992471 | -0.28183339 | -0.03410270 | -0.40158558 |
| 0.07960928 | -0.14448631 | -0.30520352 | -0.33984977 | -0.22154101 | 1.99992425 |
| 1.99992425 | 1.99992425 | 0.48873312 | 0.10206000 | 0.08864715 | 1.09296178 |
| 0.05761952 | 0.87154550 | 0.64040828 | -0.75452759 | -0.25330359 | -0.91224536 |

| Hour | IsStationary | P-Value | Tau Statistic |
|---|---|---|---|
| 1 | TRUE | 0.01000000 | -4.917948 |
| 2 | TRUE | 0.01000000 | -5.837128 |
| 3 | TRUE | 0.01000000 | -6.839808 |
| 4 | TRUE | 0.01000000 | -6.895324 |
| 5 | TRUE | 0.01000000 | -6.445611 |
| 6 | TRUE | 0.01000000 | -7.580303 |
| 7 | TRUE | 0.01000000 | -6.412932 |
| 8 | TRUE | 0.01000000 | -7.458907 |
| 9 | TRUE | 0.01000000 | -7.478475 |
| 10 | TRUE | 0.01000000 | -7.588359 |
| 11 | TRUE | 0.01000000 | -7.111670 |
| 12 | TRUE | 0.01000000 | -4.679636 |
| 13 | TRUE | 0.01000000 | -5.266351 |
| 14 | TRUE | 0.01000000 | -5.490569 |
| 15 | TRUE | 0.01000000 | -6.621222 |
| 16 | TRUE | 0.01000000 | -7.116080 |
| 17 | TRUE | 0.01000000 | -8.229424 |
| 18 | TRUE | 0.01000000 | -6.939559 |
| 19 | TRUE | 0.01000000 | -7.507875 |
| 20 | TRUE | 0.01821757 | -3.815609 |
| 21 | TRUE | 0.04207562 | -3.445131 |
| 22 | TRUE | 0.04758142 | -2.875880 |
| 23 | TRUE | 0.01000000 | -4.254819 |
| 24 | TRUE | 0.01000000 | -5.195442 |

# AUGMENTED DICKEY–FULLER

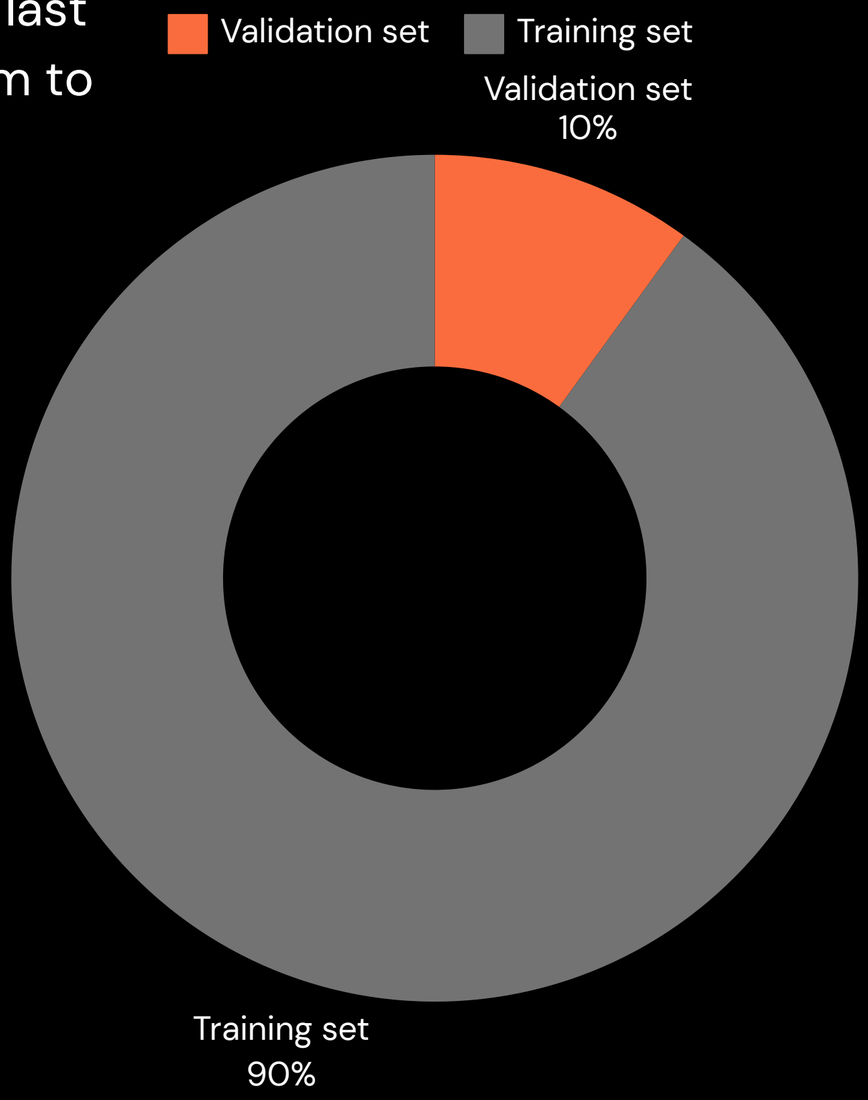
01 To check if the preprocessing performed was consistent

02 Comparing the tau statistic to a critical value at significance level (5%) which is –2.86

03 If the tau statistic is more negative than the critical value at the 5% level, the null hypothesis of non-stationarity is rejected, indicating that the series is stationary

# TRAIN AND VALIDATION SETS

- Initial hourly dataset contains 17,544  observations. But the last 744 are unknowns, as they represent the predictions we aim to make with final model
- Filled hourly dataset contains 16800 observations (17,544 - 744 = 16800 values)
- Divided into 24 daily time series, each consisted of 700 values
- Each of them finally divided into training and validation



# DUMMY VARIABLES

To improve the model's predictive accuracy, dummy variables for specific dates and holidays were introduced and one-hot encoded. These variables capture special events that could influence the time series behavior.

Christmas Eve

Boxing Day

New Year’s Day

Valentine’s Day

Assumption of Mary

Christmas Day

New Year’s Eve

Epiphany

Easter days

# EVALUATION METRIC

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

Where:

- $n$ is the number of observations in the test set,
- $y_i$ represents the actual observed value at time step $i$,
- $\hat{y}_i$ is the predicted value at time step $i$,
- $|y_i - \hat{y}_i|$ is the absolute error for each prediction.

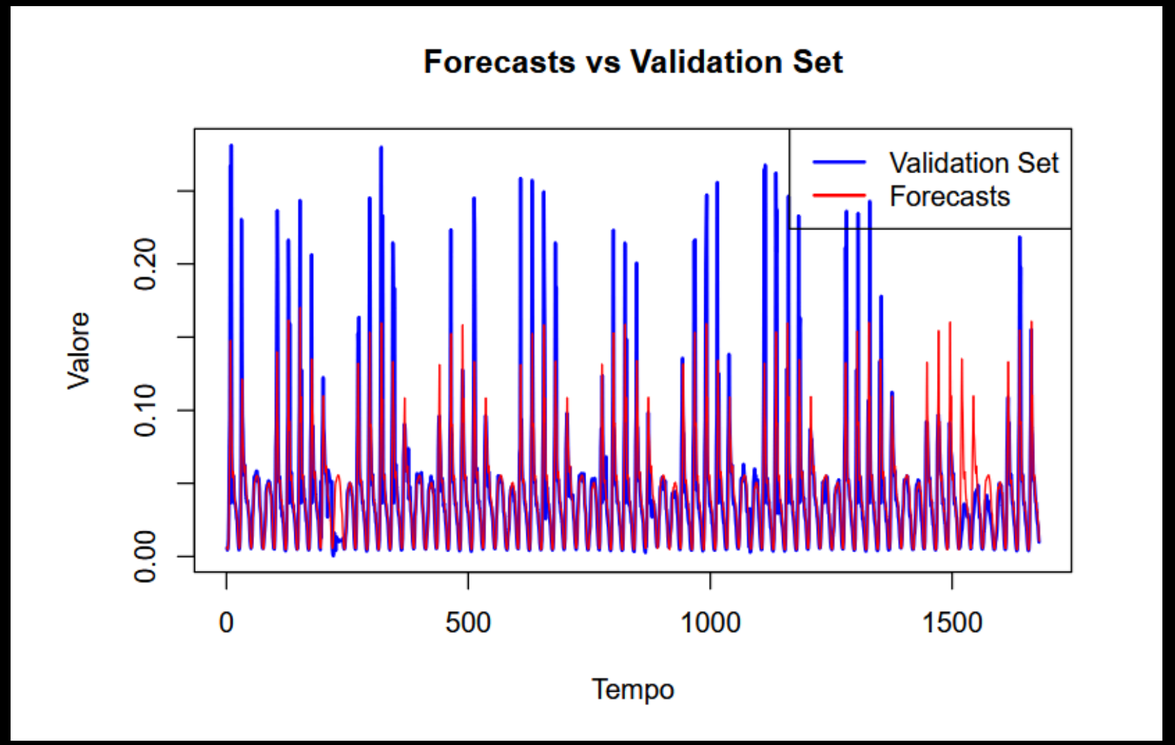
- Mean Absolute Error (MAE), measures the average magnitude of errors between predicted and actual values, providing an interpretable measure of model accuracy
- A lower MAE indicates better model performance, as it signifies that the predictions are closer to the actual observations on average

# ARIMA MODELS

After different trials, three promising models were identified and then refined by incorporating one-hot encoded dummy variables for specific dates and holidays, as detailed in the preprocessing section.

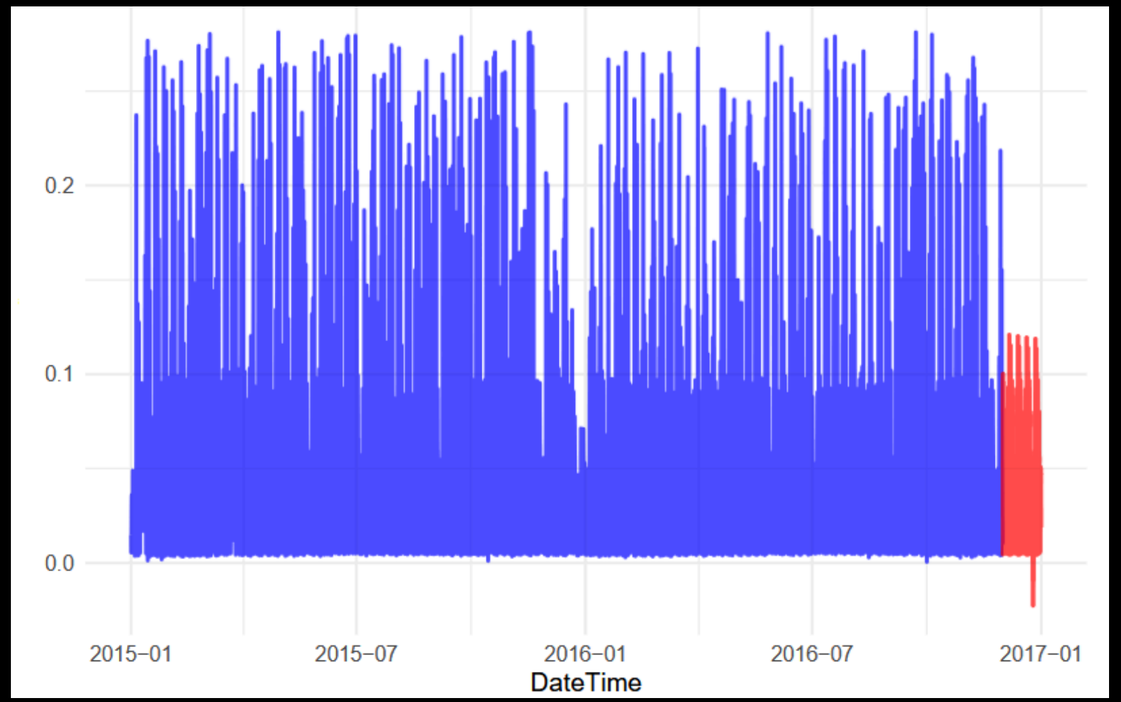
| Model | Normal | With Dummies |
|---|---|---|
| ARIMA (3,1,1) (0,1,1)(7) | 0.009563 | 0.009546 |
| ARIMA (3,1,1) (0,1,2)(7) | 0.009558 | 0.009562 |
| ARIMA (3,1,1) (1,1,1)(7) | 0.009552 | 0.009554 |

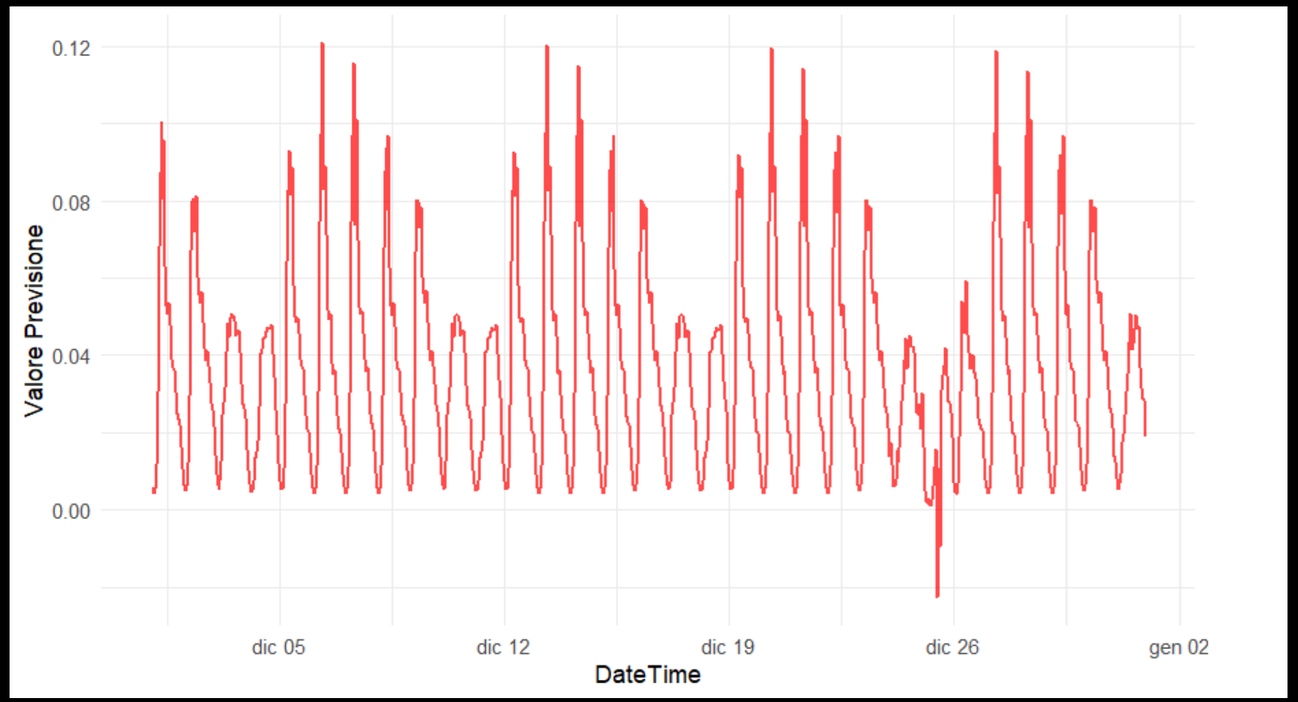
The models seem to achieve the same level of performance. Dummies influence performance very little and in some cases worsen.



# ARIMA (3,1,1) (0,1,1)(7)

This type of model was retrained using the full dataset up to 2016-11-30, and the objective was to predict the hourly data for December 2016. A plot shows the actual time series followed by the model's predicted values and the forecasts itself.

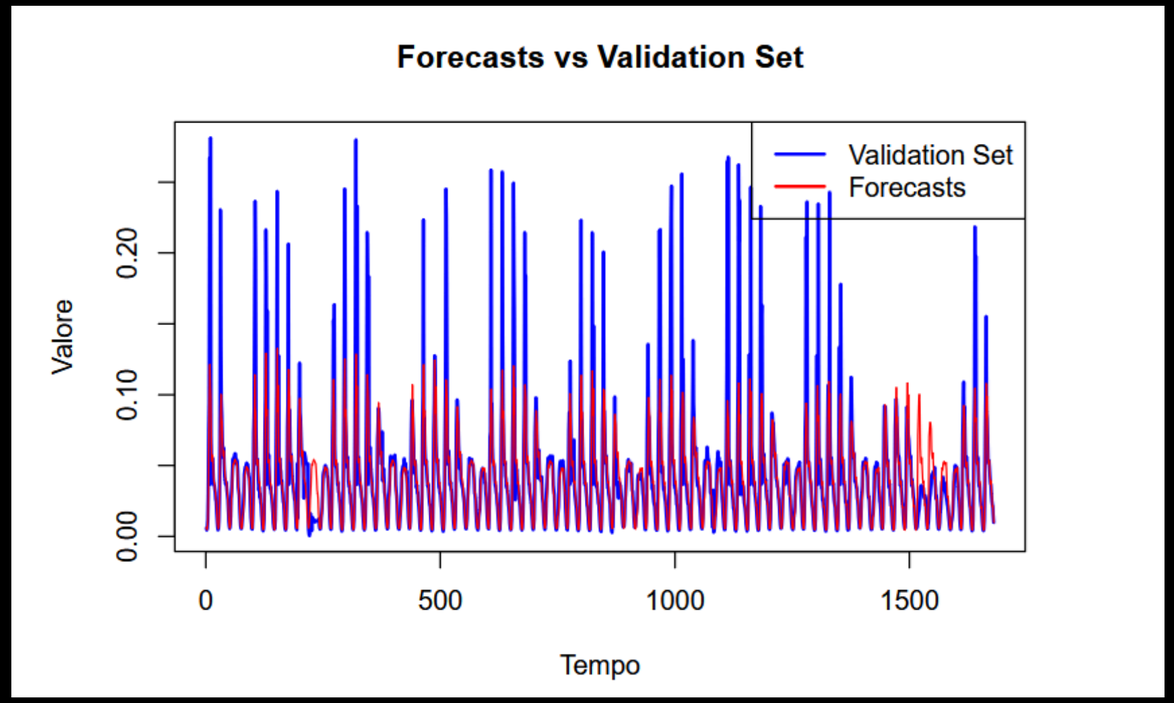


# UCM MODELS

The approach involved starting with a simpler baseline model and progressively enriching it with additional components. The preprocessing phase is the same of the previous category.

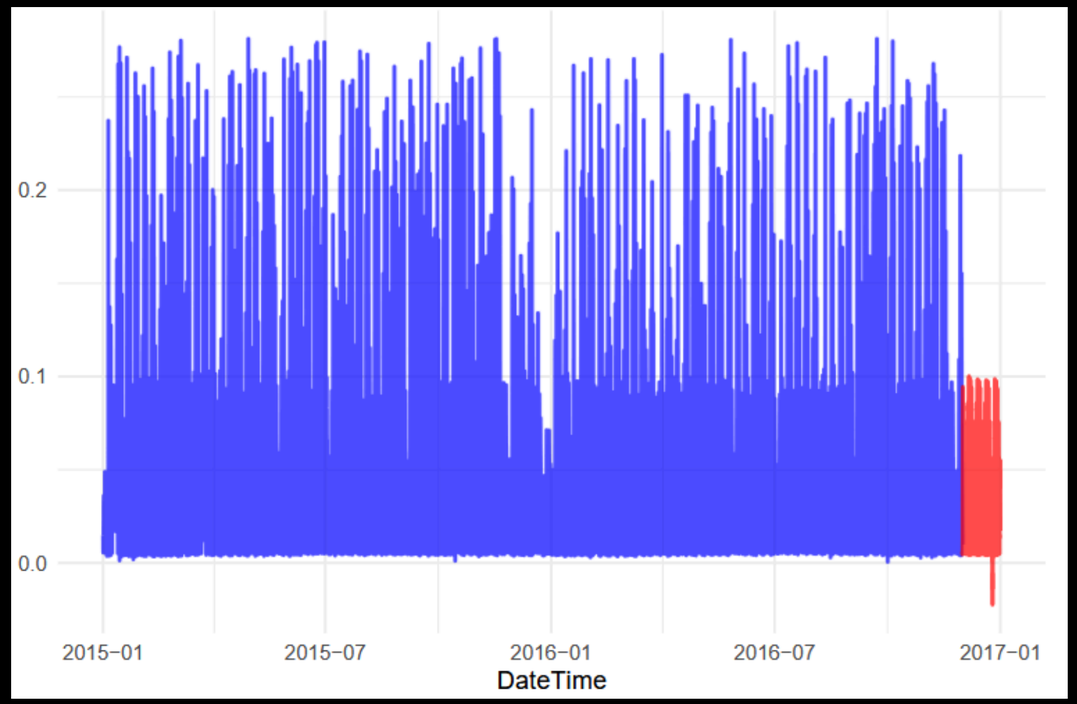
| Model | Normal | With Dummies |
|---|---|---|
| LLT, seasonal dummy (7) | 0.009738 | 0.009588 |
| LLT, trigonometric seasonality (7) | 0.009792 | 0.00958 |
| LLT, seasonal dummy (7) and one trigonometric harmonic (365) | 0.01 | NONE |
| LLT, seasonal dummy (7) and 2 harmonics (365) | 0.009514 | 0.0095 |
| LLT, seasonal dummy (7) and 3 harmonics (365) | 0.0145 | NONE |

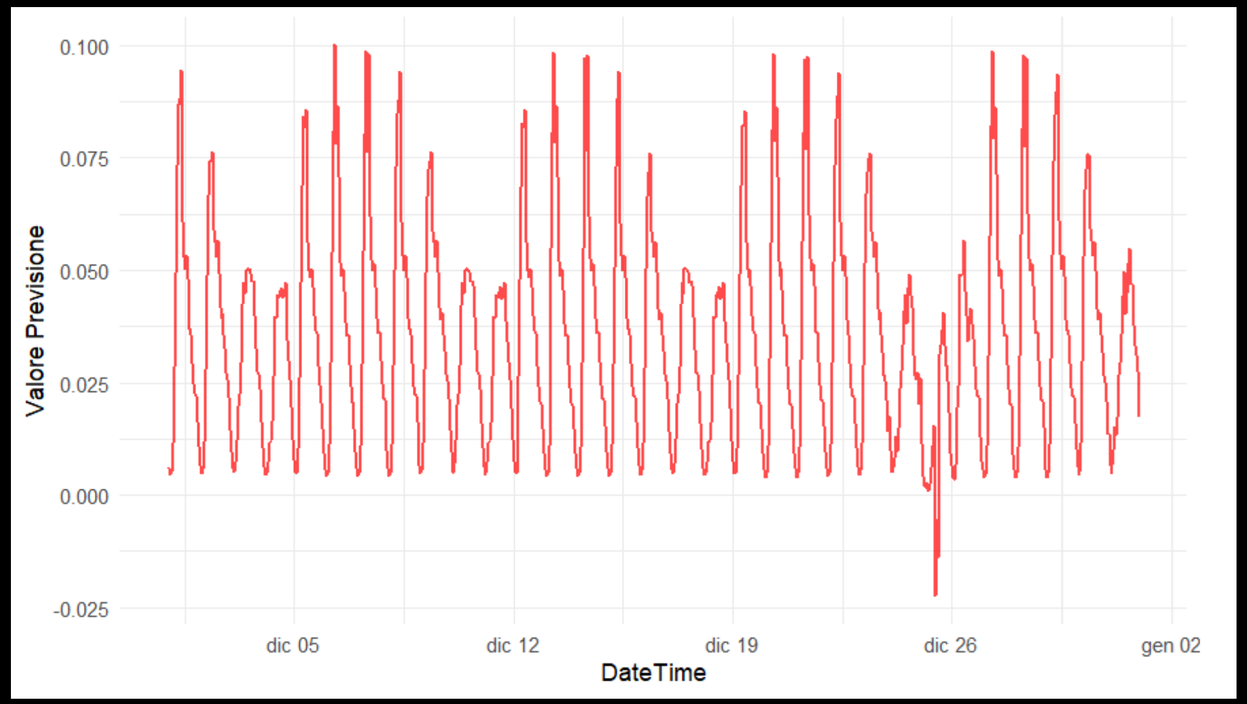



The models seem to achieve the same level of performance. Dummies have a postive influence.

# LLT, S.DUMMY(7) , 2 H. (365)

The best-performing model was the LLT with seasonal dummy (7) and 2 harmonics (365) , enhanced with dummies. It was retrained on the full dataset,using data up to 2016-11-30 to forecast December 2016. Plot shows the actual and predicted segments of the time series.

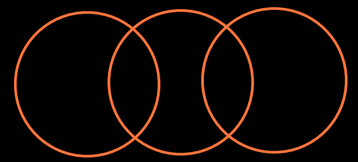


Machine learning models, typically, require a well structured featues to learn adequately. These variables are added to the dataset to capture temporal dependencies and enhance the model's ability to predict time series dynamics:

- **Lag-X**: containing the value of the previous lag
- **Lag-X7**: containing the value of the week before
- **Diff**: difference between the current value and the previous lag
- **DayOfWeek**: ordinal numeric value referring to the day of the week (from 1 to 7)
- **DayOfYear**: ordinal numeric value referring to the day of the year (from 1 to 365)
- **Diff-X7**: difference between the current value and the previous week's value

# FEATURE ENGINEER FOR ML

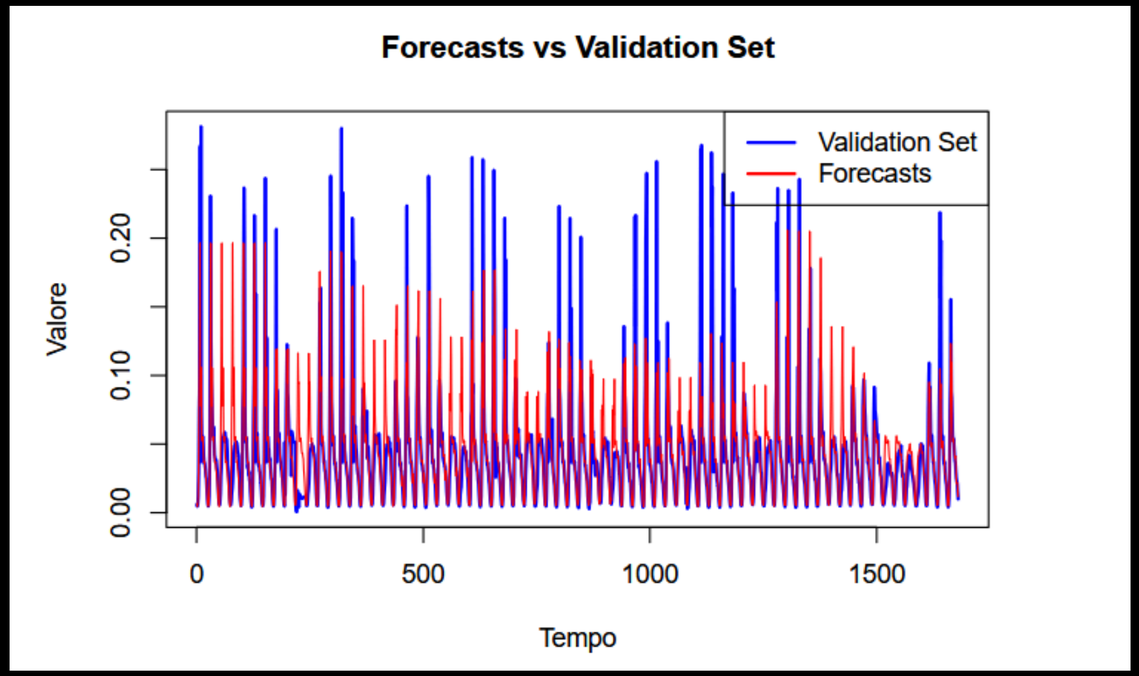


# ML MODELS

These models are applied and subsequently enhanced by integrating one-hot encoded dummy variables representing specific dates and holidays, as outlined in the preprocessing section and in feature engineer part.

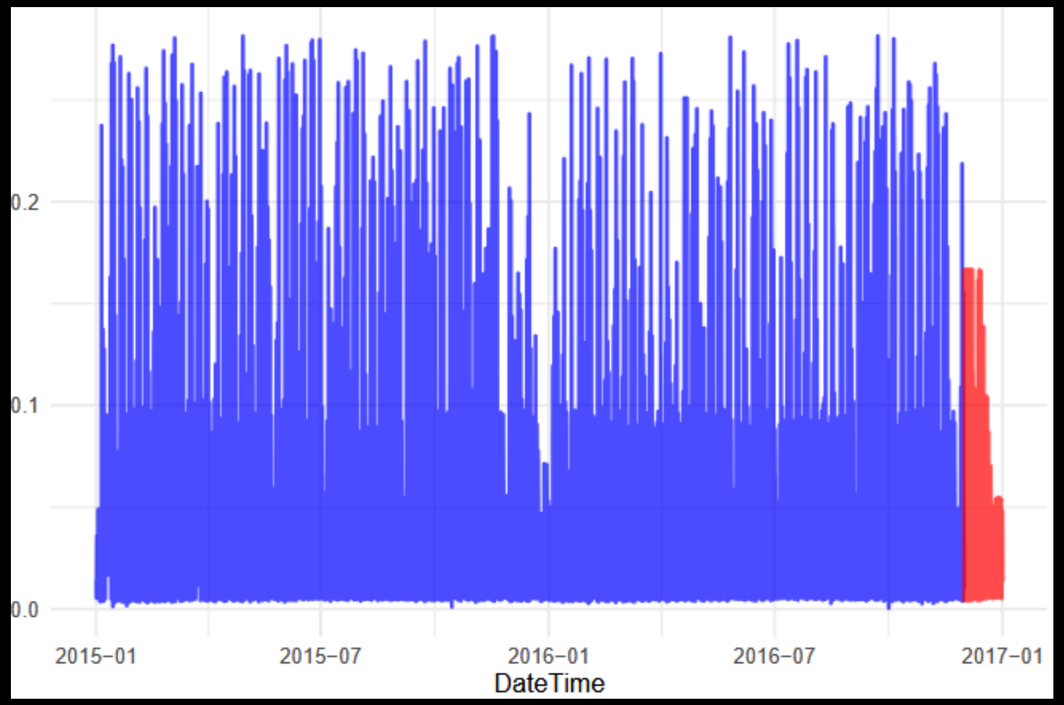
| Model | Normal | With Dummies |
|---|---|---|
| Random Forest | 0.019579 | 0.01922 |
| XGBoost | 0.024698 | 0.01931 |
| KNN | 0.013555 | 0.01353 |

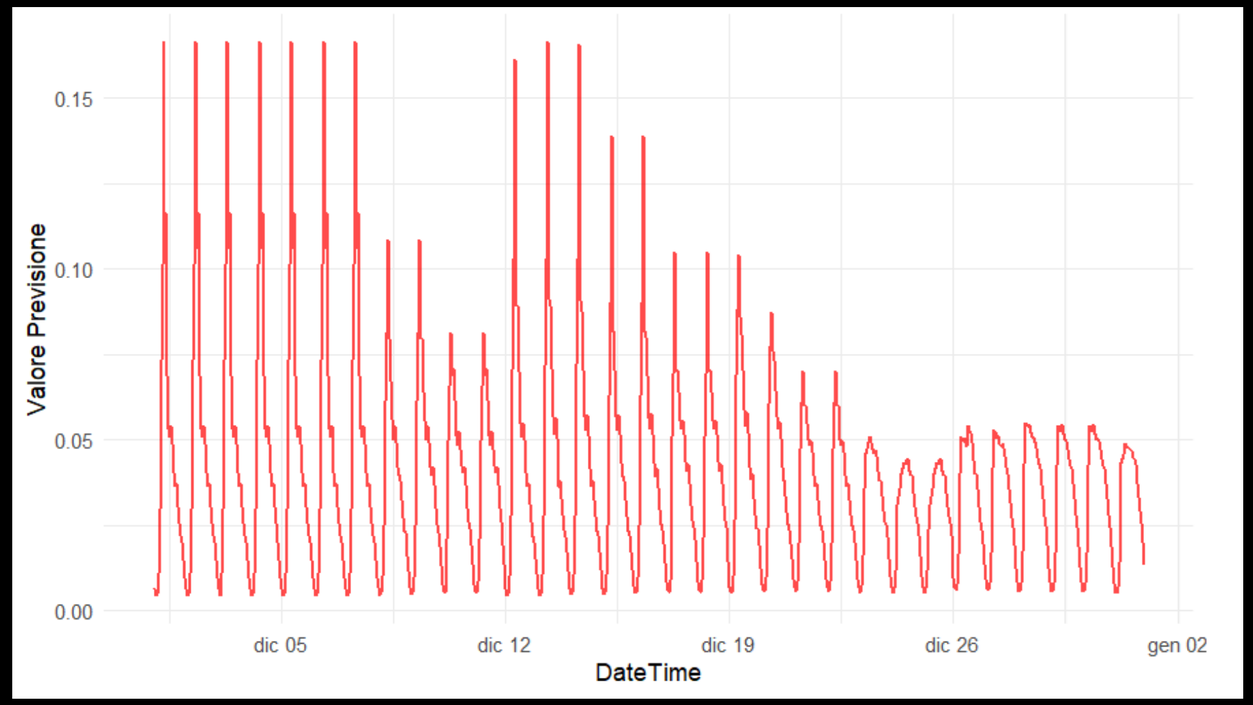
The models demonstrate similar levels of performance, where the use of dummies slightly improves accuracy.



# K-NEAREST NEIGHBORS (KNN)

The best predictive performance was achieved by the K-Nearest Neighbors (KNN) model, enhanced with dummies. The model was retrained on the complete dataset up to 2016-11-30 to predict December 2016. A plot displays the actual and predicted values for the final portion of the time series.series.



# CONCLUSION

- The models show generally a good level of performance, capturing the overall trend and seasonal patterns effectively.

- A common limitation is their inability to predict the peaks accurately, indicating a lack of granularity for abrupt changes or extreme values.

# CALL TO ACTION

- Future improvements could focus on hybrid models or peak detection-specific models, as well as incorporating additional features to better address these variations.

- Despite these limitations, the models provide valuable insights and offer a solid foundation for further refinement.

# THANKS FOR ATTENTION

Andrea D’Amicis 869008