University of St. Gallen

School of Management, Economics, Law, Social Sciences,

International Affairs and Computer Science

Data Science Fundamentals Project

Predicting Nitrogen Dioxide (NO_2) Concentrations along streets in Zurich

Nino Caduff	22-607-329
Frederick E. Safian	22-619-902
Jonah Nguyen	22-619-761

Research Report

University of St. Gallen

3,580: Workshop Fundamentals of Data Science

Prof. Dr. Lyudmila Grigoryeva, Jonathan Chassot, Hannah Bushoff

10 December 2023

Table of Contents

List of Figures

Assistance Information

The text of the paper was fully produced by us and thereafter artificial intelligence (ChatGPT 4) was used to improve the coherence and readability of the text.

1 Introduction

This paper explores the critical issue of Nitrogen Dioxide (NO_2) pollution in Zurich, a city known for its lively urban atmosphere. Despite its beauty, Zurich faces a hidden challenge: The harmful effects of NO_2 on health and the environment. Our research taps into predictive analytics to forecast NO_2 levels in the city, a crucial step in balancing urban development with environmental sustainability.

Air pollution, particularly NO_2, is a silent threat to public health and the ecological balance of cities. In Zurich, where modern urban life coexists with environmental awareness, addressing NO_2 pollution is essential. Our research is a blend of environmental science and data science, offering a comprehensive approach to a widespread urban issue.

We detail our process of collecting and preparing data, and the task of merging different datasets, like traffic data, weather conditions, and urban factors, to create a predictive model. This model aims to be both precise and enlightening. Our work contributes to a future where data-driven strategies lead to healthier, more sustainable cities.

2 Preprocessing

In this chapter, we delve into the crucial first steps of our journey towards understanding and predicting NO_2 levels in Zurich. This stage is foundational, as the quality and structure of our data directly influence the effectiveness of our predictive model.

2.1 Datasets and First Steps

Our research on NO_2 levels in Zurich centers on air quality data from 1983, with a focus on 2012-2023 records, aligned with traffic data availability. Utilizing Zurich's city database, we merged annual air quality and traffic datasets into a single dataframe, streamlined by a custom function for accuracy and efficiency.

We enriched our study with meteorological data, forming a "merged_meteo" dataset that combines air quality, traffic, and weather information. To reflect time-related variations, such as increased summer traffic, we integrated dummy variables for seasonal and weekday changes.

Additionally, we included data on urban green spaces, examining trees and greenspaces near measurement stations at various distances. This aspect of our study helps assess the impact of urban greenery on NO_2 levels, offering a comprehensive view of environmental factors in Zurich's air quality.

2.2 Target Variable NO2_tomorrow

Central to our predictive model is its ability to forecast future conditions. Accordingly, we've designated the NO_2 concentration of the following day, aptly named “NO2_tomorrow,” as our target variable. This approach enables us to model how today’s environmental and urban factors influence tomorrow's air quality. To set this up, we take the NO_2 data column, shift it one row upwards, and create a new column for this target variable. This shift effectively aligns each day's data with the NO_2 concentration of the next day. Such alignment allows us to utilize the current day's comprehensive data to predict the subsequent day’s NO_2 levels, encapsulating a forward-looking perspective in our analysis.

2.3 Addressing Seasonality in the Data

While incorporating dummy variables for season and weekday into our model, we also recognized the importance of understanding the role of seasonality in our NO_2 data. To gain insights into seasonal effects, we applied seasonal decomposition to each of the three streets separately. We took weekly averages for illustration purposes. Figure 1 illustrates these seasonal trends for Rosengartenstrasse.

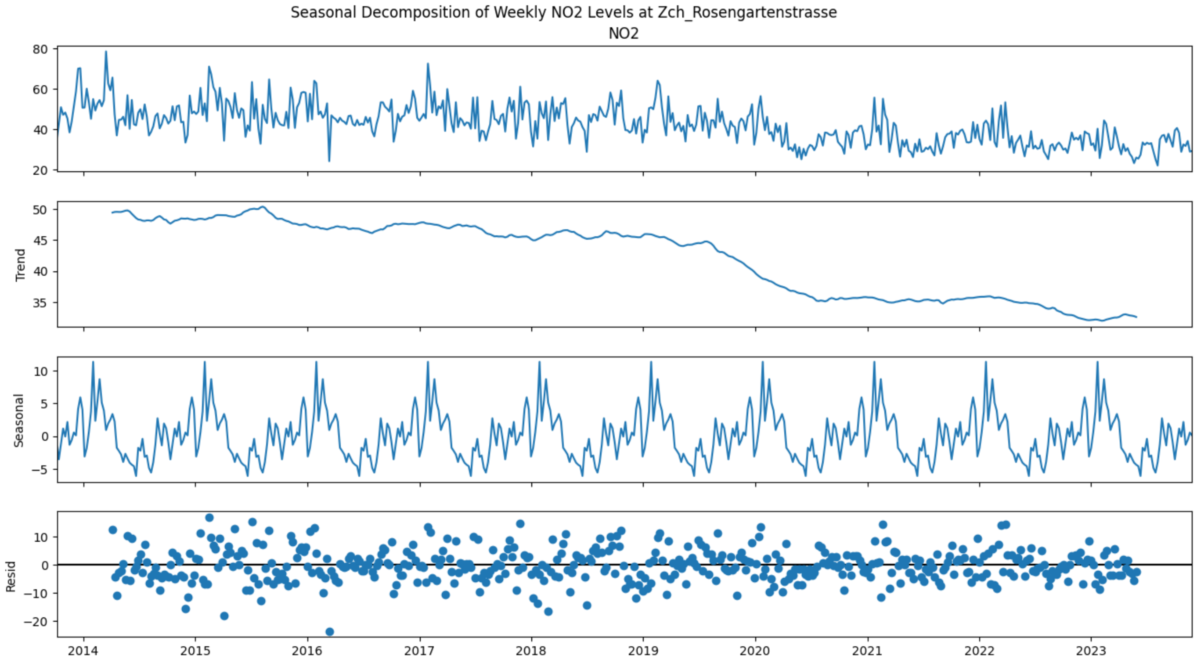

Figure 1: Seasonality at Rosengartenstrasse

Our analysis revealed a distinct pattern of seasonality as well as a downward trend across all three streets. Instead of deseasoning and detrending our data, we discovered that by having our target variable with a one-day time lag as a feature, these seasonal variations will be incorporated directly into our model.

2.4 Splitting our Dataset

To ensure the robustness and accuracy of our predictive model, we have divided our dataset into three subsets. Given the importance of temporal factors such as seasonality, as highlighted in chapter 2.3, our approach to splitting the dataset is not random but rather chronological, also known as Out-of-sample evaluation. This decision is crucial to maintain the integrity of the temporal patterns and trends within our data:

- **Training Set (2012 to 2019)**: This is the primary dataset used to train our model. It provides the foundation on which the model learns and adapts to the patterns within our data.
- **Validation Set (2020 to 2021)**: This subset plays a critical role in evaluating the model's performance. It's used to fine-tune the model's hyperparameters, ensuring optimal functioning.
- **Test Set (2022 to 2023)**: Reserved for the final phase of our analysis, the test set comprises entirely unseen data. It serves as the ultimate test of our model's predictive powers, simulating real-life scenarios to gauge its accuracy and reliability.

2.5 Handling Missing Data

In the world of data science, encountering datasets without any missing values (NaNs) is a rarity, and our project was no exception. We encountered a significant number of NaNs, necessitating a strategic approach to address these gaps:

2.5.1 Target Variable

For the target variable 'NO2_tomorrow', we opted against imputation. If a target value was missing, we chose to exclude that record, as imputing values for the variable we aim to predict could skew our model's learning process.

2.5.2 Features

In our dataset, we identified a total of 20 feature columns with missing values. Some of them revealed even missing values of more than 50%. While the straightforward option might be to drop these columns, we chose to retain them to preserve potentially valuable insights they could offer.

First, we checked for rows with a high number of NaNs. We noticed that certain rows were missing all meteorological data. Mean or median imputation in such scenarios would have introduced bias by creating uniform, non-informative rows. Rather, we employed a targeted imputation strategy, using data from surrounding days for these gaps. This approach was feasible as these rows were not positioned at the start or end of the data for a particular street, allowing us to use the surrounding rows. However, we found that for many rows, adjacent data points were also missing, primarily due to gaps in the target variable. These rows we dropped. For the rest of the NaNs, which were not due to a complete failure of the measurement station, we decided to go with mean or median imputation. Therefore, we visualized the distribution of each column through boxplots and histograms, as seen in Figure 2.

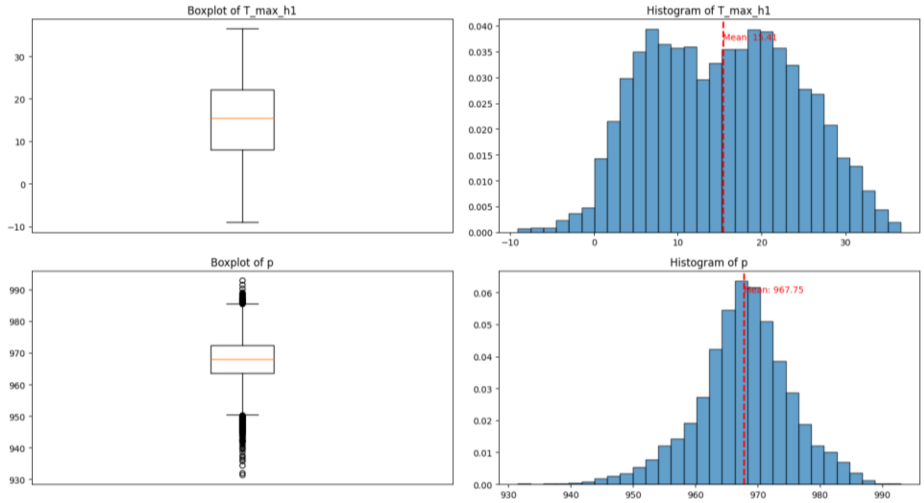

Figure 2: Boxplot and Histogram of two of the 20 Features with NaNs

For columns with little or no outliers, such as T_max_h1, we imputed missing values using the mean. This approach is suitable for these columns due to their more uniform distribution. For columns with significant outliers, including p, we used the median for imputation, as the median is more robust to outliers. In addition, we decided to apply a logarithmic transformation to these features. This transformation helps reducing the distortionary effect of outliers in our model, as many of the features we use, including the main feature NO_2, show many.

Our attention then turned to the TotalDailyTraffic column. Unlike others, it did not contain NaN values but had instances of zero values (due to the way we summed the hourly values up). Its distribution was quite special, as can be seen in Figure 3.

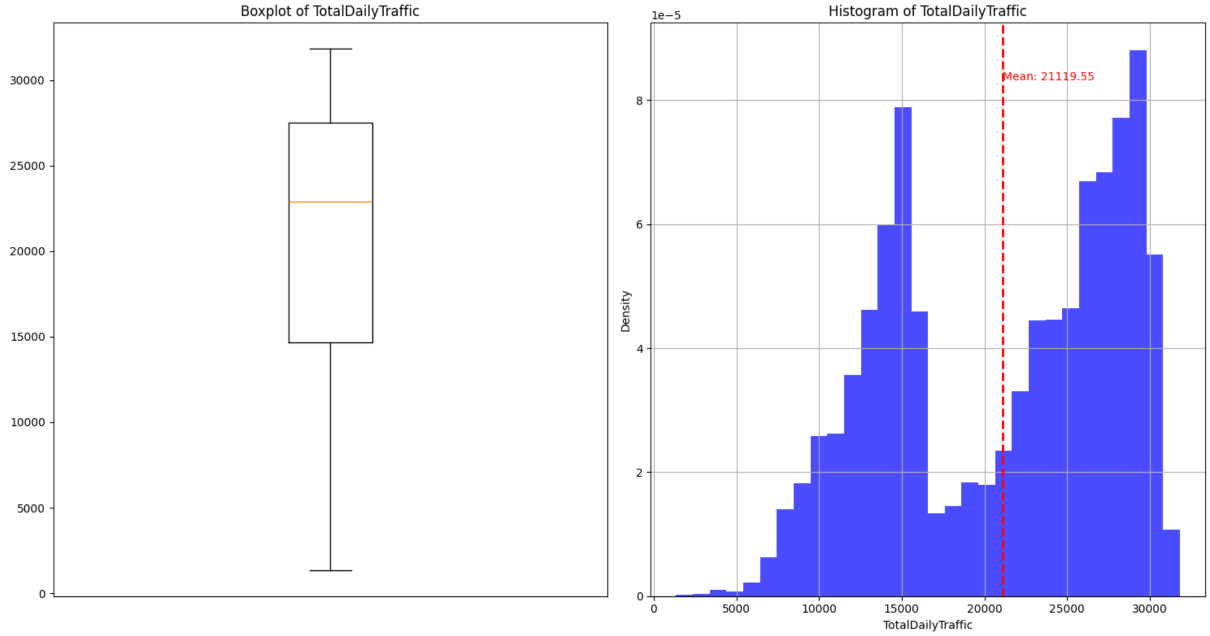

Figure 3: Boxplot and Histogram for TotalDailyTraffic

We also tried to plot the traffic separately for each day of the week to possibly find more even distributions. However, we still found the same distribution with high values around the edges. Using mean or median imputation here would have destroyed this feature's distribution. Hence, we decided to simply drop the 450 rows containing NaNs in the traffic column. As before, to reduce the distortionary effect of this distribution, we applied a log transformation for the traffic column.

2.6 Feature Engineering

In this section, we focus on refining our dataset through feature engineering, with particular attention to reducing multicollinearity and implementing polynomial features. These steps are crucial in enhancing the predictive accuracy of our model by providing a clearer, more relevant set of inputs for analysis.

2.6.1 Reducing Multicollinearity

To address the issue of multicollinearity in our regression model, which can significantly impact its performance, we employed a strategy to handle the “dummy variable trap” associated with categorical features like seasons and weekdays. The trap occurs due to high correlation among dummy variables. For instance, in the case of the 'Season' feature, we initially created dummy variables for each of the four seasons. However, since the presence of three automatically implies the absence of the fourth, having all four represented is redundant and contributes to multicollinearity. By removing one season's dummy variable, we effectively reduced this issue. The same principle was applied to the weekday variables.

To systematically identify and address multicollinearity in the numerical features, we utilized a correlation matrix. This matrix, depicted in Figure 4, helped us spot highly correlated features.

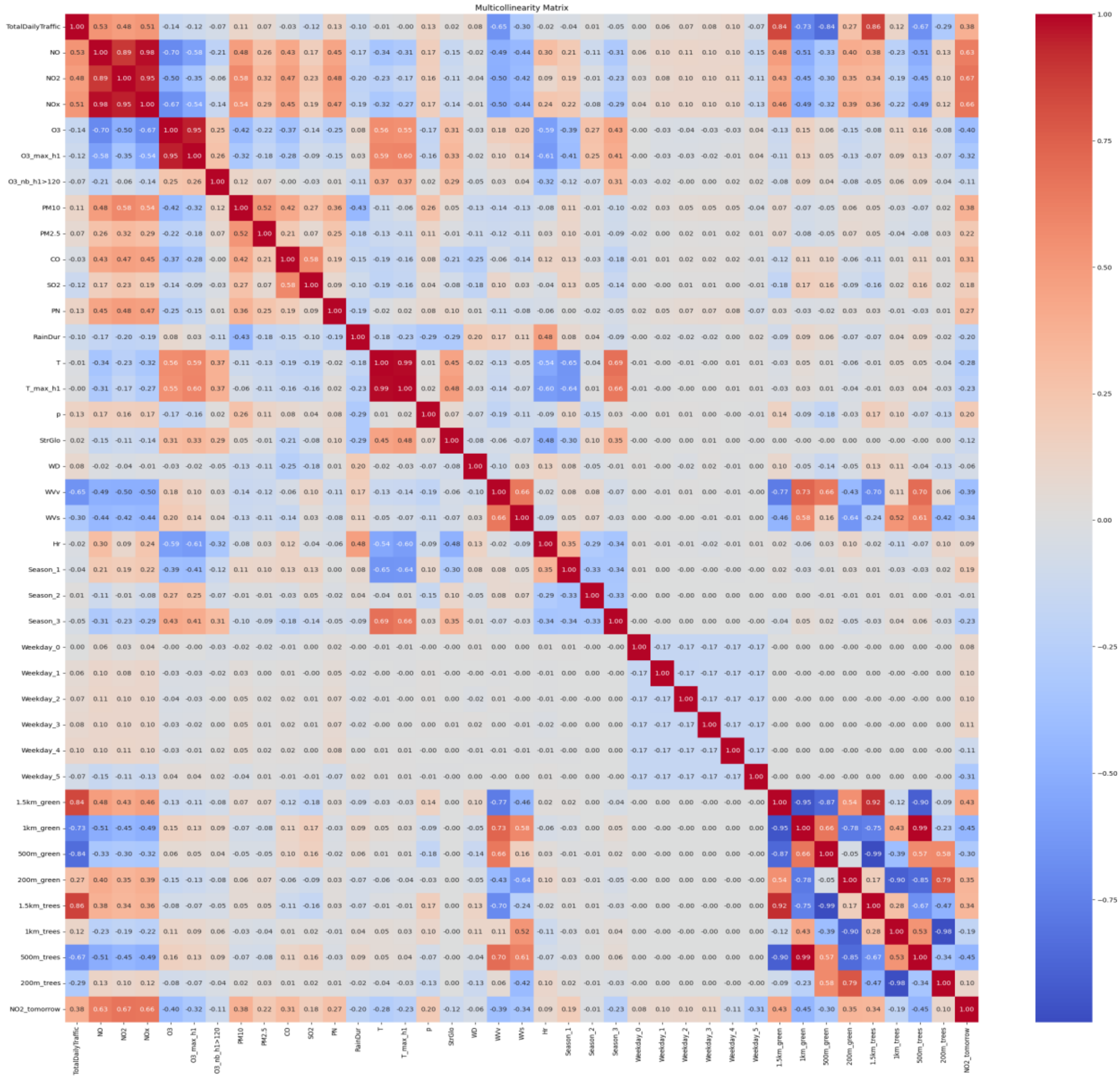

Figure 4: Multicollinearity Matrix

In the matrix, areas with dark blue or dark red indicate strong correlations; dark blue signifies an inverse relationship between features, while dark red indicates a direct relationship. We observed several such correlations in the bottom right section of the matrix. To enhance our model's performance, we decided to remove one of each pair of features with an absolute correlation higher than 0.9.

2.6.2 Polynomials

The rationale behind incorporating polynomial features into our model becomes apparent when examining Figure 5.

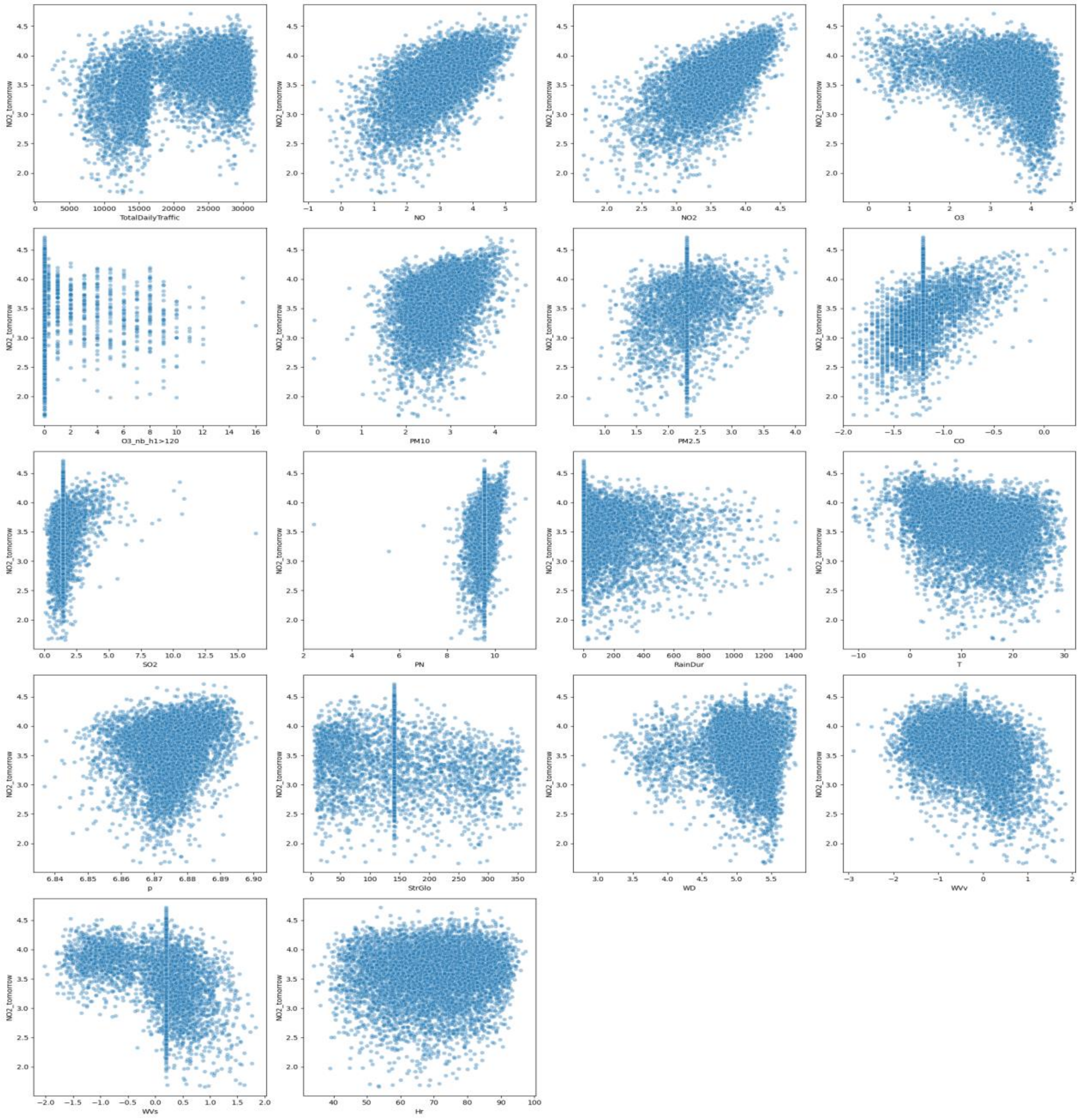

Figure 5: Relationships between NO2_tomorrow and Main Dataset Featuresmuss

Our analysis of the dataset indicates that most features do not exhibit a clear linear relationship with the target variable 'NO2_tomorrow', with the notable exception of NO_2 itself. To better capture these complex, non-linear patterns, we have introduced polynomial features. These are particularly useful for modeling the subtleties and nuances in our data that linear terms might miss. Concerning the columns with a high percentage of missing values (PM2.5, CO, SO2, PN, StrGlo, WVs) our analysis suggests a correlation with the target variable. This observation validates our decision to retain these features rather than discarding them. Building on this, we expanded our dataset to include polynomial features of the second and third degree, aiming to deepen our model's understanding of the data's underlying structure. As a result, we have developed and will utilize multiple versions of our dataset for training, validation, and testing:

- Standard datasets: train_set, validation_set, test_set
- Reduced feature datasets: train_reduced, validation_reduced, test_reduced
- Polynomial feature datasets (degree 2): train_poly_2, validation_poly_2, test_poly_2
- Reduced feature polynomial datasets (degree 2): train_reduced_poly_2, validation_reduced_poly_2, test_reduced_poly_2
- Polynomial feature datasets (degree 3): train_poly_3, validation_poly_3, test_poly_3
- Reduced feature polynomial datasets (degree 3): train_reduced_poly_3, validation_reduced_poly_3, test_reduced_poly_3

This diversified approach allows us to explore different aspects and dynamics of the data, enhancing the robustness and accuracy of our predictive model.

3 Model Testing

In this section, we evaluate the predictive performance of our models using two distinct algorithms: Linear Regression and Random Forests. Each model brings its own set of strengths and characteristics to the task of predicting NO_2 levels for our dataset.

3.1 Linear Regression Model

3.1.1 Process

For all the different dataset pairs, we first standardize them using the MinMaxScaler. This helps reducing the effect of outliers and different scales on our model by scaling everything to be between 0 and 1. Thereby, to avoid future leakage, we fit the standardizer only on the training set and then use this to standardize the validation and test set. We also experimented with other standardizers, but this one worked the best, thus proving our hypothesis.

We then apply Lasso to select the most important features. Having to many features needlessly overcomplicates a model and can lead to overfitting, especially in the case of polynomials. Lasso thus does the selection for us. To find the optimal regularization parameter, we utilize cross validation (LassoCV).

After having selected the most important features, linear regression is fitted to the training sets. Then the model’s performance on the validation and test set is evaluated based on MSE, R2, and adjusted R2. Using the adjusted R2 here makes sense, as the number of features varies greatly across all different dataset pairs.

3.1.2 Results

The following table depicts the results (on validation and test set) of our linear regression for all dataset pairs.

Dataset Pair(train/test)	**MSE(validation/test)**	**R2(validation/test)**	**Adjusted R2(validation/test)**
train_set/test_set	0.09/0.10	0.56/0.55	0.55/0.55
train_reduced/test_reduced	0.09/0.10	0.59/0.57	0.58/0.57
train_poly_2/test_poly_2	0.08/0.09	0.62/0.61	0.61/0.61
train_reduced_poly_2/test_reduced_poly_2	0.09/0.10	0.59/0.56	0.58/0.56
train_poly_3/test_poly_3	0.10/0.11	0.56/0.52	0.54/0.52
train_reduced_poly_3/test_reduced_poly_3	0.10/0.11	0.56/0.52	0.54/0.52

We observe that the best model is the one with second degree polynomials and without having kicked out the features with high multicollinearity. Figure 6 depicts the performance of the model.

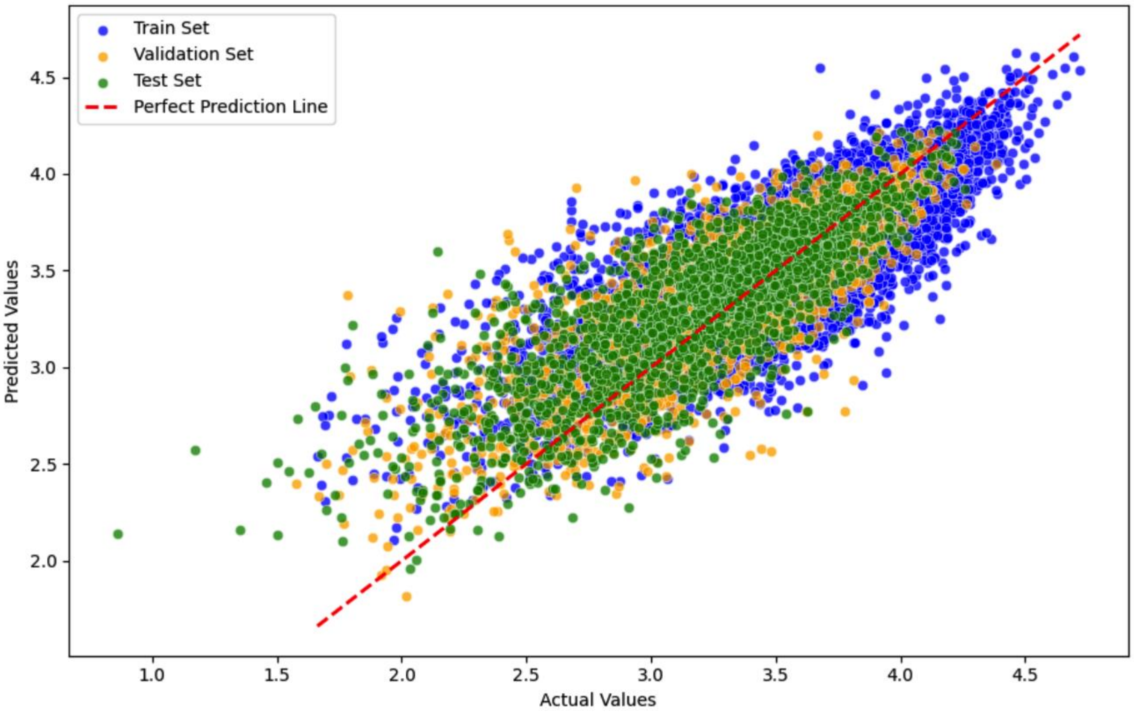

Figure 6: Linear Regression - Actual vs Predicted Values

When checking the performance of our selected model on the set we trained it with, train_poly_2, we discovered the following results: Train MSE: 0.07, Train R2 Score: 0.66. This is slightly better than the performance on the validation and test set, but not significantly. This implies that our model is generalizing pretty well. However, we thought that by additionally applying L2 regularization (Ridge) next to Lasso, we might be able to reduce the difference between the training and validation/test set even more. This hypothesis was proven wrong: When we included Ridge in our code, we obtained a slightly worse result than with just linear regression and Lasso.

In order to put our model's performance into context, we compared it to a naïve model. Namely, for our naïve model, we predicted tomorrows NO_2 values simply by taking today's value. Figure 7 shows the performance of our model compared to the naïve one.

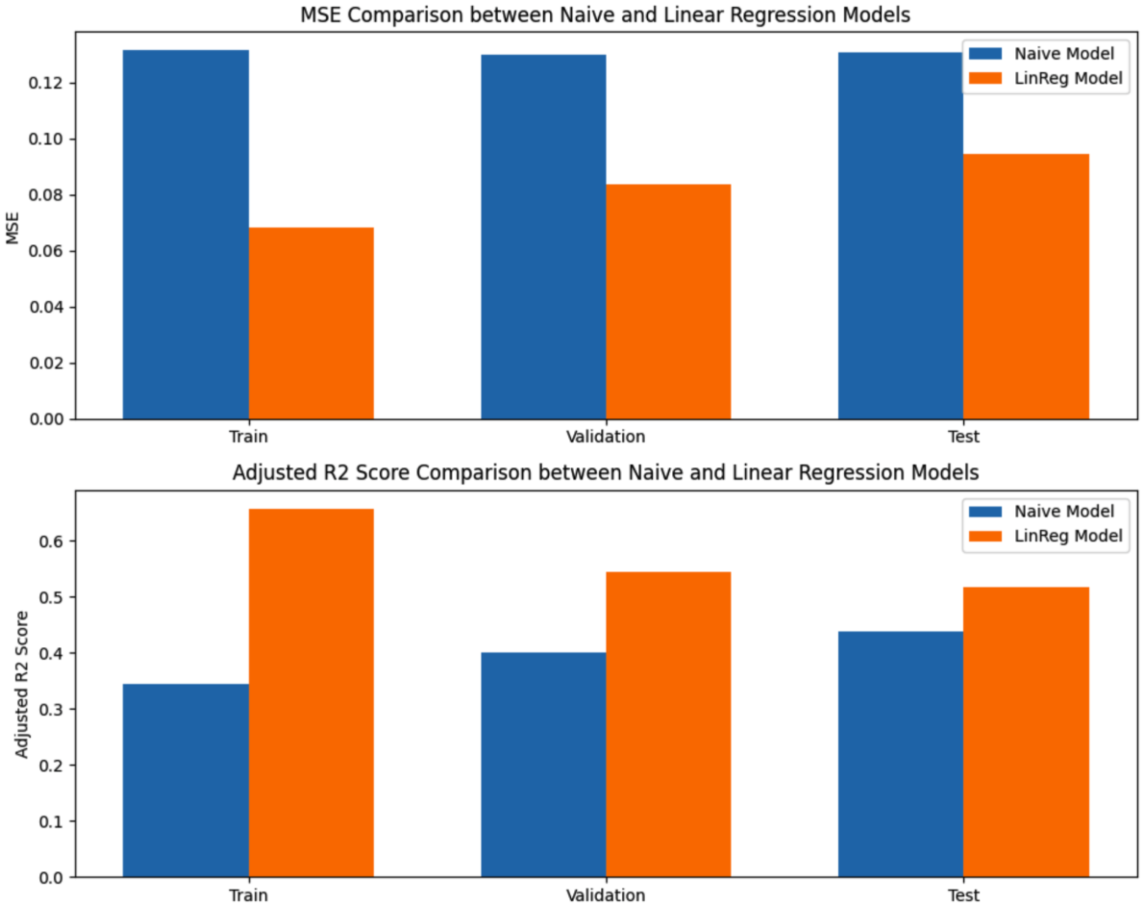

Figure 7: Comparison of Naive and Linear Regression Model

We see that our linear regression model outperforms the naive model by 17% in R2 score, thus explaining 17% more of the variance in the model. The MSE is also significantly better.

3.2 Random Forest

For our linear regression model, we were able to apply a lot of different techniques to obtain a better result. The same does not hold for the random forest regressor. Namely, a random forest does not benefit from standardization/normalization techniques. Furthermore, it can capture non-linear relationships, hence polynomials also do not help. We still wanted to include a second model (next to our linear regression model) in our project. Hence, we went for the random forest. To receive an optimal result, we first set up a random grid of parameters, which we then ran through a random search. This helps us in narrowing down the range of each hyperparameter, i.e., shows us where we need to concentrate our search. We then use the results of our random search to set up a parameter grid for our grid search. The grid search then gives us the best fit, i.e., the one which maximizes our performance. The randomized search grid is set up as follows:

- Number of Trees (n_estimators): 100, 200, 300, 400, 500
- Max number of levels in each decision tree (max_depth): None, 10, 20, 30, 40, 50
- Min number of data points placed in a node before the node is split (min_samples_split): 2, 5, 10, 15, 20
- Min number of data points allowed in a leaf node (min_samples_leaf): 1, 2, 4, 6, 8
- Max number of features considered for splitting a node (max_features): 'sqrt', 'log2'

Our grid search then revealed the following values to be optimal: n_estimators = 500, max_depth = 50, min_samples_split = 5, min_samples_leaf = 2, max_features = 'sqrt'.

The results, however, were not so satisfying. They were as follows:

- Validation Set - MSE: 0.10, R2 Score: 0.54
- Test Set - MSE: 0.12, R2 Score: 0.50

We find that our random forest performs worse than our linear regression model. Apparently, our data is not too complex. Furthermore, the Random Forest seems to overfit more. Nonetheless, our focus was on the linear regression model. The random forest regressor is included to showcase the performance of another model in comparison to the linear regression model.

Declaration of Authorship

We hereby declare that

- we have written this group paper independently;
- we have written this group paper using only the aids specified in the assistance information;
- all parts of the group paper produced with the help of aids have been precisely declared;
- we have mentioned all sources used and cited them correctly according to established academic citation rules;
- we have acquired all immaterial rights to any materials we may have used, such as images or graphics, or that these materials were created by us;
- the topic, the group paper or parts of it have not already been the object of any work or examination of another course, unless this has been expressly agreed with the faculty member in advance and is stated as such in the group paper;
- we are aware of the legal provisions regarding the publication and dissemination of parts or the entire group paper and that we comply with them accordingly;
- we are aware that our group paper can be electronically checked for plagiarism and for third-party authorship of human or technical origin and that we hereby grant the University of St. Gallen the copyright according to the Examination Regulations as far as it is necessary for the administrative actions;
- we are aware that the University will prosecute a violation of this Declaration of Authorship and that disciplinary as well as criminal consequences may result, which may lead to expulsion from the University or to the withdrawal of our titles.

By uploading this group paper, we confirm through our conclusive action that we are submitting the Declaration of Authorship, that we have read and understood it, and that it is true.

St. Gallen, 10.12.2023

NCaduff

Nino Caduff

Frederick E. Safian

Jonah Nguyen