Hochschule für Technik und Wirtschaft Berlin
Fachbereich II
Professional IT Business and Digitalization

Masters Course Project Studies(M25)
Summer 2023

Automatize Machine Learning Processes

Yash Revannavar (s0586034), Omkar Gavali(s0585840) and Nikhil Amin (s0585844)

Automatize Machine Learning Processes

Yash Revannavar (s0586034), Omkar Gavali(s0585840) and Nikhil Amin (s0585844)

September 26, 2023

Contents

1 Introduction **3**
- 1.1 Motivation . . . 3
- 1.2 Problem Description . . . 3
- 1.3 Objectives . . . 3
- 1.4 Structure of the paper/report . . . 3

2 Literature Research **5**
- 2.1 PM10 Prediction . . . 5
- 2.2 Machine Learning . . . 5
- 2.3 Auto-ML . . . 6
- 2.4 Summary . . . 6

3 Method **7**
- 3.1 Data Exploration . . . 7
- 3.2 Windowing . . . 8
- 3.3 Cyclic Feature and Scaling . . . 8
  - 3.3.1 Cyclic Feature Representation . . . 8
  - 3.3.2 Application of Cyclic Features . . . 9
- 3.4 ML and Auto-ML Techniques . . . 10
  - 3.4.1 Tensorflow Functions . . . 10
  - 3.4.2 Autosklearn Regression . . . 10
  - 3.4.3 Pycaret Regression . . . 11
- 3.5 Evaluation Techniques . . . 11

4 Implementation **12**
- 4.1 Data Cleaning and Preparation . . . 12
- 4.2 Machine Learning Methods . . . 13
- 4.3 Auto-ML Methods . . . 13
- 4.4 Evaluation Methods . . . 14

5 Results **15**
- 5.1 Data Analysis . . . 15
- 5.2 Evaluations . . . 16

6 Conclusion **19**

1 Introduction

In today's era, Machine Learning (ML) has become instrumental in enhancing various aspects of our lives, from predicting weather and climate changes to monitoring air quality. The ability to predict air quality is especially crucial as it enables timely warnings to the public when poor air quality is anticipated. These warnings empower individuals to take precautionary measures, such as wearing masks or avoiding areas with compromised air quality. In this paper, We embark on a similar path in this study, beginning with an exploration of the fundamentals of air quality measurement.

1.1 Motivation

The determination of air quality encompasses numerous parameters, including the nature and size of pollutants, as well as chemical substances present in the air. According to WHO Global Air Quality Guidelines 2021 The Air Quality Index (AQI) is a tool to communicate air quality information to the public in a way that is easy to understand and use. It is calculated using the concentrations of six key pollutants: particulate matter (PM 10 and PM 2.5), ozone, nitrogen dioxide, sulfur dioxide, and carbon monoxide.[Org21] Our specific focus will be on PM10, which represents particulate matter with a diameter of less than or equal to 10 μm, without considering the chemical composition. We have chosen PM10 due to its potential to deeply penetrate the respiratory system and its common sources, such as dust from unsealed roads, smoke from fires, vehicular emissions, and industrial processes.[Org21]

1.2 Problem Description

Predicting PM10 levels accurately using Machine Learning (ML) or Automated Machine Learning (AutoML) poses a challenging task. First and foremost, the development of predictive models suitable for PM10 forecasting requires a sound understanding of ML/AutoML techniques. Choosing appropriate libraries that effectively implement these techniques is crucial for model development and evaluation. Furthermore, the availability of a substantial amount of quality data is indispensable for training, validating, and fine-tuning the models. Thus, to predict PM10 levels accurately, we need proficient models, adept libraries, and an abundance of high-quality data to ensure optimal performance and reliability of the forecasting process.

1.3 Objectives

The primary aim of this paper is to rigorously test and evaluate various Auto-ML tools for predicting the Air Quality Index within a specific dataset. To achieve this objective, we utilize an Air Quality Index dataset from the North Rhine-Westphalia region, focusing on identifying the most efficient Auto-ML approach tailored to this dataset. The central question guiding our exploration is whether Auto-ML surpasses traditional ML methods in this context.

1.4 Structure of the paper/report

The structure of this paper is organized into six main sections. In the introduction section (Section 1), we present the motivation, problem description, objectives, and an overview of the paper's methodology. The subsequent section (Section 2) delves into the literature research, discussing PM10 prediction, machine learning, and Auto-ML techniques. Section

3 elaborates on our methodology, encompassing data exploration, windowing, cyclic feature engineering, and an array of ML and Auto-ML techniques(Section 3). In Section 4, we detail the implementation process, including data cleaning, machine learning, Auto-ML methods, and evaluation techniques(Section 4). Section 5 offers insights into the results, focusing on data analysis and model evaluations(Section 5). Finally, in Section 6, we conclude the paper, summarizing key findings and potential future directions(Section 6).

2 Literature Research

Over the recent years, there has been increasing concern surrounding the severe air pollution issue that has garnered global recognition, primarily due to its grave effects on people's health and general well-being [Cha+20]. Consequently, there has been a surge of interest among researchers in the examination of air pollution, specifically for the purpose of predicting and forecasting PM10 concentrations in the Earth's atmosphere. Over the past two decades, there has been a substantial increase in scholarly articles addressing this subject matter, rendering it more accessible to a broader spectrum of researchers [CCH19].In this section, we review the works of few of the researchers in this field and share some similarities with our work in this project.

2.1 PM10 Prediction

A prototype stochastic NN model based on neural networks developed by the Italian National Agency for New Technologies, Energy and Sustainable Economic Development (ENEA) called EnviNNet was used to predict PM10 concentration at a regulatory monitoring site in Phoenix, Arizona. This model used a three-layer MLP (multi-layer perceptrons) network. The input vector for this model consisted of Pollution data, Meteorological data and Statistical and descriptive indicators such as wind velocity and direction, working days or holidays etc. The model was trained using a selected set of appropriately preprocessed historic time series data to predict PM10 in the hindcasting mode[Fer+12]. A combination of a Multilayer Perceptron Neural Network and Clustering algorithms have also been used in the forecasting model to predict the average concentration of PM10 for a period of 24 hours in Salamanca, Mexico. Clustering algorithms used were K–means and FCM (Fuzzy c–Means algorithm). The clustering algorithms were implemented in order to find relationships between PM10 and the Meteorological data[Cor+15].

2.2 Machine Learning

There are several parameters that play an important role in model performance but they are mainly categorised as meteorological, pollutant emissions, traffic, and others. Meteorological parameters refer to the variables that characterise atmospheric chemistry. The important Meteorological variables include wind speed, wind direction, and relative humidity which can have a significant role in the dispersion and concentration of several air pollutants including PM10. Meteorological and pollutant emissions data are the most commonly used predictors by researchers in this field[CCH19]. A popular method in handling missing data is by list-wise or pair-wise deletion of predictors[CCH19]. There are a few noteworthy state-of-the-art imputation methods such as Bayesian compressive sensing (BCS) imputation methods [Wil+18], known data regression (KDR) method [FAF15], single imputation [PB06], vector autoregressive model-imputation (VAR-IM) algorithm [BW18].

Long Short-Term Memory (LSTM) has become the state-of-the-art model for several machine learning problems. The prediction and forecasting of PM10 can be performed using the Enhanced spatial, temporal sequence-improved Sparse Auto Encoder with Deep Learning (EISAE-DL), which uses a variation of LSTM. However, a known problem was identified in LSTM that was used in EISAE-DL i.e. learning of a long-term dependent sequence of the training dataset. An alternative method called Transfer learning (TL) in a Stacked Bidirectional and Unidirectional LSTM can be used to solve the learning issue in LSTM for long-term dependencies. EISAE-DL with TL and modified LSTM model is named EISAEDeep Transfer Learning (EISAE-DTL) [PAS22]. The splitting of data into the subsets, namely, the

training, and test sets is an important step in the development of ANN models. The reason why the splitting of data is an essential modelling step is that it prevents the issue where the model would memorise the data in the training subset, but will not be able to generalise to new unseen data. The training subset is used for computing the gradient and calibrating the network weights and biases. The testing subset is used to determine the generalisation ability of the developed model which means that the error from the testing subset is used to compare the predictive performance of different models [CCH19].

2.3 Auto-ML

We employed automated machine learning (Auto-ML) tools in our study, namely Autosklearn[1] and Pycaret[2], specifically tailored for regression tasks. Furthermore, in the machine learning segment of our study, we leveraged Tensorflow functions to augment the predictiveness of our models.

PyCaret is an open-source machine-learning library designed to simplify and accelerate machine-learning workflows. Our decision to employ PyCaret in this research paper is primarily attributed to its automation capabilities and its inherent simplicity, resulting in substantial reductions in the time and effort typically needed for the regression models. Furthermore, PyCaret provided us with an extensive list of regression algorithms, facilitating the identification of the most apt model for a given regression task. Auto-sklearn is a state-of-the-art hybrid AutoML method which outperforms the previous state of the art in AutoML by automatically taking into account past performance on similar datasets, and by constructing ensembles from the models evaluated during the optimization. Its search strategy of pipelines uses bayesian optimisation, meta-learning and ensemble learning. Auto-Sklearn automates the tasks in the machine learning pipeline such as data preprocessing, feature preprocessing, hyperparameter optimization, model selection and evaluation [Feu+15]. The addition of Auto-sklearn was essential in our study since it facilitated the fine tuning of hyperparameters in our machine learning models, aiming for maximum accuracy. This automated procedure of hyperparameter optimization played a pivotal role in fully utilising the latent capabilities inherent in the selected models.

2.4 Summary

The literature research has provided insights into various models employed for PM10 concentration prediction and forecasting. Notable models discussed include the stochastic neural network-based EnviNNet, which utilizes a three-layer MLP (multi-layer perceptrons) network. Additionally, the combination of a Multilayer Perceptron Neural Network and Clustering algorithms, specifically K-means and FCM (Fuzzy c-Means), was utilized for forecasting in Salamanca, Mexico. The Enhanced spatial, temporal sequence-improved Sparse Auto Encoder with Deep Learning (EISAE-DL) introduced LSTM for prediction, and a modified LSTM model with Transfer Learning (TL) was proposed as EISAEDeep Transfer Learning (EISAE-DTL). The comparative performance of Auto-ML tools like Autosklearn and PyCaret can vary depending on the specific dataset, problem, and requirements of a given machine learning task. Therefore, there is no definitive answer regarding which tool consistently performs better, as their performance may differ across different scenarios.

[1]Here is the link for the website https://automl.github.io/auto-sklearn/master/
[2]Here is the link for the website https://pycaret.org/

3 Method

In the following section, we describe the methods employed to predict the Air Quality Index (AQI) using a combination of traditional machine learning (ML) techniques and automated machine learning (Auto-ML) approaches. We begin by presenting an overview of our dataset in the section "Data Exploration," where we describe data pre-processing techniques such as windowing, cyclic feature extraction, and scaling to prepare the data for predictive modeling. Subsequently, we delve into the ML and Auto-ML techniques employed, including TensorFlow functions such as LSTM and Dropout for enhancing predictive capabilities, as well as Autosklearn and Pycaret for efficient regression modelling. Finally, we introduce the evaluation metrics utilized to assess the performance of our predictive models, providing a comprehensive analysis of the AQI forecasting results.

3.1 Data Exploration

The air quality data used in this research project was obtained through a collaboration with our University Professor and a third-party data Provider. To predict PM10 we will use a datasets from NRW. This was provided by UM. It comprises three distinct data sets

Main Database (Main Db) The Main Db serves as a repository of records from various monitoring sites, each with a unique set of attributes, including identifiers and other relevant information. These monitoring sites play a pivotal role in our analysis, serving as the spatial basis for our air quality predictions.

Chemical Database (Chemical Db) The Chemical Db contains comprehensive information regarding the chemical composition of the pollutants present in the air. These details are essential for understanding the nature and composition of the airborne contaminants, which in turn influence air quality.

Measurement Database (Measurement Db) The Measurement Db encompasses critical data related to air quality measurements. This data set includes parameters like actual PM10 readings, raw values, site identifiers, measurement time and date, measurement value and other pertinent information. The actual PM readings are of particular significance as they directly reflect the concentration of particulate matter with a diameter of 10μm or less, a key factor in air quality assessment. Our primary areas of interest within these data sets include monitoring sites, actual PM10 readings,measurement time and date, and the specific type of pollutant, which in our case, pertains to PM10.

Meteostat In addition to the air quality data, we recognized the importance of incorporating meteorological information to enhance the accuracy of our AQI predictions. We obtained essential meteorological data, including temperature, relative humidity, precipitation, and atmospheric pressure, from Metostat. These meteorological features are crucial as they contribute significantly to the predictive capabilities of our AQI model.We dropped a few parameters due to insufficient data for chosen time period, such as wind direction, wind speed, peak wind gust time, pressure, total sunshine, cloud cover, and snow. The successful fusion of air quality and meteorological data forms the foundation of our comprehensive approach to predicting AQI, as described in subsequent sections.

3.2 Windowing

Using the windowing operator we can convert a time series problem into a machine learning problem. This allows us to use all the additional tools and techniques to train and optimize models.

Feature engineering is an essential step in the pipelines used for many machine learning tasks, including time-series forecasting. Although existing Auto-ML approaches partly automate feature engineering, they do not support specialised approaches for applications on time-series data such as multi-step forecasting. Multi-step forecasting is the task of predicting a sequence of values in a time-series. [Laz20]

3.3 Cyclic Feature and Scaling

Cyclic features refer to variables that exhibit cyclical patterns, such as hours of the day, days of the week, or months in a year. These cyclic features pose a unique challenge in machine learning since they involve cyclical relationships between values. For instance, hour 23 and hour 0 on a clock are closer in proximity than, say, hour 23 and hour 12. To effectively utilize such features, it is crucial to transform them into a representation that preserves this cyclical information.CYl

3.3.1 Cyclic Feature Representation

One of the most effective methods to convert cyclic features into a suitable representation is through the use of trigonometric functions, particularly sine and cosine. These functions are periodic and repeat their values every 2π radians, making them well-suited for modelling cyclic behaviour.

In the Figure 1 , we see that the hours 0 and 11.5 obtain very similar values after the sine transformation. So how can we differentiate them? To fully code the information of the hour, we must use the sine and cosine trigonometric transformations together.

To transform cyclic variables into (x, y) coordinates using sine and cosine functions, we must first normalize them to the range of 0 to 2π radians. This normalization process ensures that the cyclical variable maps consistently to the unit circle.

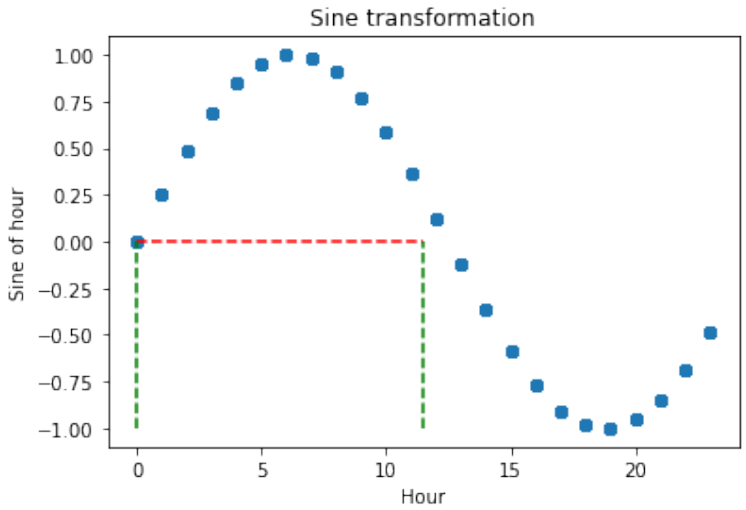

Figure 1: Transformation of each hour of one day (24h) into there Sine value representations.

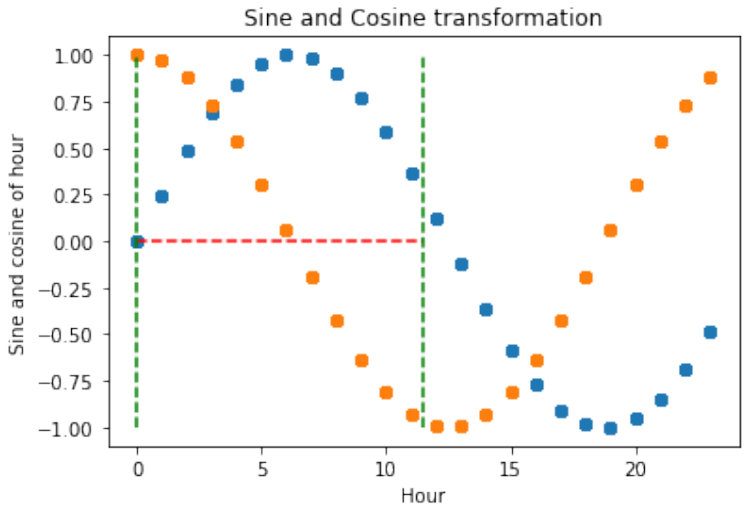

Figure 2: Transformation of each hour of one day (24h) into there Sine and Cosine value representations.

The application of both sine and cosine functions is instrumental in breaking the symmetry inherent in cyclical features. By incorporating the cosine function, which is out of phase with the sine function, a unique encoding is assigned to each cyclic value. In figure 2 we can see that , after transformation, hour 0 corresponds to the values of sine 0 and cosine 1, setting it apart from hour 11.5, which is associated with sine 0 and cosine -1. In essence, this combined approach enables us to differentiate between all observations within the original cyclic variable. In the Figure 3 we can vizualise the (x, y) circle coordinates generated by the sine and cosine features.

The method of cyclic feature transformation has significant implications in various domains, including time-series analysis, natural language processing, and environmental monitoring. In our study, we apply this technique to the Air Quality Index (AQI) data set, specifically to handle cyclic features related to time, such as hourly variations and seasonal changes.

By preserving the cyclical nature of time-related features, we aim to enhance the accuracy of our machine-learning models in predicting AQI. The application of cyclic features allows us to capture subtle temporal patterns that might be overlooked when treating time as a linear variable.

3.3.2 Application of Cyclic Features

The method of cyclic feature transformation has significant implications in various domains, including time-series analysis, natural language processing, and environmental monitoring. In our study, we apply this technique to the Air Quality Index (AQI) data set, specifically to handle cyclic features related to time, such as hourly variations and seasonal changes.

By preserving the cyclical nature of time-related features, we aim to enhance the accuracy of our machine-learning models in predicting AQI. The application of cyclic features allows us to capture subtle temporal patterns that might be overlooked when treating time as a linear variable.

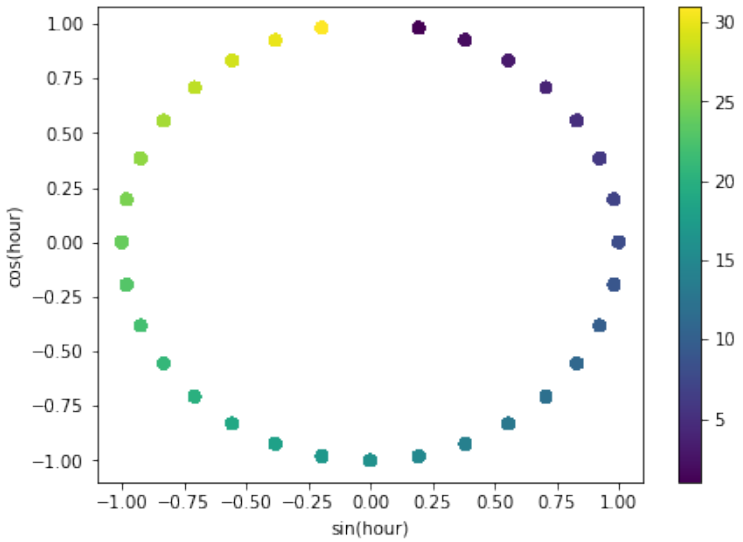

Figure 3: Illustration of Cyclic Feature Transformation.

In summary, the Cyclic Feature and Scaling method is a powerful technique for handling cyclical data, which is prevalent in various domains, including air quality prediction. By applying trigonometric functions and normalizing cyclical variables to 2π radians, we ensure that the inherent cyclic patterns are effectively preserved, enabling machine learning models to make more accurate predictions.

In the subsequent sections, we will provide a detailed account of our experimental setup, including the data set used, the machine learning algorithms employed, and the evaluation metrics considered. Our goal is to determine the most effective approach for predicting the Air Quality Index by leveraging the power of cyclic feature representation.

3.4 ML and Auto-ML Techniques

In this section, we detail the machine learning (ML) and automated machine learning (Auto-ML) techniques employed in our study. We also describe the evaluation metrics utilized to assess the performance of these techniques.

We employed a combination of traditional machine learning (ML) techniques and automated machine learning (Auto-ML) approaches to predict the Air Quality Index (AQI).

For the ML component, we utilized Tensorflow functions, specifically Long Short-Term Memory (LSTM) networks, and Dropout regularization to enhance the predictive capabilities. The LSTM network helps capture long-term dependencies in the data, which is crucial for time-series prediction tasks like AQI forecasting. Dropout regularization was employed to prevent overfitting and improve generalization.

Additionally, we utilized the Auto-ML tools Autosklearn and Pycaret for regression tasks.

3.4.1 Tensorflow Functions

Tensorflow is a widely used open-source machine learning framework. We leveraged Tensorflow functions, including LSTM and Dropout, to construct and train our predictive models.

3.4.2 Autosklearn Regression

Autosklearn is an efficient and automated machine learning tool that automates the process of selecting and optimizing machine learning models for regression tasks.

3.4.3 Pycaret Regression

Pycaret[3] is another versatile Auto-ML library that provides an extensive range of functions to streamline the model selection, hyperparameter tuning, and evaluation processes for regression tasks.

3.5 Evaluation Techniques

To assess the performance of our predictive models, we utilized the following evaluation metrics:

- Mean Absolute Error (MAE)
- Root Mean Squared Error (RMSE)
- Mean Absolute Percentage Error (MAPE)
- Coefficient of Determination (R^2)

These metrics provide a comprehensive evaluation of the predictive accuracy and precision of our models in forecasting the AQI.

[3]Here is the link for the website https://pycaret.org/

4 Implementation

In the implementation phase, we delve into the heart of our research by scrutinizing and comparing the inner workings and outcomes of six distinct models. Among these models, two belong to the realm of machine learning, while the other four fall under the umbrella of Auto-ML. Notably, in the machine learning models, we diligently selected and prepared the necessary features to feed into our algorithms. In stark contrast, the Auto-ML models streamlined this process, sparing us from the intricacies of data cleaning and preprocessing. However, we opted to conduct these steps even in the Auto-ML context, seeking to unearth valuable insights and discern disparities in the results. This section details our methodology and the intricacies of model implementation.

4.1 Data Cleaning and Preparation

In accordance with the comprehensive data collection process outlined in Section **??**, a thorough examination and refinement of the data set is imperative to extract valuable insights. This phase encompasses a series of meticulous procedures aimed at optimizing the raw data to align with our research objectives.

The key steps involved in data cleansing and preparation include:

1. **Outlier Removal**: Due to excessive use of crackers on New Year's Eve, high pollution and anomalous data spikes occurring around New Year were identified as outliers and subsequently removed from the data set. Refer Section 5.1

2. **Table Integration**: Multiple tables were consolidated, establishing coherent relationships between site names and chemical names based on their unique identifiers.

3. **Data Segmentation**: The data set was partitioned into distinct subsets corresponding to different chemical constituents.

4. **Feature Extraction for Duisburg (PM10)**: Features relevant to PM10 levels were meticulously extracted for Duisburg and stored in a separate file named `Duisburg.csv`.

5. **Meteorological Data Extraction**: Meteorological data specific to Duisburg was obtained using MeteoStat[4], an open-source Python library dedicated to weather data retrieval.

6. **Data Integration**: The MeteoStat database and `Duisburg.csv` were seamlessly combined to enrich the data set with meteorological information.

7. **Handling Missing Values**: Instances with missing data were systematically removed from the data set.

8. **Data Splitting**: The data set, now comprising a total of 32 months, was divided into two segments: 29 months(2 years and 5 months) for training and 3 months for testing.

9. **Temporal Column Transformation**: Date-time information was disassembled into various temporal components, including the number of weeks, year, hours in the 24-hour format, day of the week, day of the year, and day of the month.

10. **Windowing Method**: To facilitate the application of time series algorithms, a windowing method was employed to structure the data. Refer Section 3.2

[4]Here is the link for the Documentation https://dev.meteostat.net/python/

11. **Cyclic Encoding**: Cyclic encoding techniques, involving sine and cosine functions, were applied to periodic features such as hours and days to account for their cyclical nature. Refer Section 3.3

4.2 Machine Learning Methods

We conducted experiments with two machine learning models, both utilizing the same architecture outlined below, but with different data sets.

- **ML 1:** For the first data set, we employed the following data processing steps: removal of missing values, temporal column transformation for DateTime information, windowing techniques, and standard scaling.
- **ML 2:** In the second data set, we introduced an additional step by incorporating cyclic encoding.

Figure 4: Model Architecture used for training the (standard) machine learning models (ML1 and ML2).

```
Model: "sequential"
_________________________________________________________________
Layer (type) Output Shape Param #
=================================================================
lstm_1 (LSTM) (None, 5, 128) 72192
_________________________________________________________________
lstm_2 (LSTM) (None, 5, 64) 49408
_________________________________________________________________
lstm_3 (LSTM) (None, 32) 12416
_________________________________________________________________
dropout_1 (Dropout) (None, 32) 0
_________________________________________________________________
dense_1 (Dense) (None, 1) 33
=================================================================
Total params: 134,049
Trainable params: 134,049
Non−trainable params: 0
```

4.3 Auto-ML Methods

Our research also explored the data set using four different Auto-ML models, each with distinct configurations:

1. **Auto-ML 1:** The data set was pre-processed by removing missing values and applying temporal column transformation to DateTime data. No windowing or scaling techniques were used. The model was trained using Auto Sklearn Regressor.
2. **Auto-ML 2:** Similar to Auto-ML 1, this data set was pre-processed by removing missing values and applying temporal column transformation, with no windowing or scaling techniques. However, the model was trained using Pycaret Regressor, and Pycaret's Auto-ML function selected the Extra Trees Regressor model.

3. **Auto-ML 3:** The data set underwent pre-processing steps, including removing missing values, temporal column transformation, and scaling techniques such as normal scaling and cyclic encoding. Windowing was not applied. The model was trained using Pycaret Regressor, and the Auto-ML function chose the Extra Trees Regressor model.

4. **Auto-ML 4:** In this case, the data set was processed by removing missing values only; no temporal column transformation, windowing, or scaling techniques were applied. The model was trained using Pycaret Regressor, and the Auto-ML function selected the Random Forest Regressor model.

4.4 Evaluation Methods

To assess the performance of all six models, comprising two manual machine learning models and four Auto-ML models, we conducted a comprehensive evaluation using four key metrics:

1. **Mean Absolute Error (MAE)**: MAE provides a measure of the average absolute difference between the predicted values and the actual target values. It offers insights into the model's overall accuracy[ONR19].

2. **Root Mean Square Error (RMSE)**: RMSE quantifies the square root of the average of squared differences between the predicted and actual values. It emphasizes the model's ability to capture errors effectively[ONR19].

3. **Mean Absolute Percentage Error (MAPE)**: MAPE calculates the average percentage difference between predicted and actual values. This metric is particularly useful when evaluating errors as a percentage of the actual values[ONR19].

4. **R-Square (R2)**: R2, also known as the coefficient of determination, assesses the proportion of variance in the dependent variable that is predictable from the independent variables. It provides an indication of how well the model fits the data[ONR19].

These evaluation techniques collectively provide a comprehensive view of the performance of both manual machine learning and Auto-ML models, aiding in the comparison and selection of the most suitable model for the given data set.

5 Results

In our research, we conducted a comprehensive evaluation of six models refer in Section 4.2 and 4.3, comprising two Manual Machine Learning and four Auto-ML models, using well-established evaluation techniques, including Mean Absolute Error (MAE), Root Mean Square Error (RMSE), Mean Absolute Percentage Error (MAPE), and R-Square (R2). These techniques, detailed in Section 4.4, offer a robust framework for assessing the models' performance.

Our analysis reveals insightful trends and distinctions in the results, providing valuable insights into the suitability of each model for our data set 5.1. In the following sections, we present our findings through a series of charts and tables, which visually depict the comparative performance of these models. These visual aids serve to enhance the understanding of the results, enabling us to make informed decisions regarding the selection of the most appropriate model for our specific data set.

5.1 Data Analysis

In the process of analyzing the PM10 air quality data for Duisburg-Walsum, two distinct data sets were considered: one with outliers and another with outliers removed. The Figure 5 portrays the original data set, which exhibited notable spikes in pollution levels, particularly around New Year's Eve. However, these anomalous data spikes, primarily attributed to the excessive use of fireworks and crackers during the holiday, were identified as outliers and subsequently excluded from the data set. The second Figure, 6 represents the refined data set devoid of these irregular observations. This pre-processing step ensures that the analysis and subsequent conclusions are based on a more reliable and representative data set, enabling a more accurate assessment of PM10 air quality trends in Duisburg-Walsum.

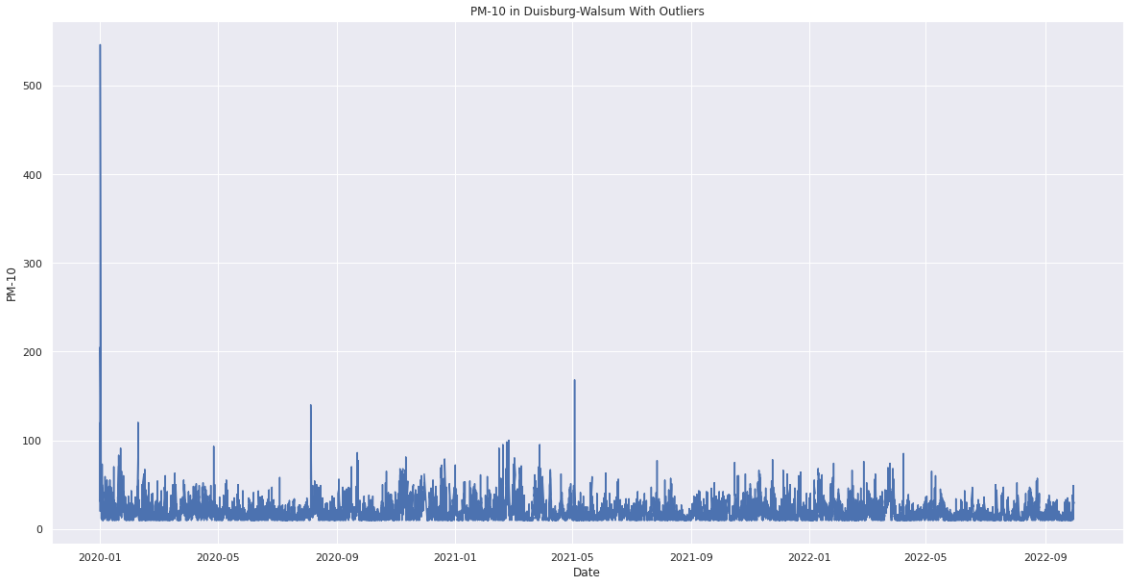

Figure 5: Concentration of the air pollutant PM10 [g/m^3] at the measurement station Duisburg-Walsum (Germany) from 01.01.2020 to 31.09.2022. The outlines are not removed.

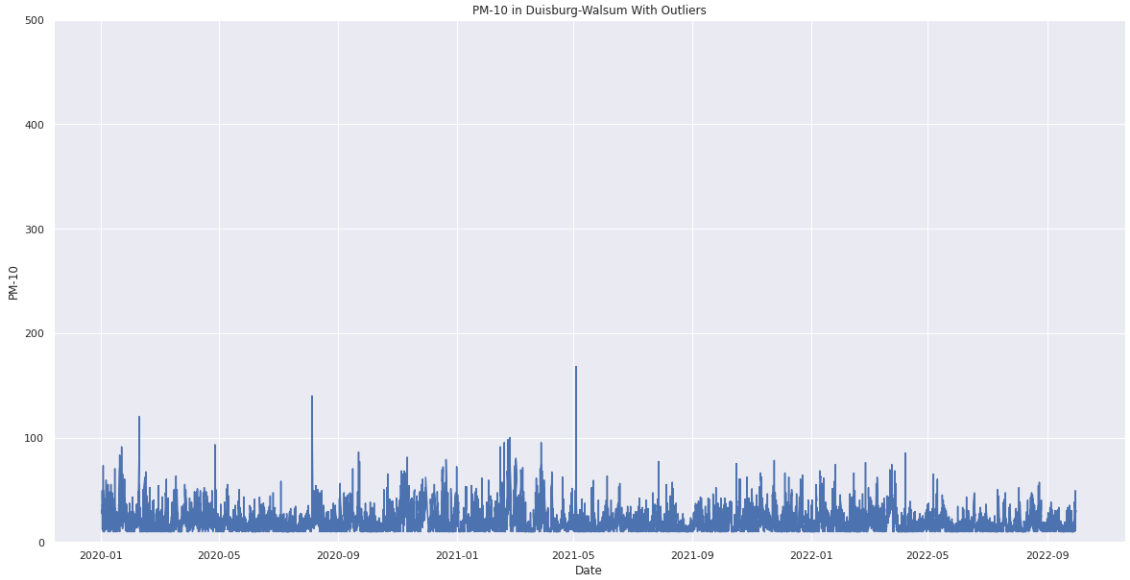

Figure 6: Concentration of the air pollutant PM10 [g/m^3] at the measurement station Duisburg-Walsum (Germany) from 01.01.2020 to 31.09.2022.The outlines are removed.

5.2 Evaluations

The performance of various models in predicting PM-10 air quality levels for Duisburg-Walsum is summarized in Table 1. Among the models evaluated, Auto-ML 02 exhibited superior performance with the lowest Mean Absolute Error (MAE) of 3.61, Root Mean Square Error (RMSE) of 5.55, and Mean Absolute Percentage Error (MAPE) of 0.1946. Notably, Auto-ML 02 achieved a substantial R-squared (R2) value of 0.7046, signifying a strong fit of the model to the data.

However, it is important to note that in the context of R2, higher values indicate a better fit, while for MAE, RMSE, and MAPE, lower values are desirable. Therefore, while Auto-ML 02 excelled in most metrics, it's essential to consider the specific evaluation criteria and application requirements. For a comprehensive assessment, the combination of these metrics provides a nuanced view of model performance.

Table 1: Summary of Model Performance

Model Name	**MAE**	**RMSE**	**MAPE**	**R2**
ML 1	4.00	32.80	0.25	0.22
ML 2	3.78	26.49	0.24	0.37
Auto-ML	4.98	43.42	0.31	-0.02
Auto-ML 02	3.61	5.55	0.1946	0.7046
Auto-ML 03	3.71	5.59	1.9504	6.902
Auto-ML 04	5.89	8.54	0.335	0.291

To visually assess the performance of different models, refer to the bar graphs in Figure 7 (MAE), Figure 8 (MAPE), Figure 9 (RMSE), and Figure 10 (R2). In these figures, the Auto-ML 02 model is highlighted with green bars, while the other models are represented in blue.

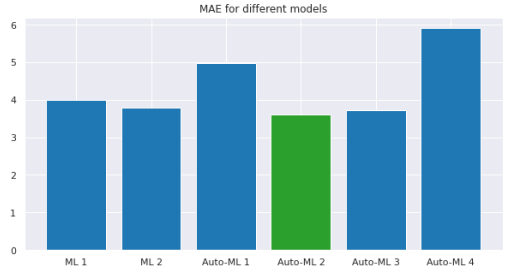

Figure 7: Bars represent the MAE score of the ML and Auto-ML models. Green bar represents the Auto-ML 2 model which has performed well overall. Note: Lower the value the better.

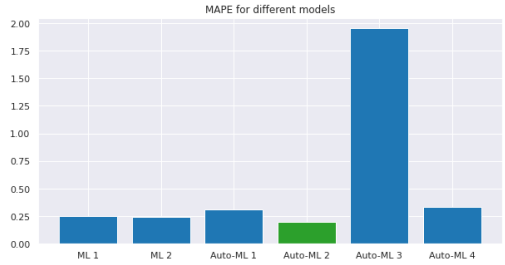

Figure 8: Bars represent the MAPE score of the ML and Auto-ML models. Green bar represents the Auto-ML 2 model which has performed well overall. Note: Lower the value the better.

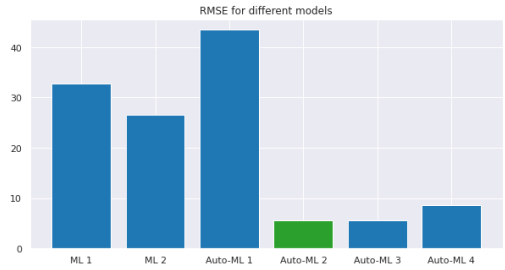

Figure 9: Bars represent the RMSE score of the ML and Auto-ML models. Green bar represents the Auto-ML 2 model which has performed well overall. Note: Lower the value the better.

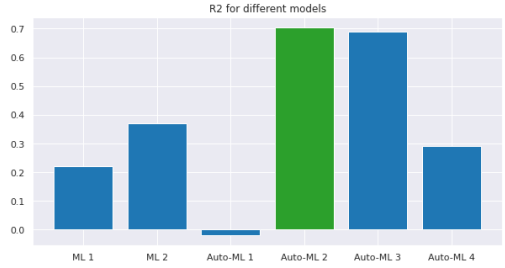

Figure 10: Bars represent the R2 score of the ML and Auto-ML models. Green bar represents the Auto-ML 2 model which has performed well overall. Note: Higher the value the better.

During our research, we uncovered a significant feature in Pycaret that sets it apart from Auto-sklearn. When configuring our data with Pycaret and initiating model training, an intriguing aspect became apparent 1. Pycaret not only trains the models but also provides real-time updates in the output window, culminating in the creation of a tabulated summary, as illustrated in Figure 11. This output table, as seen for Auto-ML 2, identifies the best-performing model according to Pycaret. It comprises essential columns, including Model Code, Model Name, Mean Absolute Error (MAE), Root Mean Square Error (RMSE), R-squared (R^2), Mean Absolute Percentage Error (MAPE), and Training Time (in seconds) and the best score is highlighted in yellow. This feature enhances model evaluation and selection, streamlining the decision-making process.

Listing 1: Setting Up Data and Comparing Models

```
s = setup(df, target='target', session_id=123)
best = s.compare_models()
```

Model Code	Model	MAE	RMSE	R2	MAPE	TT (Sec)
et	Extra Trees Regressor	3.617	5.6234	0.6918	0.1962	0.227
rf	Random Forest Regressor	4.0987	6.1647	0.6302	0.2307	0.272
lightgbm	Light Gradient Boosting Machine	5.2295	7.4118	0.4661	0.3002	166.622
knn	K Neighbors Regressor	5.242	7.7711	0.4127	0.2891	0.11
dt	Decision Tree Regressor	5.2402	8.6154	0.2781	0.2811	0.029
gbr	Gradient Boosting Regressor	6.1776	8.6793	0.2687	0.359	0.227
ridge	Ridge Regression	6.9344	9.653	0.0961	0.4053	0.251
br	Bayesian Ridge	6.9351	9.6531	0.0961	0.4054	0.253
lr	Linear Regression	6.9343	9.6539	0.0959	0.4052	2.454
omp	Orthogonal Matching Pursuit	7.1467	9.9099	0.0474	0.4201	0.263
huber	Huber Regressor	6.8142	10.0565	0.0189	0.3569	0.263
en	Elastic Net	7.3306	10.1016	0.0102	0.4316	0.256
lasso	Lasso Regression	7.3339	10.1069	0.0092	0.4318	0.262
llar	Lasso Least Angle Regression	7.365	10.1549	-0.0003	0.4333	0.263
dummy	Dummy Regressor	7.365	10.1549	-0.0003	0.4333	0.144
ada	AdaBoost Regressor	10.2812	12.3184	-0.4908	0.727	0.094
par	Passive Aggressive Regressor	27.1129	31.8619	-38.3396	1.72	0.257
lar	Least Angle Regression	505905.4882	714547.0674	-33434987659	31863.4595	0.255

Figure 11: This output table of the Auto-ML 2 represents the best performing model by Pycaret.

6 Conclusion

In conclusion, our study harnessed the power of Auto-ML with Pycaret, incorporating data preprocessing involving the handling of missing values and date splitting (refer to Section 4.1). Employing a regression model, the Extra Decision Tree exhibited commendable performance across various evaluation metrics, including Mean Absolute Error (MAE), Root Mean Square Error (RMSE), Mean Absolute Percentage Error (MAPE), and R-squared (refer to Table 1). The model's ability to accurately predict PM-10 air quality levels, evidenced by its strong performance in these metrics, underscores its effectiveness in air quality forecasting. This outcome underscores the potential of the Extra Decision Tree model as a valuable tool for air quality management and prediction.

Limitations

However, there are limitations in both of the Auto-ML libraries we tested, namely Auto-sklearn and Pycaret. In Auto-sklearn, we can only view the performance of the best model, and it lacks dedicated time series modeling capabilities, offering only regression and classification. In contrast, Pycaret provides a real-time output window displaying the performance of all tested models and allows us to print the best model. Nonetheless, its time series module has limitations, as it can only model data with a single time feature and predict a single target, which did not align with our dataset having multiple features and a single target (refer to Table 1).

Future Work

Moving forward, several avenues for future work present themselves:

1. Exploring and identifying an AutoML library capable of handling time series data with multiple features, and comparing its performance with existing models.

2. Conducting a more in-depth analysis of AutoML libraries and pollution datasets to uncover further insights and optimizations.

3. Addressing the training time challenge posed by existing models, which can take hours to execute, by investigating the use of programming languages like Mojo to reduce training times.

Acronyms

NRW North Rhine-Westphalia

AQI Air Quality Index

ML Machine Learning

Auto-ML Automated machine learning

PM10 Particulate Matter 10

Db Database

LSTM Long Short-Term Memory

MAE Mean Absolute Error

RMSE Root Mean Square Error

MAPE Mean Absolute Percentage Erro

R2 R-Square

NN Neural Network

ANN Artificial Neural Network

ENEA Energia Nucleare ed Energie Alternative

MLP MultiLayer Perceptrons

FCM Fuzzy c–Means algorithm

EISAE-DL Enhanced spatial, temporal sequence-improved Sparse Auto Encoder with Deep Learning

TL Transfer learning

EISAE-DTL Enhanced spatial, temporal sequence-improved Sparse Auto Encoder with Deep Transfer Learning

BCS Bayesian compressive sensing

KDR known data regression

VAR-IM vector autoregressive model-imputation

References

[BW18] Faraj Bashir **and** Hua-Liang Wei. "Handling missing data in multivariate time series using a vector autoregressive model-imputation (VAR-IM) algorithm". **in***Neurocomputing*: 276 (2018). Machine Learning and Data Mining Techniques for Medical Complex Data Analysis, **pages** 23–30. ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2017.03.097. URL: https://www.sciencedirect.com/science/article/pii/S0925231217315515.

[CCH19] Sheen Mclean Cabaneros, John Kaiser Calautit **and** Ben Richard Hughes. "A review of artificial neural network models for ambient air pollution prediction". **in***Environmental Modelling Software*: 119 (2019), **pages** 285–304. ISSN: 1364-8152. DOI: https://doi.org/10.1016/j.envsoft.2019.06.014. URL: https://www.sciencedirect.com/science/article/pii/S1364815218306352.

[Cha+20] Yue-Shan Chang **andothers**. "An LSTM-based aggregated model for air pollution forecasting". **in***Atmospheric Pollution Research*: 11.8 (2020), **pages** 1451–1463. ISSN: 1309-1042. DOI: https://doi.org/10.1016/j.apr.2020.05.015. URL: https://www.sciencedirect.com/science/article/pii/S1309104220301215.

[Cor+15] Maria Guadalupe Cortina–Januchs **andothers**. "Development of a model for forecasting of PM10 concentrations in Salamanca, Mexico". **in***Atmospheric Pollution Research*: 6.4 (2015), **pages** 626–634. ISSN: 1309-1042. DOI: https://doi.org/10.5094/APR.2015.071. URL: https://www.sciencedirect.com/science/article/pii/S1309104215301951.

[FAF15] A. Folch-Fortuny, F. Arteaga **and** A. Ferrer. "PCA model building with missing data: New proposals and a comparative study". **in***Chemometrics and Intelligent Laboratory Systems*: 146 (2015), **pages** 77–88. ISSN: 0169-7439. DOI: https://doi.org/10.1016/j.chemolab.2015.05.006. URL: https://www.sciencedirect.com/science/article/pii/S0169743915001197.

[Fer+12] H.J.S. Fernando **andothers**. "Forecasting PM10 in metropolitan areas: Efficacy of neural networks". **in***Environmental Pollution*: 163 (2012), **pages** 62–67. ISSN: 0269-7491. DOI: https://doi.org/10.1016/j.envpol.2011.12.018. URL: https://www.sciencedirect.com/science/article/pii/S0269749111006750.

[Feu+15] Matthias Feurer **andothers**. "Efficient and Robust Automated Machine Learning". **in***Advances in Neural Information Processing Systems*: **byeditor**C. Cortes **andothers**. **volume** 28. Curran Associates, Inc., 2015. URL: https://proceedings.neurips.cc/paper_files/paper/2015/file/11d0e6287202fced83f79975ec59a3a6-Paper.pdf.

[Laz20] F. Lazzeri. *Machine Learning for Time Series Forecasting with Python*. Wiley, 2020. ISBN: 9781119682370. URL: https://books.google.de/books?id=2VOMEAAAQBAJ.

[ONR19] P.M. Oliveira, P. Novais **and** L.P. Reis. *Progress in Artificial Intelligence: 19th EPIA Conference on Artificial Intelligence, EPIA 2019, Vila Real, Portugal, September 3–6, 2019, Proceedings, Part I*. Lecture Notes in Computer Science. Springer International Publishing, 2019. ISBN: 9783030302412. URL: https://books.google.de/books?id=O2SsDwAAQBAJ.

[Org21] World Health Organization. *WHO global air quality guidelines: particulate matter (PM2.5 and PM10), ozone, nitrogen dioxide, sulfur dioxide and carbon monoxide*. World Health Organization, 2021, xxi, 273 p.

[PAS22] Shree Nandhini Parthiban, P. Amudha **and** S. Sivakumari. "Exploitation of Advanced Deep Learning Methods and Feature Modeling for Air Quality Prediction". **in***Revue d'Intelligence Artificielle*: (2022). URL: https://api.semanticscholar.org/CorpusID:255974429.

[PB06] Antonella Plaia **and** A. Bondi. "Single imputation method of missing values in environmental pollution data sets". **in***Atmospheric Environment - ATMOS ENVIRON*: 40 (**december** 2006), **pages** 7316–7330. DOI: 10.1016/j.atmosenv.2006.06.040.

[Wil+18] D. Alexandra Williams **andothers**. "A comparison of data imputation methods using Bayesian compressive sensing and Empirical Mode Decomposition for environmental temperature data". **in***Environmental Modelling Software*: 102 (2018), **pages** 172–184. ISSN: 1364-8152. DOI: https://doi.org/10.1016/j.envsoft.2018.01.012. URL: https://www.sciencedirect.com/science/article/pii/S1364815217307338.