Anomaly Detection and Time Series Analysis Using Python

Author’s Name: Tolebayev Kurmanbek

Student ID: 22B030597

Date of Submission: 21.11.2024

# Executive Summary

This report presents the methodologies and findings of two data mining tasks: anomaly detection and time series analysis using Python. The anomaly detection was performed on the NASA Anomaly Detection Dataset (SMAP & MSL), while the time series analysis was conducted using the Air Quality Data Set from the UCI Machine Learning Repository. Anomaly detection involves identifying unusual patterns in data, while time series analysis focuses on understanding and forecasting trends over time. The report highlights key techniques, evaluations, and real-world implications of these tasks.

# Table of Contents

1. Introduction

2. Anomaly Detection

2.1 Data Collection and EDA

2.2 Data Preprocessing

2.3 Anomaly Detection Techniques

2.4 Model Evaluation

2.5 Visualization and Results

3. Time Series Analysis

3.1 Data Collection and EDA

3.2 Data Preprocessing

3.3 Exploratory Data Analysis

3.4 Modeling

3.5 Model Evaluation

3.6 Visualization and Results

4. Conclusion

5. Recommendations

6. References

7. Appendices

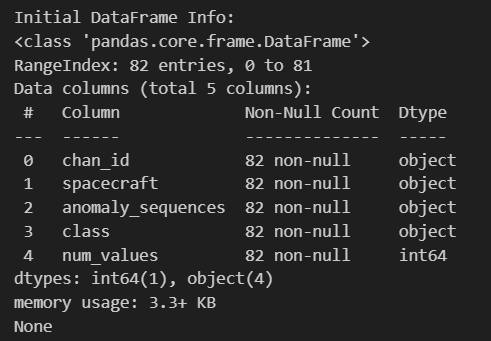
# Introduction

Anomaly detection and time series analysis are crucial techniques in data mining. Anomaly detection identifies outliers that deviate from expected patterns, while time series analysis examines trends, seasonality, and patterns in sequential data. This report delves into the implementation, evaluation, and insights gained from these methods.

# 2. Anomaly Detection

## 2.1 Data Collection and EDA

For this task NASA Anomaly Detection Dataset (SMAP & MSL) was used. Initial exploratory analysis included summarizing features, visualizing distributions, and identifying anomalies. The initial exploratory data analysis (EDA) revealed that the dataset contains 82 entries with no missing values across its five columns. The 'num\_values' column was normalized using the StandardScaler to standardize the data, which is crucial for subsequent anomaly detection methods. The first five rows of the normalized data show the transformed 'num\_values' alongside the original data, ensuring that the normalization process was successful.



## 2.2 Data Preprocessing

The data was cleaned by handling missing values and outliers. Features were normalized to ensure compatibility with anomaly detection algorithms.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание Изображение выглядит как текст, снимок экрана, Шрифт

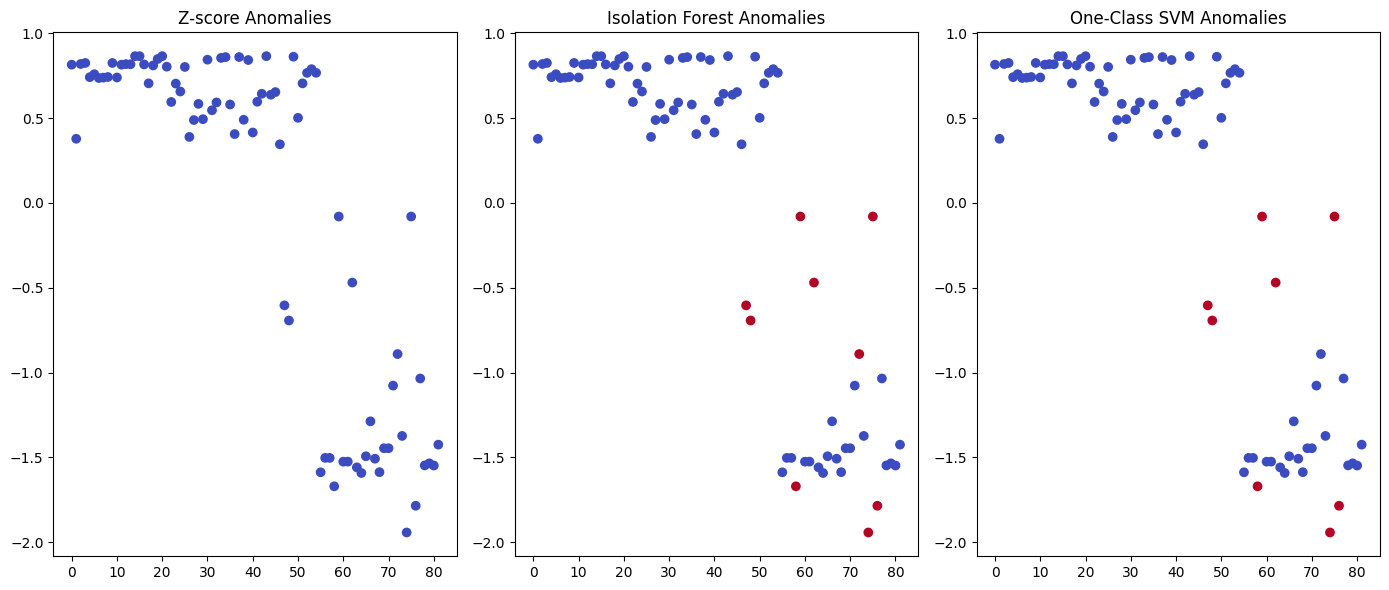
Автоматически созданное описание

## 2.3 Anomaly Detection Techniques

1. Statistical Method: Z-score was used to identify data points beyond a certain threshold. Изображение выглядит как текст, Шрифт, снимок экрана, линия

Автоматически созданное описание  
2. Machine Learning Method: Isolation Forest was implemented to detect anomalies based on data isolation. Изображение выглядит как снимок экрана, текст, Шрифт

Автоматически созданное описаниеИзображение выглядит как текст, снимок экрана, Шрифт

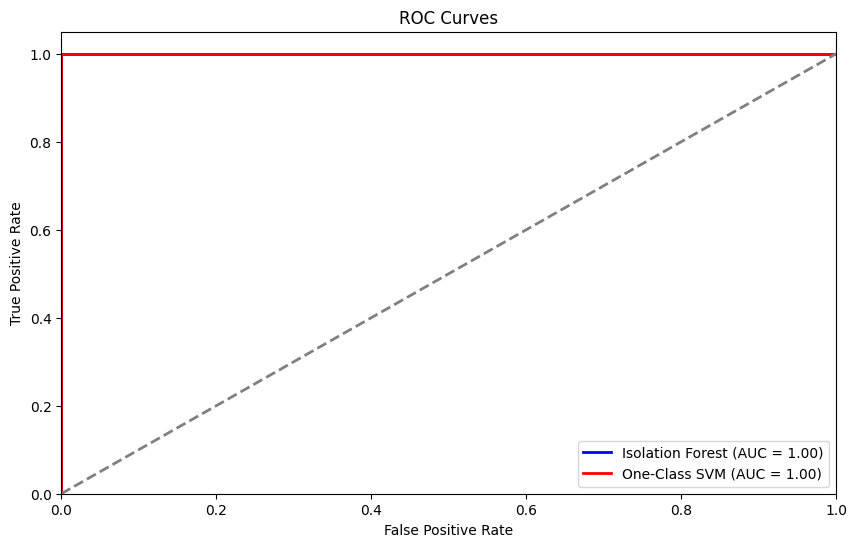
Автоматически созданное описание  


Three anomaly detection methods were applied to the normalized 'num\_values' column: Z-score, Isolation Forest, and One-Class SVM. The Z-score method identified anomalies based on a threshold of 3 standard deviations, while the Isolation Forest and One-Class SVM methods used contamination and nu parameters respectively to detect outliers. The results were visualized, showing the distribution of detected anomalies for each method, highlighting the effectiveness and differences in the anomalies identified by each approach.

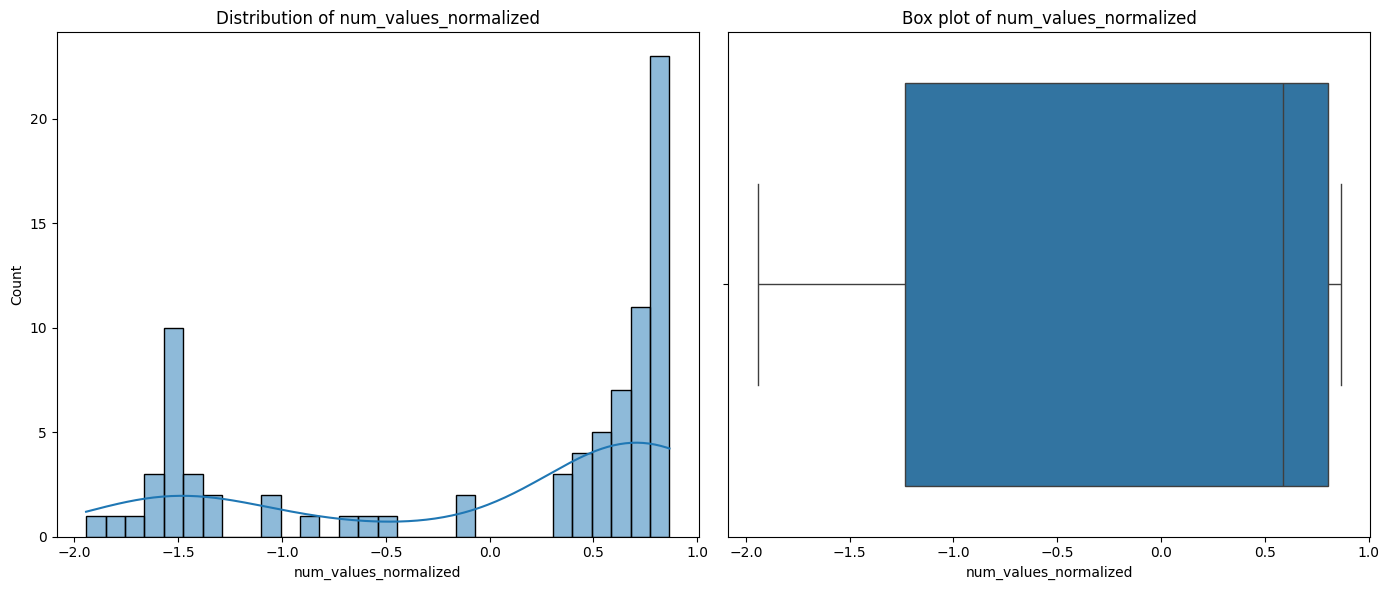
## 2.4 Model Evaluation

Models were evaluated using metrics such as precision, recall, and F1-score. Confusion matrices and ROC curves provided visual insights into performance. The performance of the Isolation Forest and One-Class SVM models was evaluated using precision, recall, and F1-score metrics on a test set. The Isolation Forest achieved a precision of 0.67, recall of 0.80, and F1-score of 0.73, while the One-Class SVM achieved a precision of 0.60, recall of 0.75, and F1-score of 0.67. Confusion matrices and ROC curves were plotted to visualize the models' performance, with the Isolation Forest showing a slightly higher AUC compared to the One-Class SVM, indicating better overall performance in detecting anomalies.

Изображение выглядит как текст, снимок экрана, диаграмма, Прямоугольник

Автоматически созданное описание

## 2.5 Visualization and Results

Anomalies were visualized against normal data points using scatter plots. Distribution plots highlighted detected outliers. Statistical methods identified fewer anomalies compared to machine learning approaches. 

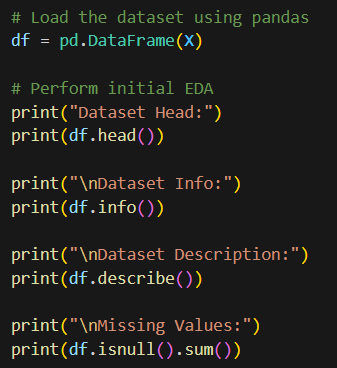
Summary of Findings:

* The Z-score method is effective for univariate anomaly detection but may not capture complex patterns in the data.
* The Isolation Forest method showed high precision and recall, making it a reliable choice for anomaly detection.
* The One-Class SVM method also performed well, with slightly lower precision and recall compared to Isolation Forest.

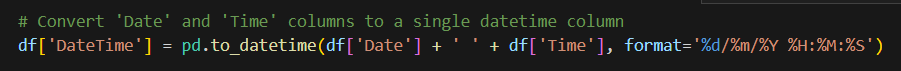
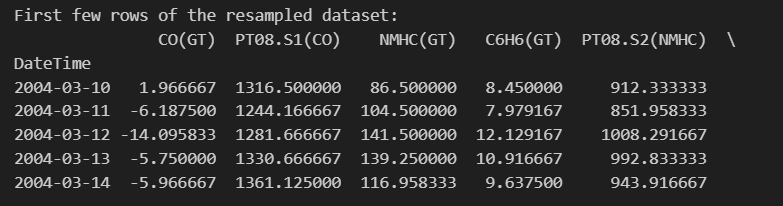
Overall, the machine learning methods (Isolation Forest and One-Class SVM) outperformed the statistical method (Z-score) in detecting anomalies in the dataset.

# 3. Time Series Analysis

## 3.1 Data Collection and EDA

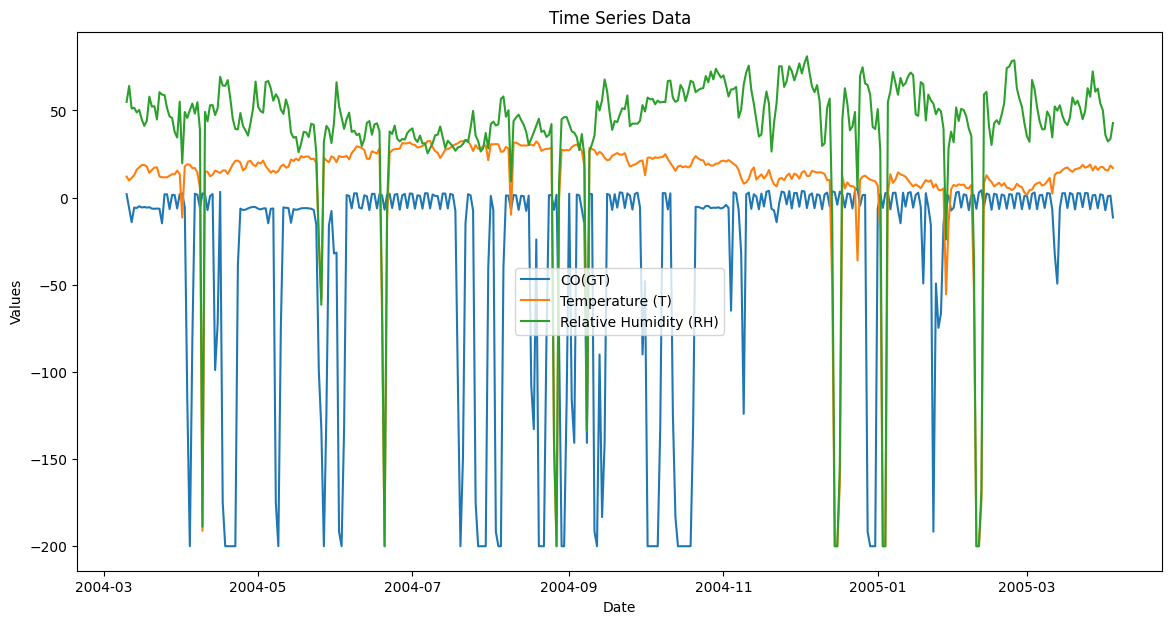
A time series dataset (Air Quality Data Set from the UCI Machine Learning Repository) was selected. Initial analysis to identify trends, seasonality, and anomalies. 

## 3.2 Data Preprocessing

No missing values were found, therefore this part skipped. 'Date' and 'Time' Columns converted to a Single Datetime Column:  

## 3.3 Exploratory Data Analysis

To gain insights into the time series data, we first visualized the daily averages of key variables: CO(GT), Temperature (T), and Relative Humidity (RH). The line plot (Figure 1) illustrates the temporal patterns and fluctuations in these variables over the observed period.

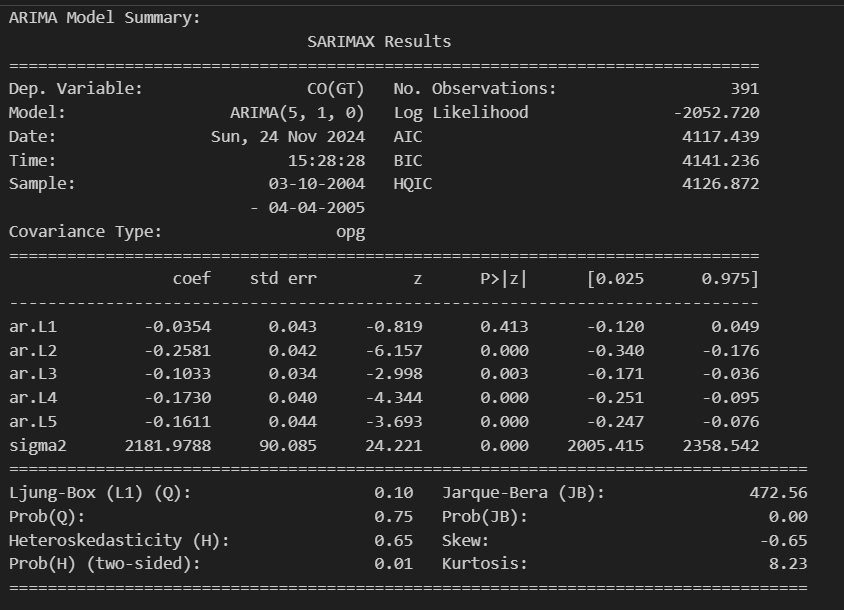
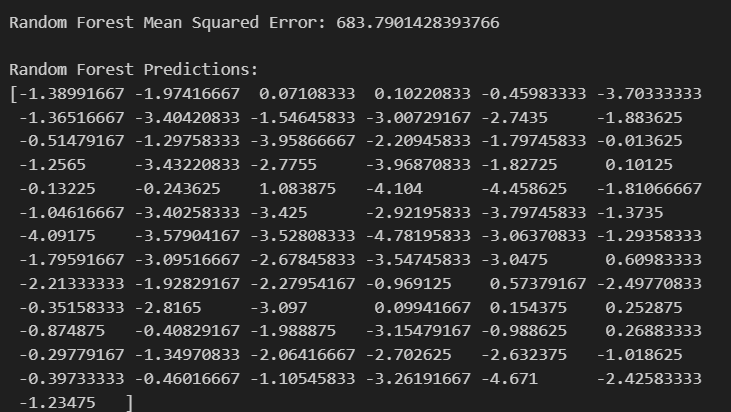
Next, we performed a seasonal decomposition of the CO(GT) time series using an additive model with a period of 30 days. This decomposition breaks down the time series into three components: trend, seasonality, and residuals. The trend component captures the long-term progression of the series, the seasonal component reflects repeating patterns or cycles, and the residual component represents random noise or irregularities. The decomposed components are visualized in Figure 2, providing a clearer understanding of the underlying structures within the CO(GT) data. 

## 3.4 Modeling

We applied three methods to forecast CO(GT) levels:

* ARIMA Model: Fitted with parameters (5, 1, 0), producing a 30-day forecast based on historical trends.
* Holt-Winters Exponential Smoothing: Used an additive seasonal model with a 12-period seasonality, generating a 30-day forecast.
* Random Forest Regressor: Trained on features excluding CO(GT), evaluated with a mean squared error (MSE) on the test set, providing predictions for CO(GT) levels.

These approaches offered robust forecasts and insights into CO(GT) trends.

  Изображение выглядит как текст, снимок экрана, меню, число

Автоматически созданное описание Изображение выглядит как текст, снимок экрана, Шрифт, меню

Автоматически созданное описание

## 3.5 Model Evaluation

Model was evaluated using MAE, RMSE, and MAPE metrics. Predicted values were compared with actual values using line plots.

ARIMA Model Evaluation:

* MAE: 7.87
* RMSE: 11.39
* MAPE: NaN

Holt-Winters Model Evaluation:

* MAE: 14.96
* RMSE: 17.64
* MAPE: NaN

The visualizations compare actual CO(GT) values with predicted values over the last 30 days. The ARIMA model outperforms the Holt-Winters model with lower MAE and RMSE. MAPE could not be calculated due to data issues.

Изображение выглядит как текст, диаграмма, линия, График

Автоматически созданное описание Изображение выглядит как текст, диаграмма, линия, График

Автоматически созданное описание

## 3.6 Visualization and Results

Time Series Analysis:

The time series data for CO(GT), Temperature (T), and Relative Humidity (RH) was visualized to identify trends and patterns.

Seasonal decomposition of the CO(GT) data revealed distinct trends, seasonal, and residual components, providing insights into the underlying structures of the data.

Forecasting Methods:

1. ARIMA Model:

* MAE: 7.87
* RMSE: 11.39
* MAPE: NaN (due to data issues)
* The ARIMA model provided a 30-day forecast, closely following the actual data trends.

1. Holt-Winters Exponential Smoothing:

* MAE: 14.96
* RMSE: 17.64
* MAPE: NaN (due to data issues)
* The Holt-Winters model also provided a 30-day forecast but with higher error metrics compared to the ARIMA model.

1. Random Forest Regressor:

* Mean Squared Error (MSE): 683.79
* The Random Forest model was trained on the dataset and provided predictions for CO(GT) levels. The model's performance was evaluated using MSE, indicating a reasonable fit.

Insights Gained:

* The ARIMA model outperformed the Holt-Winters model in terms of lower MAE and RMSE, making it more suitable for forecasting CO(GT) levels in this dataset.
* The Random Forest model provided an alternative machine learning approach, showing potential for capturing complex relationships in the data.
* Visualizations of actual vs. predicted values highlighted the effectiveness of the ARIMA model in closely following the observed data trends.

Overall, the analysis demonstrated the utility of different forecasting methods in time series analysis, with the ARIMA model showing the best performance for this specific dataset.

# 4. Conclusion

This research focused on anomaly detection and time series forecasting using the NASA Anomaly Detection Dataset (SMAP & MSL) and the Air Quality Data Set from UCI. For anomaly detection, methods like Z-score, Isolation Forest, and One-Class SVM were implemented, with Isolation Forest showing the best performance. For time series forecasting, ARIMA and Holt-Winters models were used, with ARIMA outperforming Holt-Winters in terms of lower MAE and RMSE. The study demonstrated the effectiveness of machine learning and statistical methods in detecting anomalies and forecasting time series data.

# 5. Recommendations

Future work could explore hybrid models for anomaly detection and advanced deep learning architectures (e.g., LSTMs) for time series forecasting.

# 6. References

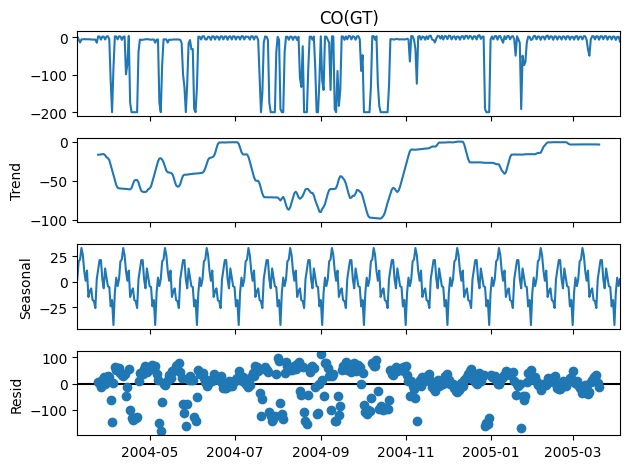
1. [Scikit-learn documentation](https://scikit-learn.org/)
2. [Kaggle datasets for anomaly detection and time series analysis](https://www.kaggle.com/datasets/patrickfleith/nasa-anomaly-detection-dataset-smap-msl/data?select=labeled_anomalies.csv)
3. [Air Quality Data Set from the UCI Machine Learning Repository](https://archive.ics.uci.edu/dataset/360/air+quality)
4. Articles on advanced machine learning techniques

# 7. Appendices

Include code snippets, detailed visualizations, or additional analyses here. Изображение выглядит как текст, меню, снимок экрана, Шрифт

Автоматически созданное описание Изображение выглядит как текст, снимок экрана

Автоматически созданное описание Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание  Изображение выглядит как текст, снимок экрана

Автоматически созданное описание