Air Quality Prediction Using Regression Analysis

MAY 7

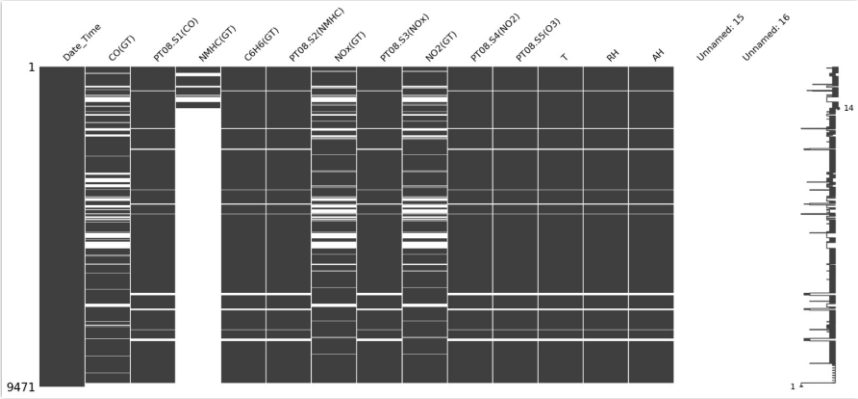
IST 707
Authored by:
Sahil Wani
Mahitha Chennamadhava
Sai Sisira Pathakamur
Garvaa Jamsran
Pavan Kumar Reddy Katasani

Introduction

In the modern world, air pollution is a significant problem. The majority of cities experience air pollution. Numerous contaminants exist, including total petroleum hydrocarbons, acrolein, asbestos, benzene, NO2, NO, SO2, NH3, PM 2.5, PM 10, polycyclic aromatic hydrocarbons, and synthetic vitreous fibers. Monitoring of air pollution has been more popular recently because it significantly affects both human health and the ecological balance. In this report, we analyze the AQI data for a city in Italy, with the aim of providing insights into the air quality of the city and informing policymakers on strategies to improve air quality and protect public health.

Data Collection and Processing

The air quality index (AQI) data analyzed in this report was obtained from the UCI machine learning repository. This data was collected from a monitoring station located within the city over a 12-month period, spanning from March 2004 to February 2005. The data has 9358 instances of an hourly averaged responses from an array of 5 metal oxide chemical sensors. To prepare the AQI dataset for analysis, we encountered several missing values represented by -200, and found that dropping rows with all NAN column values would result in a significant loss of data. Filling NAN values with linear interpolation was also not a viable option due to the complex time series patterns in the data, including cyclical and seasonal patterns. Instead, we opted to use K-nearest neighbor (KNN) imputation, a non-parametric method that estimates missing values based on the values of the k-nearest neighbors. KNN imputation is a structured dataset-friendly method, resistant to outliers and noise, making it a suitable choice for our data imputation needs. The following figure helps us understand the missing data.



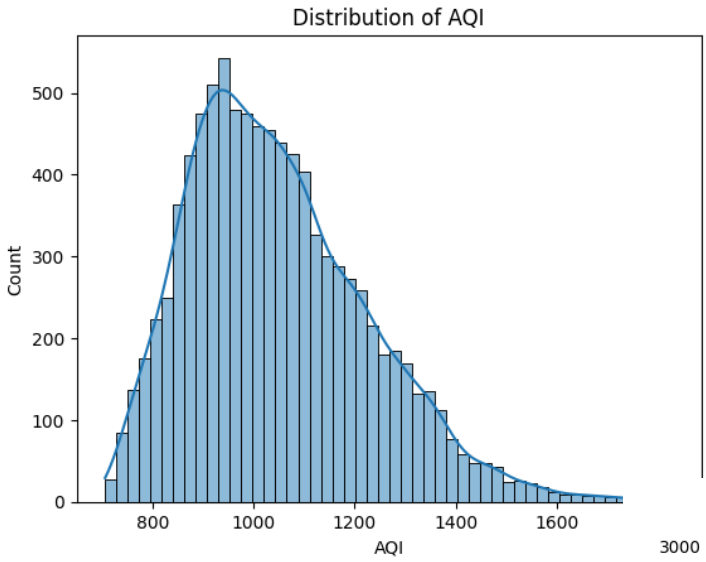
Attribute information of the dataset-
Attributes:

- Date (DD/MM/YYYY)
- Time (HH.MM.SS)
- CO(GT) — True hourly averaged concentration CO
- PT08.S1(CO) — Hourly averaged sensor response for CO
- NMHC — True hourly averaged overall Non-Metallic Hydrocarbons concentration
- C6H6(GT) — True hourly averaged Benzene concentration
- PT08.S2(NMHC) — Hourly averaged sensor response for NMHC
- NOx(GT) — True hourly averaged NOx concentration
- PT08.S3(NOx) — Hourly averaged sensor response for NO
- NO2(GT) — True hourly averaged NO2 concentration
- PT08.S4(NO2) — Hourly averaged sensor response for NO2
- PT08.S5(O3) — Hourly averaged sensor response for O3
- T — Temperature in Â°C
- RH — Relative Humidity (%)
- AH — Absolute Humidity

Data Analysis

The initial step in the data analysis process involved defining the AQI variable by utilizing a formula which incorporates the AQI values for PM10, PM2.5, O3, NO2, and SO2. The formula is AQI = [(Ipm10/ Bpm10) + (Ipm2.5/ Bpm2.5) + (Io3/ Bo3) + (Ino2/ Bno2) + (Iso2/ Bso2)]. The corresponding breakpoint concentrations were also considered in this equation, which denotes the concentrations at which pollutant levels start to have adverse effects on human health. The breakpoint concentrations for Italy are established at PM10: 50 µg/m3, PM2.5: 25 µg/m3, O3: 100 µg/m3, NO2: 200 µg/m3, and SO2: 125 µg/m3. The AQI value is then calculated based on the concentrations of these five major pollutants, as well as CO.

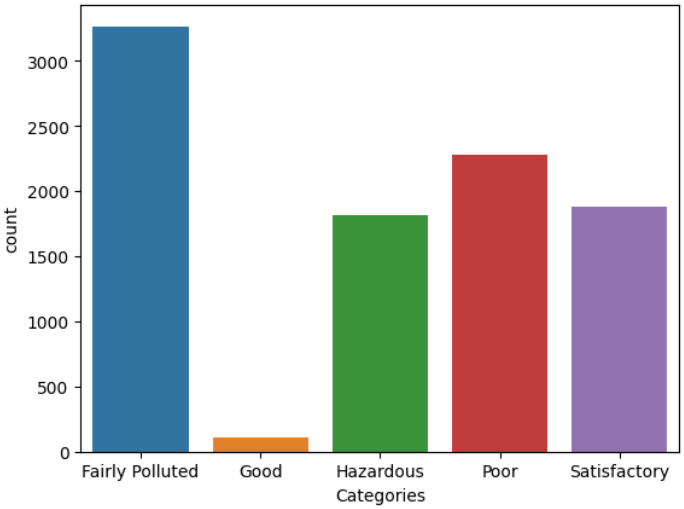
The AQI variable was subsequently categorized into five distinct categories based on the established AQI thresholds: Good (AQI < 750), Satisfactory (AQI ≤ 900), Fairly Polluted (AQI ≤ 1050), Poor (AQI ≤ 1200), and Hazardous (AQI > 1200). These categories were determined by analyzing the normal distribution of the AQI values, which confirmed the accuracy and reliability of the data. So, we defined two new columns into our dataset named AQI and category.



AQI < 750	Good
AQI <=900	Satisfactory
AQI <= 1050	Fairly Polluted
AQI <= 1200	Poor
AQI >1200	Hazardous

This plot will give us an idea about the distribution of AQI values in the dataset.

The presented plot depicts the distribution of AQI categories within the analyzed dataset. It provides an insight into the relative frequency of each AQI category.



Findings and Results

1. Decision Tree Classifiers

Decision tree classifiers builds a tree-like model to make predictions based on a sequence of decisions. The findings and results of our analysis are presented as follows. Firstly, the dataset was split into training and testing sets, with a ratio of 7:3 respectively, to train our models. We employed the decision tree algorithm with hyperparameters tuning using GridSearchCV. The best hyperparameters for the decision tree classifier were found to be max_depth=3 and min_samples_leaf=0.1, with an accuracy of 0.7763532763532763.

2. KNN Classifier

It classifies new data points based on the class of the nearest neighbors in the training set. The algorithm works by calculating the distance between the new

data point and all the instances in the training data set. The k closest instances are then used to predict the class of the new data point.

The parameter k represents the number of nearest neighbors to consider when making a prediction, and it can be tuned to optimize the performance of the algorithm.

In this project, we used GridSearchCV to perform hyperparameter tuning and find the best values for k and the weight function (uniform or distance). Results show that the best hyperparameters for your KNN model were k=5 and weighting scheme='distance'. This means that the model considered the distance between a new data point and its k=5 closest neighbors when making a classification decision, with closer neighbors having more influence on the decision.

We obtained an accuracy score of 0.9441, which means that the model correctly predicted the AQI category for 94.41% of the test data points. This is a strong performance compared to the decision tree algorithm, the KNN algorithm achieved a higher accuracy, indicating that it may be a better choice for this particular data set and problem.

3. Random Forest Classifier

Random forest is a supervised learning algorithm that builds multiple decision trees and combines their predictions to obtain a more accurate and stable result. It randomly selects subsets of features and data points to build decision trees, and then aggregates the results of all decision trees to produce the final prediction. The RandomForestClassifier model was trained and evaluated using the GridSearchCV method, with a parameter grid consisting of values for max_depth, max_features, min_samples_leaf, and n_estimators. The best estimator was then used to predict the target variable for the test data.

The accuracy score of the model on the test data was found to be 0.8736. The best parameters for the model were determined to be max_depth = 4, max_features = 4, min_samples_leaf = 5, and n_estimators = 200. The classification report for this model was not printed, but it can be generated using the same code as for the DecisionTreeClassifier model and KNeighborsClassifier model.

4. SVM Classifier

SVM classifiers separates data into classes using a hyperplane and finds the best boundary to maximize the margin between classes. We trained an SVM model using GridSearchCV to tune hyperparameters. The parameters that we varied were 'C', 'kernel', and 'degree'. The model was trained on the training dataset and

tested on the test dataset. The best parameters selected by the GridSearchCV were 'C': 0.01, 'degree': 4, and 'kernel': 'poly'.

The accuracy of the model on the test dataset was found to be 0.8977920227920227, which indicates that the model is able to predict the class labels with an accuracy of 89.78%. This is a good accuracy score for our dataset.

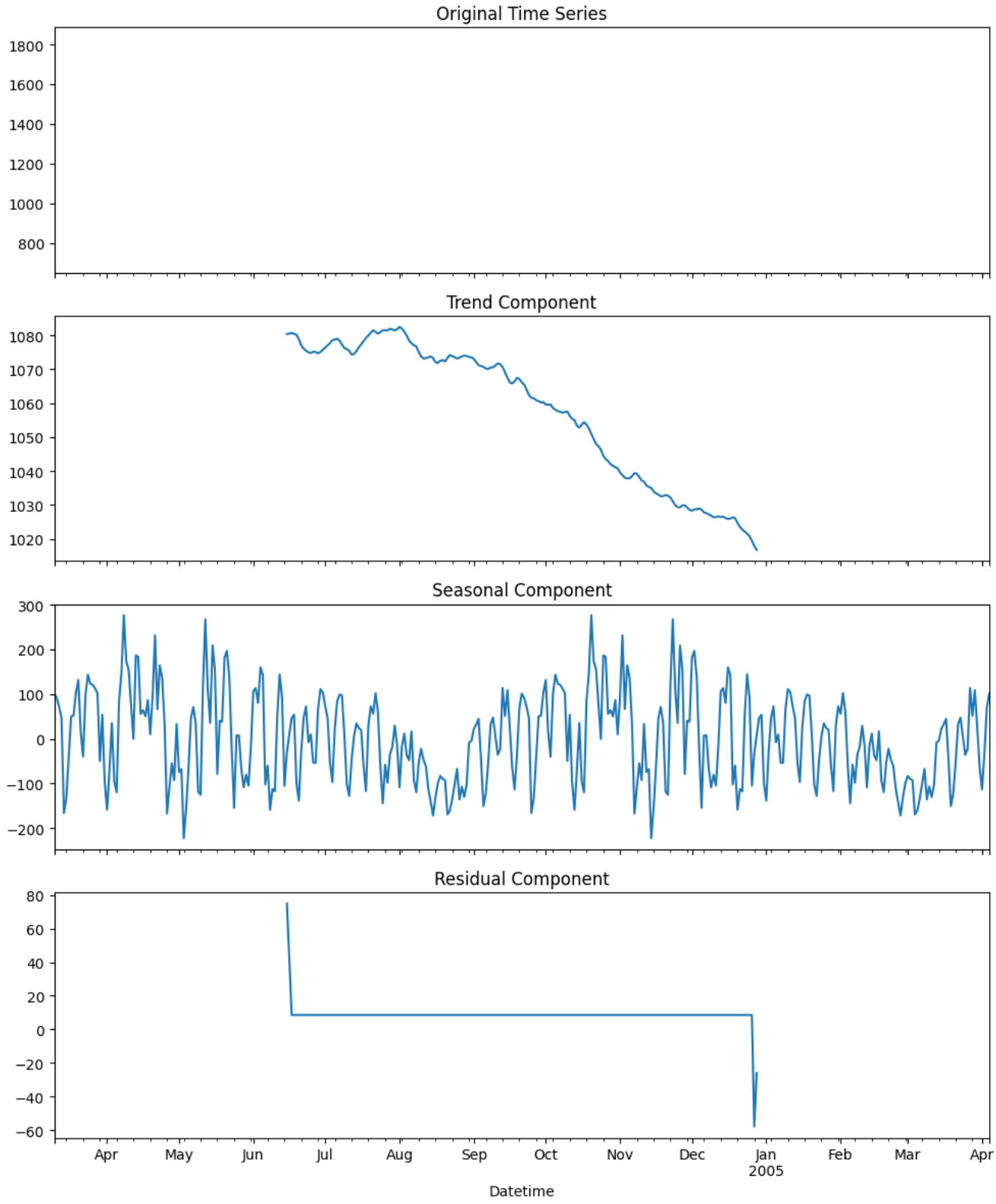
In summary, we were able to train an SVM model that can predict the class labels with good accuracy by tuning the hyperparameters using GridSearchCV.

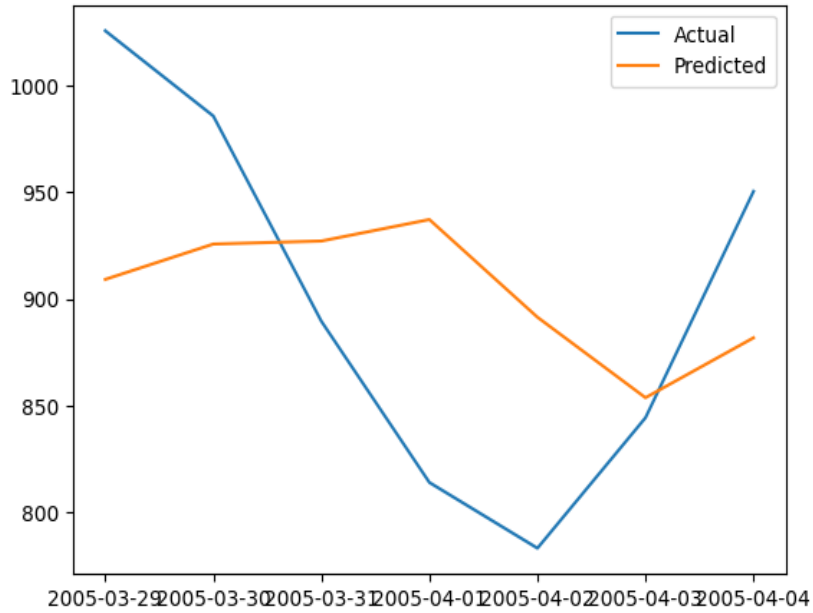
5. Time Series Forecasting

a. ARIMA Model

The next step was to decompose the time series data to recognize patterns, trends, and seasonality. This helped us understand the underlying structure of the data and identify any significant trends or patterns. We then conducted an Augmented Dickey-Fuller (ADF) test to determine the stationarity of the time series. The null hypothesis of the ADF test is non-stationarity with a trend which means it has a unit root and exhibits a trend, while the alternative is stationarity without a trend which means it has no unit root and does not exhibit a trend. The p-value of the ADF test was found to be less than 0.05, which indicated rejection of the null hypothesis and confirmed stationarity of the time series with a 95% confidence level. This stationary data was used for fitting a time series model.

In this project, we used the seasonal decomposition analysis (SDA) method to analyze the air quality index (AQI) time series data. The SDA is a time series analysis technique used to break down a time series into its underlying trend, seasonal, and residual components. We applied the additive model of SDA on our AQI time series data with a period of 195 days. The decomposition revealed that the AQI data had a noticeable seasonality pattern and a slightly decreasing trend over time. The residual plot showed that there were some fluctuations in the AQI data that could not be explained by the trend and seasonality components. Overall, the SDA method was useful in identifying the underlying patterns and fluctuations in our AQI time series data.


Original Time Series
1800
1600
1400
1200
1000
800
Trend Component
1080
1070
1060
1050
1040
1030
1020
Seasonal Component
300
200
100
0
−100
−200
Residual Component
80
60
40
20
0
−20
−40
−60
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Jan
2005
Feb
Mar
Apr
Datetime



To predict the next 10 days, an auto ARIMA model was used, which resulted in a mean absolute error of 74.98 and a root mean square error of 84.96. These errors were higher in nature, as ARIMA models generally require more data. The model was found to be overfitting, indicating that the performance of the model on new data may not be as good.

b. Exponential Smoothening

Exponential smoothing was used as an alternative method, which has fewer parameters to estimate than ARIMA and can make it easier to select optimal smoothing parameters. The first step of the code involves splitting the data into training and test sets. The training set comprises 80% of the data, while the test set contains the remaining 20%. The Holt-Winters method is then applied to the training set using the Exponential Smoothing function from the statsmodels.tsa.holtwinters library. The function is configured to use an additive trend, an additive seasonal component with a period of 100, and to optimize the smoothing parameters using maximum likelihood estimation.

Once the model is trained on the training set, it is used to make predictions on the test set. The forecast function of the model_fit object is called with an argument specifying the length of the test set. The Mean Absolute Percentage Error (MAPE) is then calculated by computing the absolute percentage error between the actual values and the predicted values for each time step in the test set. The average of these errors is then computed as MAPE.

Finally, the trained model is used to make forecasts for the next 10 time periods using the forecast function again. These forecasts can be used to gain insights into future trends and patterns in the data.

The exponential smoothing model resulted in a MAPE of 4.51, which means that the model's predictions, on average, were off by 4.51% of the actual values in the test set. A MAPE of 4.51% is generally considered to be a good result, indicating that the model's predictions are accurate and reliable.

These are the forecasted AQI values for next 10 days.

1131.749929	1004.824410
1078.957425	989.769654
1045.808006	1009.110220
1058.827633	1112.439036
1028.478679	1135.527800

In this analysis, we explored the air quality index (AQI) data of a city in Italy. We found that the average AQI levels were moderate, with occasional spikes in pollution. Overall, this analysis highlights the need for continued monitoring and mitigation efforts to improve air quality and protect public health in the city.

References

S. De Vito, E. Massera, M. Piga, L. Martinotto, G. Di Francia, On field calibration of an electronic nose for benzene estimation in an urban pollution monitoring scenario, Sensors and Actuators B: Chemical, Volume 129, Issue 2, 22 February 2008, Pages 750-757, ISSN 0925-4005, [Web Link].

N. Srinivasa Gupta, Yashvi Mohta, Khyati Heda, Raahil Armaan, B. Valarmathi, G. Arulkumaran, "Prediction of Air Quality Index Using Machine Learning Techniques: A Comparative Analysis", Journal of Environmental and Public Health, vol. 2023, Article ID 4916267, 26 pages, 2023, [web link].

T. Madan, S. Sagar and D. Virmani, "Air Quality Prediction using Machine Learning Algorithms –A Review," 2020 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN), Greater Noida, India, 2020, pp. 140-145, doi: 10.1109/ICACCCN51052.2020.9362912.

Sarkar N, Gupta R, Keserwani PK, Govil MC. Air Quality Index prediction using an effective hybrid deep learning model. Environ Pollut. 2022 Dec 15;315:120404. doi: 10.1016/j.envpol.2022.120404. Epub 2022 Oct 11. PMID: 36240962.