Data ProSolveRS

Data ProSolveRS

Pritish Saha

Rounak Nath

Srinjoy Ganguly

AIR POLLUTION AND RH

Nitrogen Oxides (NOx)

Formation of Nitric Acid Aerosols: NOx emissions can react with atmospheric moisture to form nitric acid (HNO3) aerosols, which are also hygroscopic and can contribute to an increase in relative humidity by absorbing water vapor.

Ozone (O3)

Secondary Aerosol Formation: Ozone can react with volatile organic compounds (VOCs) and other pollutants in the atmosphere to form secondary organic aerosols (SOAs), which, as mentioned earlier, can be hygroscopic and contribute to an increase in relative humidity.

Carbon Monoxide (CO) and Carbon Dioxide (CO2)

While carbon monoxide and carbon dioxide themselves do not directly affect relative humidity, their presence in the atmosphere can influence atmospheric processes that indirectly affect RH, such as temperature regulation and cloud formation.

Volatile Organic Compounds (VOCs)

Secondary Organic Aerosol Formation: VOCs can undergo photochemical reactions in the atmosphere to form secondary organic aerosols (SOAs), which can be hygroscopic and contribute to an increase in relative humidity through water vapor absorption.

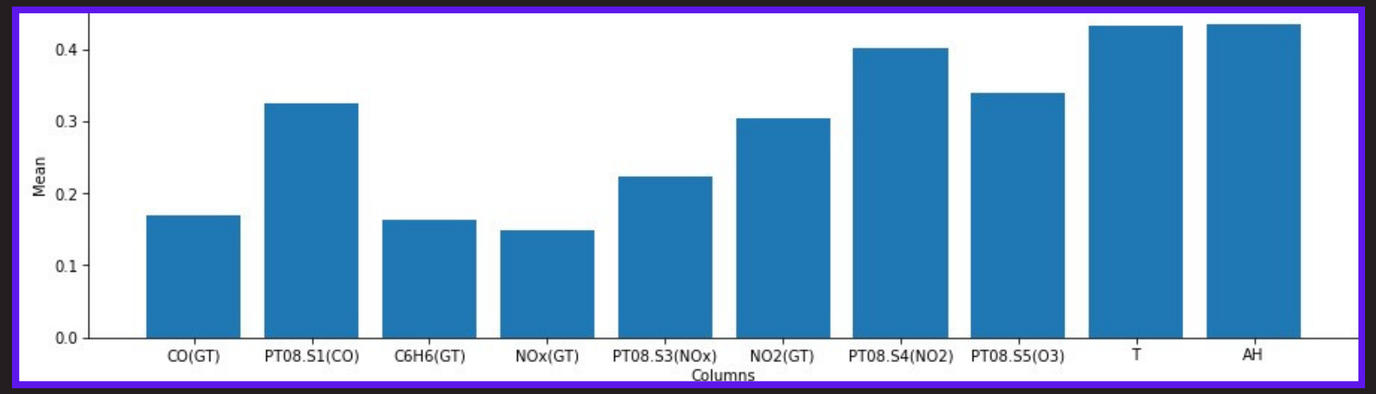
EXPLORATORY DATA ANALYSIS

Insights from EDA: Summary statistics, distribution, standard deviation, and correlations illuminate key aspects of the dataset

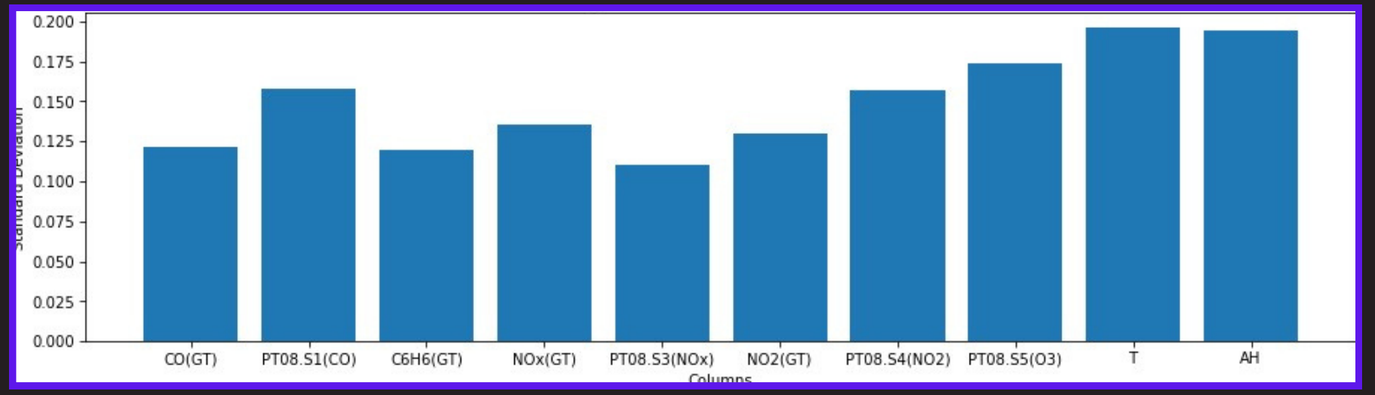
Correlation heatmap

μ	CO (GT)	PT08.S1 (CO)	C6H6 (GT)	PT08.S2 (NMHC)	NOx (GT)	PT08.S3 (NOx)	NO2 (GT)	PT08.S4 (NO2)	PT08.S5 (O3)	T	RH	AH
CO (GT)	1,000	0,844	0,895	0,895	0,865	0,128	0,815	0,810	0,824	0,582	0,460	0,543
PT08.S1 (CO)	0,844	1,000	0,883	0,883	0,837	0,095	0,785	0,881	0,871	0,610	0,519	0,629
C6H6 (GT)	0,895	0,883	1,000	1,000	0,887	0,098	0,836	0,850	0,865	0,613	0,463	0,558
PT08.S2 (NMHC)	0,895	0,883	1,000	1,000	0,887	0,098	0,836	0,850	0,865	0,613	0,463	0,558
NOx (GT)	0,865	0,837	0,887	0,887	1,000	0,157	0,857	0,793	0,845	0,561	0,480	0,528
PT08.S3 (NOx)	0,128	0,095	0,098	0,098	0,157	1,000	0,214	0,116	0,130	0,398	0,481	0,379
NO2 (GT)	0,815	0,785	0,836	0,836	0,857	0,214	1,000	0,733	0,788	0,589	0,411	0,489
PT08.S4 (NO2)	0,810	0,881	0,850	0,850	0,793	0,116	0,733	1,000	0,825	0,615	0,566	0,700
PT08.S5 (O3)	0,824	0,871	0,865	0,865	0,845	0,130	0,788	0,825	1,000	0,556	0,521	0,581
T	0,582	0,610	0,613	0,613	0,561	0,398	0,589	0,615	0,556	1,000	0,290	0,586
RH	0,460	0,519	0,463	0,463	0,480	0,481	0,411	0,566	0,521	0,290	1,000	0,704
AH	0,543	0,629	0,558	0,558	0,528	0,379	0,489	0,700	0,581	0,586	0,704	1,000

Mean of each column



Standard Deviation of Each column



DATA PREPROCESSING

Removal of columns

- The column ‘NMHC(GT)’ has been removed as the column has 7086 missing values and the removal of the column is beneficial for better fitting the model.
- The column ‘PT08.S2(NMHC)’ has been removed as the feature has a correlation value of ‘1.00’ with the column ‘C6H6(GT)’, so we need only one column to fit the model. The above mentioned correlation value is acquired from EDA.
- The columns 'Date' and 'Time' were dropped as they represent timestamp information (DD/MM/YYYY and HH.MM.SS) that may not directly contribute to predicting relative humidity levels. Instead, time-related patterns and trends can be captured using additional features derived from these timestamps during feature engineering.
- The 'ID' column was dropped as it serves as a unique identifier for each data point and does not provide any predictive value for relative humidity classification. Including such identifiers as features may introduce noise and unnecessary complexity to the model.

Label Encoding

- The RH categories ('Dry', 'Ideal', 'Slightly Elevated', 'Elevated', 'High') were encoded into numerical values for model training.

Removal of rows

- 289 rows had missing values for AH. After getting a closer look we can also notice that in those rows other 4 columns also have missing values, so these rows aren’t suitable for KNN imputation. On the other hand, due to the number of total rows(8000) being much higher than 289 and due to the fact these rows have a lot of missing values, we safely removed these rows for better fitting of the curves.

Imputation

- As the columns ‘CO(GT)’, ‘NOx(GT)’, and ‘NO2(GT)’ have almost 1600 missing values, and the total number of rows is 8000 **KNN** imputation is used to impute the missing values.

DATA PREPROCESSING

Dropping unnecessary columns

Normalizing the data using Z-score, Max(best), Min-max normalization

The dataset was split into training and testing sets. The training set contains 75% of the data, while the testing set contains the remaining 25%.

Max Normalization

- Performs a linear transformation on the original data. This technique gets all the scaled data in the range (0, 1).
- (Train Accuracy = 98.65%)
- (Test Accuracy = 97.86%)

$$X' = \frac{X}{Xmax}$$

Z-Score Normalization

- Z-score normalization transforms the data into a mean of 0 and a standard deviation 1.
- (Train Accuracy = 98.09%)
- (Test Accuracy = 96.30%)

$$z = \frac{x - \mu}{\sigma}$$

μ = Mean

σ = Standard Deviation

Min–Max Normalization

- Performs a linear transformation on the original data. This technique gets all the scaled data in the range (0, 1).
- (Train Accuracy = 97.61%)
- (Test Accuracy = 96.11%)

$$X' = \frac{X - Xmin}{Xmax - Xmin}$$

SVM

Description

Decision trees, a versatile machine learning model, were employed to predict the RH. By partitioning the feature space based on intrinsic parameters such as total concentration of CO, NOx, NO2 etc, decision trees offer an intuitive approach to making predictions, allowing for easy interpretation of the underlying decision-making process.

Accuracy

Train	Test
94.68%	93.97%

Insights

- The high accuracy scores achieved on both the train and test datasets indicate that the SVM model effectively captures the underlying patterns in the data and generalizes well to unseen data. This suggests that the decision boundaries learned by the SVM algorithm are robust and reliable for predicting RH categories.
- The high accuracy on the test dataset (93.97%) suggests that the model performs well on new, unseen data, indicating its potential for real-world applications.
- The slight drop in accuracy from the train dataset (94.68%) to the test dataset (93.97%) indicates a small degree of overfitting, but the model still demonstrates strong performance overall.

MLP

Description

In summary, this code defines a simple neural network architecture, compiles it with appropriate loss and optimizer settings, and trains it using the specified training data. The goal is likely to perform a classification task based on the given features (X_train) and labels (y_train). Certainly! Multilayer Perceptrons (MLPs), a type of feedforward neural network, consist of interconnected layers of neurons. They learn complex relationships from input data and are widely used in meteorology to predict parameters like relative humidity.

Accuracy

Train	Test
98.65%	97.86%

Insights

1. Model Architecture:
   - The code initializes a sequential neural network model using Keras (a high-level neural networks API).
   - The model is named “my_model.”
   - It consists of two layers:
     - Dense Layer 1:
       - Units: 50 (number of neurons in this layer)
       - Activation function: ReLU (Rectified Linear Unit)
     - Dense Layer 2:
       - Units: 5 (number of neurons in this layer)
       - Activation function: Linear (no activation function applied)
2. Model Compilation:
   - The model.compile() function configures the model for training.
   - Loss function: Sparse Categorical Crossentropy (used for multi-class classification tasks with integer labels)
   - Optimizer: Adam (an efficient gradient-based optimization algorithm)
   - Learning rate for the optimizer: 0.01
3. Training the Model:
   - The model.fit() function trains the model on the provided training data (X_train and y_train).
   - The training process runs for 500 epochs (iterations).

Thank You!

Data ProSolveRS