ASSIGNMENT # 01: BEE-14

CS-471 Machine Learning

Submission Deadline: 6th Oct 2025

Decision Tree Regression for Air Quality Monitoring

Air pollution is a major environmental concern worldwide, with pollutants such as carbon monoxide (CO), nitrogen oxides (NOx, NO_2), non-methane hydrocarbons (NMHC), and benzene (C_6H_6) having significant impacts on human health.

The Air Quality Dataset (collected in an Italian city from March 2004 to February 2005) provides hourly averaged responses of several chemical sensors, along with ground truth (GT) pollutant concentrations and meteorological variables.

This dataset allows us to explore how well sensor signals and weather conditions can predict the actual concentration of air pollutants.

Dataset Description

Each row in the dataset corresponds to one hourly measurement.

Main Variables:

- **Date, Time** → Timestamp of measurement.

Ground Truth Pollutants (possible targets for prediction):

- CO(GT) → True hourly averaged CO concentration (mg/m^3).
- NMHC(GT) → True hourly averaged Non-Methane Hydrocarbons (µg/m^3).
- C6H6(GT) → True hourly averaged Benzene (µg/m^3).
- NOx(GT) → True hourly averaged Nitrogen Oxides (ppb).
- NO2(GT) → True hourly averaged NO_2 (µg/m^3).

Sensor Signals (features):

- PT08.S1(CO) → CO-sensitive metal oxide sensor signal.
- PT08.S2(NMHC) → NMHC-sensitive metal oxide sensor signal.
- PT08.S3(NOx) → NOx-sensitive metal oxide sensor signal.
- PT08.S4(NO2) → NO_2-sensitive metal oxide sensor signal.
- PT08.S5(O3) → O_3-sensitive metal oxide sensor signal.

Environmental Conditions (features):

- T → Temperature (°C).
- RH → Relative Humidity (%).
- AH → Absolute Humidity.

*Further Details of dataset can be acquired from this page: https://archive.ics.uci.edu/dataset/360/air+quality

Assignment Task

Data Preparation

- Load the dataset into Python (CSV provided).
- Handle missing values
- Choose one pollutant as your **target variable**
- Select relevant features (sensors + weather conditions).

- Split dataset into training (80%) and testing (20%).

Model Training

- Train a **Decision Tree Regressor from scratch** to predict the chosen pollutant
- Experiment with different values of max_depth (e.g., 3, 5, 10, None).
- Record training and test performance for each case.

Model Evaluation

- Compute and report:
    - Mean Squared Error (MSE)
    - Root Mean Squared Error (RMSE)
    - R^2 Score
- Plot performance metrics vs. tree depth.

Model Visualization

- Plot the decision tree for a small depth (e.g., max_depth=3).
- Display feature importances: Which sensor/environment variable contributes most to predictions?

Baseline Comparison

- Train a **Linear Regression model** on the same dataset.
- Compare its performance with the Decision Tree Regressor.
- Discuss differences in capturing linear vs. non-linear patterns.

Discussion (short written answers)

- Which features were most important for predicting your chosen pollutant?
- Did deeper trees overfit the training data?
- Which performed better overall: Decision Tree or Linear Regression? Why?

Deliverables

Submit a Jupyter Notebook (.ipynb) or .py file containing:

1. Data preprocessing and handling of missing values.
2. Decision Tree model training with different depths.
3. Evaluation metrics and plots.
4. Visualizations (tree diagram, feature importances).
5. Linear Regression comparison.
6. Short discussion

Note: Your submitted code should be neat and clean with proper comments added.