GEOAI CHALLENGE FOR AIR POLLUTION SUSCEPTIBILITY MAPPING BY ITU

1. DESCRIPTION

The city of Milan is located in the Lombardy region (Italy) is a well-known pollution hotspot due to its surrounding topographical components (the Alps in the north and west, and the Apennines in the south) which contribute to low wind circulation in the Po Valley, leading to the accumulation of air pollutants.
The objective of this challenge is to use machine learning to produce air pollution susceptibility maps at the city level (5m spatial resolution) which will support decision-making to improve the resilience of the city.

2. DATASET DESCRIPTION

The data used for this challenge is provided by the competition organizers and downloaded from the competition platform. No external data is used. The datafiles used include:

1. Train.csv- This is a spatial and meteorological dataset containing time-series data of 55 weather stations in Milan over 6 years from 2016-2021. It also contains the AQI column which is the target variable. The model is trained on this dataset.
2. Test.csv- This dataset contains only spatial co-ordinates and season values and used to test the model for scoring in the challenge based on accuracy

3. FEATURES

The following features from the Train.csv are used to train the model:

1. lat: 5m resolution latitude
2. lng: 5m resolution longitude
3. date: date of record of parameters by the weather station from 2016-2021
4. aqi: aqi level recorded, ranging from 1-6 on the aqi scale

4. DATA-PREPROCESSING

4.1 Map Season values:

Season values are mapped using 'date' column based on the mapping provided by the challenge and stored in 'season' column:

Season 1: January-March

Season 2: April-June

Season 3: July-September

Season 4: October-December

4.2 Perform One-Hot Encoding:

For treating the seasons as nominal data, one-hot encoding is performed on the season column class using OneHotEncoder of sklearn.preprocessing.

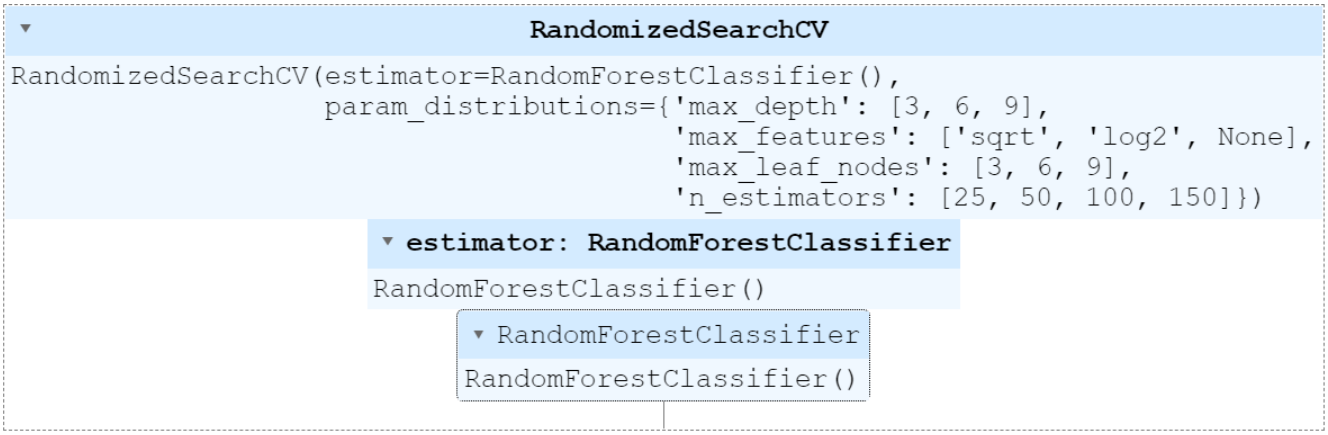
4.3 Feature Scaling:

Even though tree-based models are usually not dependent on feature scaling, StandardScaler() of sklearn.preprocessing is used for standardizing the data.

5. MODEL TRAINING

5.1 Hyper-parameter tuning

RandomizedSearchCV() of sklearn.model_selection is used to set the hyper-parameters of the RandomForestClassifier model. In this process, most appropriate hyperparameters are chosen. Since RandomForestClassifier performs k-fold cross-validation, the training data isn't split into training and validation data.



5.2 Model Training

RandomForestClassifier() of sklearn.ensemble is used to train the model with the hyper-parameter tuning's output parameters.