Random Forest for Air Pollution Susceptibility Mapping

ITU GeoAI Challenge

Isaac Oluwafemi Ogunniyi

Abstract

This work addresses the challenge of air pollution within the city of Milan (Lombardy region, Italy), a well-known pollution hotspot due to its surrounding topographical components (the Alps in the north and west, and the Apennines in the south). A machine learning approach is taken where meteorological and spatial data are used to predict air quality index (AQI) values for different locations in Milan.

1. Introduction

Air pollution is defined as the presence of toxic chemicals or compounds in the air at levels that pose a health risk. Air pollution levels can either be measured individually (per pollutant) or be represented by an index to provide a general panorama of the different pollution levels.

The objective of this work 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 Milan. The city of Milan is considered for this study because it is a well-known air pollution hotspot where topographical components contribute to low wind circulation in the Po Valley, leading to the accumulation of air pollutants.

The task of mapping air pollution susceptibility is approached as a classification problem where previous year's data for the geography whose susceptibility is to be predicted is used to forecast the AQI of the same geography in the present year.
The European Air Quality Index (AQI) is based on concentration values for up to five key pollutants, including PM10, PM2.5, O3, NO2, SO2 aggregated based on a set of guidelines [1].
Predicting AQI is important to aid authorities prepare for the accompanying health risks among other effective mitigation measures

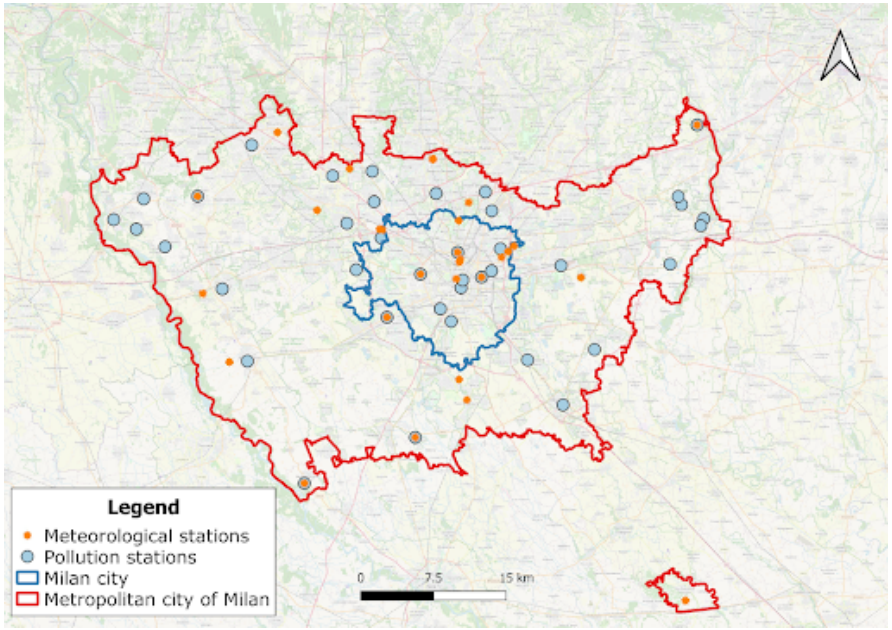

Figure 1. Location of the meteorological and air quality stations in the metropolitan city of Milan.

2. Dataset

The data used in this work is the sole property of Zindi and ITU [2]. It is made up of historical in-situ station data (meteorological and air pollution) with a minimum hourly temporal resolution.
The dataset consists of 121,653 training observations and 160 hold-out test observations for the year 2022.

The train data is made up of 62 columns and contains data of different categories.
The table below mentions some categories of data and the columns in which they are found:

Table 1. The various kinds of data columns included in the dataset

Category	Example features
Geographic	'lat', 'lng','N', 'NE', 'E', 'SE', 'S', 'SW','slope'
Meteorological	'temperature', 'precipitation', 'humidity', 'global_radiation', 'hydrometric_level'
Air Quality	'pm25_aqi', 'pm10_aqi', 'no2_aqi', 'o3_aqi', 'so2_aqi', 'aqi'

In understanding the train dataset, the following plots illustrate the yearly trend of the AQI recorded cumulatively for various geographies in the train dataset:

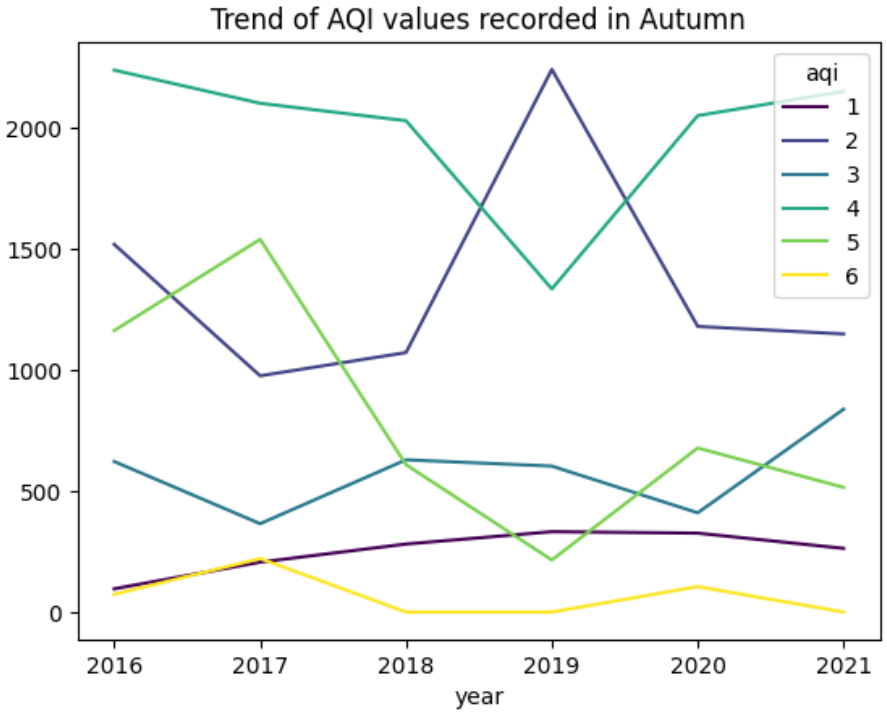

Figure 2. Trend of AQI values recorded in Autumn

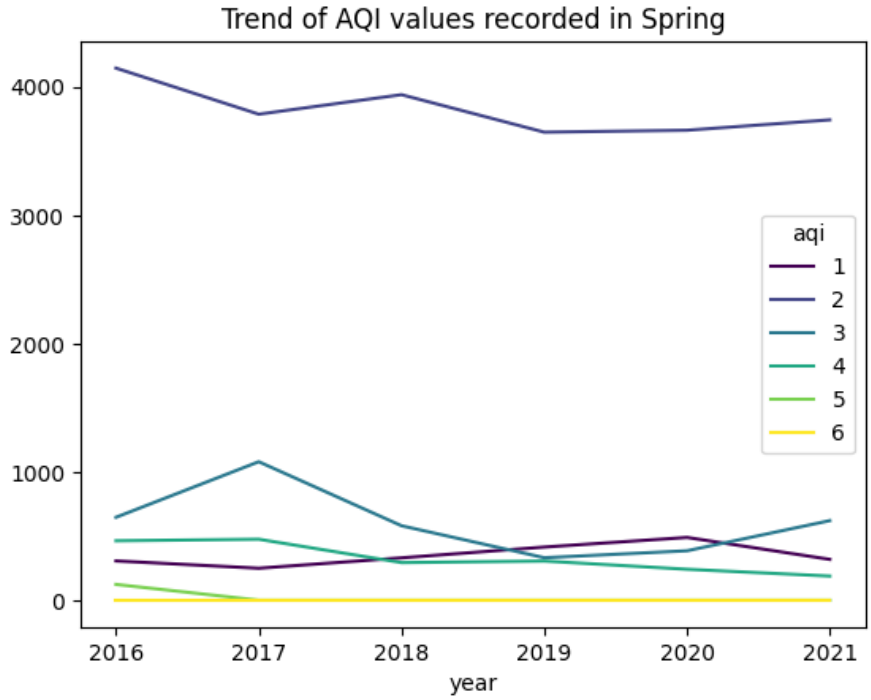

Figure 3. Trend of AQI values recorded in Spring

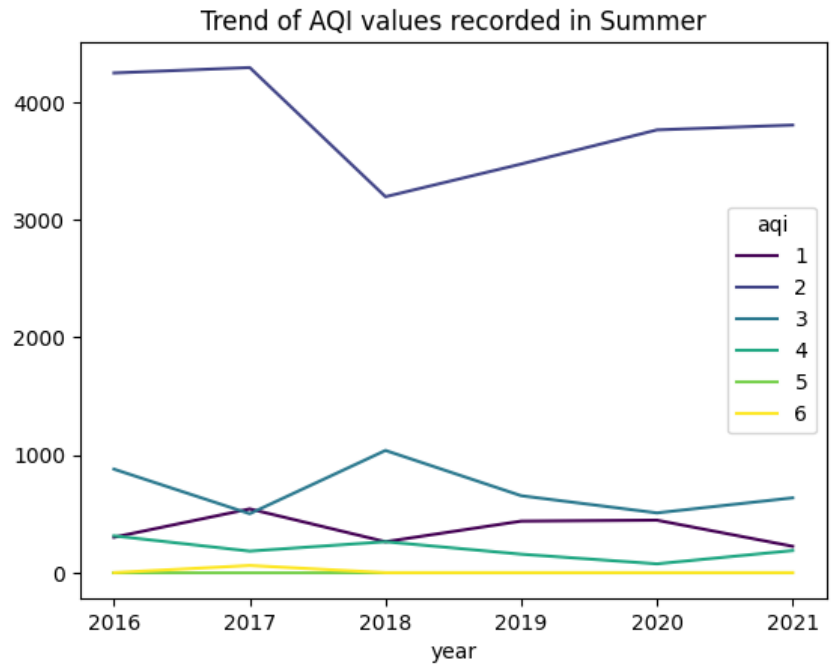

Figure 4. Trend of AQI values recorded in Summer

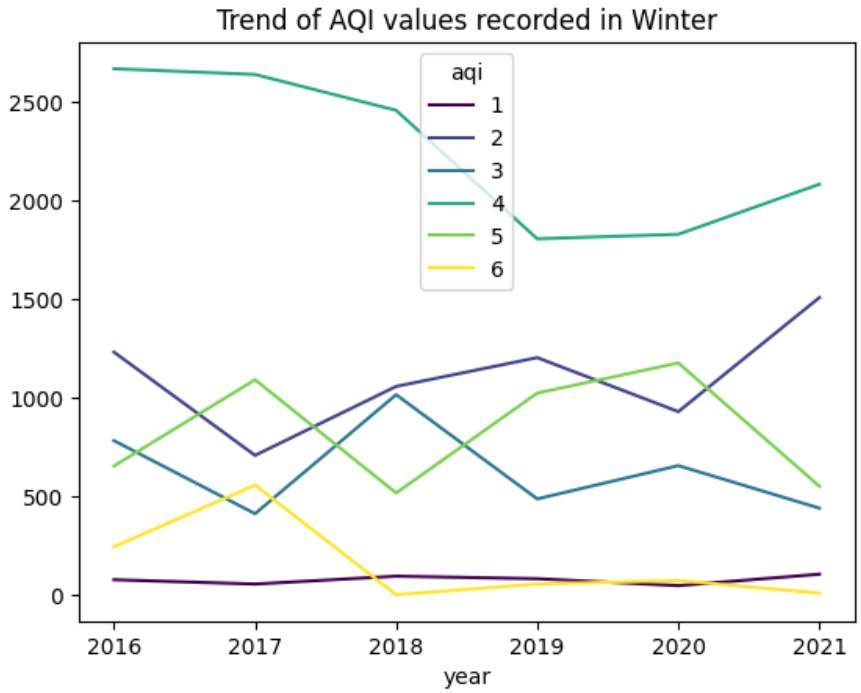

Figure 5. Trend of AQI values recorded in Winter

3. Data Preprocessing

The dataset available for training was transformed from a 121,653 by 62 dataframe to have a shape of 848x61 through a set of preprocessing steps listed below:

1. Breaking data into its constituent years from 2016 to 2021.

2. Further dividing yearly data into the distinct seasons. For the purpose of this work, the definition of the various season is shown below

   Table 2. The definition for the different seasons recorded

Season	Code	Description
Winter	1	January to March
Spring	2	April to June
Summer	3	July to September
Autumn	4	October to December

3. Condensing multiple records of a particular location across different days but within the same season and year into a single observation. To achieve this:
   - Identifiers are created for each unique location using a stringified combination of the location's longitude('lng') and latitude('lat') values.
   - Numeric components of the observations are aggregated by their mean value
   - The AQI value for the individual observations are aggregated by their modal value
4. The condensed data for each season are then combined to obtain a yearly data.

5. Yearly data were then matched with data from the previous year beginning from 2017 whose previous year is 2016.

6. Matched yearly data is then combined into the preprocessed data.
   Note that the AQI value from the matched previous year is renamed as 'aqi_y' to differentiate from the target value of 'aqi'.

4. Proposed Model

4.1 Model Structure

The proposed model is structured as a three-step scikit-learn pipeline:

1. A StandardScaler object for scaling the data and making it more suitable for machine learning algorithms.

2. A SelectKBest object to select the most important features for predicting air quality index.

3. A RandomForestClassifier instance to fit and learn the complex relationships between the processed features and the target AQI values.

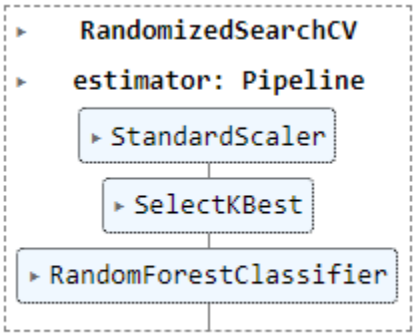

Figure 6. Structure of the Proposed Model

4.2 Training the Model

The choice of RandomForestClassifier was made as a result of searching across a number of models. To streamline the model search process, a function was created that allowed multiple models to be trained on the same data. The code snippet below shows the function.

```
# Defining function that will run the fitting of the model
def runmodel(model,tuning_params,
             scorer=make_scorer(accuracy_score),n_iter=60):
    pipe = Pipeline(steps=[
        ('sc',StandardScaler()),
```

```
        ('feature_selection',SelectKBest()),
         ('classifier',model)
       ])
    r_search = RandomizedSearchCV(pipe,tuning_params,n_jobs=-1,verbose=-1,
                                  scoring=scorer,cv=10,n_iter=n_iter,
                                  random_state=2)
    r_search.fit(X_train,y_train)
    return r_search
```

Code snippet 1. Custom function to run model fitting and hyperparameter tuning

The search process involved fitting each model to the data and tuning the hyperparameters by use of RandomizedSearchCV. In total, 11 models were trained and evaluated on the data.

4.3 Evaluating the Model

The table below shows the result of evaluating the several models on a train and validation set of data - 30% validation, 70% train. This split was achieved by a random split of the 848 processed observations.

Table 7. Evaluation results of models

	Model	Test_Scores	Train_Scores
0	RandomForestClassifier	0.976471	0.985294
1	XGBoostClassifier	0.976471	0.983824
2	VotingClassifier	0.970588	0.986765
3	GradientBoostingClassifier	0.970588	0.983824
4	SVC	0.970588	0.983802
5	LogisticRegression	0.964706	0.986765
6	KNeighbors Classifier	0.964706	0.979390
7	SGDClassifier	0.964706	0.973486
8	AdaBoost	0.952941	0.952809
9	DecisionTreeClassifier	0.947059	0.943986
10	ExtraTreesClassifier	0.947059	0.943986

Based on the evaluation of the various models, as seen above, RandomForestClassifier was selected due to its prediction accuracy on the train and validation data. After the evaluation on the final set of test data, the model predicted with an impressive accuracy score of 0.991071428.
Below is a figure with a horizontal bar plot showing the 10 features that the model depended on the most for predicting the AQI values:

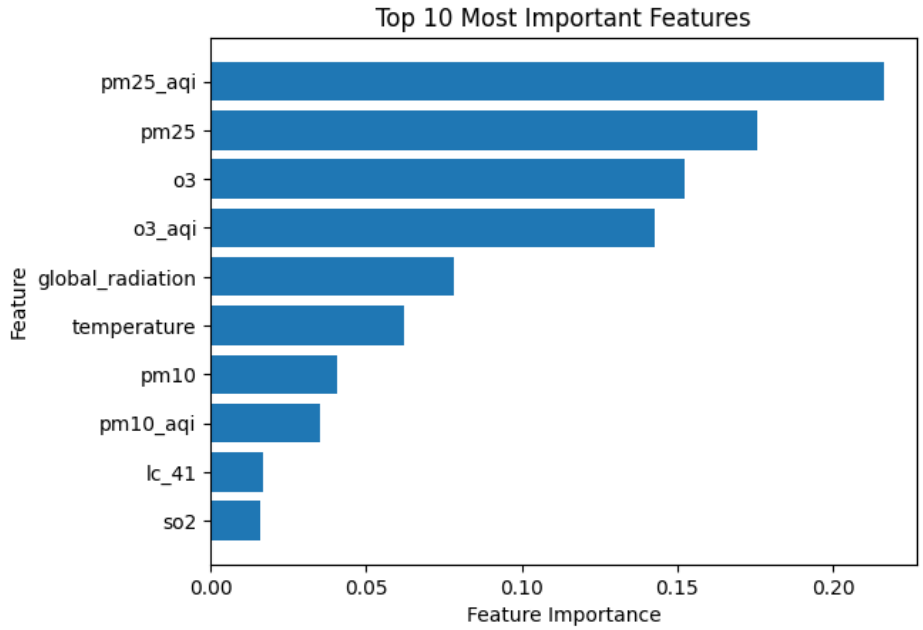

Figure 7. Top 10 important features for predicting AQI

5. Conclusion and Future Work

This solution demonstrates an effective approach to predicting air pollution susceptibility using machine learning. By utilizing meteorological and spatial data, feature engineering, and a RandomForestClassifier model, accurate predictions can be made for AQI values.

The main packages used in the Python environment for this project are:

- pandas (1.5.3) [3]
- numpy (1.23.5) [4]
- scikit-learn (1.2.2) [5]
- scipy (1.11.3) [6]

The notebook can be executed on Google Colab with a CPU runtime, and the entire process takes less than 10 minutes.

Due to the nature of the test data, we couldn't use intra-year data for predictions. For instance, data from winter 2022 wasn't used to predict AQI in summer 2022. In future work, this aspect should be considered, as using intra-year data would provide the model with more training samples, potentially improving prediction accuracy. BaggingClassifier should also be considered in future work.

References

1. European Air Quality Index
2. GeoAI Challenge for Air Pollution Susceptibility Mapping - Zindi
3. What's new in 1.5.3 (January 18, 2023) — pandas 2.1.1 documentation
4. NumPy 1.23.5 Release Notes
5. Version 1.2.2 — scikit-learn 1.3.1 documentation
6. Release Notes — SciPy v1.11.3 Manual