Predicting Air Quality Index using Machine Learning

May: 2024

Karthik kurakula
19990121-4677
kaku21@student.bth.se

***Abstract*— As air pollution refers to introducing chemicals into the atmosphere that harm human health and the environment. So, this report presents the implementation of a system designed to predict Air quality based on the affecting parameters, using machine learning techniques. As part of its approach, this system will utilize a Tkinter, to receive input for data, and, numpy, pandas, matplotlib, and Scikit-learn technologies to process, and predict air quality results efficiently.**

***Index Terms*—Air Quality Index (AQI), Prediction Model, Machine Learning, Graphical User Interface (GUI), Public Health, and Environmental Monitoring.**

I. Introduction

Air pollution is one of the top most challenging problems at present. It is important to note that many dangerous gases like CO2, NH3, and NO, are released into the atmosphere as the result of accidents, natural disasters, and many more factors. Some of the significant environmental issues that can adversely affect human health such as asthma, allergies, infections, and risk of low birth are mostly due to air pollution. Most of the health issues are linked to air pollution, particularly caused by the traffic. Therefore, acquiring data about air pollution, and measuring, assessing, and formulating evaluation models has been increasingly popular in the event of pandemics.

Air quality Index(AQI) is an assessment tool that is used for measurement. A higher level of AQI indicates that the place is most dangerous for the general public. Using traditional techniques is proven to be very complex and less effective, so the ML advanced technologies are used. In this project, we have used data science software tools and modern programming to work with large amounts of data by utilizing the specific functions and commands for data analysis, and visualization with Python language. In this procedure, the system determines the air quality if we give an input as a city name it results in the visualization of the level of all pollutants in that particular city. This technique is useful for people who travel to various cities to check the air quality as it causes multiple human diseases. Identifying the pollution sources will be helpful for environmental conservation, biodiversity, and habitat protection. So, identifying the source of pollution can assist in implementing the measures useful for improving air quality.

To make accurate predictions in the concept of air quality. we used Random Forest, logistic regression, and SVM with the help of the Scikit library, numpy, and pandas libraries. The web page implementation was done using Tkinter, a graphical User interface using Python. The rest of the report was arranged such that section II consists of the Division of labor, section III of project analysis, section IV consists of the project's design, section V consists of the implementation of the project, and section VI consists of the conclusion.

II. Division of Labour

Our DSS project "Predicting Air Quality Index Using Machine Learning Techniques" has been divided into four parts: "Programmer," "Designer," "Reviewer," and " Reviser," and issued to our team members. The tasks are given out based on individual interests and capabilities. The table below illustrates how work is divided among team members.

S No	Name	Task
1.	Karthik Kurakula	Designer, Programmer,Reviser
2.	Suma Vendrapu	Programmer, Docoumenter, Reviewer
3.	Lakshmi Vyshanvi Nerella	Designer, Documenter,Reviewer

Tasks performed during the creation of the DSS system

- The problem has been investigated and a project was picked.
- Considering the problem statement and essential solution.
- Division of tasks and time management.
- Selecting suitable datasets and machine learning models.
- Testing with different machine learning models to find the most suitable one.
- Developing the final machine learning model and program.
- Documentation for the graphical user interface of the system.
- Implementing the system's GUI with the Tkinter module and connecting it to the code.

- The feedback factor evaluates output and modifies the system.
- Schedule a meeting with the lecturer and a teammate to review performance after each task.
- Documenting and presentation.

III. PROJECT ANALYSIS

A. *Background*

The project "Predicting Air Quality Index Using Machine Learning" makes a crucial global issue: air pollution and its harmful effects on public health and the environment [1]. Since the 1970s, environmental agencies have focused much attention on air quality monitoring due to growing concerns about the health impacts of air pollution. The Air Quality Index (AQI) was established to present a comprehensible and accessible indicator of the severity of air pollution. The monitoring of air quality has advanced significantly over the years, but predicting future air quality remains difficult. Since the development of machine learning technology over the last 20 years, there is now a chance to take advantage of these advanced techniques for predicting AQI more effectively and accurately. [3]

Predicting air quality using machine learning has been increasingly popular, particularly since the early 2010s [4]. It has been demonstrated that machine learning algorithms are more efficient than conventional statistical techniques because they can handle large amounts of data and find complex trends. Machine learning algorithms can generate AQI predictions that are reliable by using historical air quality data with related variables, such as weather conditions, and industrial pollutants [2]. The goal of this project is to use these skills to create an effective predictive model that can predict air quality for various countries and cities.

The "Predicting Air Quality Index using Machine Learning and Tkinter" project prevents air pollution by combining complex analytics with a user-friendly interface [4]. It uses machine learning to predict AQI for different places by evaluating large datasets. By using a Tkinter-based GUI, these predictions are accessible and understandable to users, who can select a country and city, view AQI forecasts, and visualize data using interactive graphs. Furthermore, there is an option for users to provide feedback on prediction accuracy, which assists in improving the model. The usage of machine learning, a simple GUI, and a feedback loop improves the air quality predicted and encourages public participation in managing the environment.

B. *Problem Definition*

The project's main purpose is to create a Python GUI that displays the air quality index based on user input, such as city name. The Python GUI displays pollution levels per city, allowing users to choose a more suitable location. In addition, our GUI displays pollutant levels such as good, bad, average, hazard, and primary pollutant levels such as SO2, CO2, etc based on the city's air quality. Our project helps to decide whether people check the air quality index before going to any city

C. *Problems Objective*

- Collect and clean the data on air pollution levels, weather conditions, and other important factors.
- Identify significant information and select the most effective machine learning techniques.
- Develop the model with training data and evaluate its accuracy using evaluation criteria.
- Set up the model to provide real-time predictions and ensure they are clear and understandable.

D. *Problem Solution*

To predict air quality using machine learning, we collect and process historical data from different sources, such as pollution quantities and weather-related factors. We proceed with data exploration and feature design to choose and modify relevant features. Several machine learning models are created and evaluated using performance measures to determine which one is better. Finally, the selected model is implemented using a GUI that allows users to visualize actual predictions and historical patterns, ensuring continuous accuracy monitoring and improvement.

E. *System Criteria*

To "Predicting Air Quality Index Using Machine Learning Techniques" project used many tools and approaches to construct a decision support system.

- **Jupyter Notebook**: The Jupyter Notebook is a helpful device for running large amounts of machine-learning code. Also, machine learning libraries such as "Numpy," "Seaborn," "SKlearn," and "Pandas" can be used. Using this application, you can also keep the code and the dataset in separate folders, which reduces the workload on the user.

- **Python Programming**: Python is the most popular programming language for machine learning due to its large range of built-in libraries. These Python-based libraries offer a varied and comprehensive code experience, making them the best fit for machine-learning tasks.

- **Datasets**: We have chosen this dataset from the website **IQAir Air Quality Map** and it represents as follows
  - country_name
  - city_name
  - aqi_value
  - aqi_category
  - co_aqi_value
  - co_aqi_category

- ozone_aqi_value
- ozone_aqi_category
- no2_aqi_value
- no2_aqi_category
- pm2.5_aqi_value
- pm2.5_aqi_category

- **Graphical User Interface(GUI)**: The GUI is created using the Python module Tkinter. There is a powerful object-oriented interface provided by the Tkinter Python GUI module.

- **Feedback**: A feedback option is provided at the end of the GUI for our project. Feedback involves inquiries about the GUI and the procedure. User feedback can inform improvements in the future.

  This method reached a high level of accuracy, which directly affects the air quality of prediction. The dataset containing real-world air pollution data greatly assisted in the development of the machine learning model. A GUI for displaying air quality indices was created using the Tkinter module.

F. SWOT Analysis

SWOT(strength, weakness, opportunities, Threats) analysis conducted for the project is discussed in this section.

Strength	Weakness
Jupyter Notebook	All cities data is not available.
Affordable/ cost-effective	Delay in data updates impact more on the accuracy and predictions of the air quality.
Using advanced machine learning models that increase the accuracy and reliability of air quality forecasts	It isn't easy to integrate data from multiple sources to ensure smooth compatibility.
User-friendly interface which easily interprets and acts upon the data and predictions.	For the regular assistance and training for users, more resources are required for operating the system effectively.

opportunities	Threats
Combining various datasets and increasing the number of data sources can increase accuracy and prediction values.	False predictions due to bad data quality may result in useless outcomes.
Improving the prediction model with advanced technologies can result in sophisticated and exact predictions.	Threats associated with data manipulation or cyber attacks might affect the accuracy and quality of predictions.

IV. PROJECT DESIGN

A. Use case diagram

A use case diagram is a visual representation of the functional requirements of a system. The UML (Unified Modeling Language) shows the interactions between the users and the system to achieve a specific goal. The purpose of the use case diagrams is:

- This section of the project is useful for identifying and defining the system's functional requirements.
- It helps design the system by demonstrating the system's needs and helps the system design process for the users.
- A Use case diagram is a document that gives detailed information about the functional requirements and the interaction between the system components.

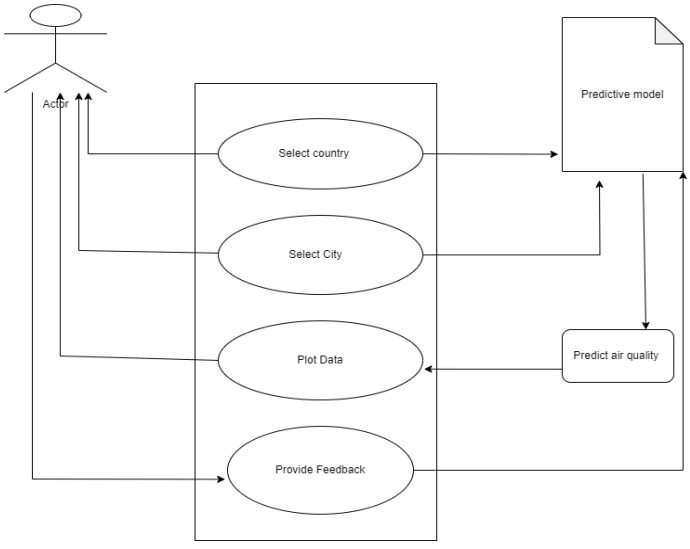

Fig. 1. Use case Diagram

B. *Activity diagram*

The flowchart is a visual representation of a process, from the data collection and preprocessing to the training, testing, development, and deployment. This flowchart is also known as an activity diagram.

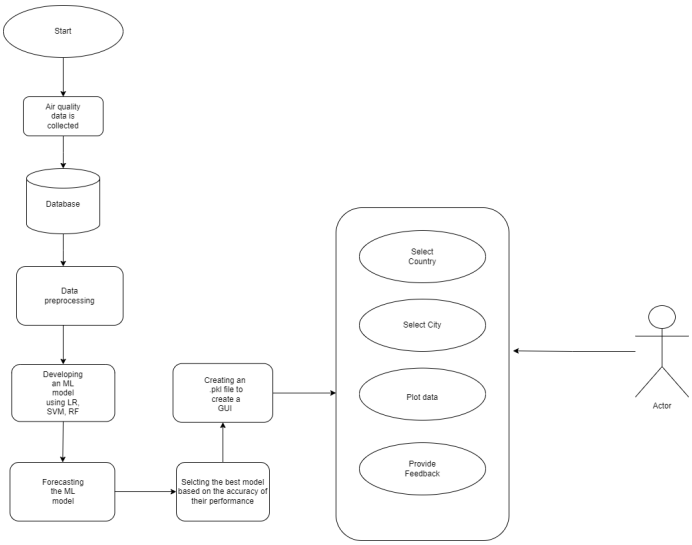

Fig. 2. Flow Chart

C. *Model Description*

Using the GUI, the user will find it easy and intuitive to navigate. The user can view AQI predictions by selecting a country and a city. The GUI provides this information to the machine learning model, which analyses using a database of historical air quality data, and weather conditions stored in a CSV file. The system predicts the AQI utilising the input and shows the results in the GUI. Users can also explore interactive graphs of historical and predicted AQI results. The GUI includes a feedback option, which allows users to constantly input the accuracy of predictions, assisting in the model's refinement and improvement in the future. Using this method, users will receive data about air quality that is accurate and actionable in an accessible format.

V. Project Implementation

Standard libraries are used in this project to analyze datasets and build models. The following is a list of the libraries that were used:

- **Pandas:** Used for data modification and analysis. It includes data structures such as Data Frames, which are required for processing structured data.

- **NumPy:** A basic Python module for scientific computation. It supports massive, multidimensional arrays and matrices, as well as a vast set of high-level mathematical functions for manipulating these arrays.

- **Matplotlib:** A Python plotting package that allows you to create static, animated, and interactive visualizations. It is commonly used for creating plots, histograms, power spectra, bar charts, error charts, scatter plots, and so on.

- **Scikit-learn:**A Python machine learning package that includes easy and fast tools for data mining and analysis. It is constructed with NumPy, SciPy, and Matplotlib.

  Train-test split divides the dataset into training and testing set .Standard Scaler standardizes characteristics by eliminating the mean and scaling to unit variance. Label Encoder encodes target labels with values ranging from 0 to n_classes-1. Random Forest Classifier, Logistic Regression, and SVM(Support Vector Machine) are used to create machine learning models. Metrics used to evaluate model performance include accuracy_score, precision_score, recall_score, f1_score, and confusion_matrix.

- **Tkinter;** Tkinter is a typical Python GUI (Graphical User Interface) module. Tkinter is the most used library for generating graphical interfaces in Python. It is used to generate the main window, frames, labels, buttons, and other widgets. The ttk (Themed Tkinter) module allows access to the Tk-themed widget set.

A. *Data Set*

We have chosen this dataset from the website **IQ Air Quality Map**. This dataset contains air quality data for cities worldwide, including AQI values and classifications for pollutants like CO, O_3, and NO_2. Each row indicates a city and contains the overall AQI, pollutant-specific AQIs, and their related classifications (e.g., Good, Not good). The data show cities with varied pollution levels, for example, Praskoveyevka, Russia, with a moderate AQI of 51, and Qalyub, Egypt, with an unhealthy AQI of 142. Using this dataset, we can examine air quality trends to develop environmental and public health approaches.

B. *Graphical User Interface(GUI)*

To make air quality data analysis more accessible, we created a graphical user interface (GUI) with Tkinter, a common Python GUI framework. The GUI is designed with usability and visual appeal in mind, and it includes many major components:

- **Full-Screen Mode:** The program launches in full-screen mode, providing a large and detailed view of the data and interface components.

- **Background Image:** A backdrop picture was added to the GUI to improve its visual appeal and provide a more engaging user experience.

- **Dropdown Menus:** The GUI has combo boxes for choosing countries and cities. This enables users to dynamically filter the data based on their preferences.

- **Data Plotting:** Users can build visual representations of the selected city's air quality measurements by clicking the 'Plot Data' button. The resultant figure shows the overall AQI as well as pollutant-specific AQIs, with comments showing air quality and the top contaminant.

- **Feedback Dialog:** Users can submit feedback using a dedicated dialog box. This feedback is taken and saved in a text file, enabling for continual enhancement of the program depending on user input.

In the figure 2 we can see the GUI generated

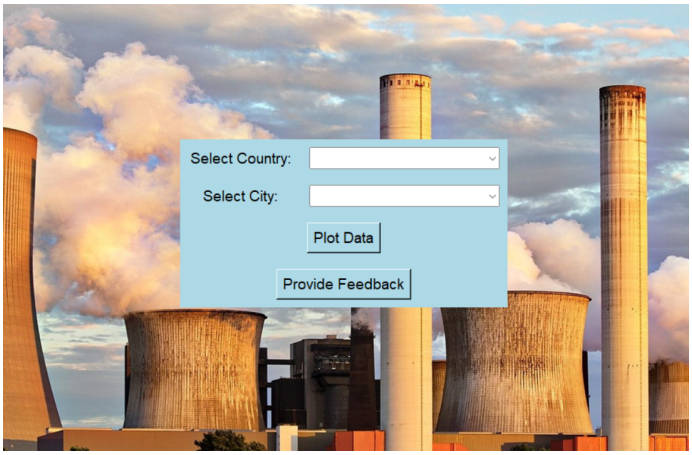

Fig. 3. GUI

C. *Machine Learning Model*

To improve the application's functioning, we added multiple machine learning models that predicted air quality classifications and provided deeper insights into the data. The implementation procedure included the following steps:

- **Data Preprocessing**
- **Encoding Categorical Features:** We employed label encoding to turn category data into numerical values that may be used for model training.
- **Feature Standardization:** The characteristics were standardized to promote uniformity in the model training process, resulting in improved model accuracy and performance.

- **Model Selection**
- Random Forest Classifier: This ensemble learning approach creates numerous decision trees during training and outputs the mode of the classes for classification problems. It is extremely accurate and can process a huge number of input variables.
- Logistic Regression: A statistical model that uses a logistic function to describe a binary dependent variable. It is useful for binary classification jobs and offers probabilistic interpretations.

- Support Vector Machine (SVM):A supervised learning model designed for classification and regression problems. SVMs are successful in high-dimensional spaces and may represent complicated feature connections.

```
models = {
    'RandomForest': RandomForestClassifier(random_state=42),
    'LogisticRegression': LogisticRegression(random_state=42, max_iter=1000),
    'SVM': SVC(random_state=42)
}

fitted_models = {name: model.fit(X_train, y_train) for name, model in models.items()}

def evaluate_model(model, X_test, y_test):
    y_pred = model.predict(X_test)
    accuracy = accuracy_score(y_test, y_pred)
    precision = precision_score(y_test, y_pred, average='weighted', zero_division=0)
    recall = recall_score(y_test, y_pred, average='weighted', zero_division=0)
    f1 = f1_score(y_test, y_pred, average='weighted', zero_division=0)
    cm = confusion_matrix(y_test, y_pred)
    return accuracy, precision, recall, f1, cm

evaluation_results = {}
for model_name, model in fitted_models.items():
    accuracy, precision, recall, f1, cm = evaluate_model(model, X_test, y_test)
    evaluation_results[model_name] = {
        'accuracy': accuracy,
        'precision': precision,
        'recall': recall,
        'f1': f1
    }
```

Fig. 4. Models Training

- **Training and Evaluation**
The dataset was partitioned into training and test sets to assess the models' performance. Each model was trained on the training set and tested on the test set, with metrics including accuracy, precision, recall, F1 score, and confusion matrix. These measures gave a full assessment of each model's performance, allowing us to choose the most effective model for making predictions.
- **Results**
By comparing the assessment indicators, we determined which model performed better. The chosen model was then applied to forecast fresh data, yielding significant insights into air quality trends in other cities. Figure 3 is the output of the city of Hyderabad, India.

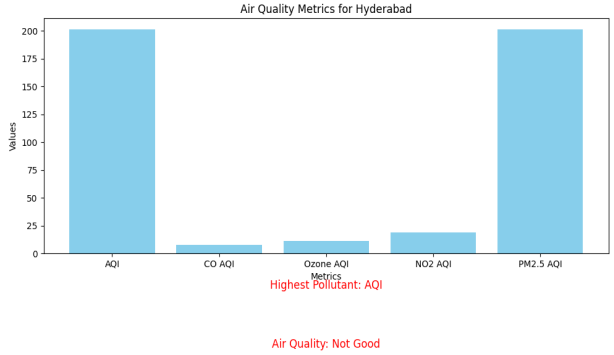

Fig. 5. Predicted Graph

VI. Conclusion

This research effectively uses machine learning and data visualization approaches to assess air quality data. The user-friendly GUI, paired with sophisticated machine learning algorithms, allows users to efficiently investigate air quality patterns across several cities. This application not only improves our understanding of air quality, but it also helps us build plans to improve environmental health and public safety. The program is a significant resource for academics, policymakers, and the general public, providing a simple and interactive method to interact with air quality data.The presentation of outcomes and data is pictorial, making it easier to interpret. Implementing the "Feedback" function allows developers to update suggestions for future versions of the application/GUI.

To improve the DSS implementation, provide a fully functional GUI and remove the local database to allow for data updates. Using various machine learning techniques and analyzing massive volumes of data helps enhance model accuracy.

References

[1] M. Bajpai, T. Jain, A. Bhardwaj, H. Kumar, and R. Sharma, "Air quality index prediction using various machine learning algorithms," in *6G Enabled Fog Computing in IoT: Applications and Opportunities*. Springer, 2023, pp. 91–110.

[2] T. Madan, S. Sagar, and D. Virmani, "Air quality prediction using machine learning algorithms–a review," in *2020 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)*. IEEE, 2020, pp. 140–145.

[3] N. N. Maltare and S. Vahora, "Air quality index prediction using machine learning for ahmedabad city," *Digital Chemical Engineering*, vol. 7, p. 100093, 2023.

[4] D. Tang, Y. Zhan, and F. Yang, "A review of machine learning for modelling air quality: Overlooked but important issues," *Atmospheric Research*, p. 107261, 2024.