Student Name : T.KIRUBAKARAN

Register Number : 422623104704

Institution :UNIVERSITY COLLEGE OF ENGINEERING

Department : COMPUTER SCIENCE AND ENGINEERING

Date of Submission : 29.04.2025

Problem Statement :

Air pollution poses a significant threat to public health, ecosystems, and the global climate. Traditional air quality monitoring systems often rely on sparse, costly sensor networks and reactive measures, which can limit timely interventions. There is a growing need for accurate, real-time, and predictive insights into air quality levels to enable proactive environmental management and policy decisions. Leveraging advanced machine learning algorithms can enhance the ability to model complex environmental data, predict pollution levels, and uncover hidden patterns. This project aims to develop a predictive model for air quality using machine learning techniques,

utilizing historical and real-time environmental data to generate accurate forecasts and actionable insights for improved public health outcomes and environmental planning.

This project aims to address this gap by using advanced machine learning algorithms to **predict future air quality levels** based on historical and environmental data. With accurate predictions, governments, organizations, and individuals can take early actions to reduce exposure and prevent health risks, making this a highly relevant and impactful problem to solve.

Objectives of the Project:

- **To design and implement machine learning models** (e.g., regression, decision trees, neural networks) to predict future air quality levels.
- **To visualize trends and predictions** in a user-friendly format (e.g., dashboards or charts).
- **To provide actionable insights** for public health officials, environmental agencies, and citizens.
- Develop a machine learning model to predict the air quality based on environmental parameters like temperature, wind, and solar radiation.
- Classify air quality into meaningful categories such as "Good," "Moderate," and "Unhealthy."
- Analyze feature importance to understand which factors most affect air quality.

- Provide actionable insights to stakeholders based on predictive analytics.

Scope of the Project :

In Scope:

- **Features Analyzed**: Ozone concentration ,solar Radiation, Wind speed, Temperature ,Month ,Day.
- **Model Used**: Random Forest Classifier
- **Output**: Air quality category prediction ("Good" ,"Moderate" ,"Unhealthy").
- **Constraints**:
  - Collection of historical air quality and weather data from reliable sources.
  - Preprocessing and cleaning of data for model training.
  - Implementation of multiple ML algorithms for comparison.
  - Forecasting air quality levels for short-term (next few hours/days).
  - Visualization and interpretation of results.

Data Sources :

- Dataset :air_quality_dataset-1.csv
- **Source**: Publicly available environmental datasets (similar to UCI Air Quality dataset).
- **Type**: Static dataset (once downloaded, no real-time updates).

- **Nature:** Pre-processed version tailored for educational and demonstration purposes.

High-Level Methodology :

Data Collection :

- **Source**: I will use publicly available datasets such as:
- **UCI Air Quality Dataset** (downloadable CSV)
- Load the dataset manually using python's pandas library

Data Cleaning :

- **Missing Values**: Will handle using techniques like mean/median imputation, interpolation, or deletion depending on the context.
- **Duplicates**: Identify and remove any repeated rows to avoid skewing the results.
- **Inconsistent Formats**: Standardize units (e.g., $\mu g/m^3$), date-time formats, and column names.
- **Outliers**: Detect using statistical methods (IQR or Z-score) and treat or remove based on domain knowledge.
- **Drop**: drop missing values values using dropna().
- Remove irrelevant columns like "rownames".

Exploratory Data Analysis (EDA) :

- Techiquies:
  - Use seaborn and matplotlib for visualizations.

- Analyze relationships between Ozone levels and other environmental factoers.
- Generate bar charts,scatter plots,and correlation heatmaps.

➢ Feature Engineering :

❖ New Features :

- **Datetime Features**: Extract hour, day, month, weekday, etc.
- **Feature:**Create a new feature called AQI_Category based on Ozone levels
- Categorize air quality into “Good”,”Moderate” ,”Unhealthy”.

➢ Model Building :

❖ Algorithms to Use :

- **Linear Regression**: For a simple baseline.
- **Random Forest Regression**: Use Random forest Classifier due to its robustness ,ability to handle overfitting, and feature importance extraction.
- Experiment with hyperparameters (number of trees ,depth).
- Ensemble models like Random Forest are robust and handle noisy, real-world data well.
- Time-series specific models (like LSTM) are ideal if long-term forecasting is a key goal.

➢ Model Evaluation :

❖ Evaluation Model performance using :

- Accurate Score
- Classification Report(Precision,Recall,F1-Score)

➢ Visualization & Interpretation :

❖ Tools :

- Matplotlib, Seaborn, Plotly for interactive charts.

❖ Visuals :

- Line charts for predictions vs actual values.
- Feature importance graphs (for Random Forest/XGBoost).
- Heatmaps and bar charts for pollutant trends.
- Dashboards (optional) using Streamlit or Tableau for interactive results.
- Use confusion matrices and bar charts for results interpretation.

➢ Deployment :

- Use **Streamlit** or **Flask** to build a simple web app where users can input current data and get air quality predictions.
- Alternatively, create a **Jupyter Notebook** with clean output cells, interactive plots, and documentation for sharing insights.

Tools and Technologies :

➢ Programming Language :

❖ Python :

- Chosen for its simplicity, readability, and wide range of libraries for data analysis and machine learning.

❖ Google Colab :

- Cloud-based, free to use, supports GPU acceleration, and allows easy collaboration.
- Alternatively, you can use Jupyter Notebook or VS Code if working locally.

❖ Libraries :

- pandas – For data manipulation and analysis.
- numpy – For numerical operations.
- datetime – To handle and process date-time features.

❖ Data Visualization :

- matplotlib – For creating basic plots (line, bar, scatter, etc.).
- seaborn – For enhanced statistical visualizations like heatmaps, boxplots, etc.
- plotly – For interactive and dynamic visualizations

❖ Machine Learning & Modeling :

- scikit-learn – For preprocessing, regression models, and evaluation metrics.
- statsmodels – For basic statistical modeling (optional)

❖ Optional Deployement:

- Deployement is optional.
- If deployed,Streamlit or flask can be consider to create a simple web-based application.

Team members and roles:

NAME	ROLE	RESPONSIBILITIES
A.Abibrisckilla	Project leader	Oversee project development,coordinate team activities,ensure timely delivery of milestone,and contribute to documentation and final presentation
R.Ragul	Data manager	Collect data from APIs (e.g:twitter),manage dataset storage,clean and preprocess

		text data,and ensure quality of input data.
T.Jayasudha	EDA and visualization specialist	Performing exploratory data analysis(EDA)-creating detailedgraphs(bar plot,pie,chart,heatmap,scatter plot)-interpreting trends and relationships in data
N.Dawoodkhan	Research and Documentation Manager	Writing the problem statement,objectives,and methodology preparating and organizing the final report-researching best practices and algorithms
T.Kirubakaran	Model Evaluation and Deployment Assistant	Evalution model performing using metrics like accuracy and classification report-Assisting with deployment readiness(if needed)-Summarizing model results

and suggesting improvements.

A.ABIBRISCKILLA

R.RAGUL

T.JAYASUDHA

N.DAWOODKHAN

T.KIRUBAKARAN

- Gradio – To create simple user interfaces for ML models (optional).
- FastAPI – For building high-performance APIs (optional, more advanced).

Team Members and Roles :