WEBAI AQI - THE AIR QUALITY MONITORING PLATFORM

Team: FusionX

Abhijeet Rajhans
Manipal University Jaipur
abhijeetrajhans.ar@gmail.com

Shreyasi Ray
Manipal University Jaipur
shreyasiray11@gmail.com

IIT KANPUR

BuzzOnEarth
Mainstreaming Sustainability

intel

BuzzOnEarth India Hackathon 2024

PROBLEM STATEMENT

AIR QUALITY MONITORING AND PREDICTION

Implement an AI system that continuously monitors air quality across the city at real time, predicts pollution levels, and suggests actionable measures to mitigate the effects of poor air quality on residents' health.

- Air pollution in urban areas poses serious *health risks* while also hindering *sustainable development*.
- Accurate *monitoring* and timely intervention are crucial for minimizing environmental impact and ensuring cities remain livable and productive.
- Thus, the project caters to these issues by utilizing AI techniques to predict pollution levels by *calculating AQI* and provide *real-time alerts* to users.
- Serving as both an *information hub* and an interactive platform, the project leverages data-driven *models* and user-friendly interfaces to keep citizens informed, enabling timely actions and fostering *discussions* among users. Ultimately, the project contributes to better environmental and public health outcomes through collaboration and awareness

SOCIAL, ECOLOGICAL AND ECONOMIC IMPACT

The project fosters community action through *real-time AQI updates*, Reducing AQI *suggestions,* and anonymous *discussion* forums.

The *chatbot* and *visualizations* make information accessible, promoting environmental awareness

Socially, it raises awareness, helping citizens make informed decisions to protect their health.

Ecologically, it promotes pollution reduction and a cleaner environment by targeting key pollutants.

Economically, it can lower healthcare costs by reducing pollution-related illnesses and encourage sustainable practices that boost productivity.

PROMOTING SUSTAINABILITY

- For every AQI level, offering tailored suggestions to reduce emissions at different AQI levels.
- Enabling anonymous collaboration in forums, driving collective action.
- Providing real-time updates and notifications to promote immediate response.
- Visualizing data to increase awareness and drive long-term behavioral change
- Educating users on pollutants and their sources to reduce future emissions.

ESTIMATED IMPACT AND SCALE

- 20% reduction in peak AQI levels through collective action and awareness.
- 10% decrease in healthcare costs related to air pollution
- Improved air quality and ecosystem recovery, leading to more sustainable urban environments.

SOLUTION DESCRIPTION

- Developed an *Air Quality Index (AQI)* Predictor and *Health Advisor* application utilizing *Random Forest Regression* to forecast PM2.5 and PM10 levels based on environmental factors such as CO, SO2, NH3, and more.

- The application integrates *real-time* data processing to deliver actionable insights on air quality.

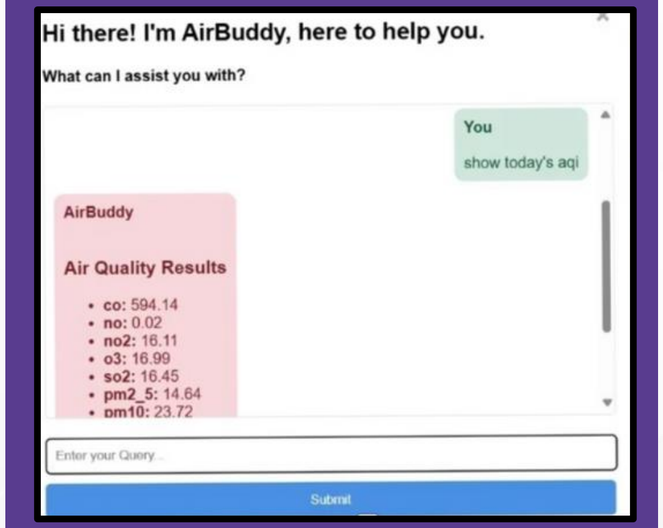
PROBLEM-SOLVING APPROACH AND CORE INNOVATION

Real-Time Insights

- Provides immediate feedback on air quality, empowering users to make informed decisions regarding their health and environment.

- Includes a *severity classification system* that ranks air quality on a scale of 1 to 5, guiding users on necessary precautions.

AI-DRIVEN AQI PREDICTOR AND HEALTH ADVISOR



User Engagement

Features an *email notification system* to alert registered users about current air quality conditions, enhancing community awareness.

LEVERAGING AI IN THE PROJECT

Interactive Chatbot

An AI-powered chatbot responds to general queries about air quality and health, providing users with personalized support and information.

Community Forum with Summarization:

Facilitates a platform for users to discuss air quality issues, with the capability to summarize discussions, allowing for efficient information dissemination and user engagement.

TECHNICAL ARCHITECTURE

General Python Libraries

- *API Interaction*:
  - fetch API: Makes HTTP requests to fetch data (e.g., AQI data).
  - requests: Sends HTTP requests to web APIs for real-time AQI data.
  - json: Parses JSON data for handling API responses.

- *Database & Email*:
  - sqlite3: Interacts with SQLite databases for storing AQI and user data.
  - smtplib: Sends emails via SMTP for AQI updates.
  - email.mime: Formats and sends multipart emails (plain text and HTML).

- *Data Handling*:
  - os: Interacts with the operating system (files, paths).
  - re: Utilizes regular expressions for basic NLP tasks.
  - collections.Counter: Counts hashable objects, useful for tallying pollutants or interactions.

Machine Learning Libraries

- *Data Manipulation*:
  - **numpy**: Performs numerical computations and matrix operations.
  - **pandas**: Manages datasets (e.g., AQI data), cleaning, and organizing for analysis.

- *Modeling & Evaluation*:
  - **scikit-learn**:
    - **train_test_split**: Splits datasets into training/testing sets.
    - **StandardScaler**: Scales data for machine learning models.
    - **mean_squared_error**, **r2_score**: Metrics for evaluating regression models.

- *Regression Models*:
  - **Linear Regression**: Basic linear regression model.
  - **Decision Tree Regressor**: Tree-based regression model.
  - **Random Forest Regressor**: Ensemble model using multiple decision trees.
  - **Gradient Boosting Regressor**: Boosts decision trees for regression.
  - **SVR**: Predicts AQI levels using support vector machines.
  - **MLP Regressor**: Neural network-based regressor for complex relationships.

- *Model Persistence:*
  - **joblib**: Saves and loads trained machine learning models.

TECHNICAL ARCHITECTURE

Intel's Scikit-learn Extension

Optimization Libraries:

- sklearnex: Intel Extension for Scikit-learn that accelerates machine learning workflows using optimized algorithms and Intel hardware.

Patching Scikit-learn:

patch_sklearn():

- Patches Scikit-learn to use Intel-optimized versions of its algorithms.
- Applied by importing and running `patch_sklearn(), speeding up training and inference without modifying existing code.

Optimized Linear Models:

sklearnex.linear_model:

- Contains Intel-optimized versions of linear models, such as `LinearRegression`, `Ridge`, `Lasso`.
- Improves performance on Intel hardware for large datasets while maintaining the standard Scikit-learn interface.

TECHNICAL ARCHITECTURE

NLP & Transformers

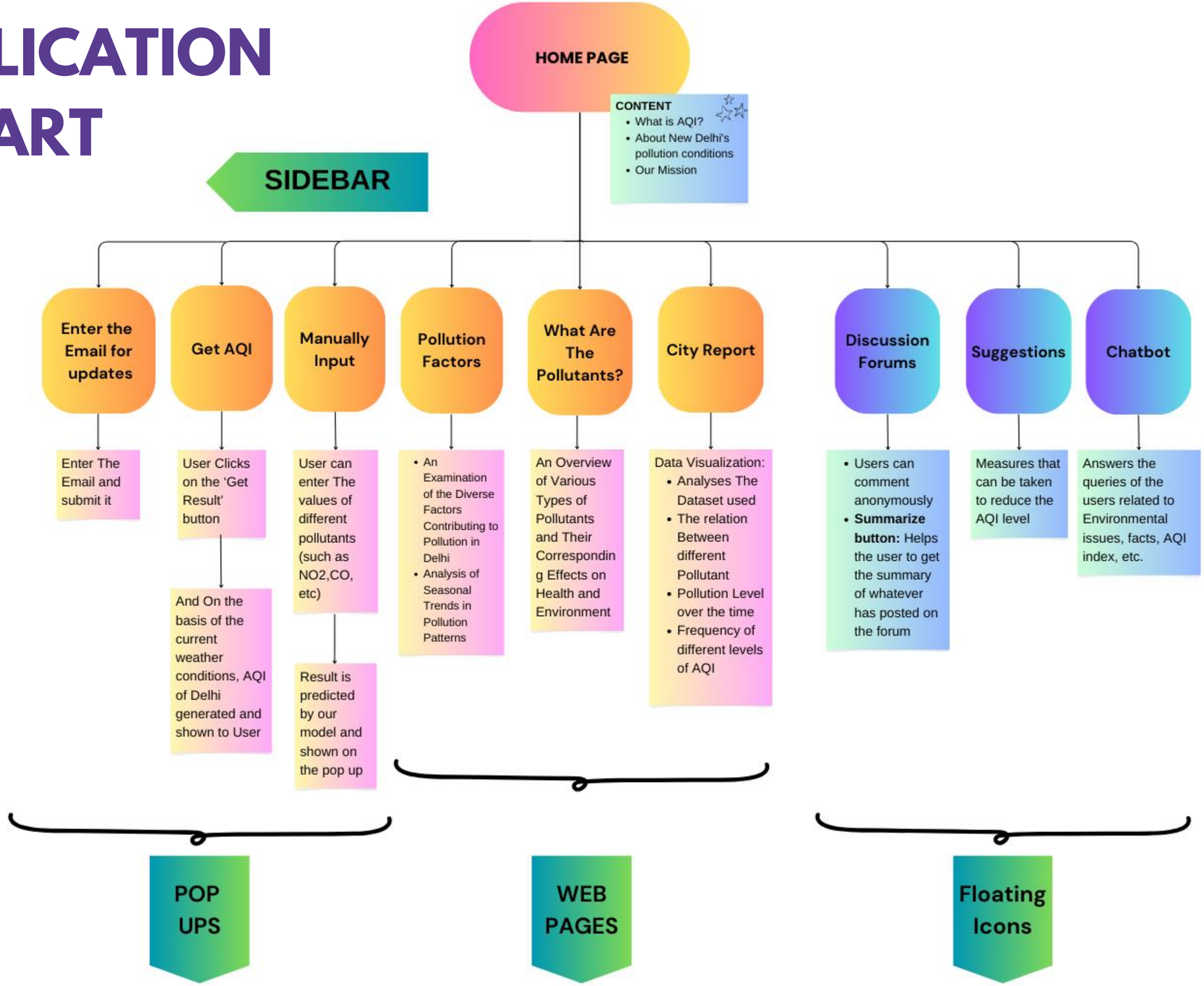
- *Transformers Library*:
  - Provides powerful NLP models for text generation and summarization.

- *Text Generation Models*:
  - **Falconsai/text_summarization**: Summarizes comments or reports.
  - **google/flan-t5-large:** Used for summarization and generating chatbot responses.
  - **gpt2**: Generates real-time textual outputs.
  - **intel/distilgpt2-wikitext2**: Efficient text generation.
  - **Intel/neural-chat-7b-v3-3**: Conversational model for chatbot interactions.

External APIs

- *OpenWeatherMap API*: Fetches real-time AQI and weather data.
- *llma3 API*: Used for NLP tasks like summarization and conversational AI.

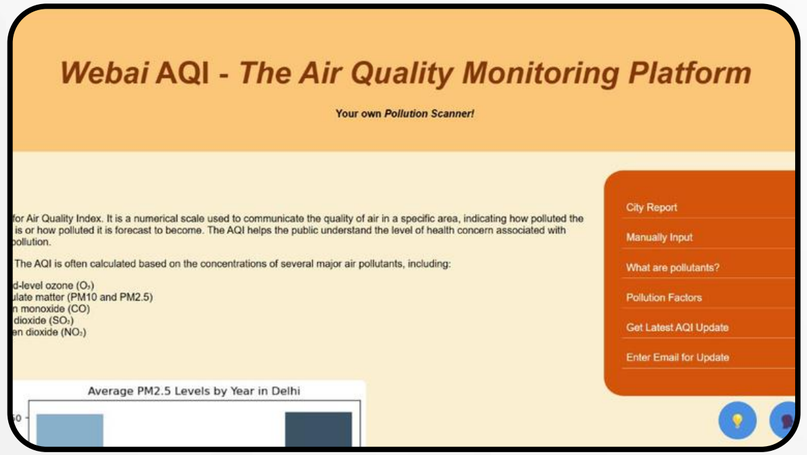
OUR APPLICATION FLOWCHART



FUTURE POTENTIAL AND SCALABILITY

- **Market Opportunity:** Rising demand for air quality monitoring solutions in urban areas due to increasing environmental regulations.
- **Geographic Expansion:** Can easily scale beyond Delhi by integrating local datasets and APIs, adaptable for global deployment.
- **Real-Time Data Integration**: Incorporates IoT-based air quality sensors for hyper-local AQI updates and insights.

POTENTIAL FOR REAL-WORLD DEPLOY



- The system's architecture can be designed for flexibility, allowing it to integrate with existing **air quality sensors**, APIs, and **government portals**. It can also be adapted for **mobile applications**, further increasing its accessibility and user base.
- **Public Health** Initiatives: Easily integrates with healthcare systems to issue alerts for at-risk populations (e.g., children, elderly, and those with respiratory conditions) during high AQI levels.
- **Public Transportation** Systems: Integrates with local transportation to offer pollution-reducing tips, like suggesting public transport usage on high AQI days.

Intel® Tiber Cloud

- Purpose: Scalable AI infrastructure optimized for deep learning and NLP workloads, ideal for high-demand tasks.
- Usage: Powers large models for AQI prediction, text summarization, and real-time data processing.

Intel® neural-chat-7b-v3-3

- Purpose: Advanced conversational model for generating human-like responses.
- Usage: Enhances the chatbot (AirBuddy) for AQI-related queries, creating engaging, contextually relevant conversations.

Intel® distilgpt2-wikitext2

- Purpose: Efficient, lightweight version of GPT-2 for fast text generation.
- Usage: Generates AQI updates and pollution reduction suggestions in real time.

Intel® fid_flan_t5_base_nq

- Purpose: T5 model fine-tuned for answering open-ended questions and summarization.
- Usage: Provides accurate responses and summaries in the chatbot and forums.

Intel® distilgpt2

- Purpose: Optimized GPT-2 for faster, efficient text generation.
- Usage: Improves chatbot speed and ensures real-time AQI updates.

Intel's sklearnex (Intel Extension for Scikit-learn) is a library that optimizes and accelerates Scikit-learn workflows by utilizing Intel's hardware (e.g., CPUs with Intel oneAPI libraries). It provides faster execution for machine learning algorithms and pipelines using Scikit-learn, especially for larger datasets.

patch_sklearn():

- *Purpose*: This function patches Scikit-learn with optimized versions of its algorithms, using Intel's DAAL (Data Analytics Acceleration Library).
- *Usage*: After calling patch_sklearn(), Scikit-learn uses optimized algorithms for training and inference, often speeding up processes without code modification. Once patched, the rest of the code remains identical to standard Scikit-learn usage.

sklearnex.linear_model:

- *Purpose*: This module contains optimized linear models from Scikit-learn, such as Linear Regression, Ridge, Lasso, and more.
- *Performance*: These models are optimized for performance on Intel hardware, providing faster training times compared to the default Scikit-learn models, especially on large datasets.

Benefits:

- *No code changes*: Simply import and patch.
- *Improved performance*: Especially useful on Intel processors with large datasets.

BuzzOnEarth India Hackathon

Prepared by Abhijeet Rajhans and Shreyasi Ray
of Manipal University Jaipur

THANK YOU