WATER QUALITY PREDICTION

By: AKSHAR NATH GORAIN

Registration ID:
STU6829b3f24d2691747563506

AICTE Internship ID:
INTERNSHIP_17464164681834e0e35d8

Learning Objectives

- To develop an AI-powered system for predicting water pollutants using machine learning.
- To utilize historical station-based monitoring data for environmental analysis and forecasting.
- To implement a multi-output machine learning model for pollutant prediction.
- To deploy the prediction model as a user-friendly web application using Gradio.
- Understand how to use Python libraries like Pandas, NumPy, Scikit-learn for analysis and model building.
- Understand how to use Matplotlib, Seaborn for visualization.
- Understand how to use GitHub for project version control and uploading.
- Understand the web deployment process via Gradio in Google form Colab.

Source : www.freepik.com/

Tools and Technology used

Tools used :

Coding Language : Python

Environment used : Collab

- Libraries used :

Numpy :For handling arrays & mathematical operations.

Pandas :For data analysis, cleaning and filtering.

Matplotlib :For visualisation.

Seaborn : For making attractive graphs.

Sklearn : Used in ML(Regression , classification).

- Models used :

Linear regression : Used in SML for predicting continuous numeric values.

Random forest regressor :SML algorithm used for regression task, good for overfitting problem.

- Evolution metrices :

RMSE(Root Mean Squared Error), MAE(Mean Absolute Error), R2 Score.

Methodology

- Data Collection: Used PB_All_2000_2021.csv dataset, covering the time period 2000–2021.
- Data Cleaning: Cleaned null values and filtered key stations from the dataset.
- EDA & Visualization: Conducted exploratory data analysis and visualized pollutant trends.
- Model Building: Implemented a MultiOutputRegressor with RandomForestRegressor.
    - Input Features: Year, Station ID
    - Output Pollutants: O_2, NO_3, NO_2, SO_4, PO_4, Cl^-
- Model Persistence: Saved the trained model as pollution_model.pkl and column features as model_columns.pkl using Joblib.
- Web Deployment: Converted the application to Gradio for web deployment in Google Colab.

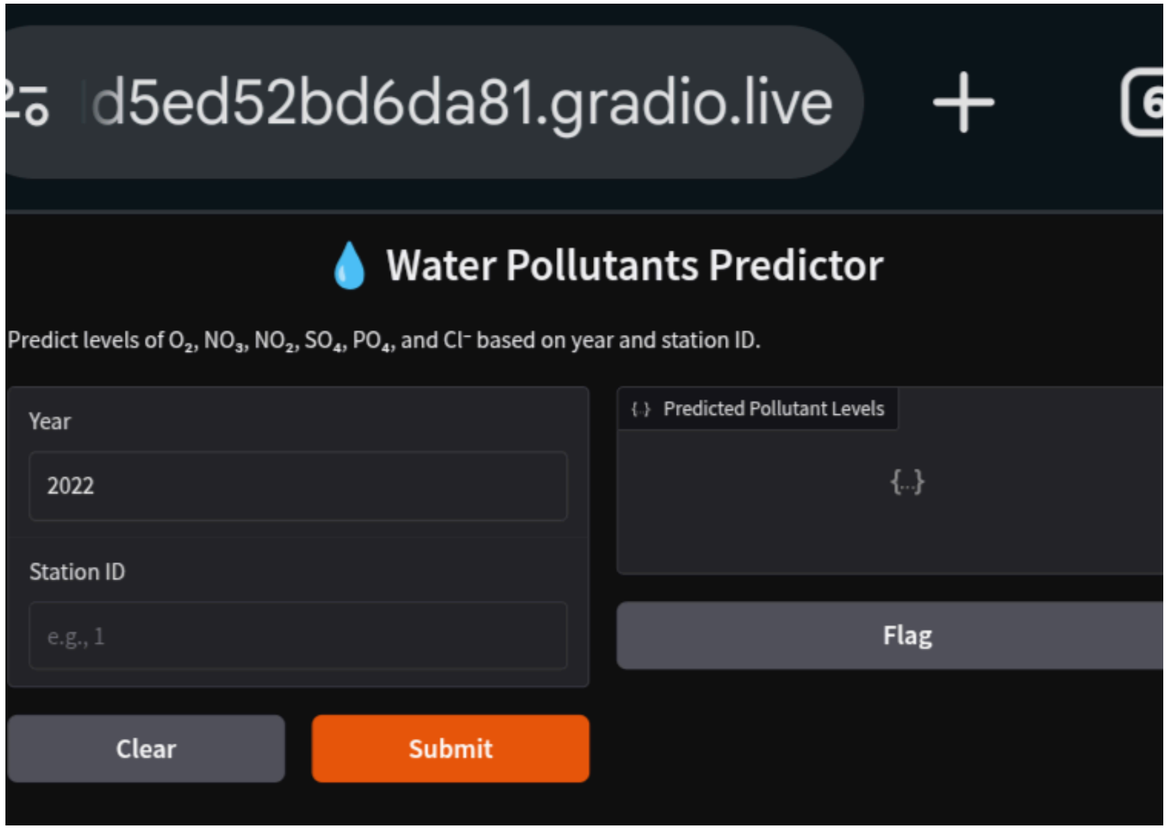
Problem Statement:

- Lack of a proactive system to forecast water pollutant levels, hindering effective environmental management.
- Difficulty in rapidly assessing potential water quality issues based on historical trends and station-specific data.
- Need for an accessible and user-friendly tool that can provide quick insights into future water quality to aid decision-makers.

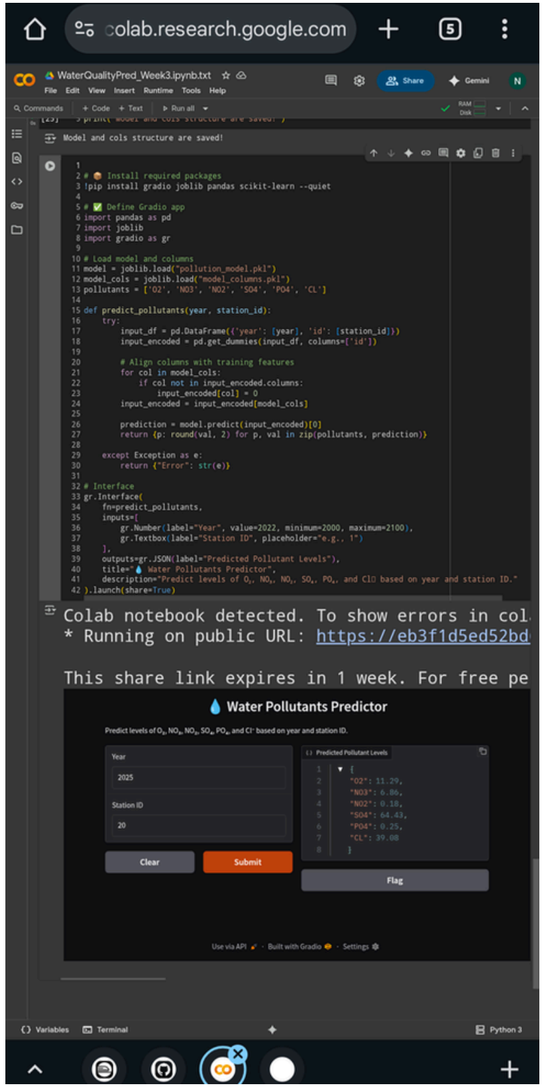
Solution:

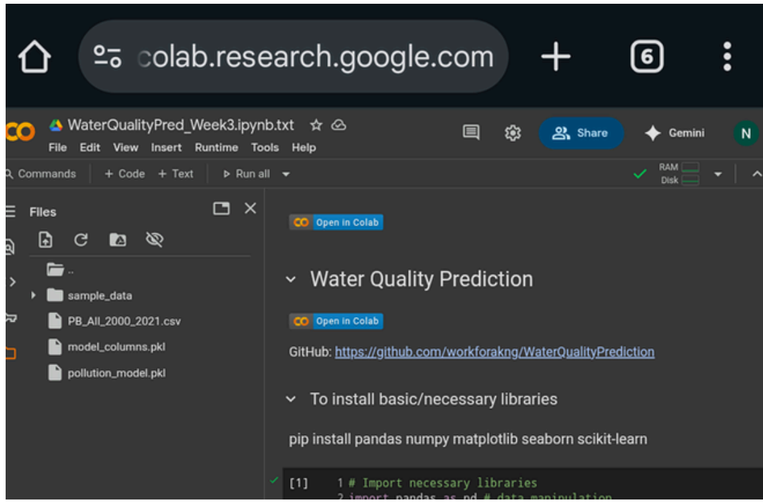
- Developed an AI-powered machine learning model capable of predicting six key water pollutants (O_2, NO_3, NO_2, SO_4, PO_4, Cl^-).
- The model leverages historical year and station_id data for predictions.
- Created an interactive web application using Gradio, allowing users to input a year and station ID to get immediate pollutant level predictions.
- Provided a deployment solution via Google Colab, making the prediction tool easily accessible for environmental analysis and forecasting.

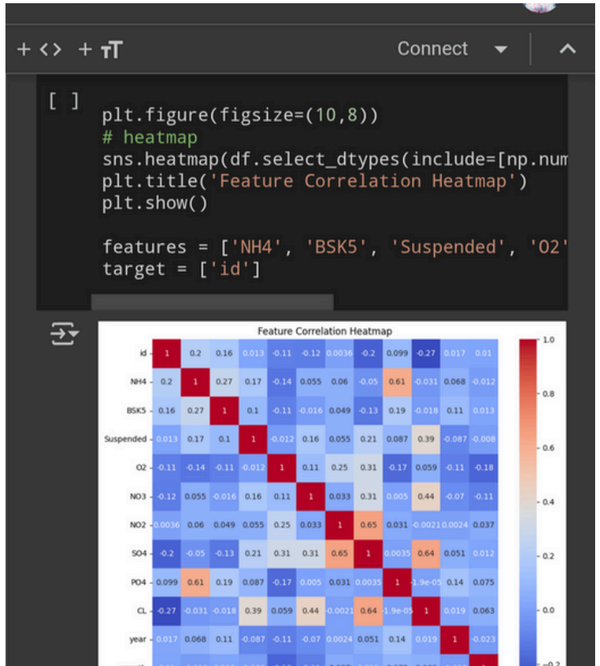
Screenshot of Output:



Gradio Interface







Conclusion:

- Successfully developed and deployed an AI-powered water quality prediction model using historical monitoring data.
- The Multi-output Random Forest Regressor effectively predicts six major water pollutants (O_2, NO_3, NO_2, SO_4, PO_4, Cl^-).
- The Gradio web application provides an accessible and intuitive tool for environmental analysis and forecasting.
- This project demonstrates the significant potential of machine learning to support proactive environmental management and decision-making for a healthier future.

Click to open GitHub Link : https://github.com/workforakng/WaterQualityPrediction.git

GITHUB:

GitHub Repository: https://github.com/workforakng/WaterQualityPrediction.git

- Find the full project code, dataset, and trained model files on GitHub.

Click to open Repository

Run the App in Google Colab:

- Click below to launch the prediction app:

- 👉 Run on Colab

Run Locally using Gradio:

- pip install -r requirements.txt

- python app.py