Problem : Air Quality Level Prediction (ML)

Cities in Pakistan struggle with hazardous air pollution (like high PM2.5), which affects health and daily life. Predicting future air quality helps authorities issue warnings early.

Problem Statement

Build a machine learning model to **predict air quality levels (e.g., AQI category)** using historical data from Pakistani cities. The goal is to forecast future pollution levels so authorities can act in time.

Objective

- Predict **daily AQI category** (Good, Moderate, Unhealthy, etc.)
- Forecast a few days ahead (e.g., the next 3 days)
- Handle time-series data and real environmental features

Dataset Options (Pakistan)

1. **Pakistan Air Quality & Weather Data (hourly, 2021–2024)**

— Air quality and weather data from major cities like Islamabad and more. (Kaggle)

2. **Islamabad Air Quality Data (EPA source)**

— Detailed pollution data for Islamabad. (Kaggle)

3. **Air Quality Index — Pakistan (various cities, daily)**

— Daily AQI values for cities like Lahore, Karachi, etc. (Kaggle)

4. **Air Quality Monitoring Dataset (Pakistan 2018)**

— Smaller dataset useful for initial training or comparison. (Kaggle)

You can **combine these or enrich with weather data** too.

Tasks

1. **Clean and preprocess time-series data** (handle missing values).
2. **Feature engineering** (lag features, rolling averages, weather features).
3. Train models like **Random Forest, XGBoost, LSTM, or Temporal CNN**.
4. Predict **daily AQI levels** or pollutant concentration.

Evaluation

- **Accuracy / F1-Score** for predicted AQI category
- **RMSE / MAE** if predicting numeric AQI or pollutant levels
- **Prediction lead time** (how far ahead you can forecast accurately)

Final Submission (Required)

1. Code & Model

- Training notebook or script (`.ipynb` or `.py`)
- Saved trained model (`.pkl` or `.joblib`)
- Preprocessing pipeline (if separate)

2. Prediction Output

- CSV file with:
  - City
  - Date
  - Predicted AQI category (or AQI value)
- Forecast for **next N days**

3. Streamlit App

- Simple Streamlit app (`app.py`) that:
  - Allows city selection
  - Shows AQI forecast for next days
  - Displays alerts for **Unhealthy air**
- App must load the **trained model**

4. Short Report (1–2 pages or README)

- Dataset used
- Features created
- Model chosen and why
- Evaluation results
- Limitations

Bonus

- **Visual forecast dashboard**
- **City-wise risk ranking**
- **Explainable ML insights**

Problem: Student Dropout Early Warning System (ML)

A university is losing students every semester.

They want an **early warning system** so advisors can help students **before** they drop out.

Your Task

You are hired as a **data scientist**.

Build a machine learning system that:

1. **Predicts which students are at risk of dropping out**
2. **Flags high-risk students early in the semester**

What You Must Do

- Use academic activity and basic student data
- Train a model to predict **Dropout / Continue**
- Focus on **early detection**, not end-semester data

Dataset

https://www.kaggle.com/datasets/aljarah/xAPI-Edu-Data

Expected Output

- A **risk score** for each student
- A **list of high-risk students** for advisors

Success Criteria

- Catch as many dropout students as possible
- Keep false alarms reasonable
- Model should be easy to explain to non-technical staff

Real-World Constraints

- Data is noisy and incomplete
- Advisors need results **early**
- Decisions must be **fair and transparent**

Final Deliverables

- Trained ML model
- Prediction results
- Short explanation: *why students are at risk*

Final Submission (Required)

1) Code + Model

- Training notebook/script (`.ipynb` or `.py`)
- Saved trained model (`model.pkl` / `model.joblib`)
- Saved preprocessing pipeline (recommended as one pipeline file)

2) Predictions File

Submit a CSV like:

- `student_id` (or row index)
- `risk_score`
- `risk_label` (Low/Medium/High)
- `predicted_dropout` (0/1)

3) Streamlit App

Submit `app.py` that:

- Uploads a CSV student file
- Shows a **Top High-Risk Students** table (top 20)
- Shows a selected student's **risk score + risk label**
- Shows **top reasons** (simple feature importance is fine)

4) Short Report (README)

Include:

- How you cleaned data
- Features used
- Model choice + metrics
- How you set the risk thresholds
- Key reasons behind dropout predictions

Bonus (Optional)

- Student risk levels (Low / Medium / High)
- Simple action suggestions for advisors