**🌍 Project “Dhulo Watch”: Air Quality & Pollution Risk Prediction (Nepal)**

Monitoring Nepal’s air, one breath at a time.

**1. Problem Definition**

* **Goal:** Predict AQI (Air Quality Index) values using weather & environmental features.
* **Bonus:** Cluster regions/days into “Low Risk / Medium Risk / High Risk” pollution categories.
* **Why it matters:** Kathmandu often ranks among the most polluted cities globally; your model could show patterns and policy implications.

**2. Data Collection**

You can combine sources to make a rich dataset:

* **Primary Data:**
  + IQAir – Nepal Air Quality (can scrape daily AQI by city)
  + OpenAQ API (free AQI & PM2.5 data for Kathmandu, Bhaktapur, Lalitpur)
* **Secondary Data (features):**
  + Weather: OpenWeatherMap API (temp, humidity, wind speed, rainfall)
  + Traffic (proxy): Kathmandu traffic police reports, or Google traffic APIs (optional).
  + Geographic: District altitude/valley information (can add as categorical).

**3. Data Preprocessing**

* Merge AQI + weather by **timestamp** and **location**.
* Handle missing values (interpolation).
* Feature engineering ideas:
  + **Lag features:** yesterday’s AQI → today’s AQI.
  + **Season indicator:** festival months (Tihar/Chhath), winter vs. monsoon.
  + **Wind chill / humidity index** as derived features.

**4. Exploratory Data Analysis (EDA)**

* Plot **time-series AQI** (daily/monthly trend).
* Compare AQI across cities (Kathmandu vs. Pokhara vs. Terai).
* Correlation heatmap between AQI and weather factors.
* Identify “peak pollution months” (usually Dec–Feb).

**5. Model Building**

You can approach in **two tracks**:

**A. Supervised Learning (Prediction)**

* **Target:** AQI (continuous variable → regression).
* **Models to try:**
  + Linear Regression (baseline)
  + Random Forest Regressor
  + Gradient Boosting (XGBoost/LightGBM)
* **Evaluation:** RMSE, MAE, R².

**B. Unsupervised Learning (Clustering Risk Zones)**

* **Target:** None (unsupervised).
* Use features: [AQI, PM2.5, PM10, Temp, Humidity, Wind].
* Apply **KMeans / DBSCAN / Hierarchical Clustering**.
* Label clusters as **Low/Medium/High Pollution Risk**.
* Visualize clusters on Nepal’s district map.

**6. Model Deployment / Presentation**

* Create a **dashboard in Streamlit** where:
  + User selects location & weather conditions → model predicts AQI.
  + Clusters are visualized on a simple map (folium or geopandas).
* Or write a clean **Jupyter Notebook report** with plots + model comparisons.

**7. Extensions (to stand out)**

* Forecast AQI **7 days ahead** using regression with time-series features.
* Compare **Kathmandu vs. Delhi AQI** to show regional trends.
* Add **health advisory labels**:
  + AQI < 50 → Good
  + 51–100 → Moderate
  + 101–200 → Unhealthy
  + 200 → Hazardous

✅ **Skills you’ll showcase:**

* Regression + Clustering
* Data integration (multiple APIs)
* EDA + storytelling with visualization
* Applied ML to a real socio-environmental problem in Nepal

air-quality-nepal/

├─ data/

│ ├─ raw/ # raw API downloads / CSVs

│ └─ processed/ # cleaned / merged datasets

├─ notebooks/ # EDA and experiments

├─ src/

│ ├─ data\_fetch.py # functions to pull APIs

│ ├─ preprocess.py # cleaning & feature engineering

│ ├─ models.py # train/eval models

│ ├─ clustering.py # clustering pipeline

│ └─ utils.py

├─ app/ # streamlit or flask app

├─ models/ # saved model files (.joblib)

├─ reports/ # final report / slides

├─ requirements.txt

└─ README.md

**Plan:**

**Phase 1: Data Preparation**

✅ **Goal:** Load, clean, and prepare your air quality dataset.  
**Tasks:**

1. Import dataset (CSV or collected data) into pandas.
2. Parse dates correctly (date column → datetime).
3. Check for missing or invalid values (pm2\_5, pm10, etc.).
4. Handle missing data (e.g., interpolate, forward-fill).
5. Ensure numeric types are correct (float, not string).
6. Optional: convert station to categorical type.
7. Save cleaned dataset (data/cleaned\_aqi.csv).

**Phase 2: Exploratory Data Analysis (EDA)**

✅ **Goal:** Understand patterns, correlations, and trends.  
**Tasks:**

1. Summary statistics for each pollutant.
2. Correlation matrix (heatmap) between pollutants and weather variables.
3. Temporal analysis:
   * Monthly and seasonal trends.
   * Daily/hourly variation (if time resolution allows).
4. Station-wise analysis:
   * Compare pollution levels across stations.
   * Top 3 most/least polluted stations.
5. Trend plots:
   * Line plots of pm2\_5, pm10, AQI over time.
   * Rolling mean (7-day, 30-day) for smooth trends.
6. Geographic visualization:
   * **Folium map**: show station locations colored by AQI.
   * Use **CircleMarker** or **MarkerCluster**.

**Phase 3: Feature Engineering**

✅ **Goal:** Prepare features for forecasting.  
**Tasks:**

1. Create year, month, day, and dayofweek features.
2. Add lag features (e.g., yesterday’s PM2.5 → predict today).
3. Compute moving averages (e.g., 3-day or 7-day rolling mean).
4. Encode station as dummy variable (pd.get\_dummies()).
5. Define target variable (e.g., pm2\_5).

**Phase 4: Modeling (Prediction Prototype)**

✅ **Goal:** Predict future AQI or PM2.5 for each station.  
**Tasks:**

1. Train/test split (e.g., last 20% of data for testing).
2. Try baseline models:
   * Linear Regression
   * Random Forest Regressor
   * XGBoost (optional, for comparison)
3. Evaluate performance (RMSE, MAE, R²).
4. Plot predicted vs actual for test data.

**Phase 5: Visualization Dashboard / Presentation**

✅ **Goal:** Impress recruiters with storytelling & clarity.  
**Tasks:**

1. Interactive map of stations with average AQI.
2. Plot trends (using Matplotlib/Seaborn/Plotly).
3. Highlight insights like:
   * “Winter months show highest PM2.5.”
   * “Kathmandu consistently exceeds safe limits.”
4. Optional mini dashboard (Streamlit or Jupyter Notebook).

**Phase 6: Documentation & Presentation**

✅ **Goal:** Make it portfolio-ready.  
**Tasks:**

1. Write a clean **README.md** with:
   * Problem statement
   * Dataset summary
   * Analysis highlights
   * Model performance
   * Tools used
   * Future improvements (mention retraining pipeline plan)
2. Add screenshots of maps/graphs.
3. Export notebook as **HTML** or **PDF** for recruiters.

**🔧 Tech Stack (Prototype)**

* **Python Libraries:** pandas, numpy, matplotlib, seaborn, scikit-learn, folium
* **Optional:** plotly, streamlit
* **File Structure:**
* dhulowatch/
* ├─ data/
* │ ├─ raw/
* │ └─ cleaned\_aqi.csv
* ├─ notebooks/
* │ └─ 01\_EDA\_and\_Modeling.ipynb
* ├─ visuals/
* │ ├─ plots/
* │ └─ maps/
* ├─ README.md

└─ requirements.txt