Water Quality Classification Project

1-Week Intensive Data Science Challenge

Project Overview

This 1-week intensive project challenges you to build an end-to-end machine learning solution for water quality classification using Python. You will perform exploratory data analysis, develop predictive models, and deploy an interactive Streamlit application.

Problem Statement

How do physicochemical and biological parameters influence water quality classification and public health safety across monitoring stations in Maharashtra? Your task is to analyze water quality data, build predictive models, and create a deployable application for stakeholders.

Dataset

File: NWMP_August2025_MPCB_0.csv

The dataset contains water quality measurements from monitoring stations across Maharashtra, India, collected by the Maharashtra Pollution Control Board (MPCB). It includes 54 columns covering chemical, biological, and physical parameters.

Download Dataset: Google Drive Link

Water Quality Classes:

Class	Description
A	Drinking water source (post-disinfection)
B	Outdoor bathing
C	Drinking water (with treatment)
E	Irrigation, industrial cooling

Key Parameters:

• **Chemical:** pH, BOD, Dissolved Oxygen, Conductivity, TDS, COD
• **Biological:** Fecal Coliform, Total Coliform
• **Physical:** Temperature, Turbidity, Color
• **Geographic:** Station, District, River Basin

1-Week Project Timeline

Day	Phase	Deliverables
1-2	EDA & Data Understanding	• Data quality report • Exploratory visualizations • Statistical analysis • Missing value analysis
3-4	Preprocessing & Modeling	• Cleaned dataset • Feature engineering • Baseline model • Tuned models (2+) • Model comparison
5-6	Streamlit Deployment	• Streamlit application • Prediction interface • Interactive dashboards • Model documentation
7	Presentation & Submission	• Final presentation (10 mins) • Code repository • Demo video • Technical report

Technical Requirements

Category	Requirements
Python Libraries	pandas, numpy, matplotlib, seaborn, scikit-learn, streamlit
EDA	Distribution analysis, correlation heatmaps, missing value analysis, outlier detection, geogr
Preprocessing	Handle missing values (BDL/ND/NA), encode categorical features, scale numeric features,
Modeling	Train 3+ classifiers (Logistic, Random Forest, XGBoost), handle class imbalance, hyperpar
Evaluation	Accuracy, F1-score, confusion matrix, classification report, cross-validation
Streamlit App	Prediction page, EDA dashboard, model comparison, user-friendly interface

Detailed Phase Breakdown

Phase 1: EDA & Data Understanding (Days 1-2)

Objectives:

• Load and inspect the dataset (shape, columns, data types)
• Analyze missing values and data quality issues
• Visualize distributions of key parameters
• Explore relationships between features and target classes
• Identify geographic patterns across districts
• Generate statistical summaries and correlation analysis

Deliverables:

• Jupyter notebook with comprehensive EDA
• 10+ visualizations (distributions, heatmaps, geographic plots)
• Written insights document (key findings, data quality issues)

Phase 2: Preprocessing & Modeling (Days 3-4)

Objectives:

• Clean data (handle missing values, outliers, invalid entries)
• Engineer features (ratios, thresholds, pollution indicators)
• Encode target variable and categorical features
• Split data (train/validation/test)
• Train baseline model (Logistic Regression)
• Train advanced models (Random Forest, XGBoost, SVM)
• Perform hyperparameter tuning
• Handle class imbalance (SMOTE, class weights)

Deliverables:

• Preprocessing pipeline (Python scripts)
• Trained models (saved as .pkl files)
• Model evaluation report (metrics, confusion matrices)
• Feature importance analysis

Phase 3: Streamlit Deployment (Days 5-6)

Objectives:

• Design Streamlit app structure (multi-page layout)
• Create prediction page (input form → model inference)
• Build EDA dashboard (interactive visualizations)

• Add model comparison page
• Implement user input validation
• Add documentation and instructions

Required Pages:

• **Home:** Project overview and instructions
• **Predict:** Input form for water quality prediction
• **EDA:** Interactive data visualizations
• **Models:** Model comparison and metrics

Deliverables:

• Fully functional Streamlit application
• Deployed link (Streamlit Cloud / local instructions)
• User guide / README

Phase 4: Presentation & Submission (Day 7)

Objectives:

• Prepare 10-minute presentation
• Create demo video of Streamlit app
• Write technical report
• Organize code repository
• Submit all deliverables

Deliverables:

• Presentation slides (10-12 slides)
• Demo video (3-5 minutes)
• GitHub repository (organized, documented)
• Technical report (PDF, 3-5 pages)

Evaluation Criteria

Criteria	Weight	What We Look For
EDA Quality	20%	Depth of analysis, quality of visualizations, insights
Preprocessing	15%	Data cleaning thoroughness, feature engineering creativity
Model Performance	25%	Accuracy, F1-score, model comparison, tuning
Streamlit App	25%	Functionality, UI/UX, interactivity, deployment
Presentation	15%	Clarity, storytelling, technical depth, time management

Submission Guidelines

Required Files:

1. **GitHub Repository:** All code, notebooks, and documentation
2. **Streamlit App:** Deployed link or local run instructions
3. **Presentation:** PDF slides
4. **Demo Video:** MP4 file (uploaded to Drive/YouTube)
5. **Technical Report:** PDF document

Repository Structure:

```
water-quality-project/
■■■ data/ (dataset files)
■■■ notebooks/ (EDA and modeling notebooks)
■■■ src/ (preprocessing, training scripts)
■■■ models/ (saved model files)
■■■ app/ (Streamlit application)
■■■ README.md
■■■ requirements.txt
```

Tips for Success

• **Start Early:** Don't wait until the last day
• **Version Control:** Commit code frequently to GitHub
• **Documentation:** Write clear comments and README
• **Time Management:** Stick to the daily schedule
• **Ask Questions:** Reach out if you're stuck
• **Test Thoroughly:** Make sure your app works before submission
• **Focus on Quality:** Better to have fewer features done well

Good luck with your project! Remember, this is a learning opportunity to demonstrate your end-to-end data science skills.