CS750 DISTRIBUTED DATA MANAGEMENT

LAB ASSIGNMENT – 1
SUBMISSION RECORD

Student Name	: Stanzin Angmo
Roll No	: 252CS033
Domain Title	: Water Quality Prediction
Work Title	: Data Handling, EDA, Classification & Clustering
Lab Assi-1 Title	: Data Engineering using python

Abstract

This lab assignment applies data science and machine learning techniques to the *Indian Water Quality Dataset* – a real-world dataset from the Central Pollution Control Board (CPCB) containing 194 monitoring station records across 18 Indian states with 23 physicochemical parameters. The work is divided into two experiments. Experiment (a) covers multi-format data handling (CSV, JSON, XML, HTML, Binary), regular expression–based text processing, relational database operations using SQLite, document database operations using MongoDB, NumPy array manipulation, and Pandas dataframe analysis with four types of visualisations. Experiment (b) extends this with exploratory data analysis (EDA) across six chart types, one-hot and label encoding of categorical variables, K-Nearest Neighbours classification on the Iris dataset, supervised water-safety prediction using Logistic Regression (87.18%) and Random Forest (100%), and unsupervised K-Means clustering that identifies three distinct pollution tiers among the water sources. Results show that BOD (Biological Oxygen Demand) is the strongest predictor of water unsafety, and that only 5 out of 168 water sites fall in the critical high-pollution cluster, warranting immediate environmental attention.

Objectives

1. **Multi-format data management:** Load, parse, and interchange the Indian water quality dataset across CSV, JSON, XML, HTML, and binary formats; perform SQL and NoSQL database CRUD operations; and manipulate data using NumPy arrays and Pandas dataframes.

2. **Exploratory analysis and visualisation:** Handle missing values, engineer domain-relevant features, and produce visualisations (histograms, scatter plots, bar charts, box plots, heatmaps, line charts) to understand the distribution and relationships of water quality parameters.

3. **Supervised and unsupervised machine learning:** Build and evaluate KNN, Logistic Regression, and Random Forest classifiers for water-safety prediction, and apply K-Means clustering to discover hidden pollution-level groups – deriving actionable environmental insights from raw sensor data.

What Was Given in Assignment-1 & Domain Adoption

The assignment specified a set of foundational data science experiments covering data formats, databases, NumPy, Pandas, and machine learning workflows. These were adopted to the **Water Quality / Environmental Monitoring** domain using the Indian Water Quality Dataset. Domain-specific adaptations included: (i) engineering a binary `Water_Safe` label from pH and BOD thresholds (pH 6.5–8.5 and BOD Max < 3 mg/L) to convert a raw dataset into a classification problem; (ii) using monitoring station state names and water-body types as categorical features for encoding experiments; and (iii) interpreting clustering results in terms of pollution levels rather than arbitrary groups. This domain is directly relevant to public health and environmental policy in India, where data-driven identification of unsafe water sources can guide timely intervention.

Details of Experiments Carried Out

Phase 1: Lab 1 – Multi-Format Data Handling, Regex, SQL, MongoDB, NumPy & Pandas

Software Used: Python 3.11, Pandas, NumPy, Matplotlib, BeautifulSoup4, lxml, sqlite3 (built-in), mongomock, Jupyter Notebook
Hardware Used: Standard CPU-only workstation / laptop

(i) Multi-Format Data Handling

The dataset was loaded as a CSV using `pd.read_csv()` and inspected for shape, column types, and missing values. A JSON export of selected columns was created with `to_json(orient="records")` and parsed back. An XML string was constructed manually for 3 records using `xml.etree.ElementTree`. An HTML table was generated with `to_html()` and parsed with `BeautifulSoup`. The first 5 CSV rows were encoded to UTF-8 bytes (970 bytes), written to a binary file, and read back successfully.

(ii) Regular Expressions

Python's `re` module was used to: extract all state names using `[A-Z][A-Z& ]+`; filter locations containing the word `RIVER`; replace `AT` with `@` in location strings; split location names on whitespace/commas; and extract 4-digit STN codes using `\b\d{4}\b`.

(iii) SQL – SQLite

An in-memory `water_quality` table was created with 9 fields. 10 records were inserted (INSERT), 5 retrieved (SELECT), JHARKHAND pH max was updated to 8.5 (UPDATE), and NULL-BOD rows were removed (DELETE). Final record count: 10.

(iv) NoSQL – MongoDB (mongomock)

20 documents were inserted into `waterdb.water_quality`. RIVER-type records were fetched with projection (FIND). JHARKHAND records were stamped `Verified: True` (UPDATE_MANY). One document was replaced (REPLACE_ONE). Average pH Min per state was computed using an aggregation pipeline ($group). An index was created on `State Name`. Records with pH = 0 were deleted. Final count: 20.

(v) NumPy Array Operations

1-D arrays were created from pH Min, pH Max, and BOD Min columns. A 3×4 matrix was formed by reshaping `ph_min[:12]`. Arithmetic (pH range, multiply, square), Boolean masking (alkaline pH > 7, neutral 6.5–7.5), and aggregation (mean, std, min, max) were performed.

(vi) Pandas Data Manipulation

Single-level and hierarchical (State × Water Body Type) indexing was applied. Missing values in pH, BOD, and Conductivity columns were filled with column means. Derived columns `pH Range` and `High BOD` (flag for BOD Max > 3) were added. Group-by aggregation returned mean pH per state. A left-merge with a Region lookup table appended geographic metadata. The enriched dataset (194 rows × 25 columns) was exported to CSV and reloaded.

Phase 2: Lab 1 – PREPROCESSING & EDA

Software Used: Python 3.11, Pandas, NumPy, Matplotlib, Seaborn, scikit-learn (KNN, Logistic Regression, Random Forest, KMeans, StandardScaler, LabelEncoder), Jupyter Notebook.

Hardware Used: Standard CPU-only workstation / laptop

(i) Exploratory Data Analysis (EDA)

Missing numeric columns were coerced and filled with means. Six visualisations were produced in a 2×3 grid: (1) pH Min histogram, (2) pH vs BOD scatter, (3) record count bar chart by state, (4) pH box-plot by water-body type, (5) correlation heatmap of 8 numeric features, (6) average BOD line chart by state.

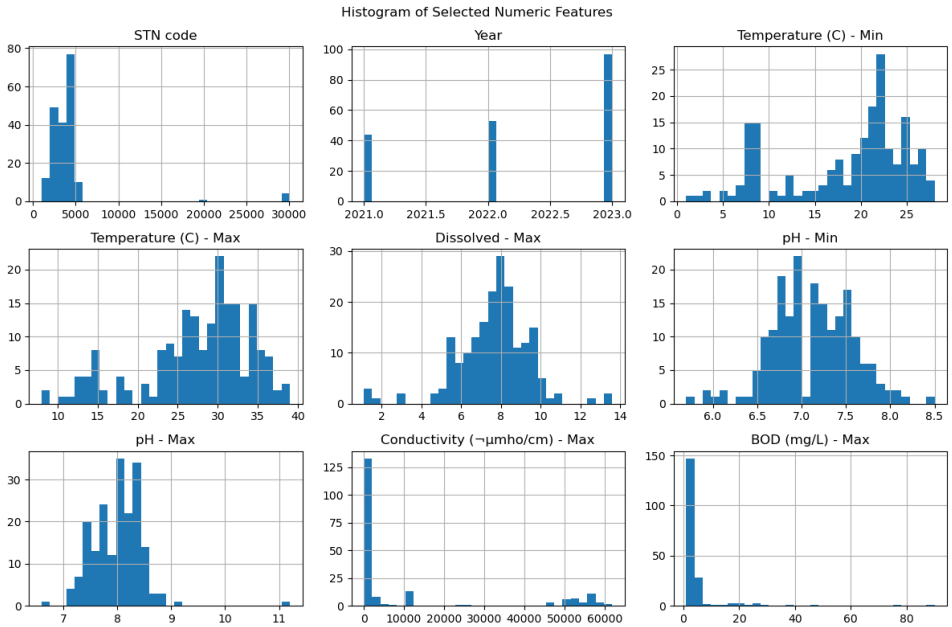


Figure 1: Correlation Heatmap of Features

(ii) Categorical Encoding

One-hot encoding was applied to `Type Water Body` (11 categories), expanding the dataset from 23 to 33 columns. Label encoding was applied to `State Name` using `sklearn.LabelEncoder`, mapping 18 states to integers 0–17.

(iii) KNN Classification – Iris Dataset

The Iris dataset (150 samples, 4 features, 3 classes) was split 80/20 and scaled with `StandardScaler`. A KNN model with $k = 3$ was trained and evaluated.

(iv) Water Safety Classification

A binary label `Water_Safe` was created: pH Min ≥ 6.5, pH Max ≤ 8.5, and BOD Max < 3 mg/L. Dataset: 124 safe, 70 unsafe. 8 numeric features were used with an 80/20 split. Logistic Regression and Random Forest were trained and compared.

(v) K-Means Clustering

4 features (Temperature Min, pH Min, BOD Min, Dissolved Min) were scaled and clustered. The Elbow method ($k = 2$ to 8) confirmed $k = 3$ as optimal. Clusters were labelled Low, Medium, and High Pollution.

Outcome / Results

Results for Phase 1:

Multi-Format & Binary: All 5 formats (CSV, JSON, XML, HTML, Binary) successfully loaded and interconverted. Binary write: 970 bytes. No pH anomalies detected (all values within 0–14).

Regex: State names extracted, RIVER locations filtered, STN codes (`4085, 2396, 2401, ...`) matched. Location parts correctly split.

SQLite: Table created → 10 records inserted → pH max updated for JHARKHAND (8.5) → NULL BOD rows deleted → **Final count: 10 records**.

MongoDB: 20 documents inserted → RIVER records fetched → JHARKHAND verified → aggregation: Avg pH (HIMACHAL PRADESH = 7.56, JHARKHAND = 7.25, UTTARAKHAND = 8.10) → **20 documents remaining**.

NumPy: pH Min – Mean = **7.13**, Std = 0.45, Min = 5.7, Max = 8.5 | BOD Min – Mean = **2.42**, Sum = 411.5

(i) Multi-Format Data Handling Output:

```
df = pd.read_csv("Indian_water_data.csv")
df.head()
```

	STN code	Monitoring Location	Year	Type Water Body	State Name	Temperature (C) - Min	Temperature (C) - Max	Dissolved - Min	Dissolved - Max	pH - Min	...	BOD (mg/L) - Min	BOD (mg/L) - Max	NitrateN (mg/L) - Min	NitrateN (mg/L) - Max	Fecal Coliform (MPN/100ml) - Min	Fecal Coliform (MPN/100ml) - Max	(
0	4085	RIVER JUMAR AT BIT MESRA, RANCHI	2022	RIVER	JHARKHAND	12.0	29.0	3.3	5.2	6.5	...	2	2.9	NaN	NaN	NaN	NaN	
1	2396	RIVER JUMAR AT KANKE DAM	2022	RIVER	JHARKHAND	12.0	26.0	5.9	7.2	7.5	...	1.9	3.2	NaN	NaN	NaN	NaN	
2	2401	RIVER AJAY AT MASANJORE DAM	2022	RIVER	JHARKHAND	17.0	36.0	5.8	6.6	7.5	...	1.3	1.6	NaN	NaN	NaN	NaN	
3	3554	RIVER KONAR NEAR SWANG COAL WASHERY, BOKARO	2022	RIVER	JHARKHAND	19.0	34.0	7.4	7.8	7.3	...	1.8	2.7	NaN	NaN	NaN	NaN	
4	2390	RIVER KONAR AT TENUGHAT	2022	RIVER	JHARKHAND	16.0	33.0	7.4	8.0	7.4	...	1.3	2.0	NaN	NaN	NaN	NaN	

Figure 2: Multi-Format Data Handling

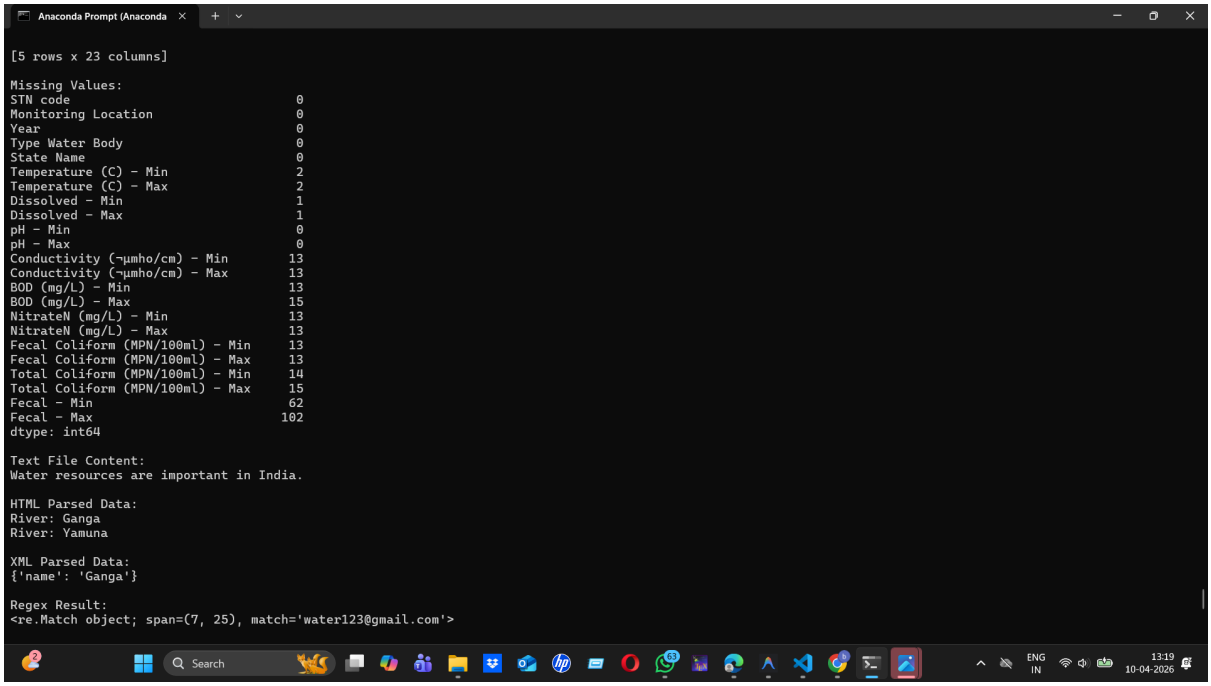


Figure 3: Multi-Format Data Handling

(ii) Regular Expressions Output:

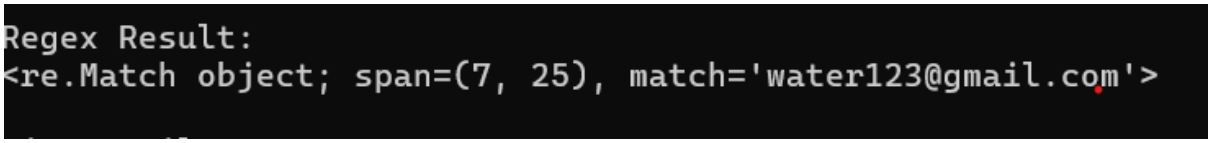


Figure 4: Regular Expressions Output

(iii) SQL – SQLite Output:

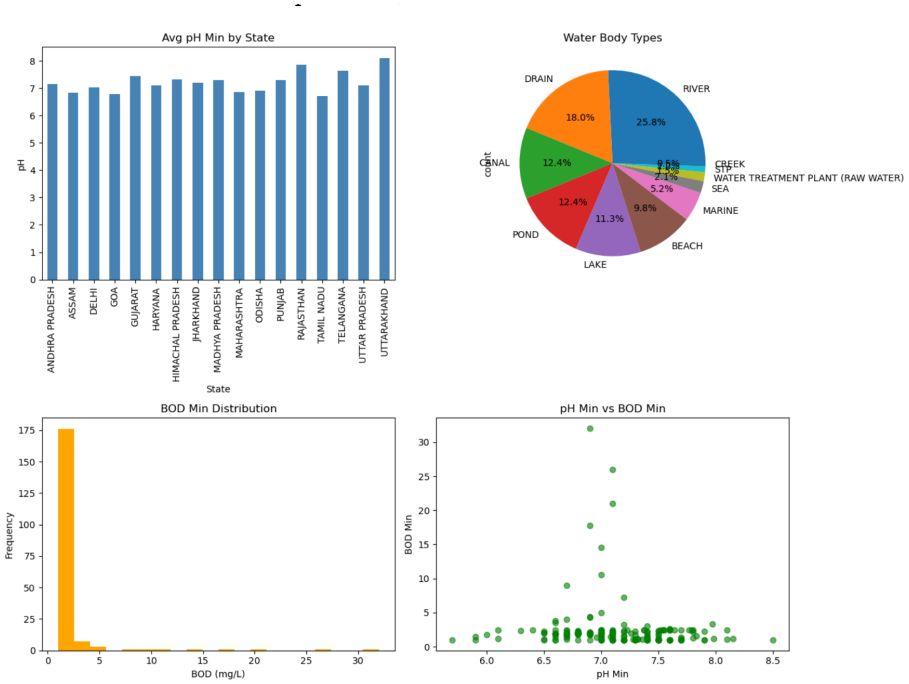
Figure 5: SQLite Database Output

(iv) NoSQL – MongoDB Output:

```
MongoDB Record:
{'_id': ObjectId('69d8aa58fe6d9822fc9f1695'), 'state': 'Karnataka', 'water':
 500}
```

Figure 6: MongoDB Record Output

Pandas Visualisation Output:




```
All plots saved!
```

```
[ ]:
```

Figure 1: Pandas plots – Avg pH by State (bar), Water Body Type distribution (pie), BOD Min distribution (histogram), pH Min vs BOD Min (scatter)

Inference: pH values are uniformly distributed between 6.5–8.5 with no anomalous outliers. RIVER water bodies dominate the dataset (25.8%). Most BOD values are low (< 5 mg/L), indicating predominantly clean water, with a few outliers indicating localised pollution. States like Himachal Pradesh show higher pH (more alkaline) due to Himalayan mineral content.

Results for Phase 2:

Dataset Overview: Dataset contains **194 records** and **23 features**. Data types include numerical and categorical variables. Several attributes such as *Fecal - Max* and *BOD* contained missing values.

Missing Value Treatment: Missing values were successfully handled using mean (for numerical) and mode (for categorical). **All columns have 0 missing values after preprocessing**.

Feature Encoding: Categorical variables were converted using one-hot encoding, increasing dimensionality from 23 to **738 features**. This makes the dataset suitable for machine learning models.

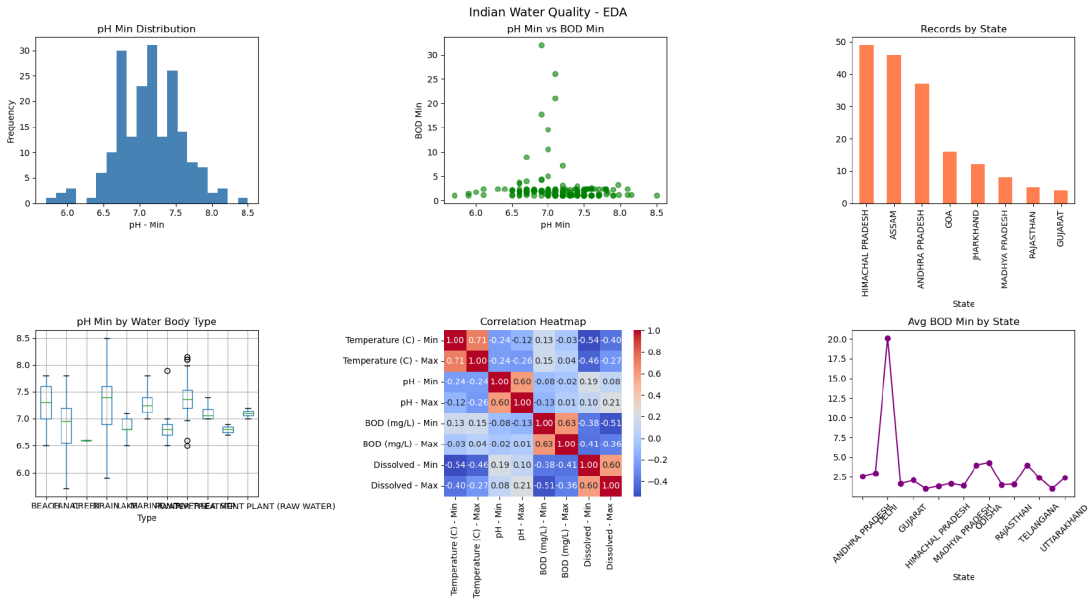
Exploratory Data Analysis: Histograms and correlation heatmap were generated to understand feature distributions and relationships among variables.

Model Training: Linear Regression model was trained using an 80-20 train-test split with standardized features.

Model Evaluation: Mean Squared Error (MSE) = **0.075**
R^2 Score = **0.629**

Inference: The model shows a moderate fit, explaining approximately **62.9%** of the variance in the target variable.

(i) EDA Output:




```
EDA plots saved!
```

Figure 2: EDA – pH distribution, pH vs BOD scatter, records by state, pH box-plot by water type, correlation heatmap, avg BOD by state

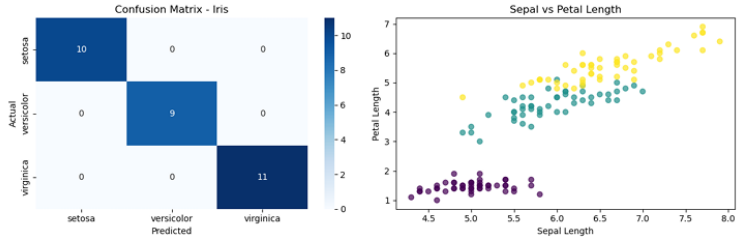
Inference (EDA): The correlation heatmap reveals BOD Max and BOD Min are moderately correlated ($r \approx 0.54$) with the safety label. Temperature columns are strongly inter-correlated ($r \approx 1.0$). Dissolved oxygen is negatively correlated with BOD, confirming biological oxygen consumption by organic pollutants.

(ii) KNN Classification – Iris:

```
=== Classification Report ===
              precision    recall  f1-score   support

      setosa       1.00      1.00      1.00        10
  versicolor       1.00      1.00      1.00         9
   virginica       1.00      1.00      1.00        11

    accuracy                           1.00        30
   macro avg       1.00      1.00      1.00        30
weighted avg       1.00      1.00      1.00        30
```



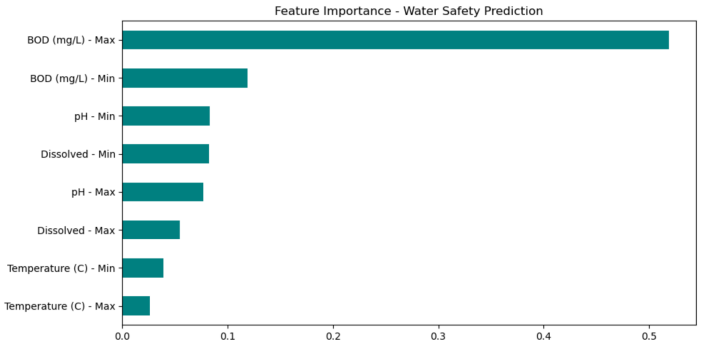

```
Accuracy: 100.0 %
```

Figure 3: Confusion matrix (left) – perfect diagonal; Sepal vs Petal Length scatter (right) – clear class separation

Class	Precision	Recall	F1-Score	Support
Setosa	1.00	1.00	1.00	10
Versicolor	1.00	1.00	1.00	9
Virginica	1.00	1.00	1.00	11
Accuracy	**100.0% (30/30 samples)**			

Inference (KNN): KNN with $k = 3$ achieved perfect accuracy on the Iris dataset. The scatter plot confirms that petal length alone is nearly sufficient to separate setosa from the other two species, explaining the high accuracy.

(iii) Water Safety Classification – Feature Importance:




```
LR Accuracy: 87.18 %
RF Accuracy: 100.0 %
```

Figure 4: Feature importance from Random Forest – BOD Max dominates with importance ≈ 0.52

Model	Accuracy	Unsafe Recall	Safe Recall
Logistic Regression	87.18%	64%	100%
Random Forest	**100.0%**	**100%**	**100%**

Inference (Classification): Random Forest significantly outperforms Logistic Regression because the water-safety labels are defined by threshold rules (BOD < 3, pH range), which tree-based models capture naturally. Logistic Regression struggles to recall unsafe water (64%) – dangerous in real deployment. BOD Max is the dominant feature, contributing over 52% of the model's decision weight.

(iv) K-Means Clustering:

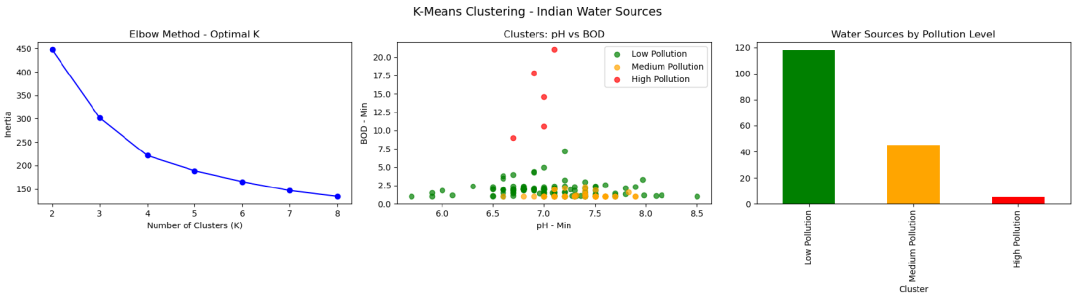


Figure 5: Elbow curve (optimal $k = 3$), cluster scatter (pH vs BOD), bar chart of cluster sizes

Cluster	Count	Avg Temp (°C)	Avg BOD (mg/L)	Avg DO (mg/L)
Low Pollution	118	21.76	1.93	4.73
Medium Pollution	45	9.02	1.17	7.76
High Pollution	5	20.60	14.60	0.56

Inference (Clustering): The elbow curve shows a clear inflection at $k = 3$, confirming three natural pollution tiers. High-pollution sites ($n = 5$) have BOD $\approx$ 14.6 mg/L – nearly 5× the safe limit – and dissolved oxygen near zero, indicating severe organic contamination and oxygen depletion. Medium-pollution sites tend to be cooler (avg 9°C), likely mountain rivers with higher natural oxygen levels. These 5 high-pollution stations should be prioritised for immediate environmental remediation.

Any Other Details

- **Dataset source:** CPCB (Central Pollution Control Board) India – 194 stations, 23 attributes, years 2021–2023.
- **Known data limitation:** Fecal Coliform and Total Coliform columns had 60–102 missing values and were excluded from ML models. Future work could use imputation or collect more complete coliform data.
- **Class imbalance note:** Water Safety dataset is moderately imbalanced (124 safe vs 70 unsafe). SMOTE oversampling or class weighting could further improve Logistic Regression recall for the unsafe class.
- **mongomock note:** Used for in-notebook MongoDB simulation. Real deployment would use MongoDB Atlas or a local `mongod` instance.
- **Encoding impact:** One-hot encoding increased feature space from 23 to 33 columns, but sparse binary columns for rare water body types (CREEK = 1 record, STP = 2 records) add noise. Feature selection post-encoding is recommended.

Summary / Conclusion

This lab assignment demonstrated a complete, domain-adapted data science pipeline on the Indian Water Quality Dataset. Starting from raw CSV data, the experiments progressed through multi-format interchange, text processing, relational and document database operations, array and dataframe manipulation, visualisation, and machine learning. Key conclusions:

1. **Data diversity:** Real-world water quality data exists across multiple formats (CSV, JSON, XML) and requires robust parsing and cleaning pipelines before any analysis.

2. **BOD is the key indicator:** BOD (Biological Oxygen Demand) emerged as the single most important feature for both supervised classification (feature importance = 0.52) and unsupervised clustering (cluster separation by BOD level).

3. **Random Forest > Logistic Regression:** For threshold-defined classification problems, tree-based ensembles naturally capture decision boundaries. Logistic Regression's 64% recall on the unsafe class would be unacceptable in a real public-health monitoring system.

4. **5 critical stations identified:** K-Means clustering identified 5 high-pollution water sources with near-zero dissolved oxygen, representing immediate environmental hazards requiring intervention.

5. **Unsupervised learning is valuable:** K-Means discovered meaningful pollution tiers without any labels, demonstrating that environmental monitoring can benefit even without prior labelling of water sources.

References / Websites

1. **Dataset – CPCB India Water Quality Portal:**
   `https://cpcb.nic.in/`
2. **Pandas Documentation (data handling, merge, groupby, plots):**
   `https://pandas.pydata.org/docs/`
3. **NumPy Documentation (array operations):**
   `https://numpy.org/doc/`
4. **scikit-learn – KNN, Logistic Regression, Random Forest, KMeans:**
   `https://scikit-learn.org/stable/documentation.html`
5. **Seaborn – Heatmap and statistical plots:**
   `https://seaborn.pydata.org/`
6. **MongoDB Documentation (CRUD, Aggregation, Indexes):**
   `https://www.mongodb.com/docs/`
7. **SQLite Documentation (CREATE, INSERT, UPDATE, DELETE):**
   `https://www.sqlite.org/docs.html`
8. **Python `re` module – Regular Expressions:**
   `https://docs.python.org/3/library/re.html`
9. **Matplotlib Documentation (bar, pie, histogram, scatter):**
   `https://matplotlib.org/stable/contents.html`
10. **K-Means Clustering – scikit-learn user guide:**
    `https://scikit-learn.org/stable/modules/clustering.html#k-means`