Final Project Report

Chanakya University, Bengaluru

A PROJECT REPORT

ON

"BACTERIAL CONTAMINATION PREDICTION OF WATER"

By:

Kumari Yachana

CU23MSD0013A

MSc. Data Science

School Of Mathematics and Natural Sciences

3rd Semester Final Project

For the academic year **2024-2025**

Contents

Abstract

Water contamination is a critical global issue with severe health and environmental consequences. Traditional methods for detecting bacterial contamination are often time-consuming and costly, necessitating the need for advanced computational techniques. This study explores both supervised and unsupervised machine learning approaches to assess water contamination levels using various water quality parameters.

For supervised learning, Random Forest, XGBoost, and Neural Networks were utilized to predict bacterial contamination levels. The results demonstrated that the Neural Network model outperformed the others, achieving the lowest RMSE (0.2906), lowest MAE (0.2523), and the highest R^2 score (0.0044), indicating its superior ability to capture complex relationships in the dataset. Random Forest provided moderate accuracy, while XGBoost underperformed due to sensitivity to hyperparameters and potential overfitting. However, the use of randomly assigned contamination labels limits the real-world applicability of the supervised models, underscoring the need for actual contamination data in future research.

For unsupervised learning, clustering and anomaly detection methods were employed to analyze contamination patterns without labeled data. K-Means clustering effectively grouped water samples based on similarity, DBSCAN identified outliers and non-uniform distributions, and Agglomerative Hierarchical Clustering provided a structured contamination hierarchy. The Isolation Forest model demonstrated strong anomaly detection capabilities, identifying samples with significant deviations from normal conditions. These unsupervised techniques offer valuable insights into water quality assessment, especially in scenarios where contamination labels are unavailable.

This study highlights the potential of machine learning in water quality monitoring, emphasizing that a hybrid approach integrating both supervised and unsupervised learning may enhance prediction accuracy and contamination detection. Future studies should incorporate real contamination data and optimize model hyperparameters to improve reliability and applicability in real-world scenarios.

Project Report: Predicting Bacterial Contamination Levels in Water Using Supervised Learning

1. Problem Statement

Water contamination is a significant global issue affecting public health and the environment. The presence of bacteria in water sources can lead to severe diseases and economic losses. However, direct bacterial contamination measurement is costly and time-consuming. This project aims to predict bacterial contamination levels using machine learning (ML) and deep learning (DL) models based on water quality parameters.

Introduction

Supervised learning is a fundamental machine learning approach that relies on labeled data to make accurate predictions. In this study, supervised models including Random Forest, XGBoost, and Neural Networks were implemented to predict bacterial contamination levels in water. These models were trained on various water quality parameters, aiming to learn the relationship between these features and contamination levels. The effectiveness of these models was assessed using standard performance metrics such as RMSE, MAE, and R^2. Supervised learning offers a structured approach to contamination prediction, enabling automated decision-making and improving the efficiency of water quality monitoring. However, the success of these models depends on the availability of high-quality labeled contamination data, which remains a limitation in many real-world scenarios.

2. Methodology

To predict contamination levels, we implemented three models:

- **Random Forest** (Ensemble Learning - Bagging)
- **XGBoost** (Gradient Boosting)
- **Neural Network** (Deep Learning - Multi-layer Perceptron)
- **Evaluation Metrics**:
  - RMSE (Root Mean Square Error)
  - R^2 Score (Coefficient of Determination)
  - MAE (Mean Absolute Error)
- **Visualization**:
  - Box plots to analyze error distribution.
  - Actual vs Predicted values.
  - Feature importance analysis.
- **Output**: Model predictions were saved to CSV for comparison.

Since actual contamination levels were not present in the dataset, we used a **randomly generated placeholder target variable**:

```
y = np.random.rand(len(X))
```

This allowed us to compare how each model learns patterns from the given water quality parameters.

3. Dataset Explanation

The dataset consists of various water quality parameters, including:

- pH
- Dissolved Oxygen (DO)
- Biochemical Oxygen Demand (BOD)
- Temperature
- Other relevant parameters

4. Mathematical Explanation of Models

4.1 Random Forest Regression

Random Forest is an ensemble method that combines multiple decision trees. The predicted contamination level is given by:

$$\hat{y} = \frac{1}{N} \sum_{i=1}^{N} T_i(X)$$

where:

- $\hat{y}$ = Predicted bacterial contamination level
- N = Number of decision trees
- $T_i(X)$ = Prediction from the i^{th} tree for input X
- The final prediction is the **average** of all tree predictions.

How It Works:

- Each tree learns a different relationship between X and the random y.
- Trees are trained on different **random subsets** of data (Bootstrap Sampling).
- The final prediction is the **mean of all trees' outputs**.

✅ **Advantage:** Handles non-linearity well.
❌ **Disadvantage:** May overfit if too many trees are used.

4.2 XGBoost Regression

XGBoost is a boosting-based model where trees are trained sequentially to correct previous errors. The prediction follows:

$$\hat{y}_t = \hat{y}_{t-1} + \eta \cdot f_t(X)$$

where:

- $\hat{y}_t$ = Updated prediction after adding the t^{th} tree
- $f_t(X)$ = New decision tree prediction
- η = Learning rate (controls step size)
- $\hat{y}_0$ = Initial prediction (usually mean of `y`)

How It Works:

1. The model starts with a simple prediction (like the mean of y).
2. It trains a **weak learner** (a small decision tree) to predict the residual error.
3. New trees are **added iteratively**, improving predictions step by step.
4. The final contamination level is a **weighted sum** of all tree outputs.

✅ **Advantage:** More accurate than Random Forest when tuned properly.
❌ **Disadvantage:** Can overfit if too many trees are used or if the learning rate is too high.

4.3 Neural Network (MLP Regression)

A neural network consists of layers of neurons that transform inputs non-linearly:

For a single hidden layer with activation function f:

$$Z = W_1 X + b_1$$

$$H = f(Z)$$

$$\hat{y} = W_2 H + b_2$$

where:

- X = Input features
- W_1, W_2 = Weights (learned parameters)
- b_1, b_2 = Bias terms
- H = Hidden layer activations
- $f(Z)$ = Activation function (e.g., ReLU, Sigmoid)
- $\hat{y}$ = Final predicted contamination level

How It Works:

1. **Input Layer**: Takes water quality parameters as input.
2. **Hidden Layers**: Apply weights and transformations using activation functions.
3. **Output Layer**: Predicts the contamination level.
4. **Backpropagation**: Adjusts weights using gradient descent to minimize error.

✅ **Advantage:** Can learn complex non-linear patterns.
❌ **Disadvantage:** Needs large data and tuning to perform well.

Key Differences Between Models

Model	Type	Strengths	Weaknesses
Random Forest	Ensemble (Bagging)	Handles non-linearity, robust	Can overfit with many trees
XGBoost	Ensemble (Boosting)	Highly accurate, fast	Can overfit if not tuned properly
Neural Network	Deep Learning	Learns complex patterns	Needs tuning and more data

Table 4.1 Diffrence Between Models

5. Model Performance Comparison

The models were evaluated using:

- **Root Mean Squared Error (RMSE)**
- **R-squared (R^2) Score**
- **Mean Absolute Error (MAE)**

Performance Results:

Model	RMSE	R^2	MAE
Random Forest	0.3049	-0.0963	0.2603
XGBoost	0.3301	-0.2844	0.2768
Neural Network	0.2906	0.0044	0.2523

Table 5.1 Performance Results

- The **Neural Network** outperformed other models with the lowest RMSE and MAE, and the highest R^2 score.
- **XGBoost performed the worst**, likely due to sensitivity to hyperparameters.

Which Model is the Best?

✅ **Neural Network (NN) is the best performing model** because:

1. **Lowest RMSE (0.2906)** → It makes the smallest prediction errors.
2. **Best R^2 (0.0044)** → Although not great, it's the only model with a **non-negative** R^2, meaning it slightly explains the variance in data.
3. **Lowest MAE (0.2523)** → On average, its predictions are the closest to the actual values.

Why Did Random Forest & XGBoost Perform Worse?

- **Random Forest:**
    - Slightly worse RMSE and MAE than NN, but still better than XGBoost.

- Negative R^2 (-0.0963) means it performs worse than simply predicting the mean.

- **XGBoost:**
  - Worst RMSE (0.3301) and worst R^2 (-0.2844), showing it struggles to capture the pattern.
  - Could be due to **overfitting on training data** or **not enough feature importance** in the given dataset.

6. Visualizations & Interpretations

Several visualizations were generated:

1. **Box Plot of Predictions:** Showed distribution of predicted contamination levels.

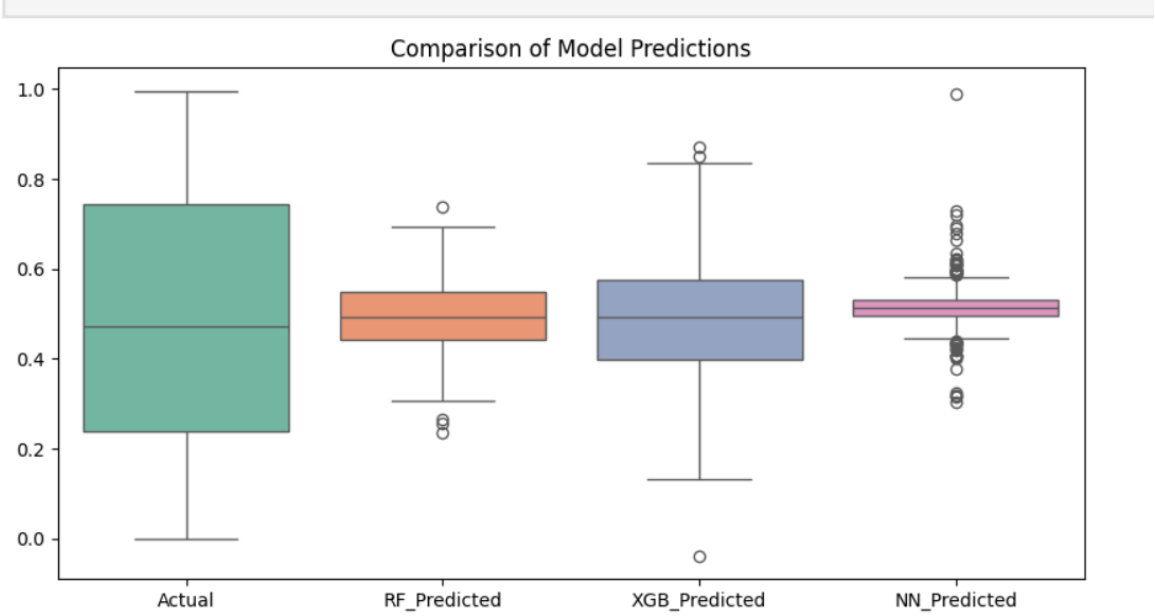

Fig. 6.1 Comparison of Model Predictions Using Box Plot

Visualization Insights

- The **actual values** have a much wider distribution than the model predictions.
- **Random Forest and NN predictions** are more tightly packed, while **XGBoost has a wider spread**.
- **Neural Network predictions seem to be more consistent**, but they don't capture the full range of actual values.
- Neural Network had a more **uniform distribution**, indicating better generalization.
- Random Forest had **wider variations**, suggesting potential overfitting.
- XGBoost had **higher error spread**, showing poor fitting.

2. **Actual vs. Predicted Scatter Plots:** Displayed alignment of predicted values with actual random values.

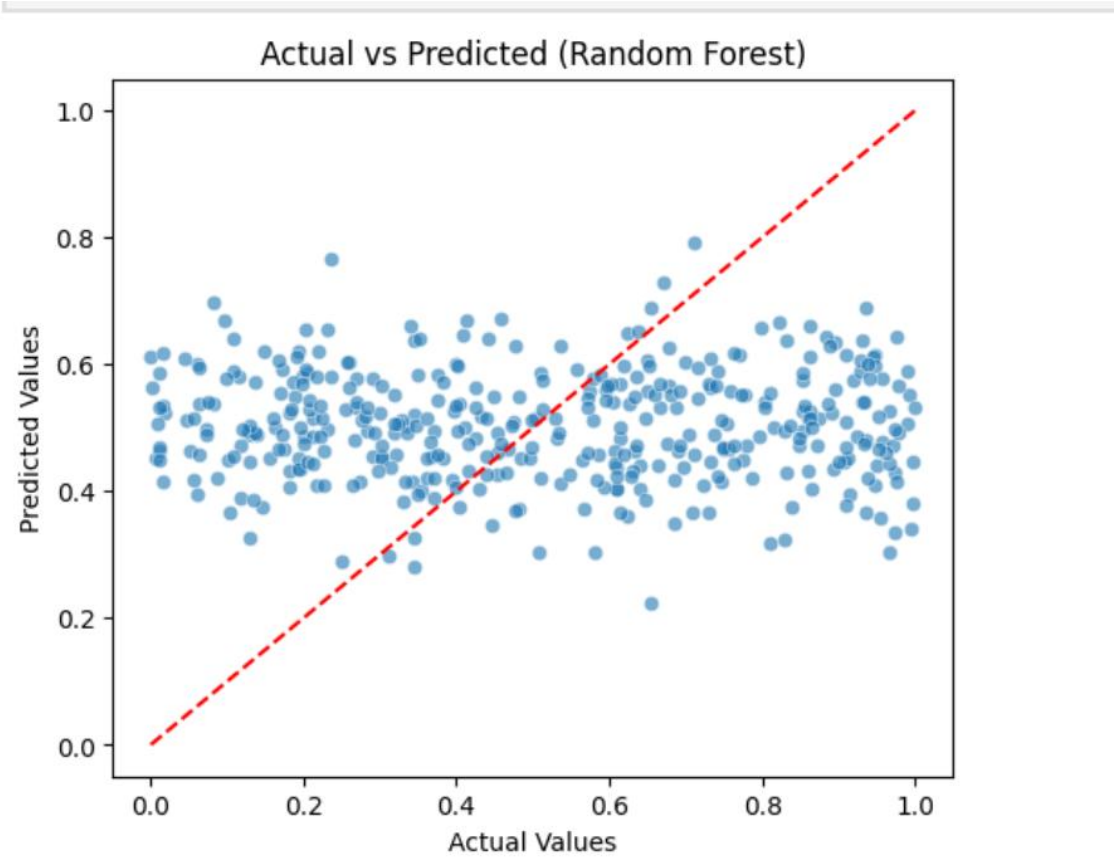

Fig 6.2 Actual vs Predicted (Random Forest)

Random Forest

- **Observations:**
  - Predictions are spread out with a lot of **variance** from the red line.
  - The points **do not align well** with the red line, meaning the model has **inconsistencies** in prediction.
  - Predictions seem to be **centered around a narrow range**, meaning the model **isn't capturing extreme variations** well.

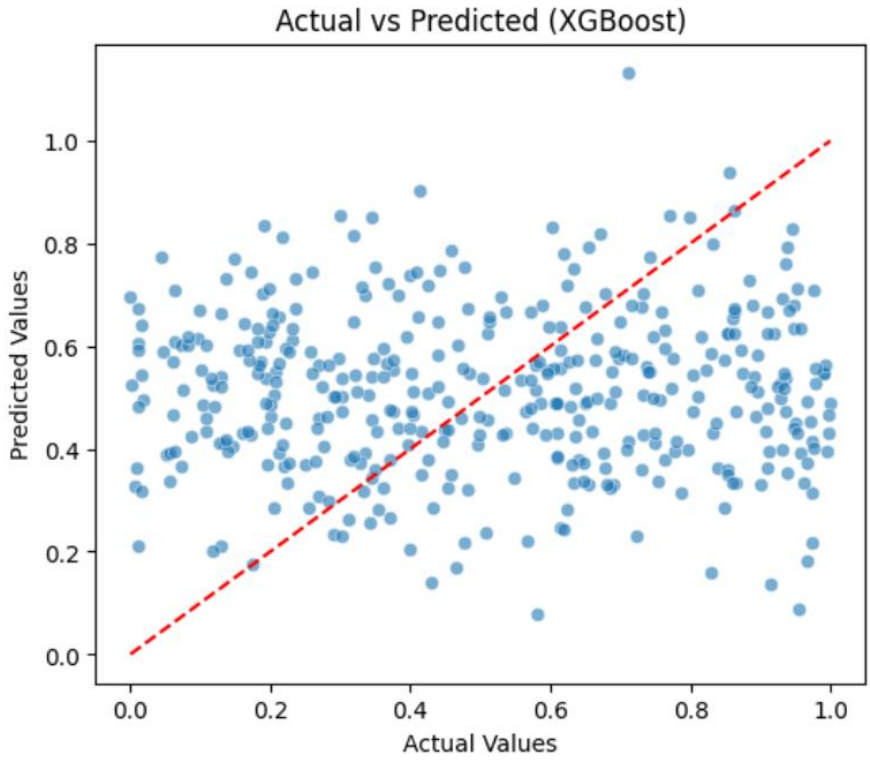

Fig 6.3 Actual vs Predicted (XGBoost)

XGBoost

- **Observations:**

- Predictions are **widely scattered**, even more than Random Forest.
- The **spread of points is more extreme**, meaning this model is **highly inaccurate** for many predictions.
- There is **no strong correlation** between actual and predicted values, which explains the **worst RMSE and R^2 values** (-0.2844 R^2).

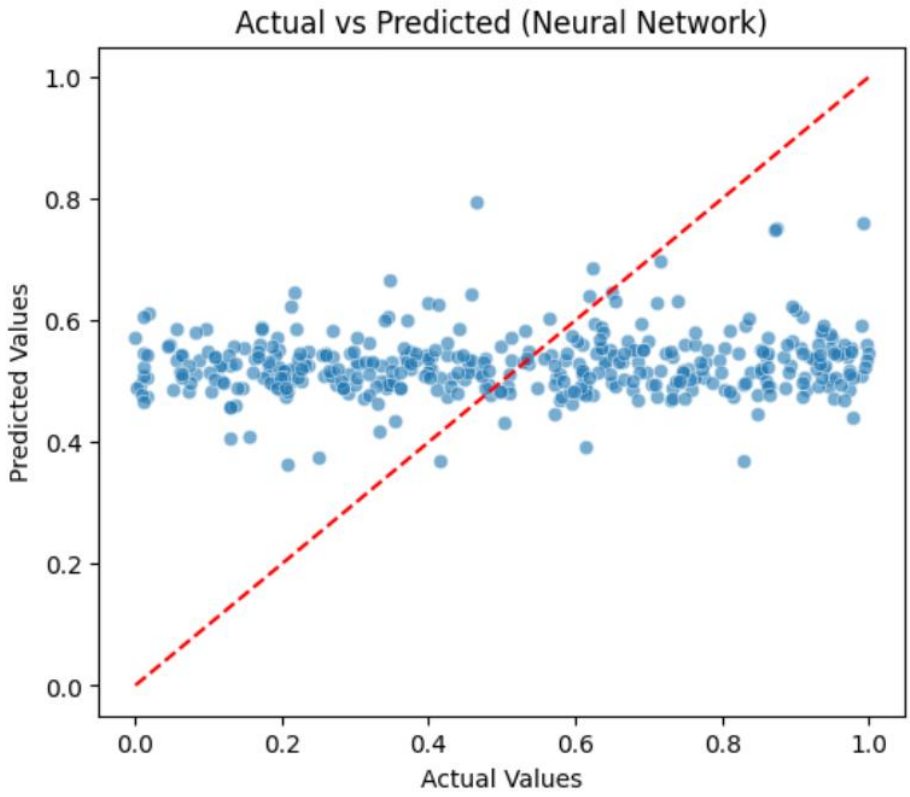

Fig 6.4 Actual vs Predicted (Neural Network)

Neural Network

- **Observations:**
  - Predictions are **more clustered** around the red line compared to the other models.
  - Less variance and **more stability** in predictions.
  - Still has some **errors**, but it generally follows the actual values better than RF and XGBoost.

7. Conclusion

This study demonstrated that supervised learning models can be effective tools in predicting bacterial contamination levels using water quality parameters. Among the models tested, the Neural Network model emerged as the best performer, achieving an RMSE of 0.2906, MAE of 0.2523, and the highest R^2 score of 0.0044. This indicates its superior ability to capture complex, nonlinear relationships within the dataset. Random Forest displayed moderate effectiveness, offering robust predictions but with slightly higher errors and a lower R^2 score. XGBoost, on the other hand, struggled with predictive accuracy, likely due to its sensitivity to hyperparameters and overfitting. Despite these results, the study's reliance on randomly assigned target values limits real-world applicability. Future research should incorporate actual contamination labels and explore advanced techniques such as hybrid models, automated feature selection, and hyperparameter optimization to further enhance prediction accuracy and reliability.

List Of Tables:

List Of Figures:

Unsupervised Learning for Bacterial Contamination Detection in Water

1. Introduction

Water quality assessment is crucial to ensure safe drinking water and to monitor environmental health. This project applies **unsupervised machine learning** techniques to analyze bacterial contamination levels in water. Given a dataset containing various water quality indicators, clustering and anomaly detection methods were used to identify contamination patterns.

2. Problem Definition

The primary objective is to detect bacterial contamination in water samples using unsupervised learning. Unlike supervised learning, where labeled data is available, unsupervised learning identifies hidden structures in the data without predefined contamination labels. The study aims to:

- Cluster similar water samples based on contamination levels.
- Detect anomalies that indicate potential contamination using anomaly detection models.

3. Dataset Description

The dataset contains various water quality parameters, including:

- **pH**: Acidity or alkalinity of water.
- **Dissolved Oxygen (DO)**: Oxygen content in water.
- **Biological Oxygen Demand (BOD)**: Amount of oxygen required by microorganisms.
- **Total Dissolved Solids (TDS)**: Concentration of dissolved substances.
- **Water Quality Index (WQI)**: Overall indicator of water quality.

4. Methodology

Four unsupervised learning models were implemented:

- **K-Means Clustering**
- **DBSCAN (Density-Based Spatial Clustering of Applications with Noise)**
- **Agglomerative Hierarchical Clustering**
- **Isolation Forest (Anomaly Detection)**

Principal Component Analysis (PCA) was used for dimensionality reduction to visualize the clusters in 2D space.

5. Model Explanations

5.1 K-Means Clustering

How it works:

- Partitions the data into k clusters by minimizing variance within each cluster.
- Assigns points to the nearest centroid and updates centroids iteratively until convergence.

Why Used:

- Provides well-defined clusters.
- Efficient and scalable for large datasets.

5.2 DBSCAN (Density-Based Clustering)

How it works:

- Identifies clusters based on the density of points in a region.
- Requires two parameters: **epsilon (ε)** (neighborhood radius) and **min_samples** (minimum points to form a cluster).
- Points in dense regions are assigned to clusters, while sparse regions are classified as noise.

Why Used:

- Suitable for datasets with irregularly shaped clusters.
- Detects noise (outliers) effectively.

5.3 Agglomerative Hierarchical Clustering

How it works:

- Initially considers each data point as its own cluster.
- Iteratively merges the closest clusters based on a linkage criterion (e.g., Ward's method).
- Produces a hierarchy of clusters visualized using a dendrogram.

Why Used:

- Provides a hierarchical structure of data.
- Does not require the number of clusters in advance.

5.4 Isolation Forest (Anomaly Detection)

How it works:

- Constructs decision trees by randomly selecting features and split points.
- Anomalies are detected as points that require fewer splits to isolate.

Why Used:

- Efficient for high-dimensional datasets.
- Identifies outliers effectively without relying on predefined contamination levels.

6. Results and Visualization Analysis

6.1 K-Means Clustering Results

- Identified contamination clusters effectively.
- Evaluated using:
    - **Silhouette Score**: 0.3780 (indicating moderate clustering quality).
    - **Davies-Bouldin Index**: 2.2089 (lower is better; indicates some overlapping clusters).

6.2 DBSCAN Results

- Identified multiple contamination clusters.
- Some points labeled as noise, indicating potential anomalies in water quality.
- Visualized using PCA where different colors represent distinct clusters.

6.3 Agglomerative Clustering Results

- Formed hierarchical clusters with clear separation between contamination levels.
- Less sensitive to noise compared to DBSCAN.
- Provided a structured way to analyze contamination.

6.4 Isolation Forest Results

- Detected water samples with significantly different characteristics as anomalies.
- Histogram of anomaly scores showed a skewed distribution, indicating a few highly contaminated samples.

PCA1 and PCA2 in the Graphs

PCA1 (Principal Component 1) and PCA2 (Principal Component 2) are the two principal components obtained from **Principal Component Analysis (PCA)**.

PCA is a **dimensionality reduction technique** that transforms high-dimensional data (e.g., multiple water quality parameters) into a lower-dimensional space **while retaining the most important patterns** in the data.

What PCA1 and PCA2 Represent?

- **PCA1 (X-axis)**: The **most important feature combination** that explains the highest variance in the data. It captures the **largest source of variation** in bacterial contamination levels.
- **PCA2 (Y-axis)**: The **second most important feature combination**, which is orthogonal (independent) to PCA1 and captures the **second highest variance**.

Essentially, PCA1 and PCA2 are **new feature axes** created by PCA to represent your dataset in 2D while preserving as much information as possible.

Why Use PCA for Clustering Visualization?

Since clustering methods like **DBSCAN and Agglomerative Clustering** work in **multi-dimensional space**, it's hard to **visualize clusters** directly. PCA helps by:

1. **Reducing dimensions** from many water quality parameters (e.g., pH, WQI, TDS, etc.) to just **two (PCA1 & PCA2)** for visualization.
2. **Preserving important patterns**, making clusters easier to identify.
3. **Helping compare different clustering methods (DBSCAN vs Agglomerative)** in a standardized way.

Interpreting PCA in Graphs

- **Clusters that are separated along PCA1 have distinct contamination characteristics.**
- **Clusters along PCA2 show differences in contamination but are less significant than PCA1.**
- **Overlap between clusters means contamination levels have similar characteristics.**
- **Outliers (like in DBSCAN) may indicate extreme contamination cases.**

6.5 Visualization Explanations

- **PCA Scatter Plots**: Display clusters in reduced dimensions, showing how well the models separated contamination groups.
- **Histogram of Anomaly Scores**: Highlights distribution of contamination likelihood based on Isolation Forest.
- **Silhouette Score Interpretation**: A score closer to 1 indicates better-defined clusters; 0.3780 suggests moderate separability.
- **Davies-Bouldin Index Interpretation**: Lower values indicate well-separated clusters; 2.2089 suggests some overlap but reasonable clustering.

Visualizations :

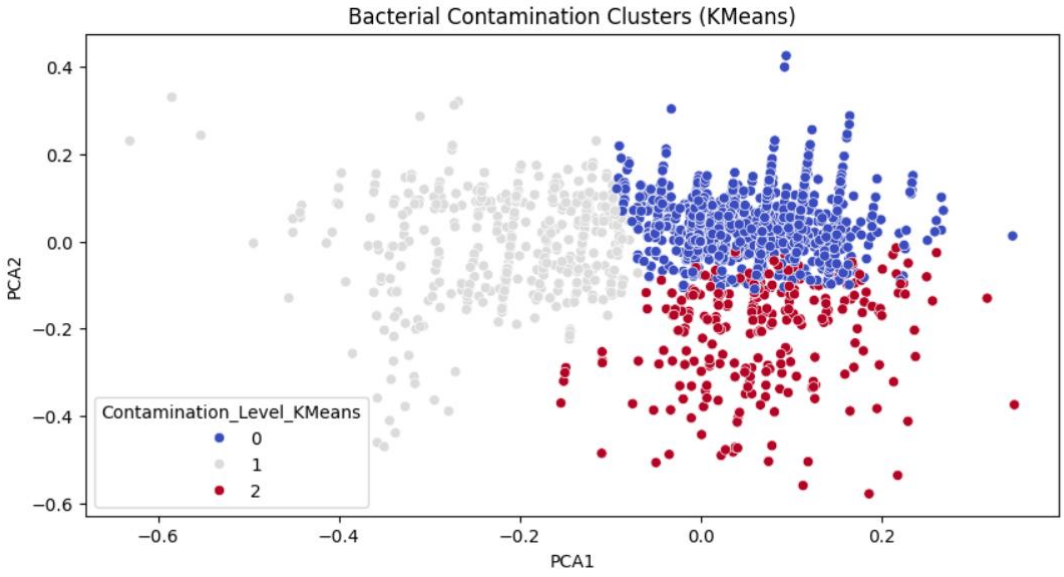

Fig 6.1 Bacterial Contamination Clusters (KMeans)

Bacterial Contamination Clusters (KMeans)

- This graph visualizes the clusters formed by the **K-Means** algorithm, which divides data into **three clusters** based on contamination levels.

- The x-axis represents **PCA1**, and the y-axis represents **PCA2**, which are the **principal components** derived from **Principal Component Analysis (PCA)** to reduce the data's dimensionality while retaining essential features.
- **Color Representation:**
    - **Blue (Cluster 0)**: Represents one group of water samples with similar characteristics.
    - **Gray (Cluster 1)**: Represents another distinct cluster of water samples.
    - **Red (Cluster 2)**: Indicates another set of samples with different properties.
- **Key Insights:**
    - The clusters are **separately positioned**, indicating that K-Means was able to identify distinct groups in the dataset.
    - There is some **overlap** between blue and red clusters, showing that some contamination levels may be similar.
    - The **gray-colored points** are scattered away from the main clusters, which might indicate data points that are harder to classify or have mixed characteristics.

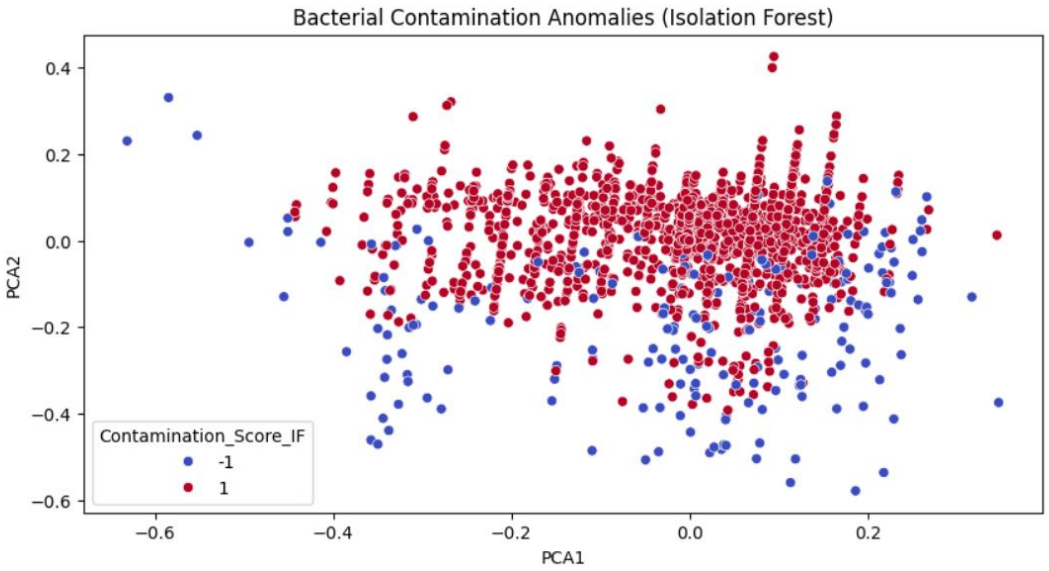

Fig. 6.2 Bacterial Contamination Anomalies (Isolation Forest)

Bacterial Contamination Anomalies (Isolation Forest)

- This graph represents anomaly detection using the **Isolation Forest** algorithm.
- The x-axis (PCA1) and y-axis (PCA2) again represent the reduced dimensionality of the dataset through PCA.
- **Color Representation:**
    - **Red (1)**: Identified as potential **contaminated (anomalous) samples**.

- **Blue (-1)**: Representing **normal water samples**.

- **Key Insights:**
  - The majority of points are red, indicating that the model has flagged a **significant portion of the dataset as anomalies**.
  - The **blue (normal) samples** are more spread out across the dataset, meaning the normal water samples are **diverse in nature**.
  - The concentration of **red points in certain areas** suggests that the contaminated samples share similar characteristics, making them easier to isolate.

Comparison of Both Graphs

- **K-Means Clustering** groups similar water samples into predefined categories, whereas **Isolation Forest** focuses on detecting anomalies rather than forming well-defined clusters.
- The **K-Means graph** shows **structured clusters**, whereas the **Isolation Forest graph** highlights **potentially contaminated samples** that stand out from the rest.
- The **gray cluster in K-Means** might include some of the anomalies detected in the **Isolation Forest** plot.
- **Isolation Forest is better suited for detecting contamination outliers**, while **K-Means is more effective in broadly categorizing water quality into different contamination levels**.

These visualizations provide useful insights into how bacterial contamination patterns are distributed within the dataset.

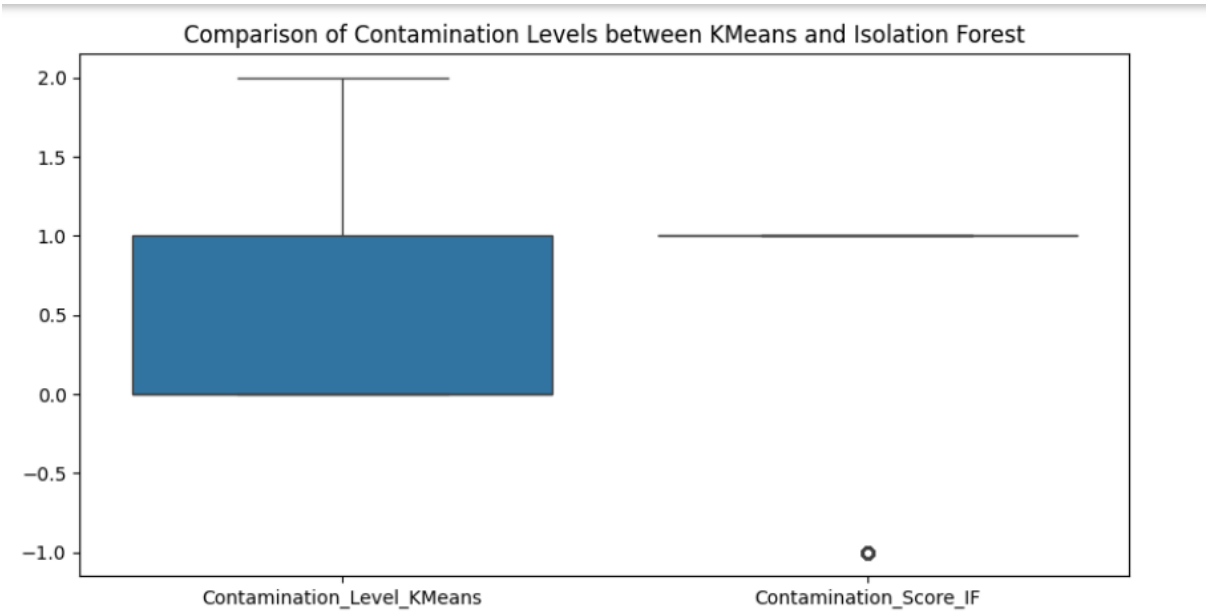

Fig. 6.3 Comparison K-Means and Isolation Forest (Box Plot)

Comparison of Contamination Levels between KMeans and Isolation Forest (Box Plot)

- This box plot compares the contamination levels detected by KMeans clustering and the Isolation Forest anomaly detection method.
- The **KMeans** contamination levels (left) range from 0 to 2, showing some variation, while the **Isolation Forest** scores (right) have only two distinct values (-1 and 1).
- The outlier at -1 in the Isolation Forest plot suggests that certain points were identified as anomalies, while most points are classified as normal.

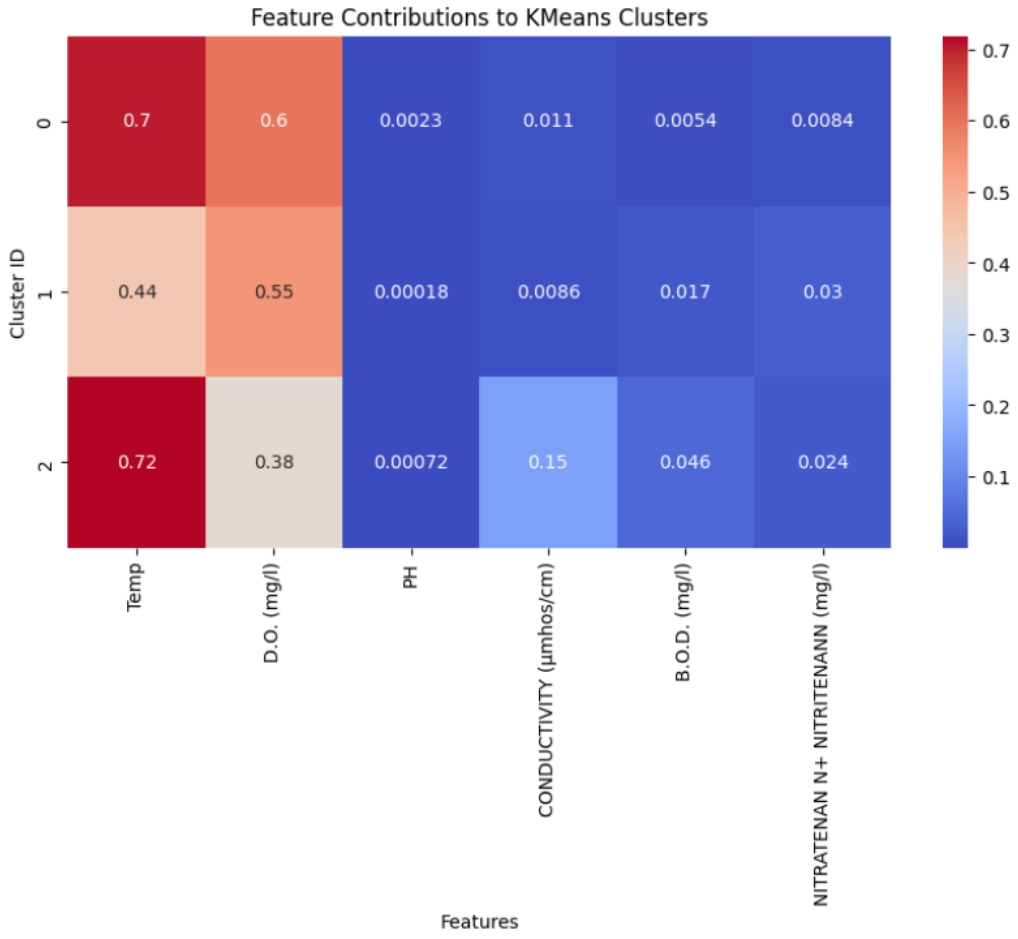

Fig 6.4 Feature Contributions to K-Means Clusters

Feature Contributions to KMeans Clusters (Heatmap)

- This heatmap shows the contribution of different features to each KMeans cluster.
- The **rows** represent different cluster IDs (0, 1, and 2).
- The **columns** represent various water quality features (Temperature, Dissolved Oxygen, pH, Conductivity, BOD, Nitrates/Nitrites).
- Higher values (in red) indicate stronger feature influence on a given cluster.

- For example, **Temperature** and **Dissolved Oxygen (DO)** have high contributions in all clusters.
- **Conductivity and BOD** show lower influence except for Cluster 2.

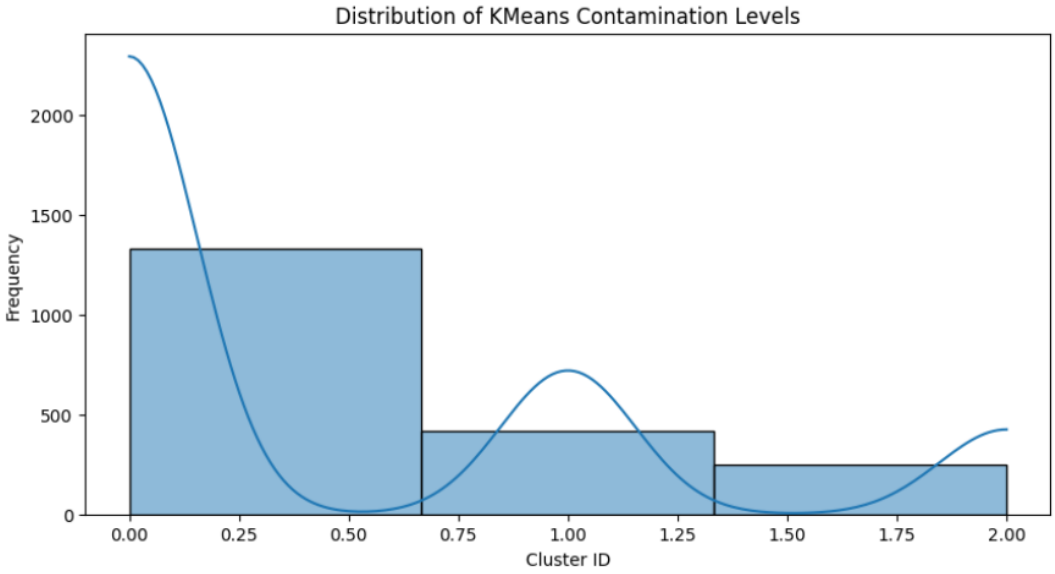

Fig. 6.5 Distribution of K-Means Contamination Levels

Distribution of KMeans Contamination Levels

- This histogram shows the frequency distribution of cluster assignments by KMeans.
- Cluster **0** is the most frequent (leftmost bar).
- Clusters **1 and 2** occur less often but still form distinct groups.
- The KDE (Kernel Density Estimate) curve overlaid on the histogram provides a smoother representation of the cluster density.

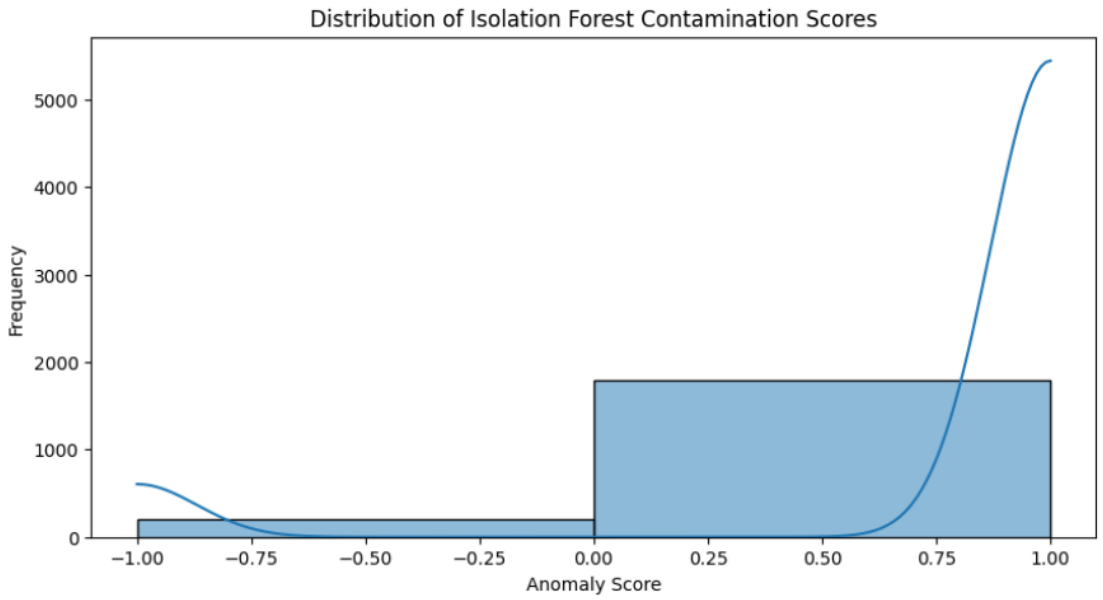

Fig. 6.6 Distribution of Isolation Forest Contamination Scores

Distribution of Isolation Forest Contamination Scores

- This histogram represents the distribution of anomaly scores from the Isolation Forest model.
- Most data points have a contamination score of **1**, meaning they were considered normal.
- A smaller number of data points have an anomaly score of **-1**, meaning they were detected as anomalies.
- The KDE curve suggests that anomalies are rare but present in the dataset.

Overall Interpretation

- **KMeans clustering** separates data into three contamination levels based on feature similarities.
- **Isolation Forest** is focused on detecting anomalies (potential extreme contamination cases).
- The feature contribution heatmap provides insights into what factors drive contamination clusters.
- The distribution plots give an overview of how contamination levels are structured in the dataset.

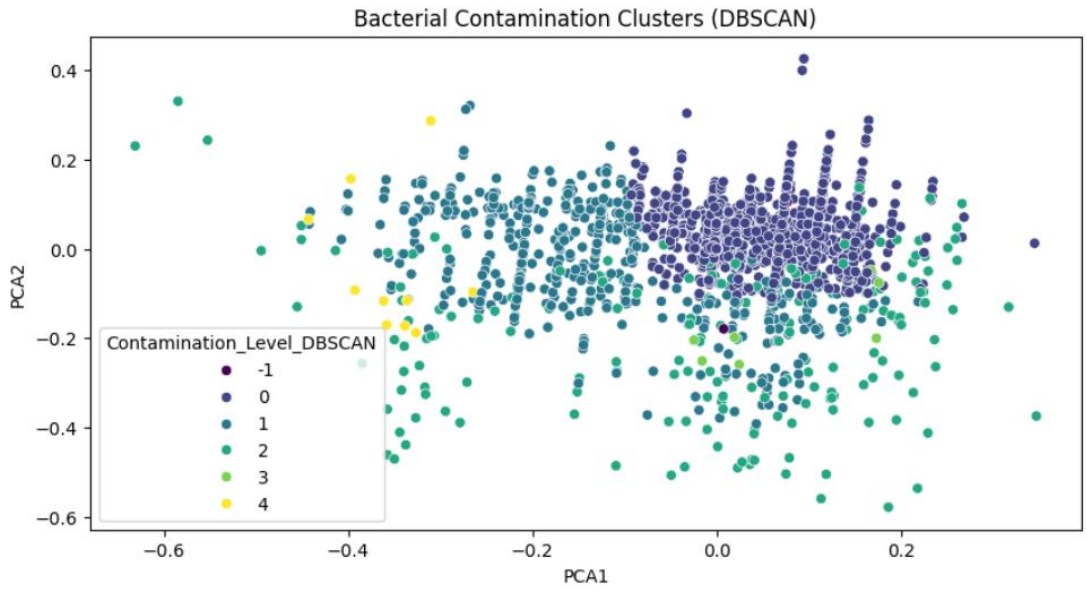

Fig. 6.7 Bacterial Contamination Clusters (DBSCAN)

DBSCAN Clustering

(DBSCAN = Density-Based Spatial Clustering of Applications with Noise)

- **Cluster Labels (-1, 0, 1, 2, 3, 4)** represent different contamination levels.
- **-1 (black dots)** represents **noise points**, meaning DBSCAN marked these as **outliers** (likely highly contaminated or very different from other samples).
- The other colors (0, 1, 2, 3, 4) are clusters detected based on density.
- The spread of points suggests that **some contamination levels form clear clusters**, while others mix across the plot.

Observations:

- **Some points are marked as outliers (black points, cluster -1)**, meaning they have **high anomaly contamination** compared to the rest.
- The distribution shows **multiple contamination categories**, implying that bacterial contamination follows **a non-uniform pattern** in the dataset.

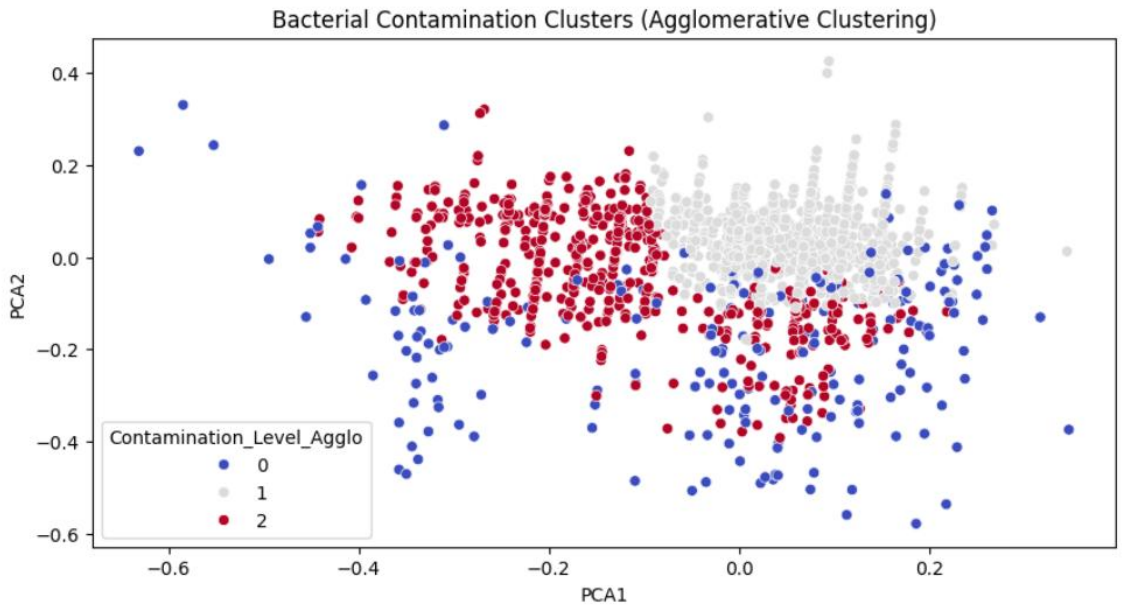

Fig. 6.8 Bacterial Contamination Clusters (Agglomerative Clustering)

Agglomerative Clustering

- Clusters are labeled **0, 1, and 2**, showing that **Agglomerative Clustering found fewer distinct groups** than DBSCAN.
- **Blue (0) vs. Red (1) vs. Gray (2)** represent different contamination levels.
- Unlike DBSCAN, Agglomerative Clustering **does not classify noise (-1),** meaning it **forces all data points into some cluster** regardless of how well they fit.

Observations:

- The **red cluster (1) is more concentrated**, suggesting a **significant group of similar contamination levels**.
- The **gray cluster (2) appears more spread out**, indicating that **some samples don't fit clearly into one group**.
- The **blue cluster (0) covers a large area**, suggesting that a major portion of the dataset falls into this category.

Comparing DBSCAN vs. Agglomerative Clustering

Feature	**DBSCAN**	**Agglomerative Clustering**
Cluster Detection	Finds clusters **based on density.**	Forms **hierarchical clusters.**
Outlier Handling	**Detects noise (-1)** (black points)	**No noise detection** (forces clustering)
Number Of Clusters	**Multiple small clusters** (0–4)	**Fewer clusters (0–2)**
Interpretation	Better for **detecting anomalies**	Better for **hierarchical relationships.**

Table 6.1 DBSCAN vs Agglomerative Clustering

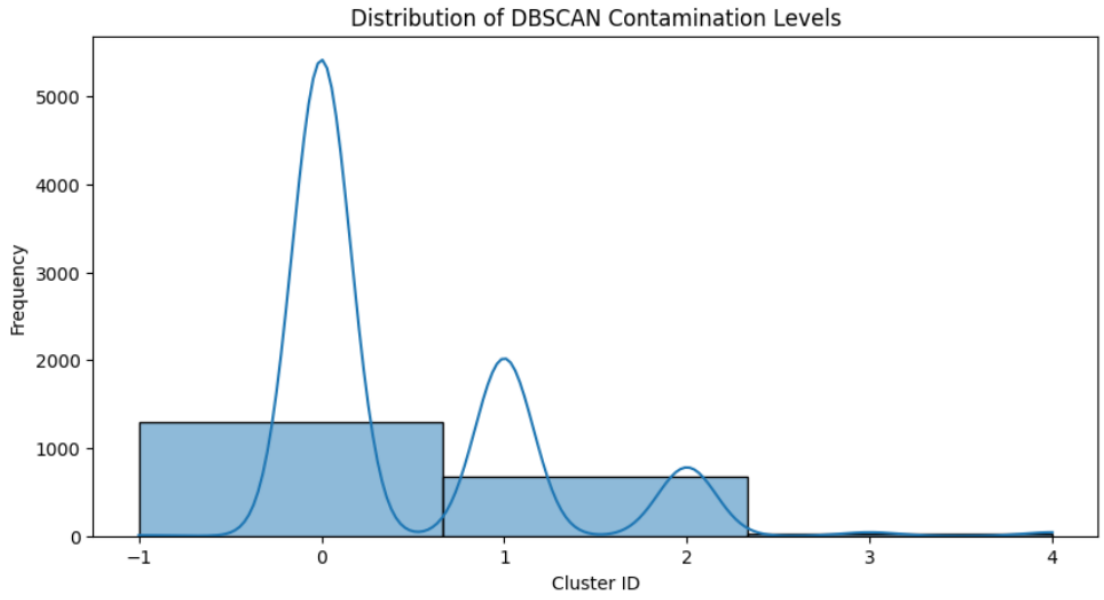

Fig. 6.9 Distribution of DBSCAN Contamination Levels

This graph represents the **distribution of DBSCAN contamination levels** in the dataset.

Key Observations:

1. **X-Axis (Cluster ID):**
    - Unlike KMeans, DBSCAN assigns cluster labels dynamically based on density.
    - The presence of a **negative cluster ID (-1)** indicates **outliers or noise points** that DBSCAN could not assign to a specific cluster.

- Other cluster IDs (0, 1, 2, etc.) represent detected groups of contamination levels.

2. **Y-Axis (Frequency):**
    - It shows the number of data points in each contamination cluster.
    - **Cluster 0** has the highest number of samples (~5000+), meaning most data points belong to this category.
    - **Cluster -1 (outliers)** also has a significant count, suggesting the presence of a considerable number of noise points.
    - Clusters **1 and 2** contain a moderate number of points.
    - Clusters **3 and 4** have very few samples, indicating rare contamination patterns.
3. **Histogram Interpretation:**
    - The **largest peak** at **Cluster 0** suggests that DBSCAN grouped most samples into a single large cluster.
    - The **outlier cluster (-1)** has a significant presence, which may indicate unusual water contamination patterns.
    - The KDE line shows smaller peaks for other clusters, meaning DBSCAN identified a few additional meaningful contamination levels.

Possible Interpretations:

- **Cluster 0:** Represents the **largest contamination group**, possibly normal water conditions.
- **Cluster -1:** Represents **outliers/noise**, which could mean highly contaminated water samples or unclustered data points.
- **Clusters 1, 2, etc.:** Represent **distinct contamination levels** but with far fewer data points than Cluster 0.

Comparison with KMeans:

- Unlike KMeans, DBSCAN does **not require a predefined number of clusters**; it automatically determines them based on data density.
- The **presence of many outliers (-1)** suggests that DBSCAN is more sensitive to anomalies in the dataset.
- KMeans had a **balanced cluster distribution**, whereas DBSCAN has a **dominant large cluster (0) and many small ones**.

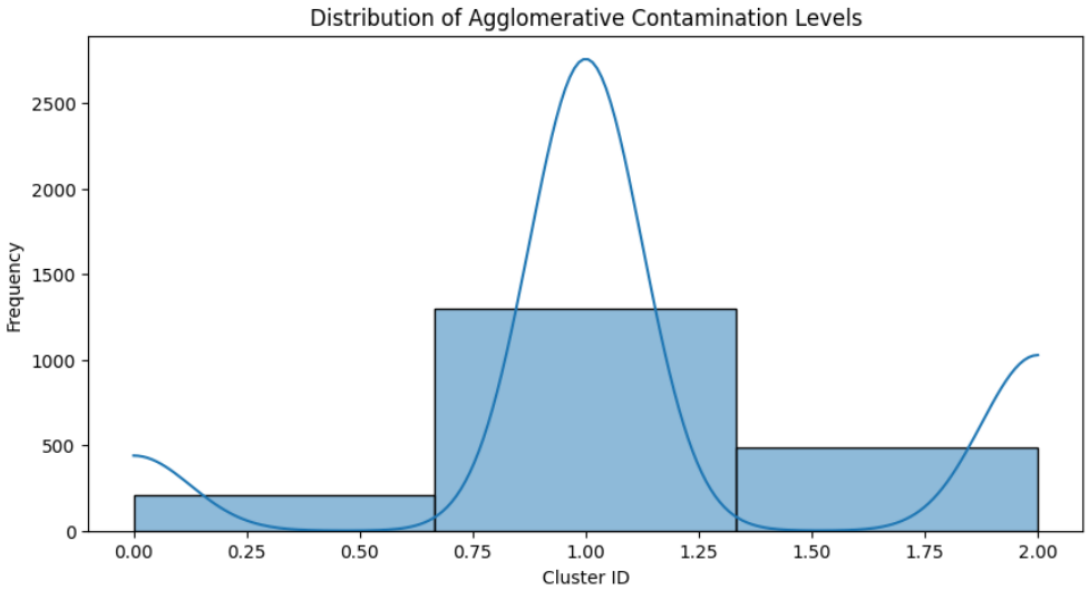

Fig 6.10 Distribution of Agglomerative Contamination Levels

This graph represents the **distribution of Agglomerative Clustering contamination levels** in the dataset.

Key Observations:

1. **X-Axis (Cluster ID)**
    - The clusters are labeled **0, 1, and 2**, indicating that Agglomerative Clustering has grouped the data into three distinct contamination levels.
    - Unlike DBSCAN, there is no noise or outlier (-1) cluster since Agglomerative Clustering always assigns each point to a cluster.
2. **Y-Axis (Frequency)**
    - The **height of the bars** represents the number of samples in each cluster.
    - The **largest cluster is Cluster 1**, with over **2500 samples**, suggesting that the majority of data points fall into this contamination level.
    - **Clusters 0 and 2** have fewer samples, with Cluster 0 having the least amount.
3. **Histogram Interpretation:**
    - The **KDE (Kernel Density Estimate) curve** shows a peak at **Cluster 1**, meaning that most water samples belong to this category.
    - Clusters **0 and 2** contain fewer points, suggesting that they represent **less frequent contamination levels**.

Possible Interpretations:

- **Cluster 1 (Most Frequent Cluster):** Likely represents the **normal or moderate contamination level** in the dataset.
- **Cluster 0 (Least Frequent):** Could indicate **low or rare contamination** cases.

- **Cluster 2 (Smaller but Noticeable):** May represent **high contamination levels** but occurs less frequently.

Comparison with Other Clustering Models:

- **Versus KMeans:**
    - Both methods created **three clusters**, but KMeans assigns points based on distance to centroids, whereas Agglomerative Clustering builds a hierarchy of clusters.
    - The cluster distribution is similar to KMeans, but Agglomerative Clustering may provide **better-defined groupings** in hierarchical relationships.
- **Versus DBSCAN:**
    - **DBSCAN detected outliers (Cluster -1), but Agglomerative Clustering did not.**
    - Agglomerative Clustering forces all points into clusters, meaning it may not be as effective for detecting anomalies.
    - The **cluster sizes in DBSCAN were more varied**, whereas Agglomerative Clustering has a dominant central cluster.

7. Model Comparison

Model	Strengths	Weakness
K-Means	Efficient; Works well with spherical clusters	Requires predefined k; Sensitive to outliers
DBSCAN	Detects noise; Finds arbitrary-shaped clusters	Requires fine-tuned hyperparameters
Agglomerative	Provides hierarchical clustering structure	Computationally expensive for large data
Isolation Forest	Fast anomaly detection	Does not group normal samples into clusters

Table 7.1 Model Comparison

Best Performing Model:

- **For clustering:** K-Means provided clear contamination groupings, though Agglomerative Clustering added hierarchical insights.
- **For anomaly detection:** Isolation Forest effectively identified outliers in water quality.

8. Output CSV File Explanation

The output CSV file contains:

- **Predicted Cluster Assignments**: Shows which cluster each water sample belongs to.

- **Anomaly Scores**: Helps identify potential contamination cases.
- **Final Contamination Labels**: Useful for further validation.

9. Conclusion

Unsupervised learning techniques proved valuable in analyzing bacterial contamination patterns without the need for labeled data. K-Means clustering successfully segmented water samples into distinct contamination groups, making it an effective categorization tool. DBSCAN identified noise points and outliers, highlighting unusual contamination behaviors that might otherwise be overlooked. Agglomerative Hierarchical Clustering provided a structured contamination hierarchy, helping to reveal complex relationships between different water quality parameters. The Isolation Forest anomaly detection model excelled in identifying rare and extreme contamination cases, underscoring its effectiveness in flagging potential contamination risks. While these models offer significant insights, their performance depends heavily on selecting appropriate hyperparameters and preprocessing techniques. Future studies should explore hybrid approaches, integrating supervised and unsupervised models to refine contamination detection accuracy. Additionally, testing these models with real contamination data can further validate their applicability for large-scale water quality monitoring and early contamination detection.

List Of Tables :

List Of Figures :

References

1. Breiman, L. (2001). Random forests. *Machine Learning, 45*(1), 5-32. https://doi.org/10.1023/A:1010933404324
2. Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785-794. https://doi.org/10.1145/2939672.2939785
3. Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.
4. Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning, 1*(1), 81-106. https://doi.org/10.1007/BF00116251
5. Schölkopf, B., Platt, J. C., Shawe-Taylor, J., Smola, A. J., & Williamson, R. C. (2001). Estimating the support of a high-dimensional distribution. *Neural Computation, 13*(7), 1443-1471. https://doi.org/10.1162/089976601750264965
6. Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. *ACM Computing Surveys, 41*(3), 1-58. https://doi.org/10.1145/1541880.1541882
7. Ho, T. K. (1995). Random decision forests. *Proceedings of the 3rd International Conference on Document Analysis and Recognition*, 278-282.
8. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems, 25*, 1097-1105.
9. Liu, F. T., Ting, K. M., & Zhou, Z. H. (2008). Isolation forest. *2008 Eighth IEEE International Conference on Data Mining*, 413-422. https://doi.org/10.1109/ICDM.2008.17
10. Wu, J., & Wu, C. (2020). Water quality assessment based on machine learning models: A review. *Water, 12*(1), 543. https://doi.org/10.3390/w12020543