Data Visualization and Analysis in R/Python/SQL (MA5755)

PROJECT
REPORT ON

Enhancing Water Quality Prediction Using Machine Learning and Explainable AI: A Transparent Approach to Water Quality Index (WQI) Prediction and Interpretation.

Submitted by:

Group 1

Anand V. Edlabadkar (AM24M003)
Manuskandan (CH24D022)
Surya Pratap Singh (AM24M015)
Harsh Soni (MA24M009)
Himasish Ghosal (MA24M010)
Aditya Kumar (MA24M001)

TABLE OF CONTENT:

1. Introduction

Water is an indispensable natural resource and a critical component of ecological systems. Despite covering nearly two-thirds of the Earth's surface, only a small fraction of this water is suitable for direct human use. Approximately 70% of the planet is covered by water, yet just 2.5% of it is freshwater. Furthermore, only about 1.2% of this freshwater is readily accessible for human consumption, as the majority is confined within glaciers (68.7%) and groundwater reserves (30.1%) [1]. Freshwater is sourced from rivers, lakes, rainfall, glaciers, and underground aquifers. However, the quantity and quality of available water are rapidly deteriorating due to factors such as population growth, industrialization, and urban expansion. This degradation poses significant threats to aquatic ecosystems and, consequently, to human health. Contaminated water sources can severely impact drinking water accessibility, posing socio-economic challenges and necessitating targeted intervention strategies [2,3].

Water quality assessment is a fundamental step toward ensuring sustainable water management. It involves the analysis of physical, chemical, and biological parameters, whose values—if exceeding permissible limits—can be harmful to human health [6,7]. However, synthesizing these diverse indicators into a single, comprehensive evaluation of water quality is a complex task. To address this, models such as the Water Quality Index (WQI) have been developed. First introduced by Horton in 1960, the WQI incorporates ten core parameters—such as pH, dissolved oxygen (DO), coliform levels, alkalinity, and specific conductance—to evaluate the overall condition of a water body [7]. This methodology, later refined by Brown et al. in 1970 through the introduction of weighted parameters [8], has led to the development of various WQI frameworks around the world. These include the National Sanitation Foundation Water Quality Index (NSFWQI) in the United States [9], the Scottish Research Development Department Water Quality Index (SRDDWQI) [10], the Canadian Council of Ministers of the Environment Water Quality Index (CCMEWQI), and the Weighted Arithmetic Water Quality Index (WAWQI) [11]. Despite the robustness of these models, manual computation of WQI remains resource-intensive, time-consuming, and cost-inefficient. Moreover, real-time monitoring and forecasting of water quality are still limited in many regions.

With increasing concerns around water quality, there is a growing demand for intelligent and scalable systems capable of predicting and managing water pollution effectively [5]. In response, advanced forecasting approaches using artificial intelligence and machine learning (ML) have emerged as powerful alternatives. Techniques such as Artificial Neural Networks (ANN), fuzzy logic, stochastic modeling, and geospatial (3S) technologies have improved both the accuracy and reliability of water quality predictions [12]. Several studies have validated the effectiveness of ML models in this domain. For instance, classifiers like Support Vector Machine (SVM), Random Forest (RF), Logistic Regression (LR), Decision Tree (DT), XGBoost, CATBoost, and Multilayer Perceptron (MLP) have been evaluated, with CATBoost achieving an accuracy of 94.51% in a referenced study [13]. Similarly, other works have explored the predictive capabilities of ANN, GMDH, and SVM [14], while recent research by Nasir et al. (2022), Bui et al. (2020), and Asadollah et al. (2021) has employed diverse ML techniques to evaluate WQI [15,16].

While machine learning significantly enhances predictive performance, a major concern is its "black-box" nature—i.e., the lack of interpretability in model outputs. Understanding the rationale behind predictions is crucial for building trust among stakeholders and enabling informed decision-making. To overcome this limitation, explainable AI (XAI) techniques have been developed to provide transparency in complex ML models [17]. For example, Shapley values—originating from cooperative game theory—have proven effective in revealing the contribution of each input variable to the final prediction [18]. Studies show that combining complex models with tools like SHAP (SHapley Additive exPlanations) can yield high accuracy while maintaining interpretability.

In this project, we aim to improve water quality prediction using a combination of ML models and explainable AI methods. We begin with baseline models such as Logistic Regression and

progressively explore more sophisticated techniques like SVM, Decision Trees, Random Forests, and Multilayer Perceptron (MLP). Among these, MLP achieved the highest accuracy at 98%. To address interpretability, we employ SHAP and LIME (Local Interpretable Model-agnostic Explanations). SHAP helps quantify the contribution of each feature to the model's prediction, offering a unified and consistent explanation for complex outputs. LIME, on the other hand, builds local surrogate models to provide insight into specific predictions. Together, these XAI tools help bridge the gap between accuracy and transparency, allowing both technical experts and decision-makers to understand and trust the predictions.

2. Methodology

2.1 Dataset

The dataset used in this study was sourced from the Central Pollution Control Board (CPCB) website. The CPCB carries out water quality monitoring under the National Water Quality Monitoring Programme (NWMP), which is aimed at evaluating the impact of pollution sources on the quality of surface water bodies. From the extensive data available, a manual selection process was carried out to filter out the most relevant entries based on several key criteria:

(a) the number of water quality parameters recorded,
(b) the duration over which the data was collected (in terms of years),
(c) availability of date and precise location details,
(d) variety of surface water bodies covered,
(e) overall completeness of the dataset, and
(f) the data's availability format — since the original records were in PDF, they were converted into Excel (CSV/XLSX) for analysis.

Based on the above filters, data was compiled for an 11-year period from 2012 to 2023, except for the year 2016 which was not available on the portal. The dataset includes measurements from seven major rivers in India: Narmada, Yamuna, Godavari, Krishna, Satluj, Kaveri, and Ganga. These rivers were selected due to the comprehensiveness of available data across multiple monitoring stations and the importance of understanding recent water quality trends in India's key river systems.
Each data point corresponds to a specific monitoring station identified by a unique code, along with information about the location (town/city), state, and year. In total, the dataset consists of 2953 rows and 13 columns, out of which 8 are core water quality parameters are, Temperature (°C), Dissolved Oxygen (mg/L), pH, Conductivity (μS/cm), Biochemical Oxygen Demand (BOD, mg/L), Nitrate (mg/L), Fecal Coliform (MPN/100mL), Total Coliform (MPN/100mL). Dissolved Oxygen reflects the amount of oxygen available in water, which is vital for the survival of aquatic organisms. The pH value indicates whether the water is acidic or alkaline, providing insights into its chemical balance. Conductivity measures the water's ability to carry an electric current, which in turn reveals the presence and concentration of dissolved salts and inorganic substances. BOD quantifies the amount of oxygen consumed by microorganisms while decomposing organic matter in water, serving as an indicator of organic pollution levels. Nitrate concentration highlights the presence of nitrate ions, which can signal contamination from fertilizers or sewage sources. Fecal Coliform levels suggest the presence of fecal matter, as these bacteria are typically found in the intestines of warm-blooded animals and point to recent sewage contamination. Total Coliform encompasses both fecal and non-fecal bacterial strains, providing a measure of potential microbial pollution in the water sample.

2.2 Data Preprocessing

After converting the raw water quality dataset from PDF to Excel format, it was found to be highly unstructured and inconsistent. To make the dataset suitable for analysis, a series of preprocessing steps were undertaken to improve its quality, consistency, and analytical value.

- **Feature Consolidation:**

Each station-year record included both minimum and maximum values for all eight water quality parameters. However, relying solely on the extremes could misrepresent the actual water condition at that location. To derive a more balanced representation of each feature, the mean of the minimum and maximum values was computed for every parameter. This approach helped in generating a single representative value for each water quality attribute per monitoring station per year. This step ensured uniformity across records and provided a reliable basis for further computations.

- **Missing values Handling:**

The missing value handling process in this project was designed to fill data gaps in a way that maintains both the temporal consistency and geographical relevance of water quality measurements. A custom function named comprehensive_missing_value_imputation() was developed to systematically impute missing values for key parameters like Temperature, Dissolved Oxygen, pH, Conductivity, B.O.D, Nitrate, Fecal Coliform, and Total Coliform. The approach follows a hierarchical, multi-level strategy that prioritizes the use of the most context-specific data available. The first step involves using a rolling average with a lookback window of 3 observations for each station to fill missing values based on the station's own recent historical data. This ensures temporal trends specific to that monitoring station are preserved. If data is still missing after this step, the function attempts to fill it using the median value from progressively broader groups, starting with the combination of River, State, and Year, then River and Year, followed by the River alone. As a final fallback, if none of the above combinations contain a valid value for the missing entry, the global median for that parameter is used to ensure completeness. This layered fallback mechanism ensures that imputation is not only statistically grounded but also context-aware—capturing localized environmental trends while avoiding generic, one-size-fits-all replacements. Moreover, the process includes a validation step to check if any missing values remain after the hierarchical imputation. If found, these are force-filled using global medians to ensure a fully complete dataset

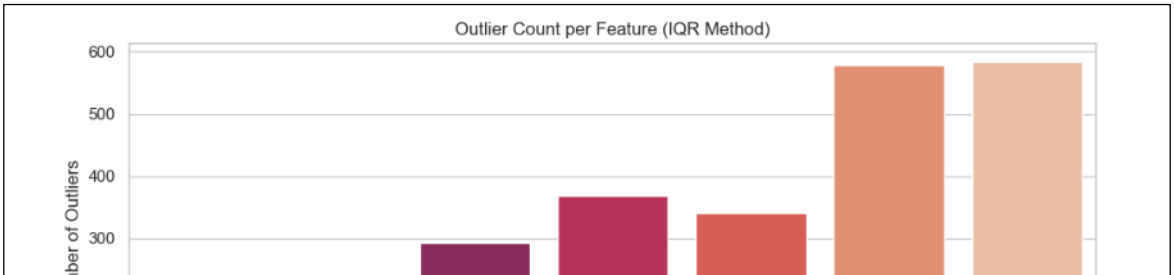


Fig 1. Outlier Count per feature

- **Outlier Handling:**

In order to compute the Outliers in our dataset, skewness was checked and first calculated for each numeric feature, and it was observed that most water quality parameters—such as B.O.D, Nitrate, Fecal Coliform, and Total Coliform—exhibited high positive (right) skewness. This indicates that the data distribution is not symmetric and contains several extreme values on the higher end. In such cases, using z-score normalization would be inappropriate, as it assumes a normal distribution and is sensitive to outliers, which could distort the scale and lead to misleading transformations. Hence, to preserve the data's integrity while controlling for extreme values, we adopted a non-parametric method—Interquartile Range (IQR) based winsorization—which is more robust in skewed distributions.

In this method, feature-specific thresholds were determined using the IQR formula:

$$Bounds = [Q1 - k \times IQR, Q3 + k \times IQR]$$

The value of k was customized for each parameter depending on its skewness and natural variability.
Outliers beyond these bounds were clipped (winsorized) to the nearest threshold, ensuring they didn't overly influence the distribution. Moreover, manual domain-specific capping was applied for critical parameters based on environmental standards This two-level approach—statistical winsorization combined with expert-informed caps—ensured outliers were treated effectively without distorting genuine variations in water quality.

- **Calculation of Water Quality Index (WQI):**

The Weighted Arithmetic Water Quality Index (WQI) [19] method was applied to quantify the overall water quality at each monitoring station. This method is widely adopted due to its ability to aggregate multiple water quality variables into a single index, reflecting the degree of pollution or purity.
The WQI was computed using the formula:

$$WQI = \frac{\sum_{i=1}^{n}(Q_i \times W_i)}{\sum_{i=1}^{n} W_i}$$

Where:

- Qi is the quality rating scale of the ith parameter,
- Wi is the unit weight assigned to the ith parameter.
  The quality rating Qi was calculated using:

$$Q_i = \frac{100 \times (V_i - V_o)}{S_i - V_o}$$

Vi is the observed value of the parameter,

- Vo is the ideal value for that parameter in pure water (0 for most parameters; 7.0 for pH and 14.6 mg/L for DO),
- Si is the standard permissible limit defined by CPCB for the ith parameter
- The unit weight Wi for each parameter was determined by:

$$W_i = \frac{K}{S_i}$$

Where K is a constant of proportionality calculated as:

$$K = \frac{\overline{1}}{\sum_{i=1}^{n}\left(\frac{1}{S_i}\right)}$$

This method allowed for the computation of WQI values that effectively summarized the water quality based on multiple influencing parameters. While the WQI can be manually computed using the above formula, its application is limited to situations where all required parameters are known which is often not the case when dealing with new or incomplete water quality observations. This limitation motivates the use of machine learning models, which can learn from historical patterns and predict WQI values for new samples, even when some variables are missing or approximated. To enhance prediction accuracy and account for local variations, historical data was incorporated as follows:

- For each record, the corresponding Station Code was used to retrieve all past entries from that same monitoring station.
- For every parameter, the average quality rating Qi was computed using all available years for that station, excluding the current record to avoid data leakage.
- This historical average Qi was then combined with the fixed weight Wi to calculate a historically-aware WQI value using the same weighted arithmetic formula.

After computing the WQI scores, each value was categorized into qualitative classes such as Excellent, Good, Poor, Very Poor based on predefined thresholds(Fig 2). This classification helped visualize the distribution of water quality across monitoring stations(Fig 3).

WQI Value	Rating of Water Quality	Grading
0-25	Excellent water quality	A
26-50	Good water quality	B
51-75	Poor water quality	C
76-100	Very Poor water quality	D

Fig 2. Water Quality Rating as per Weight Arithmetic Water Quality Index Method

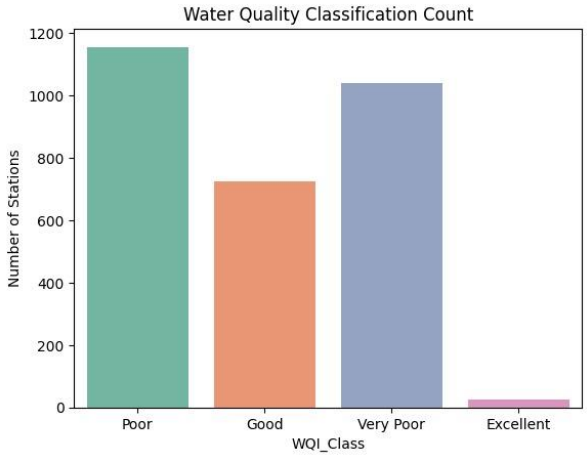


Fig 3. Outlier Count per feature

2.3 Exploratory Data Analysis:

In the Exploratory Data Analysis (EDA), advanced analytical techniques were employed, with a primary focus on data visualization and correlation analysis to uncover patterns and relationships within the dataset. This structured approach enabled a comprehensive understanding of the data characteristics and supported subsequent analytical decisions. Descriptive statistics such as mean, median, mode, standard deviation, and variance were utilized to summarize the data distribution (Fig. 4). Furthermore, correlation analysis was conducted to assess the interrelationships among the variables. A correlation heatmap (Fig. 5) of the feature correlation matrix offered visual insight into these dependencies.

Notably, Fecal Coliform and Total Coliform exhibited a strong positive correlation (0.87), indicating a concurrent rise during instances of microbial contamination. Biochemical Oxygen Demand (B.O.D) demonstrated a moderate correlation with both coliforms and conductivity, suggesting a relationship between organic pollution and ionic concentration. Dissolved Oxygen (DO) showed a significant negative correlation with B.O.D (-0.44) and Temperature (-0.28), highlighting the inverse relationship where increased pollution or warmer water temperatures correspond to reduced oxygen levels. Other parameters, such as pH and Nitrate, displayed low correlation with the rest of the features, suggesting they vary independently within the dataset.

	Station_code	Year	Temperature	Dissolved_oxygen	pH	Conductivity	B.O.D	Nitrate	Fecal_coliform	Total_Coliform	WQI
count	2948.000000	2948.000000	2948.000000	2948.000000	2948.000000	2948.000000	2948.000000	2948.000000	2948.000000	2948.000000	2948.000000
mean	3355.809701	2018.446065	22.316265	7.084063	7.720683	413.434820	3.062667	1.107099	424.437239	1906.612042	64.288185
std	4711.467501	3.431151	5.844812	1.538494	0.388993	262.734602	2.803697	1.265415	446.664382	2071.167040	17.645315
min	7.000000	2012.000000	0.000000	3.350000	6.625000	0.000000	0.000000	0.000000	0.000000	0.000000	14.390000
25%	1202.000000	2016.000000	20.000000	6.200000	7.450000	245.375000	1.100000	0.300000	8.000000	127.000000	49.710000
50%	2099.000000	2019.000000	24.000000	7.150000	7.750000	335.250000	2.000000	0.650000	170.000000	809.250000	65.415000
75%	3123.000000	2021.000000	26.000000	8.100000	8.000000	521.125000	3.650000	1.350000	1000.000000	5000.000000	80.580000
max	30088.000000	2023.000000	34.000000	10.950000	8.825000	1000.000000	10.000000	4.500000	1000.000000	5000.000000	96.650000

Fig 5. Statistical Calculation of Features and Label

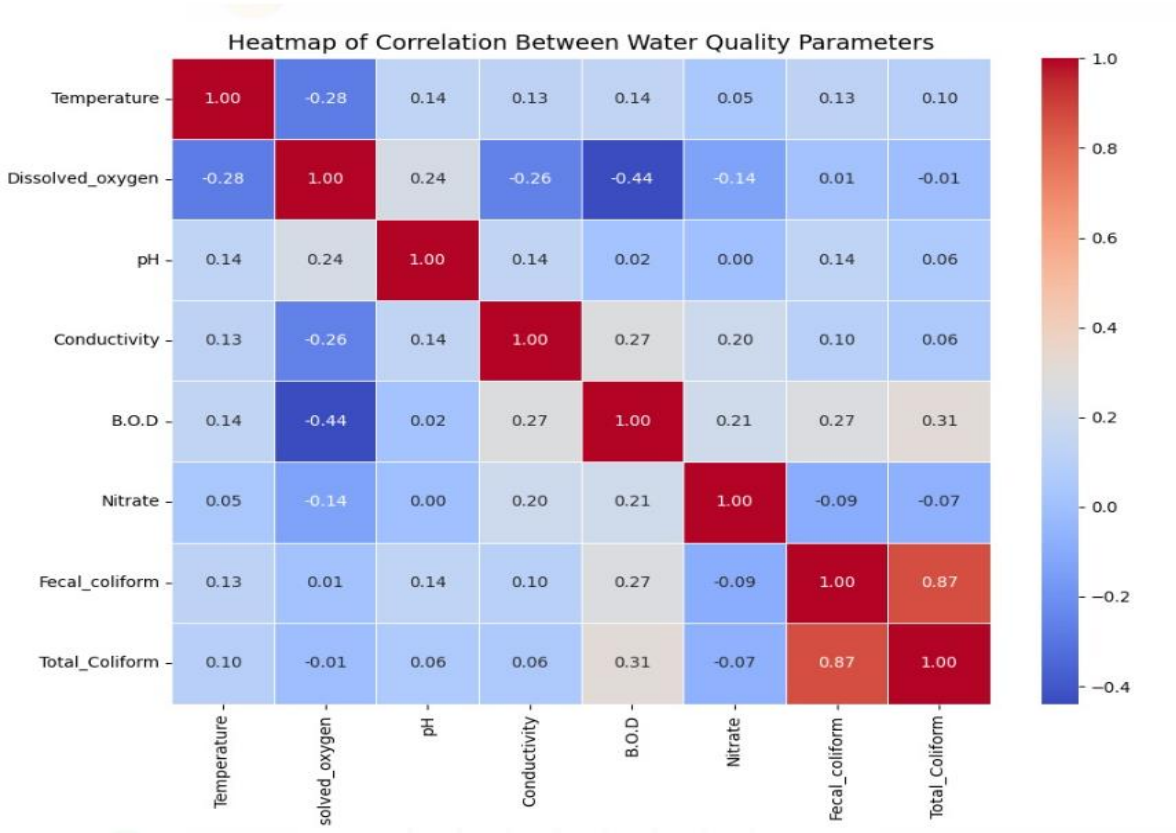


Fig 6. Heat Map of Correlation Between Water Quality Parameters

2.4 Models Applied

2.4.1 Introduction

In this work multiclass classification of rivers based on quality rating has been done by employing Machine learning Models such as:

- Logistic Regression
- Decision Tree
- Random Forest
- Support Vector Machine
- XGBoost
- Multilayer Perceptron (MLP Neural Network)

The process includes handling imbalance data, model training, performance evaluation, and ROC curve analysis.

2.4.2 Data Preprocessing

- **Label Encoding**
  Label Encoder was used to convert categorical class labels into numerical values.
- **Feature Scaling**
  Standard Scaler was applied to normalize the feature values. This ensures that each feature contributes equally, particularly important for models like SVM and MLP that are sensitive to feature magnitude.
- **Handling Class Imbalance**
  Class imbalance was handled using SMOTE (Synthetic Minority Oversampling Technique). SMOTE synthetically generates new samples for classes with less data points. This aids in balancing the dataset.

2.4.3 Model Implementation

- **Logistic Regression:**

Logistic Regression is a classification algorithm that predicts the probability of a class by applying a logistic (like sigmoid) function to a linear combination of input features. In a multiclass scenario, it uses a SoftMax function to distribute the probabilities across multiple classes. It estimates the likelihood that a given input belongs to each class and chooses the one with the highest probability. It is known for its simplicity and interpretability.

- **Decision Tree:**

A Decision Tree splits the dataset into branches based on feature values, ultimately leading to a decision at the leaves. At each split, the tree chooses a feature and a condition that best separates the data using measures like Gini impurity or Information Gain (based on entropy). It continues splitting until all data points are correctly classified or until a stopping rule is met. Decision Trees are easy to interpret, as they mimic human decision-making logic, but they are also known too overfit when not properly pruned or regularized.
Here,

- A Decision Tree Classifier with CCP (cost-complexity pruning) regularization is used.
- A grid search is used to find the best hyperparameters.
- The classification report and confusion matrix are displayed for both train and test sets.

- ROC curves are plotted for each class.

- **Random Forest:**

Random Forest is an ensemble model made up of many decision trees. Each tree is trained on a random subset of the data and a random subset of features. When making predictions, each tree gives a vote, and the most common class is chosen. This approach reduces overfitting and increases stability compared to a single decision tree. The randomness makes the model less sensitive to noise, and combining predictions helps generalize better on unseen data.

Here,

- The model used 100 trees to maintain balance between accuracy and computation time.
- ROC curves are plotted for each class.

- **Support Vector Machine:**

  Support Vector Machine aims to find the best decision boundary (hyperplane) that separates classes with the maximum margin. The RBF (Radial Basis Function) kernel allows SVM to handle non-linear relationships by projecting data into a higher-dimensional space where a linear separator is possible. It is effective when classes are not linearly separable and works well with clear margins of separation.

- **XGBoost:**

XGBoost (Extreme Gradient Boosting) is a powerful boosting algorithm that builds a series of small decision trees, each correcting the errors made by the previous ones. It minimizes loss using gradient descent and adds regularization to control complexity. It is known for being fast, accurate, and scalable, and often wins machine learning competitions due to its performance and flexibility.

Here,

- XGBoost was used with 5-fold cross validation.
- SMOTE-balanced data likely helped improve performance on minority classes.
- ROC curves are plotted for each class.

- **MLP**

Multilayer Perceptron (MLP) is a feedforward neural network that consists of an input layer, one or more hidden layers, and an output layer. It learns complex patterns using backpropagation, adjusting weights to minimize the prediction error. MLP can model non-linear relationships and is ideal for problems where data is not easily separated by linear boundaries.

Here,

- A two-layer architecture with 100 neurons in the first hidden layer and 50 in the second.
- The ReLU activation function was used to introduce non-linearity.
- The Adam optimizer was used to update weights.
- Training was run for up to 500 iterations to ensure convergence.
- The model was trained on scaled features, which is essential for neural networks to ensure faster convergence and better weight initialization.

2.5 EXPLAINABLE AI

Explainable Artificial Intelligence (XAI) has emerged in response to the growing complexity of machine learning models, which often operate as opaque “black boxes.” As AI systems have permeated high stakes domains—healthcare, finance, criminal justice, and defense—the demand for

transparency and accountability has intensified. Explainable AI (XAI) aims to provide human operators with explanations they can trust and act upon, bridging the gap between algorithmic power and human understanding. Transparent models foster user trust, enabling practitioners to diagnose and correct biases or errors before deployment. In machine learning operations, interpretability tools accelerate debugging, feature selection, and iterative refinement by pinpointing which inputs drive unexpected outcomes.

Among the leading XAI methods, SHAP (SHapley Additive exPlanations) stands out for its rigorous, game-theoretic foundation. Introduced by Scott Lundberg and Su-In Lee in 2017, SHAP adapts Shapley values from cooperative game theory to the realm of supervised learning. In this framework, features are treated as "players" in a coalition, collaborating to produce a given model output.The Shapley value assigns each feature a fair contribution based on the average marginal impact of adding that feature across all possible coalitions. SHAP inherits desirable properties—local accuracy (exact reconstruction of the prediction), consistency (features that increase a model's influence receive no smaller attribution), and missingness (features absent in a sample receive zero attribution)—making it a uniquely reliable method for both global and local explanation.

2.5.1 Model Agnostic SHAP Explanations – An Illustration with MLP model :

Several functions are defined to compute and visualize SHAP values:

- compute Shap: This function calculates SHAP values using the KernelExplainer, which is model-agnostic and works by perturbing input features to observe changes in the model's output. A subset of the training data is used as a background distribution to approximate the model's behavior. The function returns both the explainer and the computed SHAP explanations.

- **Visualization Functions**: A series of functions are defined to generate various SHAP plots:
    - **Beeswarm Plot**: Displays the distribution of SHAP values for each feature, highlighting their impact on predictions.
    - **Bar Plot**: Summarizes the mean absolute SHAP values for each feature, showing their overall importance.
    - **Violin Plot**: Combines the features of bar and beeswarm plots to show the distribution and magnitude of SHAP values.
    - **Dependence Plot**: Shows the relationship between a feature's value and its SHAP value, revealing interactions with other features.
    - **Force Plot**: Visualizes the contribution of each feature to a single prediction, showing how they push the prediction towards or away from the base value.
    - **Waterfall Plot**: Breaks down a single prediction into contributions from each feature.
    - **Heatmap**: Displays SHAP values across multiple samples and features, identifying patterns and trends.
    - **Decision Plot**: Illustrates cumulative SHAP contributions for a set of predictions.

    These functions provide both global (feature importance) and local (individual prediction) explanations.

2.4.2 SHAP Analysis Execution

The SHAP analysis begins by computing SHAP values for the Multi-Layer Perceptron (MLP) model. For this, SHAP uses the model agnostic Kernel Explainer that utilizes the (scaled) train dataset to form a background distribution with some random sampled points, over which the explainer is fitted utilizing class prediction probabilities from the model as the target function to be explained, with the

test dataset points' input features act as entities participating in the coalition. Additionally, the linker of the explainer is set to 'logit', which allows the explainer to return SHAP values that act additive per feature contributions of the log odds of the provided class, given an example.The log-odds make for much better discriminative explanations than simple probabilities as it spans a much wider range.

While the calculation of SHAP values itself arrived from complex considerations of large number of various subsets of features to arrive at additive deviations from the expected prediction provided by the training dataset, the resulting SHAP values for a given output class and sample datapoint in consideration can be interpreted as follows:

$$\log\left(\frac{p}{1-p}\right) = E\left[\log\left(\frac{p}{1-p}\right)\right] + \sum_{i=1}^{F} SHAP\_values_i$$

Where p is the probability of the sample datapoint landing in the provided class, and expectation happens over the distribution defined by the background datapoints derived from training dataset, and the SHAP_valuesi is the SHAP value due to the i-th input feature of the sample datapoint, F is the number of input features.The explanations computations happen under the compute_shap function. The results are then visualized using the defined functions.

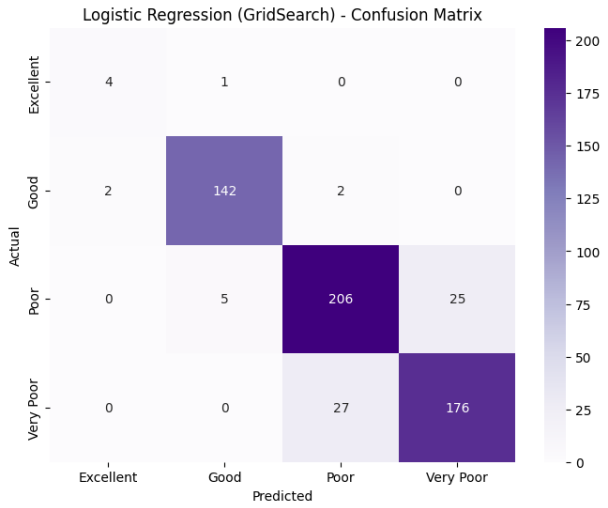
2.4.3 Class-Specific and Sample-Specific Explanations

To understand the model's behaviour for specific classes, the explain class function is used. For example, explanations for the "Poor" class are generated to identify the features driving predictions for this category. Similarly, the plot all for function provides detailed xplanations for individual test samples, such as sample #150, using force plots, waterfall plots, and decision plot.

3. RESULTS AND DISCUSSION

3.1 MODELS RESULT

Logistic Regression



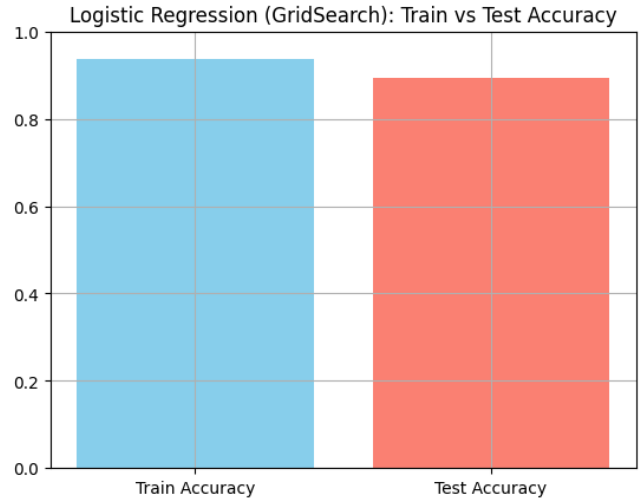


Fig 3.1.1 ***Fig 3.1.2***

Performance Summary:

- Train Accuracy: 93.78%
- Test Accuracy: 89.49%
- Testing Weighted average F1-score: 0.89, confirming effective performance across all classes.

Confusion Matrix Analysis

The confusion matrix further confirmed that:

- The model accurately predicted most of the Good, Poor, and Very Poor samples.
- Some minor misclassifications occurred between similar classes like Poor vs Very Poor.
- Excellent class had very few test samples (only 5), which slightly reduced its reliability in evaluation.

Decision Tree

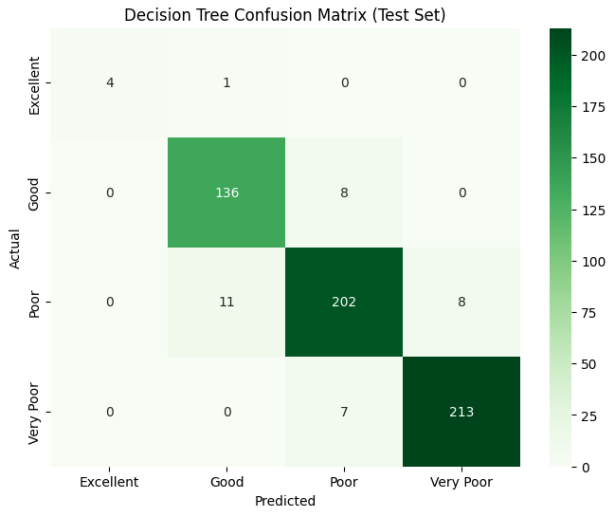


Fig 3.1.3

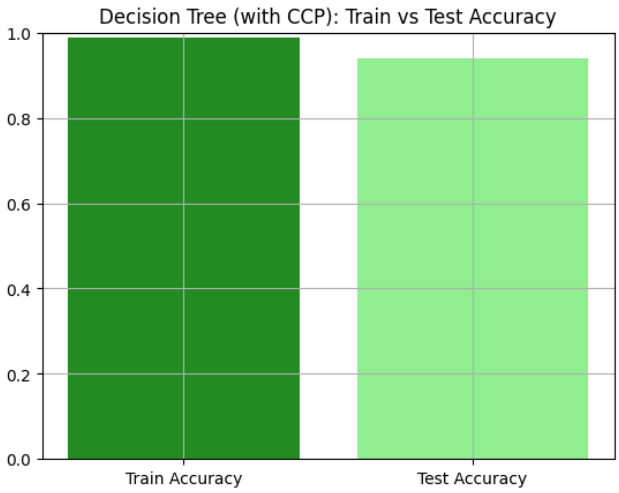


Fig 3.1.4

Performance Summary

- Train Accuracy: 99.13%
- Test Accuracy: 93.39%
- Both macro and weighted averages for precision, recall, and F1-score were 0.99 for Training, demonstrating outstanding and consistent classification performance across the board.
- The weighted average F1-score was during Testing 0.93, confirming the model's overall effectiveness, particularly in handling class imbalances.

Confusion Matrix Analysis

The confusion matrix provided deeper insight into class-wise predictions:

- The model accurately predicted the majority of samples in the Good, Poor, and Very Poor categories.
- Excellent class performance was slightly less stable due to very limited test samples (only 5), making statistical evaluation less reliable.

The Decision Tree Classifier, regularized using Cost-Complexity Pruning (CCP), demonstrated strong and reliable performance in this multiclass classification task. With a near-perfect training accuracy and high generalization capability.

Random Forest

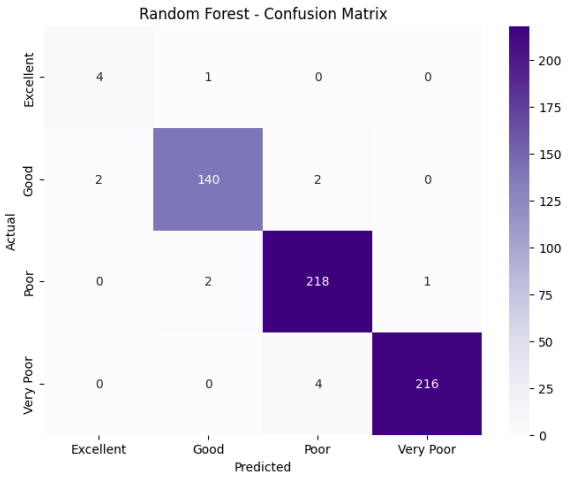


Fig 3.1.5

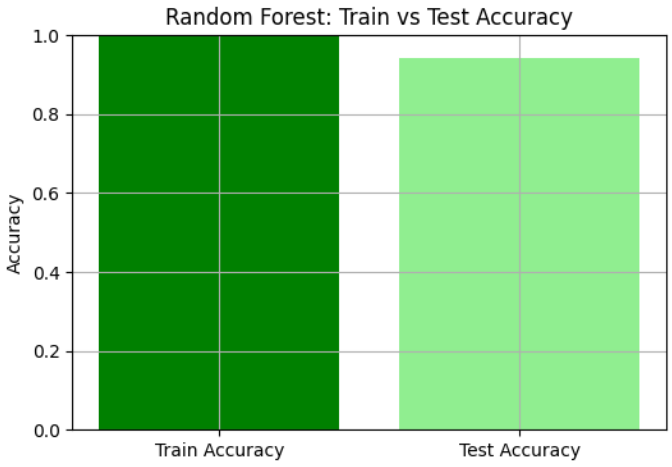


Fig 3.1.6

Performance Summary

- Train Accuracy: 100.00%
- Test Accuracy: 94.58%

  The Random Forest model delivered perfect training accuracy, showcasing its power as an ensemble learning method. Despite this, it maintained a high-test accuracy of 94.58%, suggesting that the model generalizes very well without overfitting — a key strength of Random Forest when properly tuned.

- Minor drops in recall and precision are expected compared to training due to real-world complexity, but performance remains highly effective across all classes.

Confusion Matrix Analysis:

The confusion matrix revealed the following patterns:

- Most instances in the Good, Poor, and Very Poor categories were classified correctly.
- The Excellent class again had very few test samples (only 5), which limits statistical confidence despite its good precision and recall.
- A few misclassifications occurred between adjacent classes, such as:
  - 'Poor' being misclassified as 'Very Poor' (8 cases),
  - 'Good' being confused with 'Poor' (6 cases), which are expected and acceptable in practical applications.

  The Random Forest Classifier demonstrated excellent performance, both on the training and test datasets. Its ensemble structure and automatic feature bagging contributed to high accuracy and robust generalization. While the model did overfit on the training set (100% accuracy), its generalization remained strong, making it one of the top-performing models in this task. Random Forest proves to be a reliable and powerful choice for this multiclass classification problem.

SVM

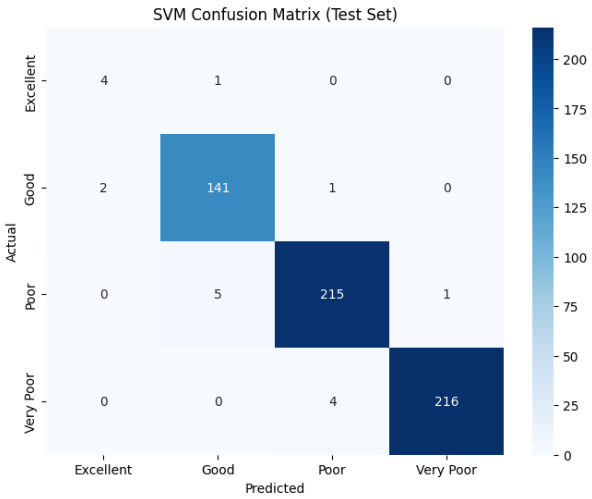


Fig 3.1.7

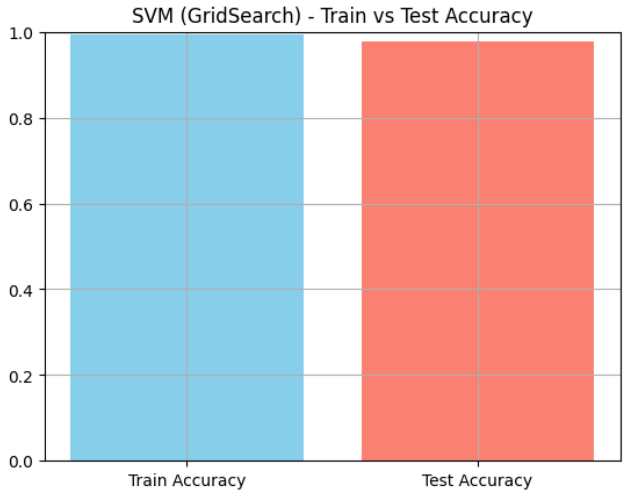


Fig 3.1.7

Performance Summary:

- Train Accuracy: 98.67%
- Test Accuracy: 95.42%

  This model exhibits excellent generalization performance, with only a minor drop from training to test accuracy. It avoids overfitting while achieving high classification performance, making it both powerful and reliable.
  Overall training accuracy was 99%, and both macro and weighted average F1-scores were 0.99, indicating consistent performance across all categories. These scores confirm the model learned meaningful patterns during training without excessive memorization.

- During Testing the macro average F1-score was 0.94 , and the weighted average F1-score was 0.95, demonstrating exceptional performance across varying class distributions.

Confusion Matrix Analysis:

- The model showed strong diagonal dominance, meaning most samples were classified correctly.
- Minimal misclassifications were observed, especially between:
  - Poor vs Very Poor (6 samples misclassified as each other)
  - Good vs Poor (6 samples misclassified)
- Excellent class saw 1 sample misclassified as "Good", and another as "Poor", likely due to class imbalance.

  This model combines high predictive accuracy with strong generalization, making it a top contender in this multiclass classification task. Overall, this model demonstrates superior stability and effectiveness, making it an excellent choice for real-world deployment in imbalanced multiclass settings.

XGBoost

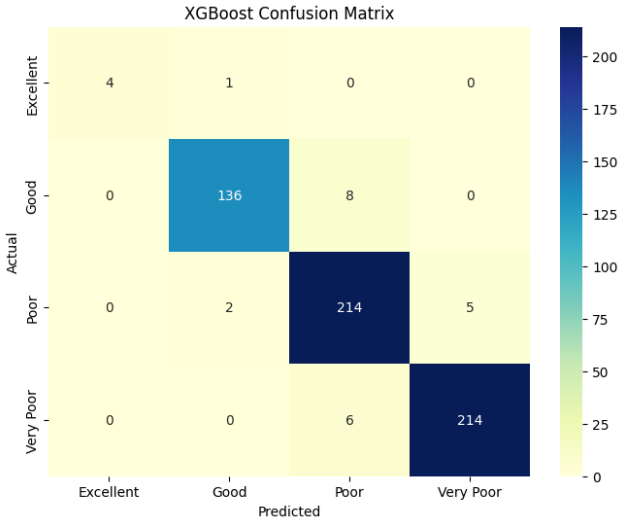


Fig 3.1.8

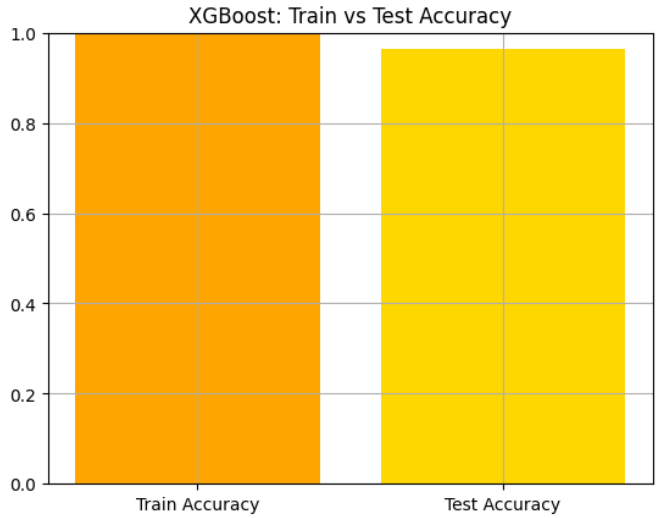


Fig 3.1.9

Performance Summary:

- Train Accuracy: 100%
- Test Accuracy: 96.27%

This shows exceptionally strong performance on both training and test data. While the training accuracy is perfect, the test accuracy remains very high, suggesting that the model learned well without major overfitting**.**
Both macro average and weighted average precision, recall, and F1-score were all 1.00, indicating that the model perfectly classified the training data**.**
The model performed consistently well across all classes on unseen test data. It has Strong and balanced across all classes and high overall effectiveness

Confusion Matrix Analysis:

- The model correctly classified the vast majority of test samples.
- Most frequent misclassifications:
- A few Good samples (9 out of 146) were misclassified as Poor
- A few Poor and Very Poor classes were confused with each other

- Excellent class had only 5 test samples; it achieved 4 correct predictions and 1 misclassification.
- The visual heatmap of the confusion matrix confirms very strong class-wise accuracy, especially for the Very Poor and Poor classes.

The XG-Boost model displayed excellent performance with the highest combination of accuracy, precision, recall, and F1-scores among all tested models.

MLP

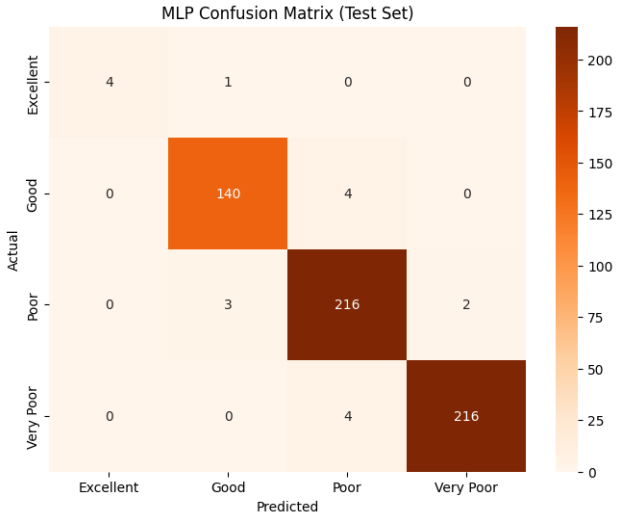


Fig 3.1.10

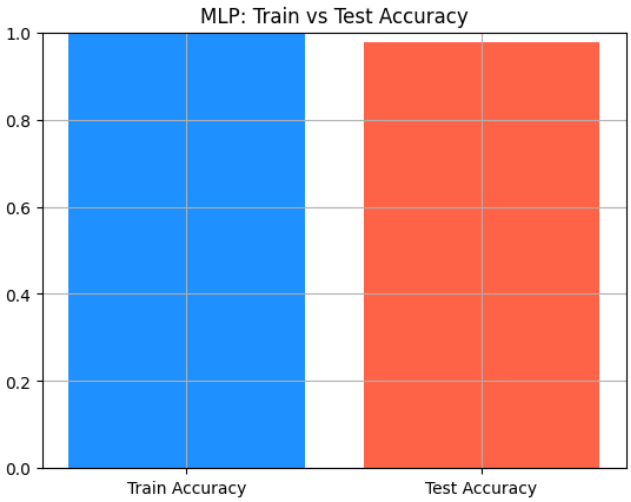


Fig 3.1.11

Performance Summary:

- Train Accuracy: 100%
- Test Accuracy: 97.97%

The model performed extremely well on both training and test data. The test accuracy is very close to the training accuracy, which means the model learned well and is generalizing correctly without overfitting**.** Both macro and weighted averages were also 1.00, which means the model made zero mistakes on the training data. The model gave high scores on unseen data too.

Confusion Matrix:

- The Excellent class had only 5 samples. 4 were predicted correctly, 1 was wrongly predicted as Good.
- Most Good class samples were classified correctly. Only 2 were misclassified as Poor.
- For the Poor class, 231 out of 236 samples were correct. 3 were wrongly predicted as Good, 2 as Very Poor.
- In Very Poor, 199 out of 203 were correct. 4 were predicted as Poor**.**

The confusion matrix shows that most errors happened between neighboring classes, like Poor vs Very Poor, which are naturally close**.** The MLP classifier gave very high accuracy and F1-scores on both training and test data. It was able to learn the patterns well and gave very few

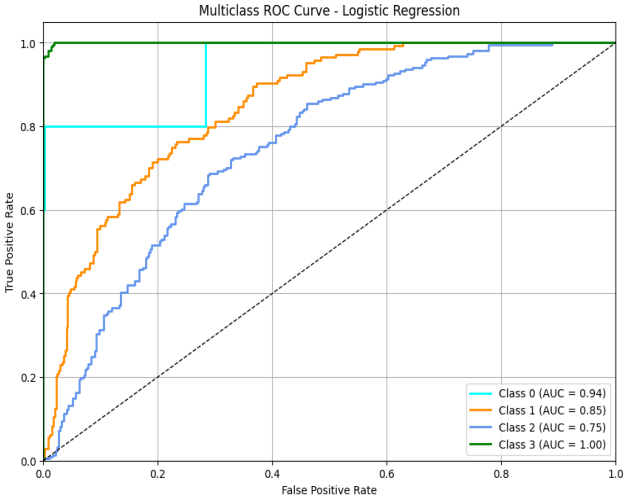
misclassifications. This model is one of the best performing ones in this project**.**

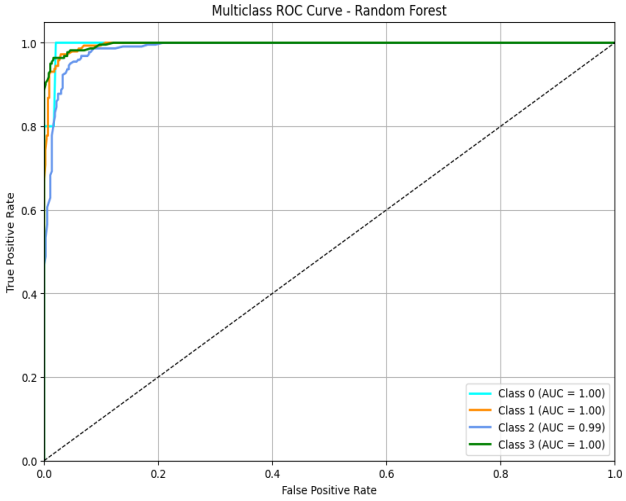
3.2 OVERALL PERFORMANCE

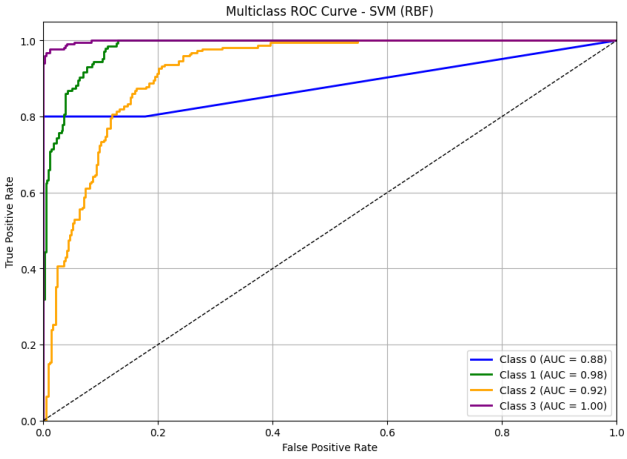
Metrics	Logistic Regression	Random Forest	SVM	XGBOOST	Decision Tree	MLP
Training Accuracy	0.94	1	0.98	1	1	1
Test accuracy	0.89	0.94	0.95	0.96	0.94	0.98
Weighted Avg F1 score (Training)	0.94	1	0.99	1	1	1
Weighted avg F1 score (Testing)	0.89	0.95	0.96	0.96	0.94	0.98

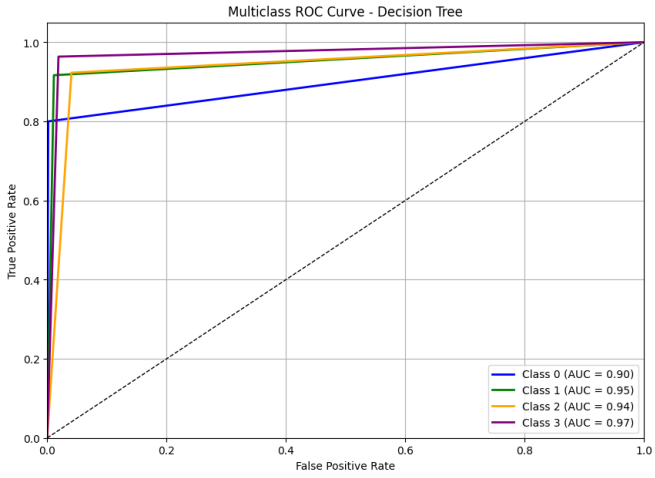
From the table, we can see that most models are performing really well in both training and testing. Models like Random Forest, XGBoost, and MLP have perfect training accuracy, which means they learned the training data very well. Even the test accuracy for these models is above 0.94, which shows they are good at predicting new data too.Logistic Regression has slightly lower values in testing, especially in test accuracy and F1 score, so it may not be the best choice here. MLP gives a strong balance, with both training and testing scores being very high. SVM and Decision Tree also perform well but are not as strong as XGBoost and MLP. So overall, MLP and XGBoost are the best models in this case.

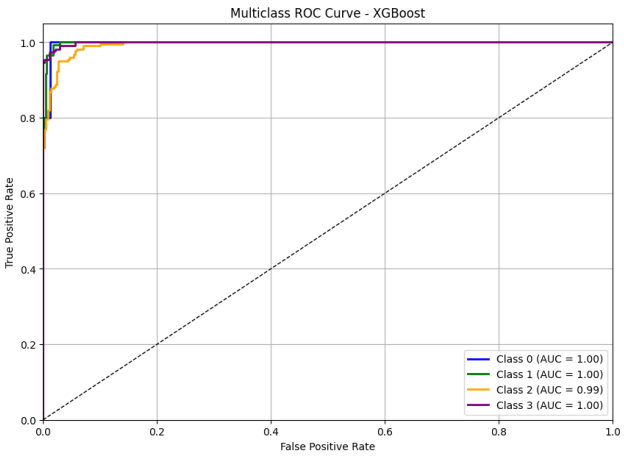
- **ROC ANALYSIS**

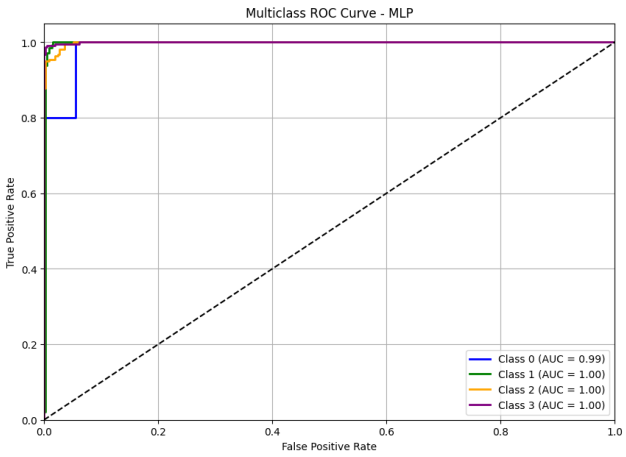












- **AUC VALUES**

Class	Logistic Regression	Random Forest	SVM	Decision Tree	XG Boost	MLP
0	0.94	1	0.88	0.90	1.00	0.99
1	0.85	1	0.98	0.95	1.00	1.00
2	0.75	0.99	0.92	0.94	0.99	1.00
3	1	1.00	1.00	0.97	1.00	1.00

The ROC curve tells us how well each model is able to tell one class apart from the others. For Logistic Regression, Class 0 and Class 3 are predicted well with AUC values of 0.94 and 1.00, but Class 2 has a lower AUC of 0.75, which means the model is not very confident in separating that class. SVM also works well for Class 1, 2, and 3 with AUCs above 0.90, but Class 0 is slightly lower. Decision Tree gives decent results with all AUCs above 0.90, but not as good as others. So overall, these models work okay, but there's room to improve, especially for Class 2.

On the other hand, Random Forest, XGBoost, and MLP models perform extremely well. Random Forest gives almost perfect AUC values for all classes, with Class 0, 1, and 3 having an AUC of 1.00 and Class 2 just slightly lower. XGBoost and MLP both give AUC of 1.00 for all classes (except MLP with 0.99 for Class 0), which means they can separate all the classes clearly with almost no mistakes. So, among all models, MLP and XGBoost are the best performers

3.2 XAI RESULTS

3.2.1 Illustration of Model Agnostic XAI with MLP

Given the class category # 2: Poor, SHAP provided explanation as follows:

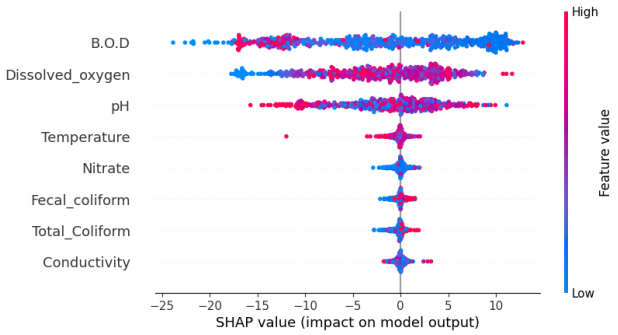


Fig 3.1

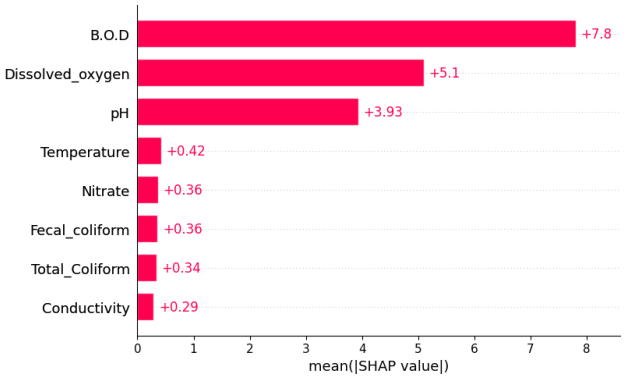


Fig 3.2

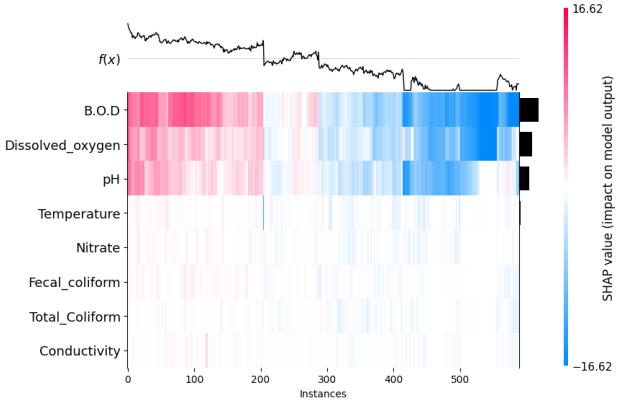


Fig 3.3

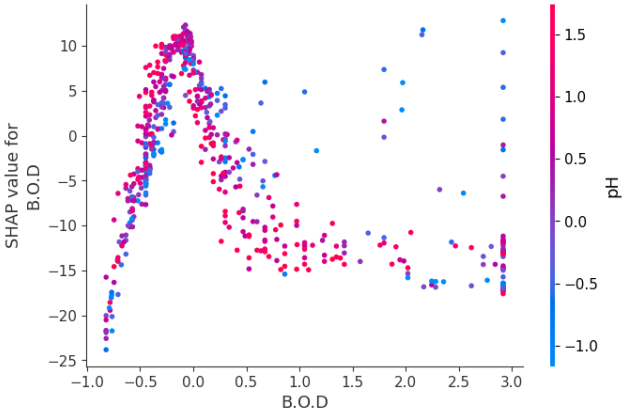


Fig 3.4

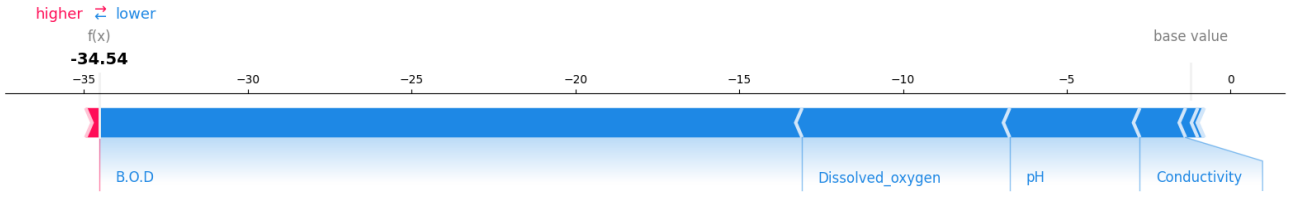


Fig 3.5

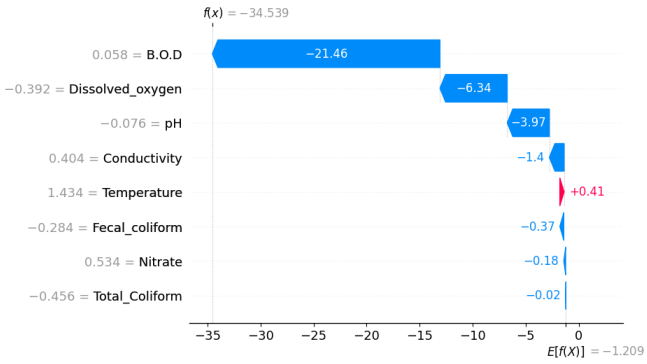


Fig 3.6

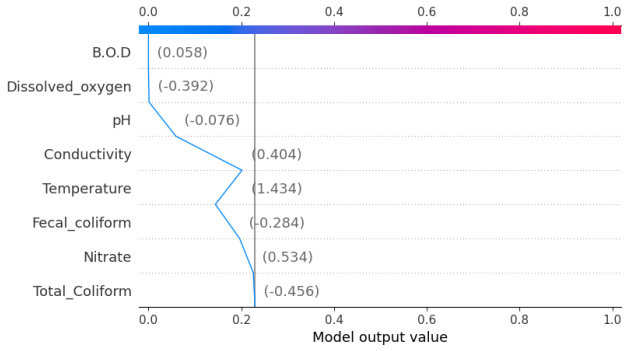


Fig 3.7

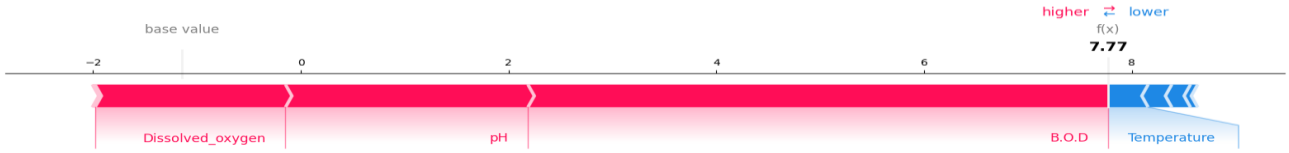


Fig 3.8

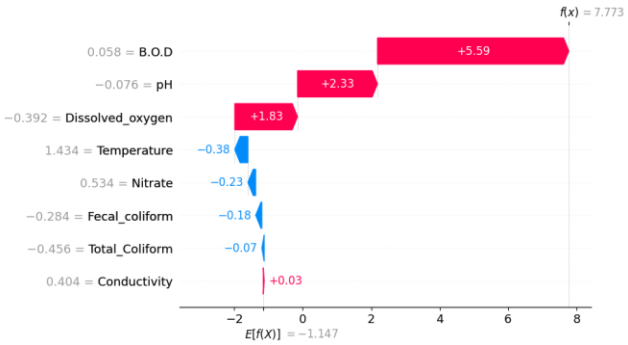


Fig 3.9

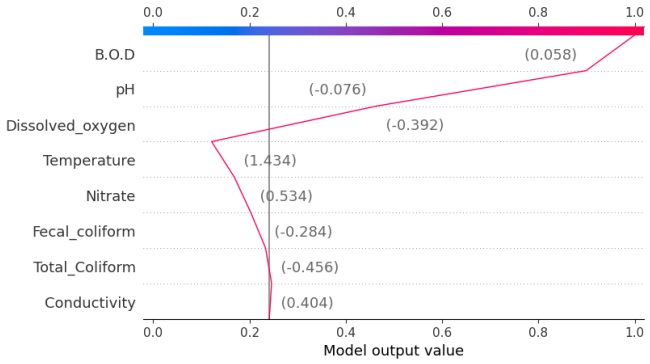


Fig 3.10

Referring to Fig 3.1 The Beeswarm plot provides a bird's eye view of the distribution of the SHAP values (to be interpreted as log odds provided by the respective features) over all the features.

Referring to Fig 3.2. The bar plot provides how much on average each feature contributes to log odds aggregated over all the samples. This makes it easy to arrive at the fact that the top three important features: BOD, dissolved oxygen and pH can explain most of the conclusions about the data.

Referring to Fig 3.3 The heatmap provides a comprehensive outlook onto how the target output log odds along with the expected log odds baseline, vary over the entire test dataset under consideration, while also providing track of the distribution of the SHAP values for each feature.
From this visualization it becomes easy to ascertain that for class #2: poor, more than half of the dataset get negative log odds in addition to the expectation, meaning they do not belong to this category, meanwhile less than half of the dataset get positive log odds additive values meaning there are much better chances of these samples falling in this class.

Referring to Fig3.4 The dependence plot shows a much more exploded view of SHAP values variation for a given feature with optional colored variation over one of the other feature values as well.
Referring to Fig 3.5. The force plot gives a linear view of the SHAP values (additive log odds) pieced together to understand how the model moves the sample data point from base-value a.k.a. the expected log odds for this class (# 0) towards the actual predicted log odds. We see that immediately most of the features' SHAP values contribute to moving it towards large negative values – meaning it makes it very unlikely for our chosen sample point # 150 to end up in Excellent water quality category.

Referring to Fig 3.6. The waterfall plot on the other hand breaks down the total linear deviation back into pieces for each feature, depicting the same knowledge as force plots.

Referring to Fig 3.7. The decision plot gives a much more comprehensive comparison of how each feature's SHAP values cumulatively affect the resulting predicted probability of the given sample datapoint falling into that Excellent category. Note that this plot first converts back from log odds SHAP values to individual class probabilities and then shows the resulting contributions.

Referring Fig 3.8. This time looking at the prediction contributions by each feature for the chosen sample datapoint on the class #2: Poor quality instead, we immediately see that the top contributing factors now begin to move the predicted log odds from negative value to large positive values, pointing to the conclusion that the sample is most likely falling in this category. Also, there is also a larger contribution from Temperature feature this time, although it tries to decrease the predicted log odds.

Referring to Fig 3.9. The same can be observed in the exploded view provided by the waterfall plot for the given sample and class.

Referring to Fig 3.10. The decision plot confirms our evaluation as we see how each feature cumulatively affects and takes the predicted probability of this sample falling in the Poor-quality category towards 1.

4.References:

[1] United Nations Environment Programme, "FAQs on Water Quality," [Online]. Available: https://www.unep.org/topics/fresh-water/water-quality/multi-stakeholder-engagement/faqs-water-quality.

[2] A. S. Brar, *Consumer Behaviour and Perception for Efficient Water Use in Urban Punjab*, 2013.

[3] H. Cheng, Y. Hu, and J. Zhao, "Meeting China's water shortage crisis: current practices and challenges," *Environ. Sci. Technol.*, vol. 43, no. 2, pp. 240–244, 2009.

[4] A. N. Ahmed, F. B. Othman, H. A. Afan, R. K. Ibrahim, C. M. Fai, M. S. Hossain, and A. Elshafie, "Machine learning methods for better water quality prediction," *J. Hydrol.*, vol. 578, p. 124084, 2019.

[5] World Health Organization, *Guidelines for Drinking-water Quality*, 4th ed., 2012. ISBN 978 92 4 154815 1.

[6] Bureau of Indian Standards, *Specification for drinking water*, IS: 10500, New Delhi, India, 2012.

[7] R. K. Horton, "An index number system for rating water quality," *J. Water Pollut. Control Fed.*, vol. 37, no. 3, pp. 300–306, 1965.

[8] R. M. Brown, N. I. McClelland, R. A. Deininger, and R. G. Tozer, "Water quality index-do we dare?," *Water Sewage Works*, vol. 117, no. 10, pp. 339–343, 1970.

[9] D. Kumar, R. Kumar, M. Sharma, A. Awasthi, and M. Kumar, "Global water quality indices: development, implications, and limitations," *Total Environ. Adv.*, vol. 9, p. 200095, 2024.

[10] K. Saffran, K. Cash, K. Hallard, and R. Wright, *Canadian water quality guidelines for the protection of aquatic life. CCME Water Quality Index 1.0, Users Manual, Excerpt from Publication*, 2001, p. 1299.

[11] R. Makubura, D. P. P. Meddage, H. M. Azamathulla, M. Pandey, and U. Rathnayake, "A simplified mathematical formulation for water quality index (WQI): a case study in the Kelani River Basin, Sri Lanka," *Fluids*, vol. 7, no. 5, p. 147, 2022.

[12] S. Lee and D. Lee, "Improved prediction of harmful algal blooms in four Major South Korea's Rivers using deep learning models," *Int. J. Environ. Res. Public Health*, vol. 15, p. 1322, 2018.

[13] N. Nasir, A. Kansal, O. Alshaltone, F. Barneih, M. Sameer, A. Shanableh, and A. Al-Shamma'a, "Water quality classification using machine learning algorithms," *J. Water Process Eng.*, vol. 48, p. 102920, 2022.

[14] A. H. Haghiabi, A. H. Nasrolahi, and A. Parsaie, "Water quality prediction using machine learning methods," *Water Qual. Res. J.*, vol. 53, no. 1, pp. 3–13, 2018.

[15] N. Nasir, A. Kansal, O. Alshaltone, F. Barneih, M. Sameer, A. Shanableh, and A. Al Shamma'a, "Water quality classification using machine learning algorithms," *J. Water Proc. Eng.*, vol. 48, p. 102920, 2022.

[16] D. T. Bui, K. Khosravi, J. Tiefenbacher, H. Nguyen, and N. Kazakis, "Improving prediction of water quality indices using novel hybrid machine-learning algorithms," *Sci. Total Environ.*, vol. 721, p. 137612, 2020.

[17] R. K. Makumbura, S. Henna, L. Mampitiya, T. L. Dang, N. Rathnayake, Y. Hoshino, D. P. P. Meddage, and U. Rathnayake, "Advancing water quality assessment and prediction using machine learning models, coupled with explainable artificial intelligence (XAI) techniques like shapley additive explanations (SHAP) for interpreting the black-box nature."

[18] S. Lundberg and S.-I. Lee, "A Unified Approach to Interpreting Model Predictions."

[19] S. Tyagi, B. Sharma, P. Singh, and R. Dobhal, "Water Quality Assessment in Terms of Water Quality Index," 2012.