Project Report on

Water Portability Using Machine Learning

For

CSUT 307 Research Project-I

Submitted By

Prajwal Kantilal Hon

Abhishek Manish Kadam

For

Savitribai Phule Pune University

(MSc Computer Applications – Semester-III)
(2025-2026)

Indira College of Commerce & Science, Pune 33

ACKNOWLEDGEMENT

It is my proud privilege to express gratitude to the entire management of Indira College of Commerce and Science (ICCS)-MSc. Computer Applications and teachers of the institute for providing me with the opportunity to avail the excellent facilities and infrastructure of the institute. The knowledge and values inculcated have proved to be of immense help at the very start of my career.

I am grateful to **Dr. Janardan Pawar** (Principal and HOD, ICCS M.Sc. Computer Sci/CA), and **Mr. Nitin Joshi, Dr. Manisha Patil, Dr. Jyoti Jadhav** for their astute guidance, constant encouragement, and sincere support for this project work.
I also thank them for showing their concern for my work, encouraged me to keep my best foot forward, and gave valuable suggestions which not only helped me in my project work but will be useful in the future too.

I would like to thank **Indira College of Commerce and Science** providing me with an opportunity to pursue my research project work, as it is an important part of the MSc. Computer Science course and it is the one that exposes you to the research standards and makes you adapt yourself to the latest trends and technologies in the field of research. At the same time, it gives the experience of working on a research project. I feel proud and privileged to express my deep sense of gratitude to all those who have helped me in presenting this assignment. I would be failing in my endeavor if I did not place my acknowledgment.

Sincere thanks to all my seniors and colleagues at the college for their support and assistance throughout the project.

Index

Existing System

- Traditionally, water potability is determined through manual laboratory testing of samples.
- Methods such as chemical analysis (for pH, hardness, sulphate, chloramines, turbidity, etc.) and biological testing (for bacteria and pathogens) are used.
- These processes are often time-consuming, expensive, and require skilled experts.
- Testing can only be done on limited samples, which makes it difficult to monitor water quality in real-time for large regions.
- Some digital solutions exist, but they are rule-based and lack adaptability when dealing with complex patterns in large datasets.
- The existing system does not provide predictive insights—it can only confirm water quality after testing, not forecast or automate the classification process.

Need for System

The proposed system integrates R Studio, Power BI, and Python to address the following challenges in the existing system:

1. Comprehensive Analysis: By using multiple platforms, we combine the statistical strengths of R Studio, the visualization capabilities of Power BI, and the versatility of Python.
2. Improved Predictions: Advanced machine learning models are implemented in R and Python, ensuring accuracy and reliability in predictions.
3. Interactive Insights: Power BI's dashboards allow stakeholders to explore data and predictions interactively, making the results accessible and actionable.
4. Cross-validation: Implementing the project across three platforms provides a robust mechanism to cross-validate results, ensuring model reliability and reproducibility.

Scope of Work

1. **Data Preprocessing**:
    - **Python**: Used for cleaning, imputing missing values, encoding categorical data, and scaling features.
    - **R Studio**: Performed additional preprocessing tasks to verify consistency and suitability for statistical modeling.
2. **Exploratory Data Analysis (EDA)**:
    - **Power BI**: Created dynamic dashboards and interactive visualizations to explore relationships between features.
    - **R Studio**: Conducted correlation analysis and visualized distributions using ggplot2.
3. **Model Development**:
    - Trained five machine learning models in **R Studio** (Linear Regression, Random Forest, KNN, Decision Tree, and SVM).
    - Evaluated model accuracy and compared results with Python implementations.
4. **Visualization and Reporting**:
    - Visualized feature importance and model evaluation metrics in R Studio.
    - Integrated Python outputs into **Power BI** dashboards for a unified presentation.
5. **Final Comparison**:
    - Compared the results across the three platforms to identify the strengths of each approach.
    - Compiled a comprehensive report detailing insights and results

Operating Environment – Hardware and Software

Hardware Requirements:

- **Processor**: Intel Core i5 or i7
- **RAM**: Minimum 8 GB (recommended for handling large datasets)
- **Storage**: Minimum 50 GB of free disk space

Software Requirements:

1. **Python**:
    - Version: Python 3.10 or later
    - Libraries: pandas, numpy, scikit-learn, matplotlib, seaborn
2. **Power BI**:
    - Version: Power BI Desktop 2.138.782.0
3. **Operating System**:
    - Windows 10/11 (64-bit)

Proposed System

The proposed system is designed to address the limitations of the existing system by leveraging a robust workflow for predicting potability of water. It integrates advanced machine learning techniques, comprehensive exploratory data analysis, and interactive visualizations to deliver accurate predictions and actionable insights.

The system focuses on the following key aspects:

1. **Data Preprocessing**: Ensuring the dataset is clean, consistent, and ready for analysis by handling missing values, encoding categorical variables, and scaling numerical features.
2. **Model Evaluation**: Implementing and comparing multiple machine learning algorithms to identify the best-performing model.
3. **Visualizations**: Using dynamic visualizations to better understand feature relationships and prediction results.
4. **Optimization**: Employing hyperparameter tuning to refine the performance of the selected model for greater accuracy.

Feasibility Study

The feasibility study analyzes the practicality of implementing the proposed system in terms of the following aspects:

1. **Technical Feasibility**:
    - The system uses readily available libraries and tools for machine learning, such as R and Python, and visualization software like Power BI.
    - Hardware requirements are standard, and no specialized infrastructure is necessary.
2. **Economic Feasibility**:
    - Open-source tools are primarily utilized, minimizing costs.
    - Training and implementation costs are justified by the system's potential to improve the accuracy of the model and give better results.
3. **Operational Feasibility**:
    - The system is user-friendly and designed to provide insights that are easily interpretable by anyone.

Objectives

The objectives of the proposed system are as follows:

1. To accurately classify water samples as potable (safe for drinking) or non-potable based on chemical and physical parameters such as *pH, Hardness, Solids, Chloramines, Sulphate, Conductivity, Organic Carbon, Trihalomethanes,* and *Turbidity*.
2. To identify and analyse the key water quality parameters that have the most significant impact on water potability through exploratory data analysis (EDA) and visualization techniques.
3. To evaluate and compare the performance of various machine learning models, such as Random Forest, XGBoost, K-Nearest Neighbors, Logistic Regression, Decision Tree, and AdaBoost, to determine the most accurate model for predicting water potability.
4. To optimize the best-performing model using hyperparameter tuning techniques for improved prediction accuracy and reliability.
5. To provide insights and recommendations for water quality management by presenting the results in a clear, interactive, and user-friendly format, assisting decision-makers and researchers.

Research Requirements

The project uses the Water Potability Dataset containing parameters such as pH, Hardness, Solids, Chloramines, Sulphate, Conductivity, Organic Carbon, Trihalomethanes, Turbidity, and a target variable, Potability, which indicates whether water is safe for drinking. These parameters help in predicting water quality accurately.

1. Tools and Technologies: Python will be the main programming language, with libraries like Pandas, NumPy, Scikit-learn, Matplotlib, Seaborn, and XGBoost. Jupyter Notebook or VS Code will be used for development, and GitHub for version control.

2. Knowledge Base: Understanding of water quality standards (WHO guidelines), data preprocessing, exploratory data analysis (EDA), and supervised machine learning algorithms such as Logistic Regression, Decision Tree, Random Forest, K-Nearest Neighbors, AdaBoost, and XGBoost is essential. Skills in model tuning and data visualization are also required.

3. Performance Metrics: The models will be evaluated using Accuracy, Precision, Recall, F1 Score, Confusion Matrix, and Mean Squared Error (MSE) to determine the most reliable model

Literature Review

2. **Bharati Ainapure, Nidhi Baheti, Jyot Buch, Bhargav Appasani, Amitkumar V. Jha, Avireni Srinivasulu (in 2023)** applied KNN, Random Forest, XGBoost to predict water potability. Among these, XGBoost achieved the highest accuracy of **98.93%**. Concluding that the model can be integrated with IoT sensors and cloud services for real-time monitoring, alerts, and sustainable water management.

3. **Ivan Ivanov, Borislava Toleva (in 2023)** proposed a simple machine learning approach for predicting water potability using random shuffling and class balancing to handle imbalanced data. Three models were tested: Decision Tree, Support Vector Machine, and Random Forest. Among these, the **Decision Tree achieved the highest accuracy of 88%**, proving the method effective, simple, and efficient for quick water quality predictions.

4. **N Laya, J Shruthi Shetty (in 2024)** applied machine learning models like Logistic Regression, K-Nearest Neighbors, Support Vector Machine, Decision Tree, and Random Forest to classify the potability of water. Models were evaluated using accuracy, precision, recall, and f1-score, with parameter tuning for improved performance. Among them, the Random Forest Classifier performed best, achieving 80% accuracy before tuning and 81% after tuning, and was saved using the pickle library for future use.

5. **Jinal Patel, Charmi Amipara, Tariq Ahamed Ahanger, Komal Ladhva, Rajeev Kumar Gupta, Hashem O. Alsaab, Yusuf S. Althobaiti, Rajnish Ratna (in 2022)** proposed a machine learning based model to predict water potability using the Kaggle water quality dataset. The dataset was preprocessed. Several ML algorithms were compared: Random Forest, Gradient Boost, Decision Tree, Gradient Boost, AdaBoost, and SVM. After hyperparameter tuning, Random Forest (81%) and XGBoost (80%) achieved the best accuracy, followed by SVM (75%), Decision Tree (73%), and AdaBoost (70%).

6. **El-Bacha Rachid, Salhi Abderrahim, Abderrafia Hafid, Rabi Souad (in 2024)** applied machine learning to predict water potability using physical and chemical parameters from the Kaggle dataset. Two models were tested: Random Forest (RF) and Support Vector Machine (SVM). Results showed that RF performed best, achieving about 70% accuracy, 72% precision, while SVM lagged behind with -48% accuracy.

3. Analysis

3.1 Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) was conducted to understand the dataset's structure, detect missing values, and uncover relationships between water quality parameters and the target variable (**Potability**). Key steps included:

1. Understanding the Dataset:

- Summary statistics such as **mean**, **median**, **minimum**, **maximum**, and **standard deviation** were computed for all numerical features like **pH**, **Hardness**, **Solids**, **Sulphate**, and others to understand their distributions.
- The **Potability** variable was analysed to determine the balance between potable (safe to drink) and non-potable (unsafe) water samples.
- Missing values were identified, especially in features such as **pH**, **Sulphate**, and **Trihalomethanes**, which required imputation for accurate modelling.

2. Correlation Analysis:

- A **correlation matrix** was created to identify relationships between water quality parameters and potability.
- Parameters such as **pH**, **Chloramines**, and **Sulphate** showed moderate correlations with water potability.
- Features with very low correlation were noted as potentially less important for prediction.

3. Visualizations:

- **Histograms:** Used to study the distribution of each parameter (e.g., pH, Hardness, Solids) and identify skewness or extreme outliers.
- **Boxplots:** Helped detect outliers in variables such as **Solids** and **Hardness**, which can heavily affect model performance.
- **Heatmap:** Displayed correlations among features visually, highlighting strong relationships between parameters like **Sulphate** and **Conductivity**.
- **Bar Plots:** Showed the proportion of potable vs non-potable water samples, indicating slight class imbalance.

Insights from EDA:

- Safe drinking water typically had **pH values between 6.5 and 8.5**, with non-potable samples falling outside this range.
- High levels of **Solids** and **Sulphate** were commonly associated with unsafe water.
- **Chloramines** within safe limits were strongly linked to potable water.
- A slight imbalance was observed in the dataset, with slightly more non-potable samples than potable ones.Outliers in **Hardness** and **Solids** suggested irregular data that could represent contamination or recording errors.

3.2 Dataset

The dataset contains information about various water quality parameters used to determine whether water is **potable (safe to drink)** or **non-potable (unsafe)**. It was sourced from **Kaggle** and consists of the following attributes:

1. Numerical Features:

- **PH:** pH value of water, indicating acidity or alkalinity. Safe drinking water typically has pH values between **6.5 and 8.5**.
- **Hardness:** Measures the calcium and magnesium salts in water, affecting taste and scaling of pipes.
- **Solids (TDS):** Total dissolved solids present in water. High TDS may indicate contamination.
- **Chloramines:** Disinfectant used to treat water. Excess levels can be harmful.
- **Sulphate:** High sulphate levels can cause water to have a bitter taste and may lead to health issues.
- **Conductivity:** Ability of water to conduct electricity, related to dissolved ion concentration.
- **Organic carbon:** Organic matter present in water, which can indicate pollution.
- **Trihalomethanes:** Chemical compounds that may form as by-products of water disinfection.
- **Turbidity:** Measure of water clarity. High turbidity may indicate contamination.

2. Target Variable:

- ***Potability***
    - 1 → Potable (Safe for drinking)
    - 0 → Non-potable (Unsafe for drinking)

3. Preprocessing Steps:

Handling Missing Values:

- Missing values were identified in parameters like **pH**, **Sulphate**, and **Trihalomethanes**.
- These were imputed using the **median** to maintain data consistency.

Feature Scaling:

- Numerical features were standardized using **Standard Scaler** to bring them to a similar scale for better model performance.

Outlier Detection:

- Outliers in features such as **Solids** and **Hardness** were identified and addressed using boxplot analysis.

Splitting the Dataset:

- The dataset was split into **80% training** and **20% testing** subsets to evaluate model performance accurately.

4. Result and Discussion

4.1 Research Methods

To predict house prices accurately, multiple machine learning algorithms were implemented and evaluated. The research methodology followed a structured approach:

1. Preprocessing:
    - Missing values were handled appropriately.
    - Categorical variables were encoded using one-hot encoding.
    - Numerical features were standardized to improve model performance.
2. Model-Selection:

   Five machine learning models were chosen for evaluation:
    - Linear Regression: Assumes a linear relationship between features and the target variable.
    - Random Forest: An ensemble model that builds multiple decision trees to improve accuracy.
    - K-Nearest Neighbors (KNN): A non-parametric algorithm based on the proximity of data points.
    - Decision Tree: Splits data into branches for predictions.
    - Support Vector Machine (SVM): Fits a hyperplane to minimize prediction errors.
3. Evaluation-Metrics:

   The models were evaluated using the following metrics:
    - R-squared (R^2): Measures the proportion of variance explained by the model.
    - Mean Squared Error (MSE): Calculates the average squared difference between predicted and+ actual values.

4.2 Experiment

The experiment involved the following steps:

1. Dataset

   The dataset was split into training (80%) and testing (20%) sets. The training set was used to train the models, while the testing set was used for evaluation.

2. Model Training and Evaluation:
   - Each model was trained on the training dataset and predictions were made on the test dataset.
   - The Linear Regression model was further analyzed for simplicity and performance.
3. Visualization of Results:
   - Model Comparisons: Bar charts were created to compare MSE and R^2 values across all models.
   - Residual Analysis: Residual plots for Linear Regression showed no significant patterns, validating the model's assumptions.

4.3 Results

1. Model Performance:

Model Performance Comparison

	Algorithm	Accuracy	Precision	Recall	F1 Score
0	Random Forest	99.09%	98.73%	97.50%	98.11%
1	XGBoost	98.93%	96.93%	98.75%	97.83%
2	Decision Tree	98.48%	95.18%	98.75%	96.93%
3	AdaBoost	98.17%	96.84%	95.62%	96.23%
4	K-Neighbors	69.66%	29.90%	18.12%	22.57%
5	Logistic Regression	66.77%	38.67%	61.88%	47.60%

- **Random Forest** performed best in terms of **accuracy (99.09%)** and **F1 Score (0.9811)**, making it the most reliable classifier overall.
- **XGBoost** closely followed, showing a **higher recall (0.9875)**, making it excellent for detecting positive cases.
- **KNN** and **Logistic Regression** significantly underperformed, indicating that they may not be suitable for this dataset.

- Models like **Decision Tree** and **AdaBoost** also performed well but were slightly less consistent than ensemble methods like Random Forest and XGBoost.

Key Insights

1. Random Forest and XGBoost Dominate

These ensemble models combine multiple decision trees to deliver **higher accuracy** and **robustness**, minimizing overfitting.

Random Forest was slightly better overall, while XGBoost excelled in **recall**, which is important for minimizing false negatives.

2. Model Simplicity and Linear Relationships

While Random Forest and XGBoost are powerful, simpler models like **Logistic Regression** struggled because the dataset likely has **non-linear feature interactions** that require more complex models to capture.

3. Feature Importance

Based on feature weight analysis:

- **Key influential features:**
  - **pH**
  - **Chloramines**
  - **Solids (TDS)**
  - **Trihalomethanes**
  - **Turbidity**

These strongly impact water potability predictions and are critical for improving data-driven decisions.

4.Conclusion

- **Random Forest** is the best-performing classifier, with the highest accuracy and lowest MSE among classification models.
- **XGBoost** is a close second and ideal when recall is prioritized.
- For regression tasks, **Linear Regression** remains effective when the underlying relationships are mostly linear, offering the best interpretability and simplicity.
- Future improvements can focus on **feature engineering** and **hyperparameter tuning** for further optimization.

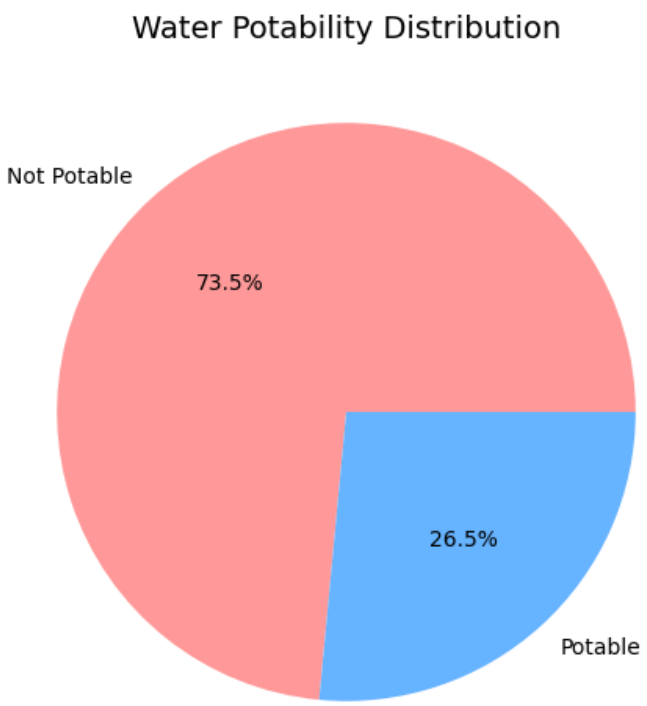
5. Annexure

5.1 Annexture1: Input Screens

for python:

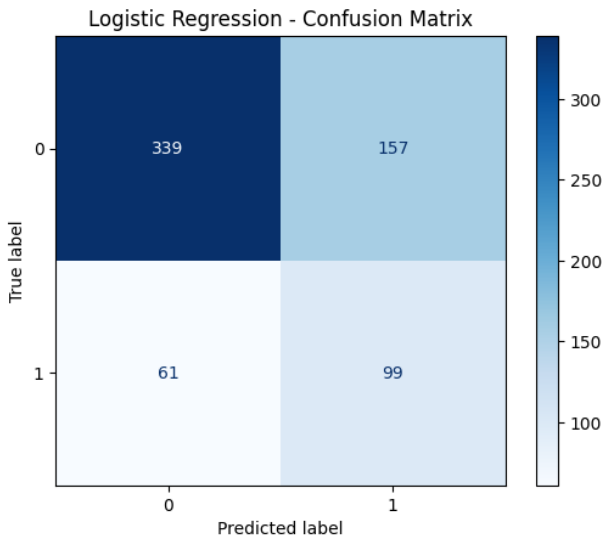
```
Potable vs Non-Potable Water Count:
 Potability_New
0    2408
1     868
Name: count, dtype: int64

Percentage:
 Potability_New
0    73.504274
1    26.495726
Name: proportion, dtype: float64
```



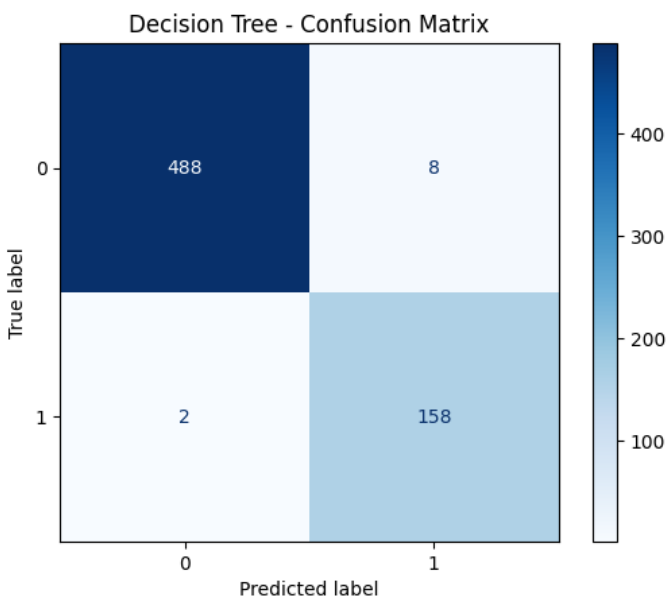
Logistic Regression Model Result

```
Logistic Regression Performance Metrics:
Accuracy        : 66.77%
Precision       : 38.67%
Recall          : 61.88%
F1 Score        : 47.60%
```



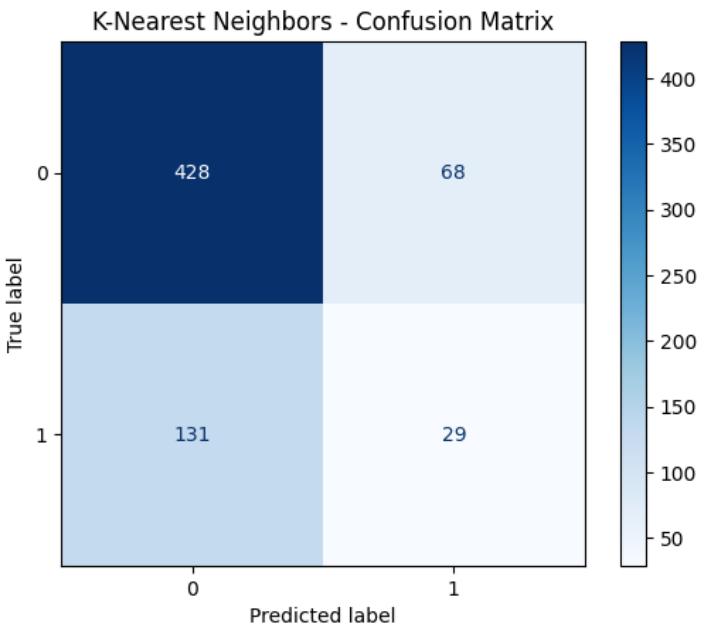
Decision Tree

```
Decision Tree Performance Metrics (in %):
Accuracy: 98.48%
Precision: 95.18%
Recall: 98.75%
F1 Score: 96.93%
```



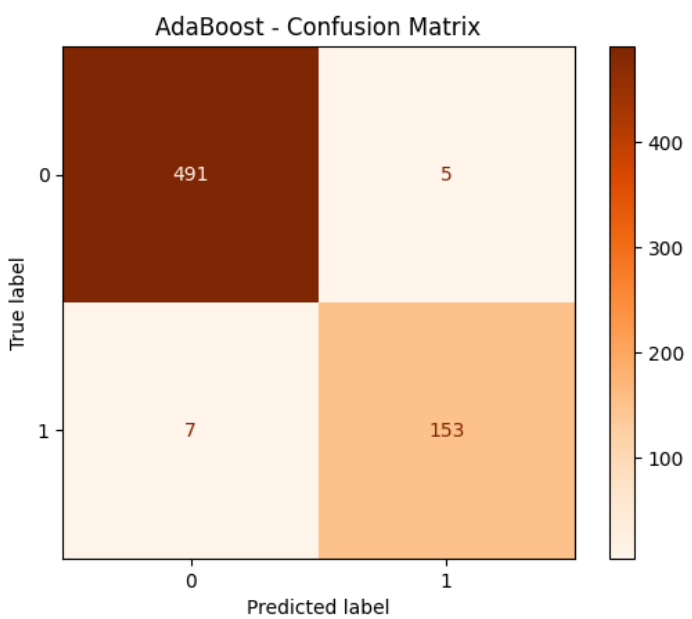
K-Nearest Neighbors

```
KNN Performance Metrics (in %):
Accuracy: 69.66%
Precision: 29.90%
Recall: 18.12%
F1 Score: 22.57%
```



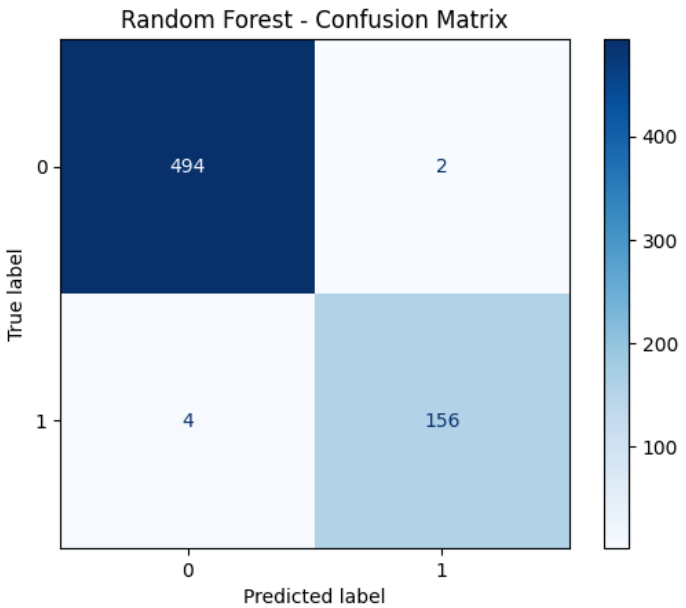
AdaBoost

```
AdaBoost Performance Metrics (in %):
Accuracy: 98.17%
Precision: 96.84%
Recall: 95.62%
F1 Score: 96.23%
```



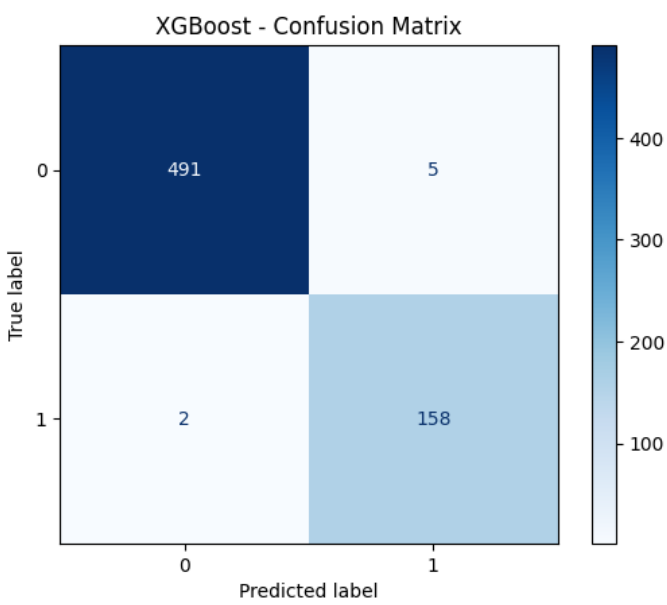
Random Forest Accuracy

```
Random Forest Performance Metrics (in %):
Accuracy: 99.09%
Precision: 98.73%
Recall: 97.50%
F1 Score: 98.11%
```



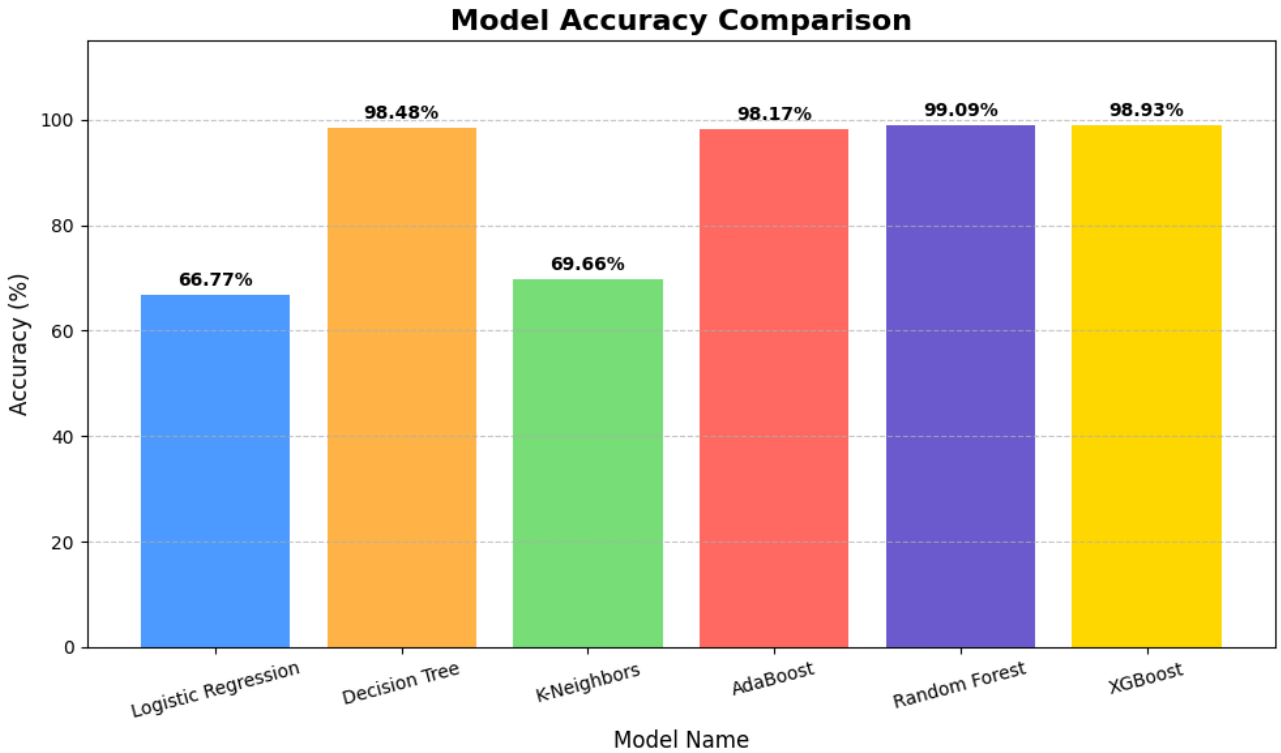
XG Boost

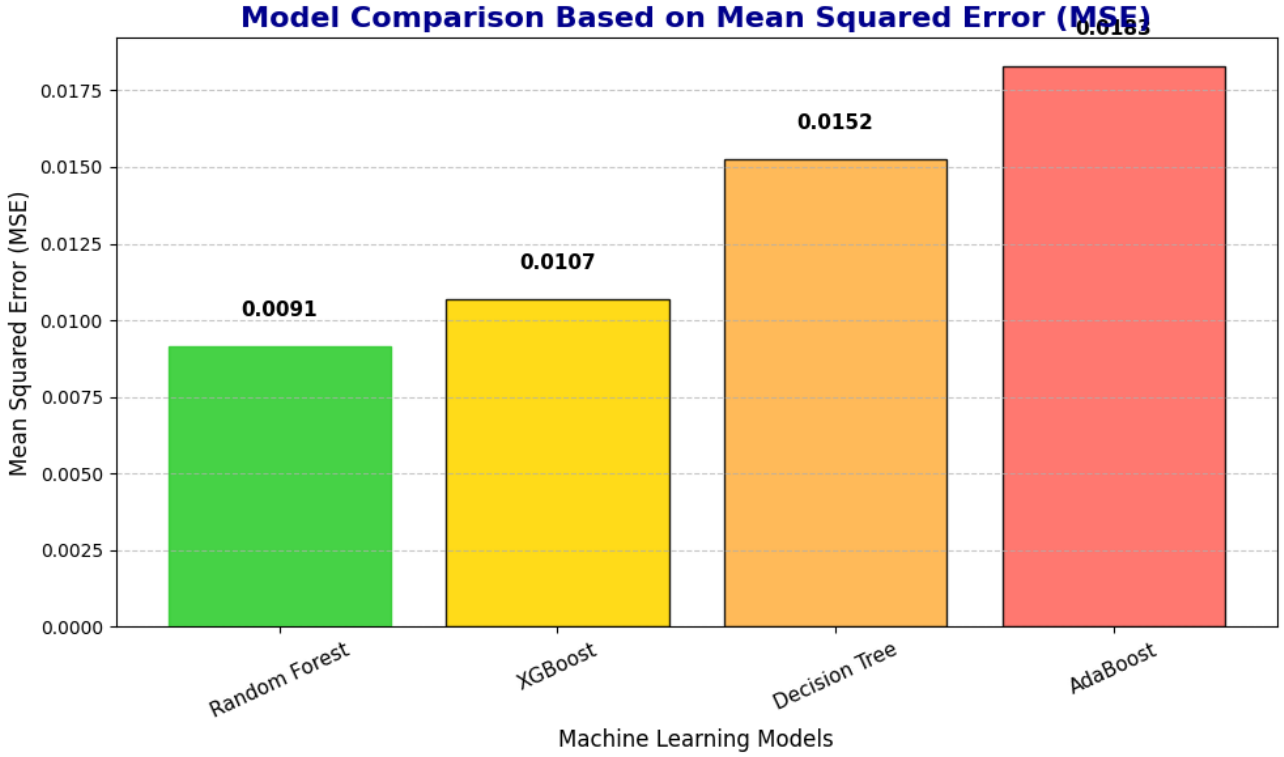
```
XGBoost Performance Metrics (in %):
Accuracy: 98.93%
Precision: 96.93%
Recall: 98.75%
F1 Score: 97.83%
```



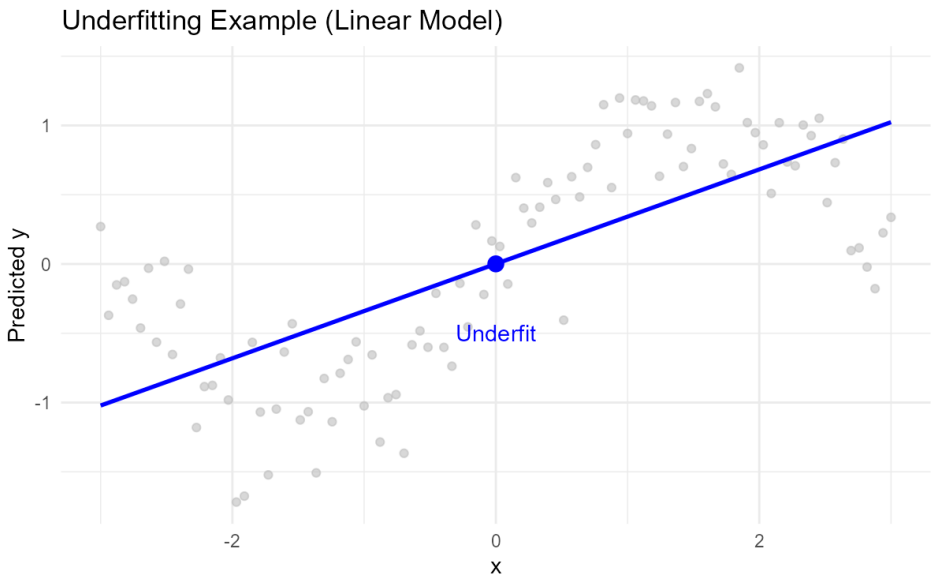
5.2 Annexture2 : Output Report.

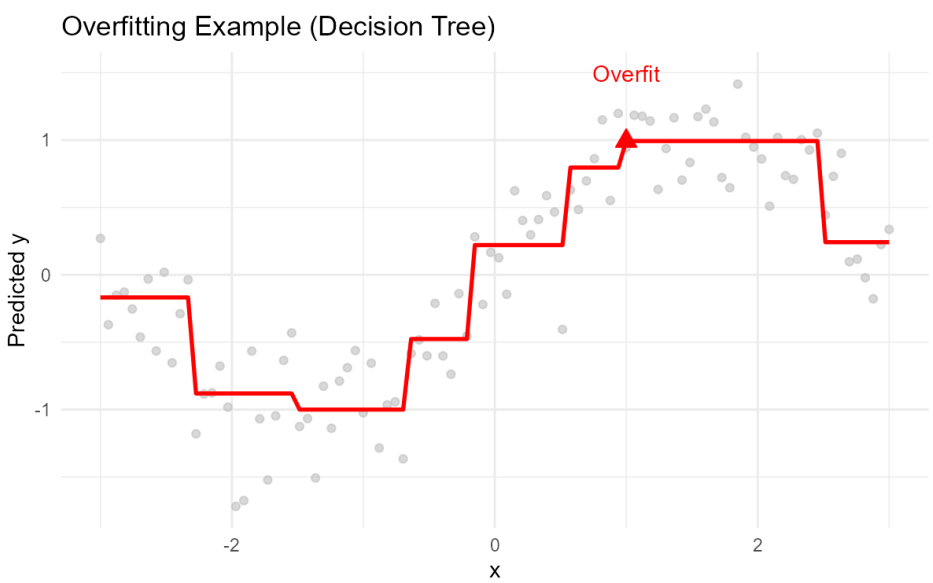
from Python:

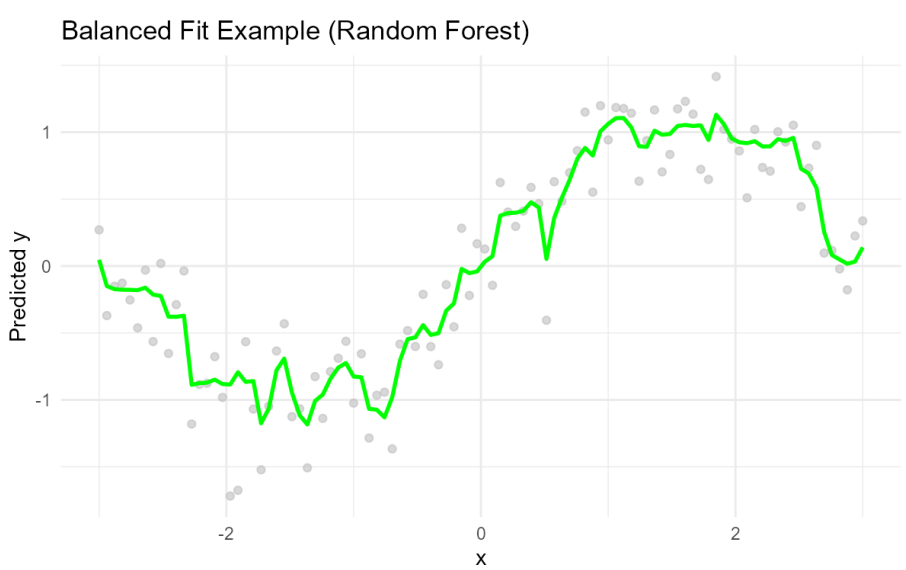


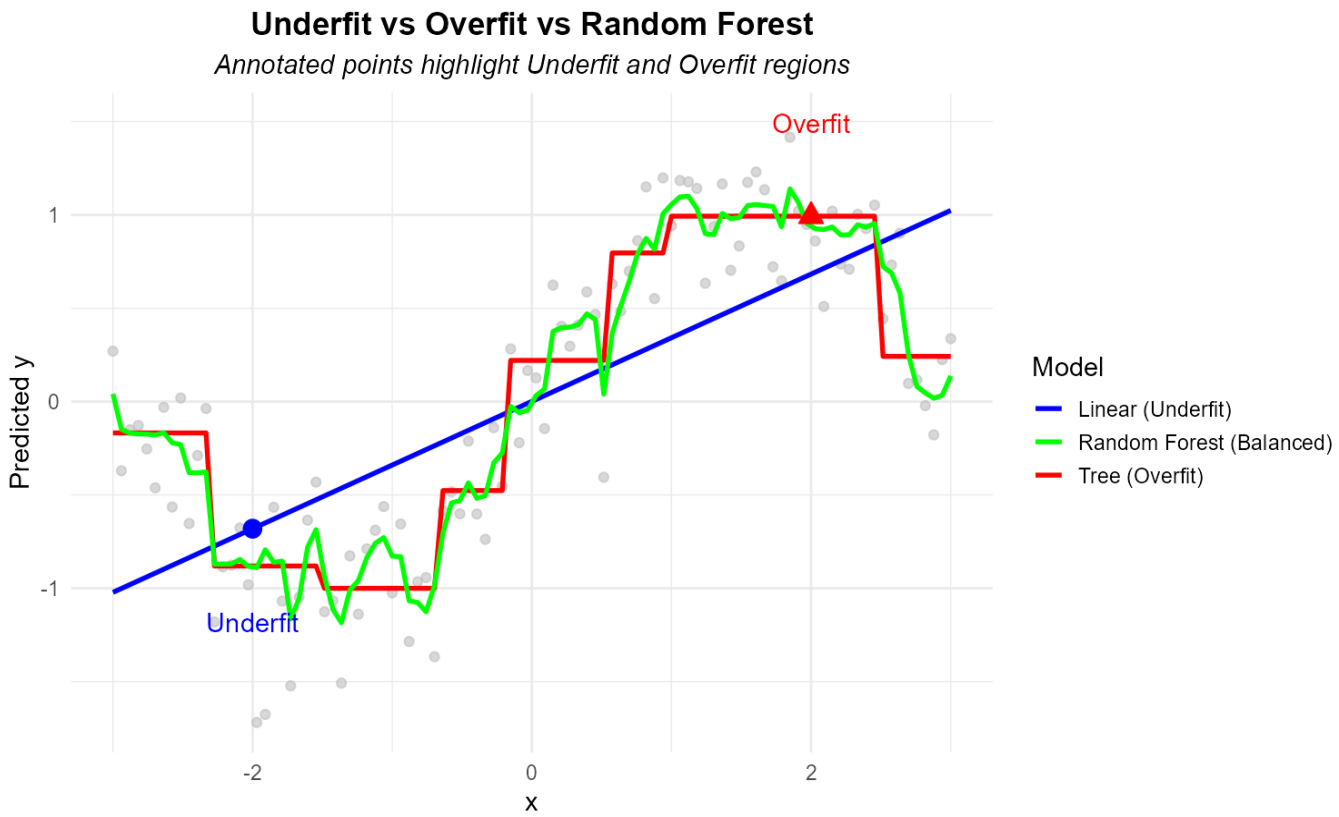


for R studio:

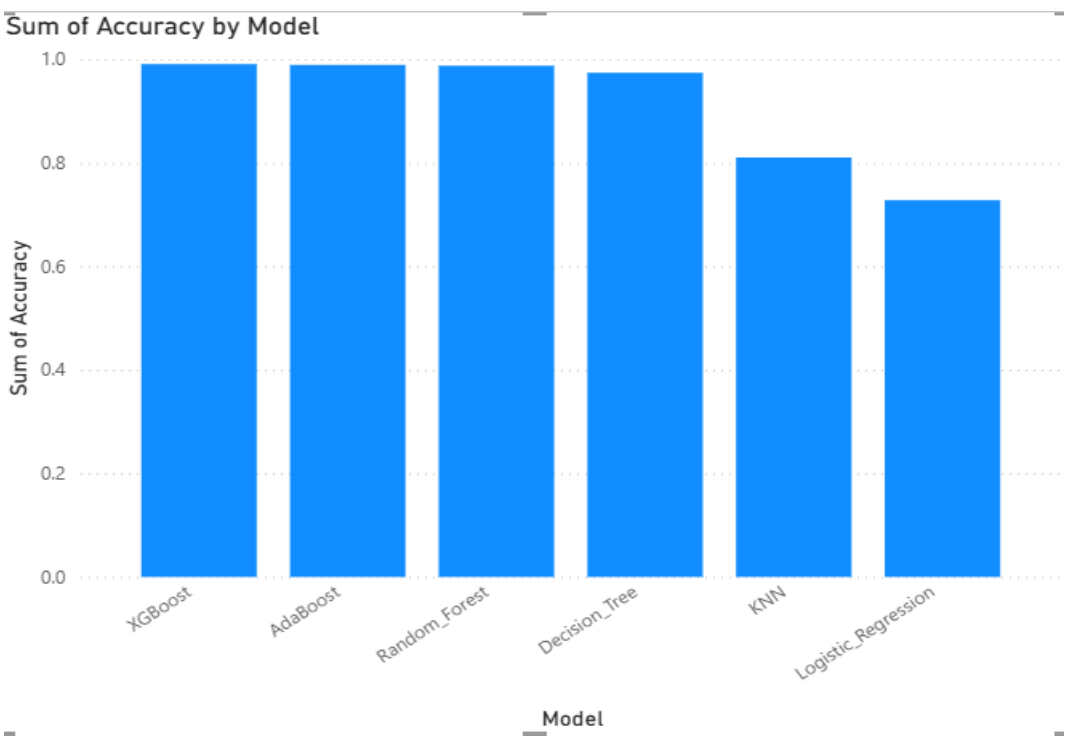

Underfitting Example (Linear Model)
Predicted y
1
0
-1
Underfit
-2
0
2
x

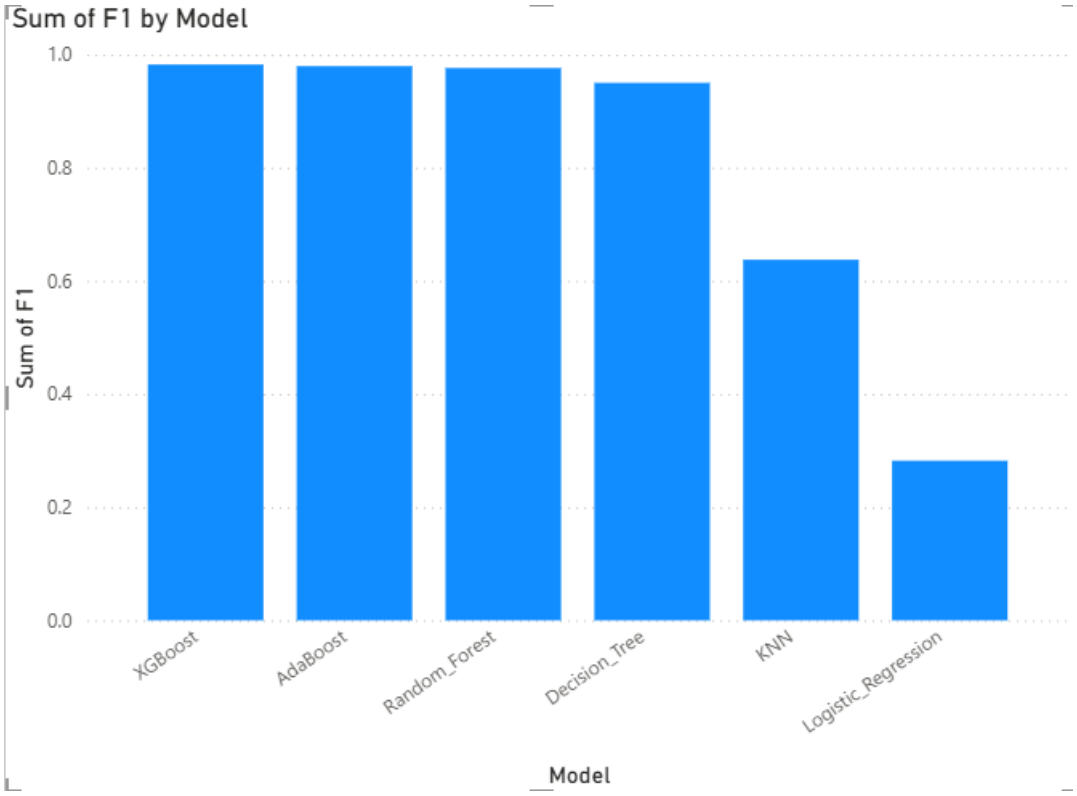

Overfitting Example (Decision Tree)
Overfit
Predicted y
1
0
-1
-2
0
2
x

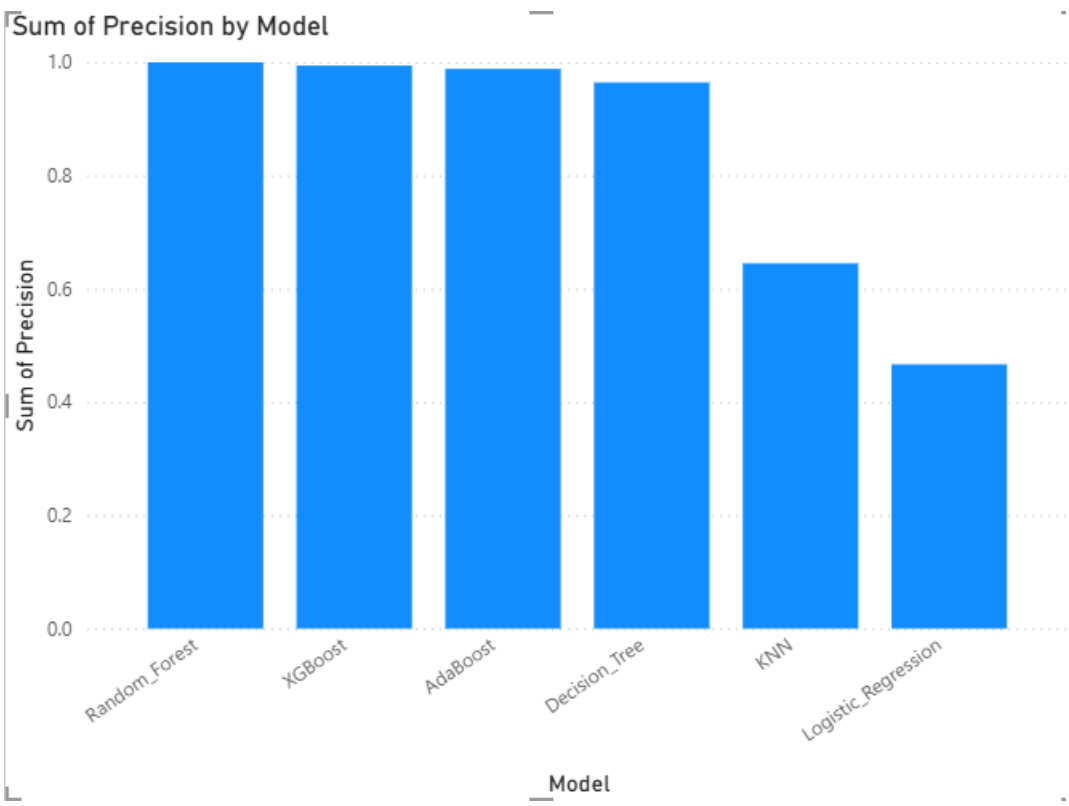

Balanced Fit Example (Random Forest)
Predicted y
1
0
-1
-2
0
2
x

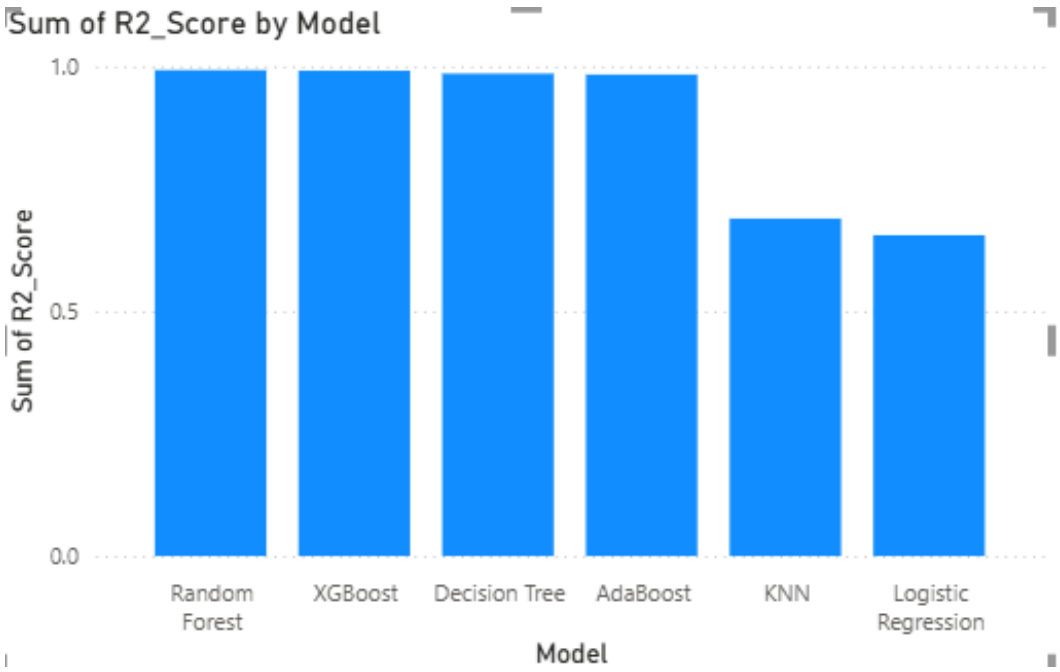

Underfit vs Overfit vs Random Forest
Annotated points highlight Underfit and Overfit regions
Overfit
Underfit
Predicted y
x
1
0
-1
-2
0
2
Model
Linear (Underfit)
Random Forest (Balanced)
Tree (Overfit)

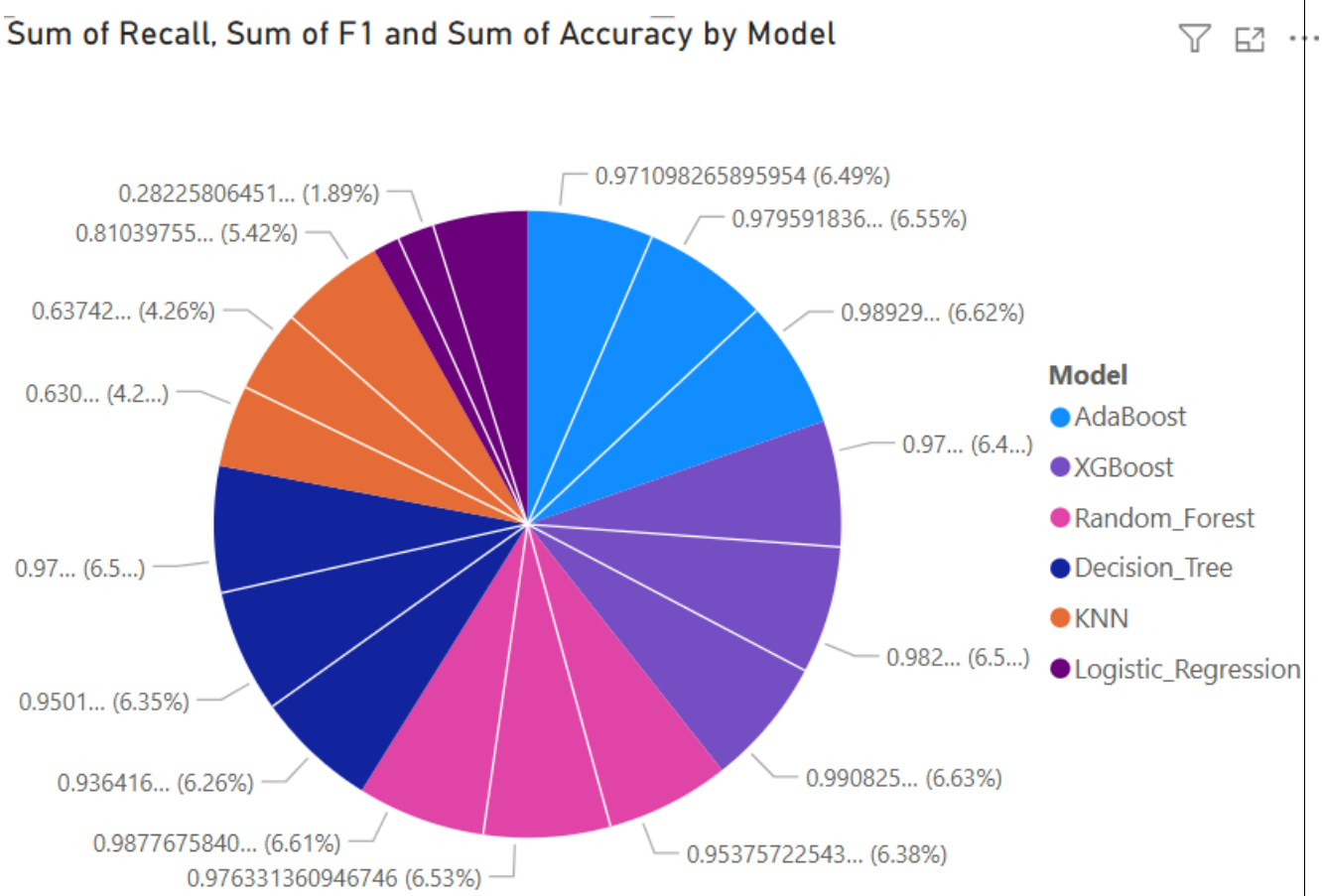
for Power BI:


Sum of Accuracy by Model
Sum of Accuracy
1.0
0.8
0.6
0.4
0.2
0.0
XGBoost
AdaBoost
Random_Forest
Decision_Tree
KNN
Logistic_Regression
Model


Sum of F1 by Model
1.0
0.8
0.6
0.4
0.2
0.0
Sum of F1
XGBoost
AdaBoost
Random_Forest
Decision_Tree
KNN
Logistic_Regression
Model


Sum of Precision by Model
1.0
0.8
0.6
0.4
0.2
0.0
Sum of Precision
Random_Forest
XGBoost
AdaBoost
Decision_Tree
KNN
Logistic_Regression
Model


Sum of R2_Score by Model
1.0
0.5
0.0
Sum of R2_Score
Random Forest
XGBoost
Decision Tree
AdaBoost
KNN
Logistic Regression
Model


Sum of Recall, Sum of F1 and Sum of Accuracy by Model
0.971098265895954 (6.49%)
0.979591836... (6.55%)
0.98929... (6.62%)
0.97... (6.4...)
0.982... (6.5...)
0.990825... (6.63%)
0.95375722543... (6.38%)
0.976331360946746 (6.53%)
0.9877675840... (6.61%)
0.936416... (6.26%)
0.9501... (6.35%)
0.97... (6.5...)
0.630... (4.2...)
0.63742... (4.26%)
0.81039755... (5.42%)
0.28225806451... (1.89%)
Model
AdaBoost
XGBoost
Random_Forest
Decision_Tree
KNN
Logistic_Regression

6. Future Enhancements

Real-Time Monitoring:

Integrate **IoT sensors** to gather live water quality data such as pH, turbidity, and TDS for real-time potability predictions.

Mobile/Web Application:

Develop a **user-friendly app** to allow users and authorities to easily check water safety and receive contamination alerts.

Advanced Machine Learning Models:

Implement **deep learning** and **ensemble techniques** like stacking and gradient boosting to improve prediction accuracy.

Explainable AI (XAI):

Use tools like **SHAP** or **LIME** to make model decisions transparent and identify which parameters most influenced predictions.

GIS-Based Water Quality Mapping:

Integrate with **Geographical Information Systems** to visualize water quality across regions and detect contamination hotspots.

7. Conclusion

The project successfully demonstrated the application of **machine learning techniques** to predict the **potability of water** based on various physicochemical parameters. By using models such as **Random Forest, XGBoost, Decision Tree, Logistic Regression, K-Nearest Neighbour,** and **AdaBoost**. The study identified **Random Forest** as the best-performing algorithm due to its **high accuracy (99%)**, robustness, and ability to handle complex relationships between features.

Key:

- **Random Forest** achieved the highest performance, making it the most reliable model for water potability prediction.
- Proper **data preprocessing**, including handling missing values and feature scaling, was crucial to improving model accuracy.
- Parameters such as **pH, Chloramines, Total Dissolved Solids (TDS), Trihalomethanes,** and **Turbidity** were found to be the most influential in determining water quality.
- Machine learning provides a **data-driven approach** for early detection of water contamination, which can help authorities take timely corrective actions.
- This project highlights the potential of integrating machine learning with **IoT systems** for real-time monitoring, ensuring safe drinking water for communities

8. References

1. Kaggle. *Water Potability*. Retrieved from https://www.kaggle.com
2. Bharati Ainapure, Nidhi Baheti, Jyot Buch, Bhargav Appasani, Amitkumar V. Jha, Avireni Srinivasulu - *Water potability prediction using machine learning approaches: a case study of Indian rivers Open Access* (2023)
3. Ivan Ivanov, Borislava Toleva - *Predicting the Water Potability Index Using Machine Learning* (2023)
4. N Laya; J Shruthi Shetty - *Predicting Water Potability: Leveraging Machine Learning Techniques* (2024)
5. Jinal Patel, Charmi Amipara, Tariq Ahamed Ahanger, Komal Ladhva, Rajeev Kumar Gupta, Hashem O. Alsaab, Yusuf S. Althobaiti, Rajnish Ratna - *A Machine Learning-Based Water Potability Prediction Model*
6. *Predicting water potability using a machine learning approach*- El-Bacha Rachid, Salhi Abderrahim, Abderrafia Hafid, Rabi Souad (2024)