Project Report: Water Potability Prediction

By Sai Vinay.

1. Introduction

Access to safe drinking water is a critical global challenge, as waterborne diseases are a major cause of illness and death. Traditionally, water quality is assessed through slow and expensive laboratory analysis. This delay means that unsafe water cannot be identified and addressed in real-time, posing a significant public health risk.

This project seeks to solve this by developing a fast, efficient, and low-cost predictive model using machine learning.

2. Project Aim and Objectives

The aim of this project is to train and evaluate a model that can accurately and automatically classify a water sample as "Potable" (safe to drink) or "Not Potable" (unsafe) using its standard chemical features (like pH, Hardness, and Sulfate).

This project will use a standard public dataset to explore, clean, and compare the performance of three baseline classification algorithms (Logistic Regression, KNN, and Random Forest) to identify the most reliable model for this task.

3. Methodology

3.1. Data Source and Preprocessing

A standard public dataset for water potability (available on Kaggle) was used. A key challenge in this dataset is the presence of significant missing data. As an initial preprocessing step, missing values in the `ph`, `Sulfate`, and `Trihalomethanes` columns were imputed using the mean value of their respective columns.

3.2. Exploratory Data Analysis (EDA)

A preliminary analysis of the dataset was performed:

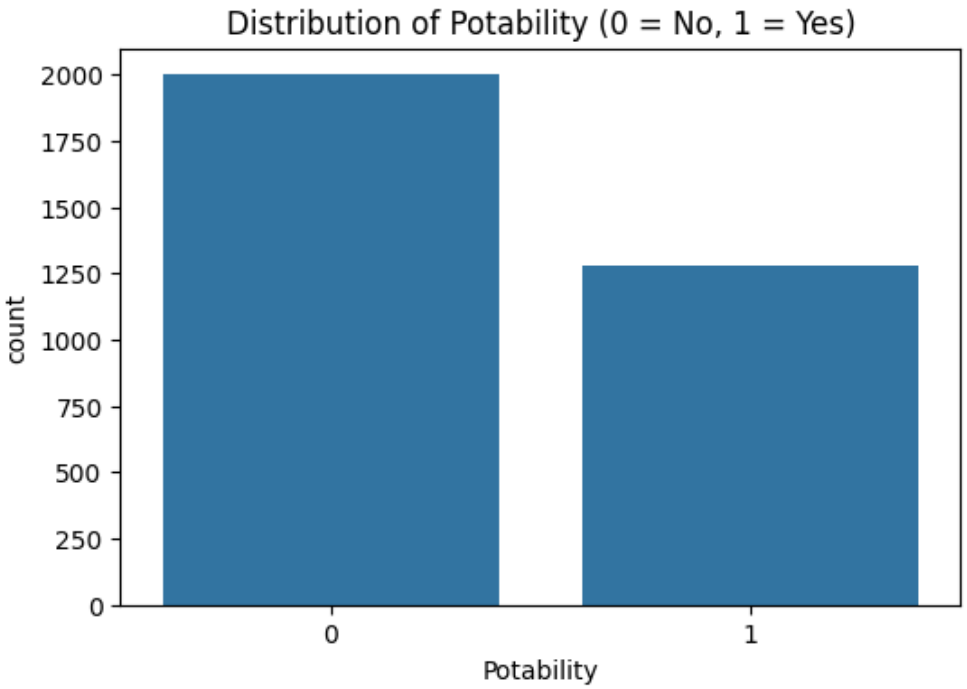
- **Class Distribution:** A count plot of the `Potability` target variable revealed that the dataset is imbalanced, with a majority of samples belonging to the "Not Potable" (Class 0) category. This is a critical finding that impacts model performance.

- **Feature Distribution:** Histograms were generated for all chemical features to understand their statistical distributions.
- **Feature Correlation:** A correlation heatmap was generated to analyze the relationships between different features and the target variable.

3.3. Model Training and Evaluation

1. **Data Splitting:** The dataset was split into training (80%) and testing (20%) sets.
2. **Feature Scaling:** The features were standardized using `StandardScaler` to ensure that all features contributed equally to model training, which is especially important for algorithms like KNN and Logistic Regression.
3. **Model Selection:** Three models were trained and evaluated:
    - Logistic Regression
    - K-Nearest Neighbors (KNN)
    - Random Forest Classifier
4. **Evaluation Metrics:** Model performance was assessed using 5-fold cross-validation on the training set and a detailed classification report (precision, recall, f1-score) and confusion matrix on the test set.

4. Results and Analysis



Distribution of potability

4.1. Cross-Validation (5-Folds) on Training Data

Cross-validation provided a baseline for model robustness. The Random Forest model showed the highest and most consistent average performance.

```
Python

Running Cross-Validation (5-Folds)
Logistic Regression 5 Scores: [0.60687023 0.60496183 0.60687023 0.60305344 0.60496183]
Logistic Regression Average Score: 60.53%


KNN 5 Scores: [0.64312977 0.62977099 0.6240458  0.61641221 0.6240458 ]
KNN Average Score: 62.75%


/Users/saivinay/Library/Python/3.9/lib/python/site-packages/sklearn/linear_model/_line
```

- **Logistic Regression Average Score:** 60.53%

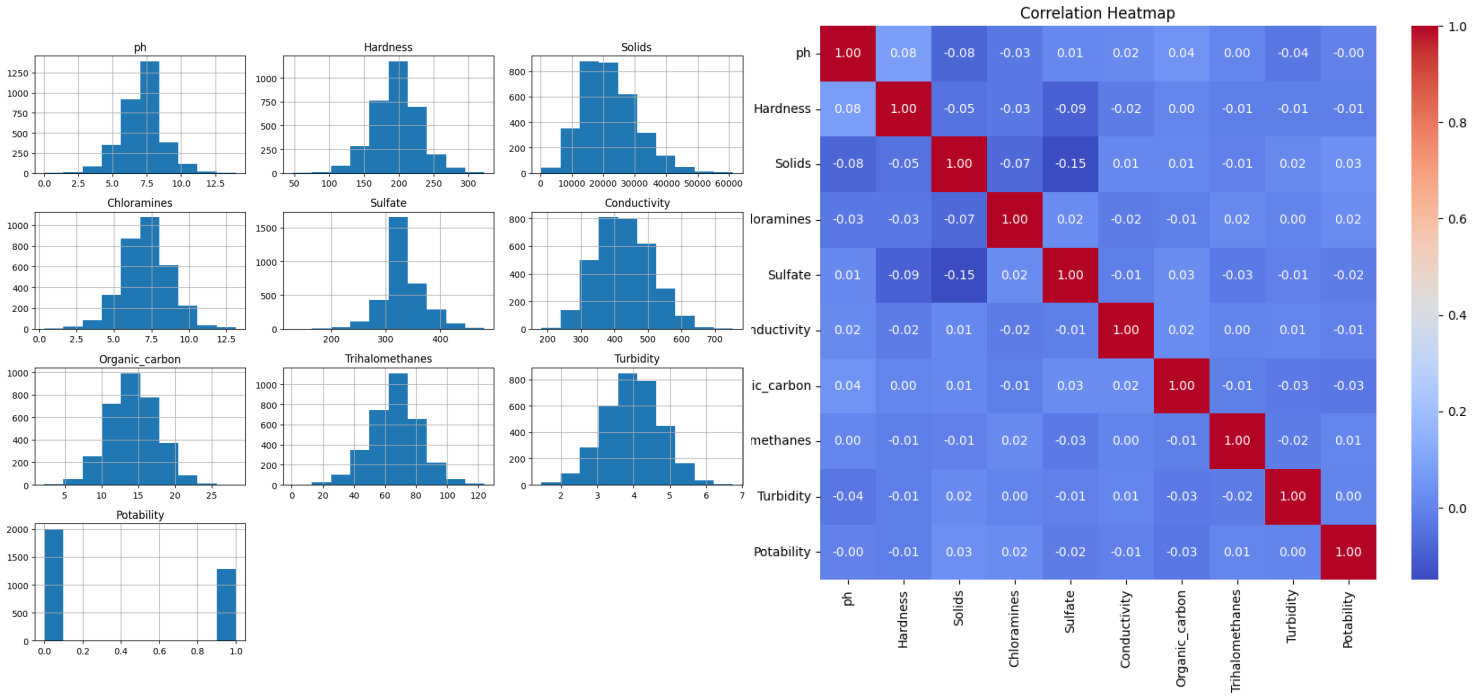
```
Model 1: Logistic Regression
Accuracy: 0.6280487804878049
              precision    recall  f1-score   support

           0       0.63      1.00      0.77       412
           1       0.00      0.00      0.00       244

    accuracy                           0.63       656
   macro avg       0.31      0.50      0.39       656
weighted avg       0.39      0.63      0.48       656
```

- **KNN Average Score:** 62.75%

```
Model 2: K-Nearest Neighbors (KNN)
Accuracy: 0.6280487804878049
              precision    recall  f1-score   support

           0       0.69      0.75      0.72       412
           1       0.50      0.42      0.46       244

    accuracy                           0.63       656
   macro avg       0.59      0.59      0.59       656
weighted avg       0.62      0.63      0.62       656
```

- **Random Forest Average Score:** 66.68%

```
...   Model 3: Random Forest
      Accuracy: 0.6814024390243902
                   precision    recall  f1-score   support

                0       0.70      0.86      0.77       412
                1       0.62      0.38      0.47       244

         accuracy                           0.68       656
        macro avg       0.66      0.62      0.62       656
     weighted avg       0.67      0.68      0.66       656
```



4.2. Final Performance on Test Set

Model 1: Logistic Regression

- **Accuracy:** 62.8%
- **Classification Report Analysis:** This model performed very poorly. It achieved a 0.00 recall and f1-score for the "Potable" (Class 1) category, meaning it failed to correctly identify *a single* potable water sample. It simply predicted every sample as "Not Potable" (Class 0), making it useless for this problem.

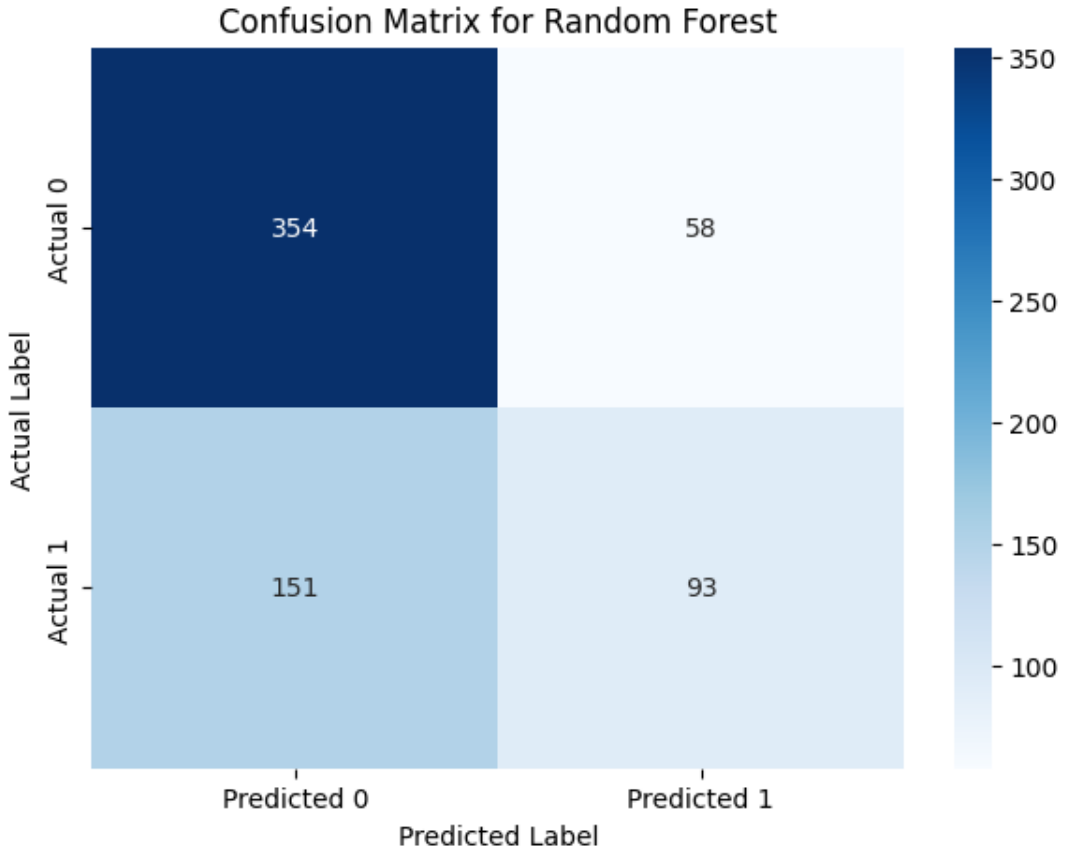
Model 2: K-Nearest Neighbors (KNN)

- **Accuracy:** 62.8%
- **Classification Report Analysis:** The KNN model showed a slight improvement, with an f1-score of 0.46 for Class 1. However, its overall accuracy is low, and its ability to identify potable water is still unreliable.

Model 3: Random Forest

- **Accuracy:** 68.1%
- **Classification Report Analysis:** Random Forest was the best-performing baseline model. However, it is still heavily biased by the imbalanced data. It was very good at identifying "Not Potable" water (Recall=0.86) but very poor at identifying "Potable" water (Recall=0.38). This means it missed 62% of all safe-to-drink water samples, classifying them as unsafe.

Analysis of Confusion Matrix (Random Forest): The confusion matrix for the Random Forest model confirmed the findings from the classification report. It showed a high number of True Negatives (correctly identifying "Not Potable") but also a very high number of False Negatives (incorrectly labeling "Potable" water as "Not Potable"). This bias makes the model unsuitable for practical use in its current state.



5. Literature Review

5.1. Status of Existing Work

Yes, a significant amount of work on this exact topic already exists. The "Water Quality / Potability" dataset from Kaggle is a popular and standard benchmark used in many academic papers and projects to compare the performance of different machine learning models. Our project, which involved cleaning missing data with the mean and testing baseline models, replicates the initial steps seen in much of this existing research.

5.2. Accuracy of Existing Work

Our baseline project achieved an accuracy of ~68.1% with the Random Forest model. This is a common baseline result.

Existing research papers achieve significantly higher accuracies, often in the 80% to 99% range. This is accomplished by applying more advanced techniques to fix the dataset's flaws. For example:

- Some studies achieve an accuracy of 99.5% using the XGBoost algorithm after applying advanced preprocessing.
- Other studies report 89% accuracy with a tuned Random Forest model after properly balancing the dataset.

The large difference in accuracy is not just from using different models, but from solving the core data-quality issues before training.

5.3. Summary of Existing Work

Predicting water potability using machine learning is a well-established research area aimed at automating and improving water quality monitoring. A review of the literature shows a clear and consistent methodology:

- **Models:** Most papers compare a standard set of classification algorithms, including Logistic Regression, K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Random Forest, and XGBoost.
- **Challenges:** The research unanimously identifies two critical challenges with this dataset:
    1. **Missing Data:** The `ph`, `Sulfate`, and `Trihalomethanes` columns contain a significant number of missing values that must be handled.
    2. **Imbalanced Data:** The dataset is "imbalanced," with a majority of samples belonging to the "Not Potable" (Class 0) category. This biases models, making them poor at identifying the "Potable" (Class 1) minority.

- **Conclusion:** The consensus is that ensemble models, specifically Random Forest and XGBoost, are the best performers. They consistently achieve the highest accuracy, especially after the data imbalance is corrected using a technique called **SMOTE**.

5.4. Identified Research Gaps (What Has Not Been Done)

While many papers have solved the accuracy problem on this static dataset, they point to several "gaps" where future work is needed. Our project is a baseline, while the gaps represent advanced, real-world extensions.

- **Gap 1:** No one has used this to predict potability from live, real-time sensor data.
- **Gap 2:** No one has built a system that recommends a fix (like "add chlorine"), it only predicts "safe/unsafe".
- **Gap 3:** No one has made a simple, user-friendly tool for the public; all the explanations are still for other data scientists.

6. Project Status and Future Work

6.1. Phase 1: Baseline Analysis (Completed)

- We took the messy, raw dataset.
- We did a simple cleaning (using `mean()`).
- We trained the standard models.
- We got a baseline accuracy score (~68.1%).
- We identified the first, most obvious problem: Our model is biased because the data is imbalanced.

6.2. Phase 2: Proposed Next Steps

Step 1: Fix the Data Imbalance with SMOTE

- **The Problem:** Our training data is "imbalanced" (it has more Class 0 than Class 1), so our model is biased.
- **The Plan:** We will apply a technique called **SMOTE (Synthetic Minority Over-sampling Technique)** to our training data.
- **What it does:** SMOTE will create new, artificial "Potable" (Class 1) samples, giving us a perfectly balanced 50/50 training set. This is the main technique used in the advanced research papers.

Step 2: Re-Train and Re-Test Our Models

- **The Plan:** We will re-train all three of our models (Logistic Regression, KNN, and Random Forest) on this new, balanced data.
- **The Test:** We will then test these new models on the original 20% test set (which we leave unchanged, as it represents the real world).

Step 3: Compare the Results

- **Expected Outcome:** We expect to see a significant improvement in our overall accuracy.
- **What to look for:** We will check the Classification Report. We should see a much higher **recall and f1-score for Class 1 (Potable)**. This will prove our new model can actually find the safe-to-drink water, which our old model could not do.

7. Conclusion

This initial project phase successfully established a baseline performance for water potability prediction. The 68.1% accuracy achieved by the Random Forest model, while the highest of the three, is misleading. A deeper analysis of the classification report and confusion matrix revealed a critical flaw: the model is heavily biased by the imbalanced dataset and is largely incapable of identifying potable water.

This finding aligns perfectly with the consensus in existing literature. The clear and immediate next step is to address the class imbalance. The plan for Phase 2 is to implement the SMOTE technique to create a balanced training set, which is expected to dramatically improve the model's recall for the "Potable" class and create a truly effective and reliable predictive tool.