Swinburne University of Technology

Faculty of Computer Science, Arificial Intelligence

Portfolio Assessment 1: "Hello Machine Learning for Engineering"

Author - ID:
Trung-Hieu Nguyen
103488337

Lecturer:
Dr. Trung Luu

Studio from 1 to 2

Ho Chi Minh, Vietnam

September 15, 2024

Contents

Chapter 1

Introduction

1.1 Motivation

As a final-year student with a strong passion for applied Computer Vision in Biomedical Engineering and Healthcare, I have interest in advancing models that can positively impact patient care and medical diagnosis. My latest project was about the topic "A deep learning approach to embryo quality assessment".

Initially, I explored using the Breast Cancer Diagnosis dataset, provided by Studio 1, for my project. However, upon reviewing it, I found that the dataset lacked essential information, such as the column names, which made it challenging to interpret and utilize effectively for machine learning tasks. Due to these limitations, I decided to switch to the Water Potability dataset. This dataset catched my interest because it aligns closely with my focus on healthcare, as clean water is a fundamental component of public health. Additionally, the dataset size was relatively larger than the others and the annotations are quite clear, making it an ideal choice for a preliminary project to warm up my course.

1.2 Dataset

1.2.1 General

The dataset is available at Kaggle. This dataset includes measurements and evaluations of water quality focused on potability, which refers to the suitability of water for human consumption. Its main purpose is to offer insights into various water quality parameters and help assess whether the water is safe to drink. Each entry in the dataset corresponds to a water sample with specific characteristics, and the "Potability" column signifies if the water is fit for consumption.

1.2.2 Variable Description

Target Variable

- **Potability:** Potability refers to whether the water is safe for drinking. A value of 1 indicates potable (safe) water, while a value of 0 means the water is not suitable for consumption.

Predictors (Input Variables)

- **pH:** The pH level is crucial for assessing the acid-base balance of water and indicates whether the water is acidic or alkaline. Organization et al. (2007), suggests a permissible pH range between 6.5 and 8.5.
- **Hardness:** Water hardness is largely caused by calcium and magnesium salts that dissolve as water passes through geological materials. The duration of contact with these materials determines the hardness level. Organization et al. (2010), suggests a permissible Hardness range between $10mg/L$ and $500mg/L$.
- **Solids (Total dissolved solids - TDS):** Water can dissolve various inorganic and organic minerals or salts such as potassium, calcium, sodium, and others. These dissolved substances affect the taste and appearance of the water. High TDS levels suggest the water is highly mineralized. The excellent TDS starts at lower than $300mg/L$, with a maximum limit of $1200mg/L$ for drinking water according to Organization et al. (2003).
- **Chloramines:** Chlorine and chloramine are common disinfectants in public water systems, with chloramines formed when ammonia is added to chlorine during water treatment. Chlorine concentrations up to $5mg/L$ and some at levels as low as $0.3mg/L$ are considered safe for drinking water according to Organization et al. (2004a).
- **Sulfate:** Sulfates are naturally occurring and found in minerals, soil, and rocks, as well as air, plants, and food. Their main use is in the chemical industry. Sulfate concentrations are typically 3 to $30mg/L$ in freshwater, though concentrations can be higher in some regions. Many diseases are commonly reported to be experienced by people consuming drinking-water containing sulfate in concentrations exceeding $600mg/L$ according to Organization et al. (2004b).
- **Conductivity:** Pure water does not conduct electricity well but gains conductivity with increasing ion concentration. Conductivity depends on the dissolved solids in the water, as they facilitate the flow of electric current. According to

WHO standards, water's electrical conductivity should not exceed 1660 $\mu S/cm$ according to C.O.B. and Nyiatagher (2009).

- **Organic_Carbon:** Total Organic Carbon (TOC) comes from decaying natural organic matter or synthetic sources. TOC measures the total carbon in organic compounds present in pure water. Moore (1998) recommends that TOC levels in treated drinking water should be below $2mg/L$ and below $4mg/L$ in source water used for treatment.

- **Trihalomethanes:** THMs are chemicals that can be found in chlorinated water. Their concentration depends on the organic material in the water, the chlorine needed for treatment, and the water temperature. According to Organization et al. (2004c), the majority of treatment facilities had relatively low total THM levels (<$50\mu g/L$), while a small number had relatively high levels (>$100\mu g/L$). Thus, the report assume safe range for THM level shoud be lower than $100\mu g/L$.

- **Turbidity:** Turbidity measures the presence of solid particles suspended in water, affecting its light-reflecting properties. It is often used to gauge the quality of waste discharge. According to Organization et al. (2017), turbidity is ideally <$1NTU$, although this may be difficult in many supplies where household water treatment is necessary to ensure the safety of drinking-water. In such cases, the aim should be to keep turbidities below $5NTU$.

1.2.3 Data

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3276 entries, 0 to 3275
Data columns (total 10 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   ph              2785 non-null   float64
 1   Hardness        3276 non-null   float64
 2   Solids          3276 non-null   float64
 3   Chloramines     3276 non-null   float64
 4   Sulfate         2495 non-null   float64
 5   Conductivity    3276 non-null   float64
 6   Organic_carbon  3276 non-null   float64
 7   Trihalomethanes 3114 non-null   float64
 8   Turbidity       3276 non-null   float64
 9   Potability      3276 non-null   int64
```

```
dtypes: float64(9), int64(1)
memory usage: 256.1 KB
```

Key points:

- The dataset has 3,276 rows, meaning there are 3,276 water samples, each representing one record of water quality measurements.
- pH (2,785 rows), Sulfate (2,495 rows), and Trihalomethanes (3,114 rows) columns contain missing values, which may require handling (e.g., imputation or removal) during data preprocessing. The remaining columns are complete, meaning no missing values.
- There was no duplicated rows or columns.
- The majority of the columns are of type float64 (representing continuous variables). Only one column, Potability, is of type int64 (representing the binary classification: 1 for potable, 0 for non-potable).

Chapter 2

Exploratory Data Analysis (EDA)

2.1 Non-processing

```
water_df.describe()
```

✓ 0.0s

[75]

	ph	Hardness	Solids	Chloramines	Sulfate	Conductivity	Organic_carbon	Trihalomethanes	Turbidity
count	2785.000000	3276.000000	3276.000000	3276.000000	2495.000000	3276.000000	3276.000000	3114.000000	3276.000000
mean	7.081738	196.392423	21957.112200	7.121794	333.867862	426.129974	14.283462	66.415219	3.966612
std	1.549632	32.017189	8592.820397	1.544126	40.450271	80.564144	3.288367	15.990523	0.776409
min	3.139631	117.125160	320.942611	3.146221	229.323489	191.647579	5.328026	23.605130	1.848797
25%	6.093092	176.850538	15666.690297	6.127421	307.699498	365.734414	12.065801	55.844536	3.439711
50%	7.036752	196.967627	20927.833607	7.130299	333.073546	421.884968	14.218338	66.622485	3.955028
75%	8.062066	216.667456	27332.762127	8.114887	359.950170	481.792304	16.557652	77.337473	4.500320
max	11.015527	276.392834	44831.869873	11.096086	438.326179	655.879140	23.295427	109.576879	6.091233

Figure 2.1: Summary statistics

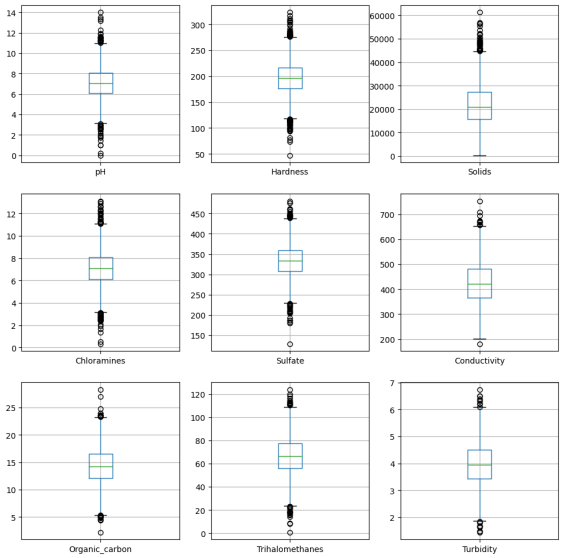

Figure 2.2: Boxplot of each feature

According to Figure 2.1 and 2.2, there are clearly errors within this dataset. There

are outliers needed to be removed, and some missing values. Besides, there is no duplicated data.

2.2 Preprocessing

When dealing with datasets that contain many features, the process of imputation can introduce significant risks in a classification task. Some risks I can think of are bias, distortion of relationships and potential outliers. Therefore, I decide to drop all rows having missing values (NaN, or Null value).

Besides, I also remove outliers using IQR method. Consequently, the dataset is in a better condition for classification tasks.

2.3 Univariate Analysis

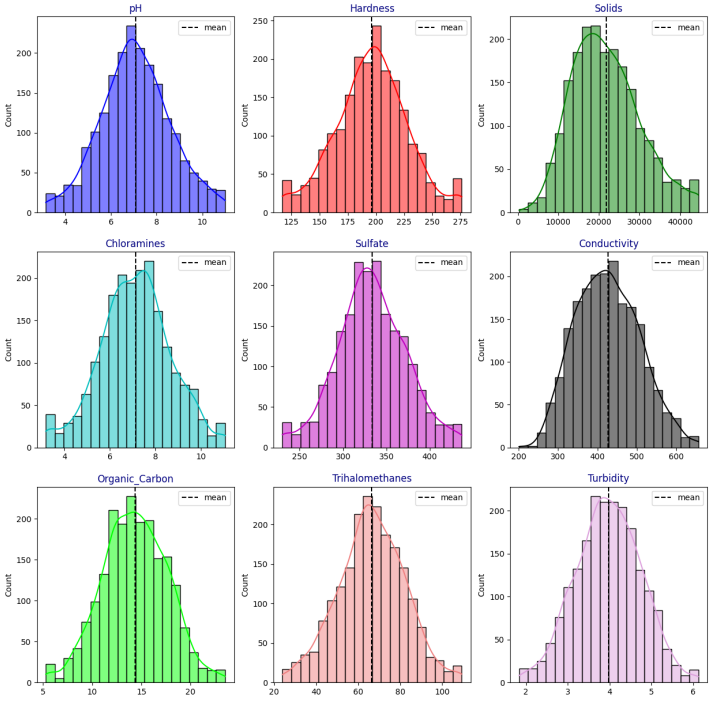

Figure 2.3: Distribution of each feature

Univariate Observation (Figure 2.1, 2.2, and 2.3):

- **Normality**: All features exhibit close-to-normal distributions.
- **Data Variability:**

- **High:** Hardness, Conductivity, Trihalomethanes, Conductivit, Trihalomethanes, Turbidity show high variability, both in terms of range and standard deviation.
- **Moderate:** pH, Solids, Chloramines, Sulfate and Organic Carbon exhibit moderate variability.

- **Skewness**: Only Solids and Conductivity appear to be slightly right skewness, while the remaining distributions are in good shape.

2.4 Multivariate Analysis

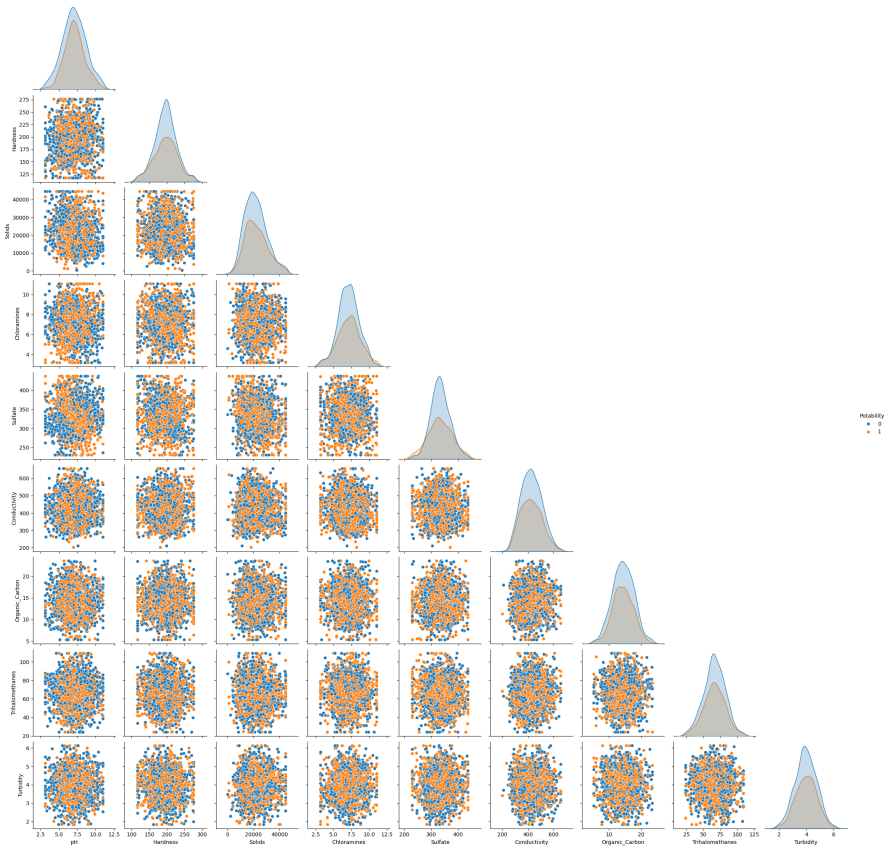

Figure 2.4: Pairplot for Correlation

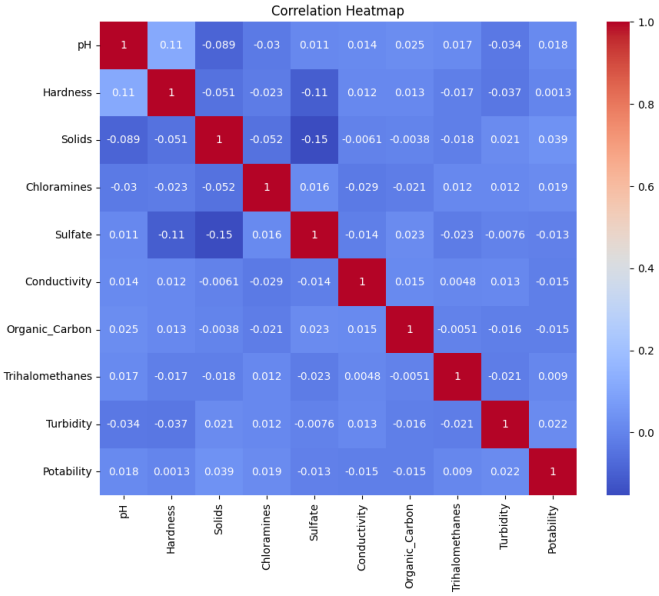

Figure 2.5: Correlation Heatmap

Multivariate Observation (Figure 2.4 and 2.5):

- **Diagonal Analysis:** The independent variables are distributed fairly widely, with no extreme skewness.
- **Off-diagonal Analysis:** The relationships between the independent variables are weak, with most scatter plots showing random distributions, suggesting that the independent variables are not highly correlated with each other.
- **Potability Relationships:** There is no obvious relationship between the individual independent attributes and the target variable "Potability". The overlap between potable and non-potable water samples across all attributes. Simple linear relationships may not suffice to predict "Potability".
- **Hisplot of Features by "Potability"** According to Figure 2.4, we can ensure that all features, and our target are in normal distributions.

2.5 Potability Distribution

The potability column is the target variable for classification tasks. The distribution of potable and non-potable water samples is in Figure 2.6.

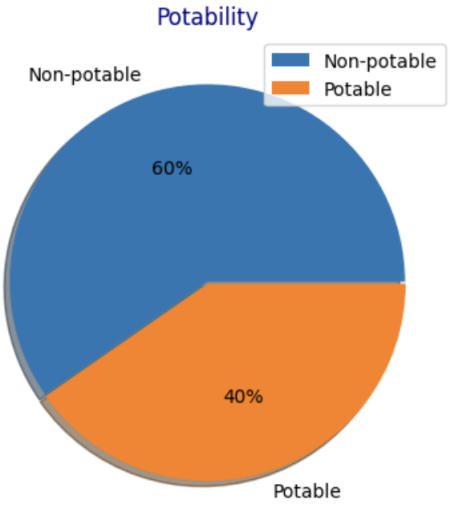

Figure 2.6: Pie chart of target distribution

Observation: The dataset is imbalanced, with more non-potable (1,200 samples: 60%) than potable (811 samples: 40%) drinks. This imbalance needs to be addressed during model building.

2.6 Ground-truth Development

Ground-truth data refers to accurate, real-world data that serves as a benchmark for evaluating model predictions. In the context of the Water Potability dataset, ground-truth development involves verifying the accuracy of the potability labels and ensuring that the independent features reflect true water quality measurements. To ensure reliable predictions, it is crucial that the potability labels (0 for non-potable and 1 for potable) are accurate. Therefore, ground-truth can be developed when comparing model predictions with results of 100 people drinking samples with certain input values.

In terms of label target variable, since the dataset has already categorized the potability of water using 1 for potable, and 0 for non-potable, there is no need for further class labelling.

2.7 EDA Summary

- We need to aware that the dataset is imbalanced, and this model needs to predict a categorical value. Therefore, multiple model evaluation metrics such as Accuracy, F1-score, Precision, Root Mean Squared Error (RMSE), and R^2 are recommended.

- Handle missing values for columns like "pH", "Sulfate", and "Trihalomethanes". It requires imputation.
- Features like "Solids" and "Conductivity" have values in much higher ranges compared to other columns like "pH" and "Organic_Carbon". This may affect the model performance where algorithms are biased towards features with larger numerical values. Normalization is recommended.
- Most features have very low correlations with the target variable (Potability). This indicates that no single feature is strongly predictive of whether the water is potable. The pair having lowest negative correlation is Solids & Organic Carbon. It can be used to composite a new feature, which can reveal hidden relationship.
- The highest correlations between features are also relatively weak, with most being close to 0. This implies there is little multicollinearity in the dataset, meaning the features are relatively independent from each other. For example:
  - "Sulfate" and "Solids" have a negative correlation of -0.15. Their interaction could capture a more complex relationship
  - "Hardness" and "Sulfate" show a weak negative correlation of -0.11. Creating an interaction term for these two might be insightful.
- The correlation between Conductivity and Potability, or Organic_Carbon and Potability are quite similar. And their correlation the other features are quite low. Therefore, a drop of either of them might provide better model performance (eliminate multicollinearity or redundancy).

Chapter 3

Feature Engineering & Selection

3.1 New Features

3.1.1 Binning (Discretization)

```
# Define the binning criteria for continuous features based on domain knowledge
bins_pH = [3, 6, 7, 8, 11]
labels_pH = ['Low', 'Best', 'Moderate', 'Risky']

# Apply binning and encode to numerical values
df['pH_binned'] = pd.cut(df['pH'], bins=bins_pH, labels=labels_pH, include_lowest=True)
df['pH_binned'] = df['pH_binned'].astype('category').cat.codes
✓ 0.0s
```

Figure 3.1: Code for classifying pH

I tend to apply this on feature with low variance and not contain long decimal values such as pH (Figure 3.1). This binning technique also require the bins to be divided equally (Figure 3.2).

- pH: Binning pH into categories such as "low" (3-6), "best" (6-7), "moderate" (7-8) and "risky" (8-11).

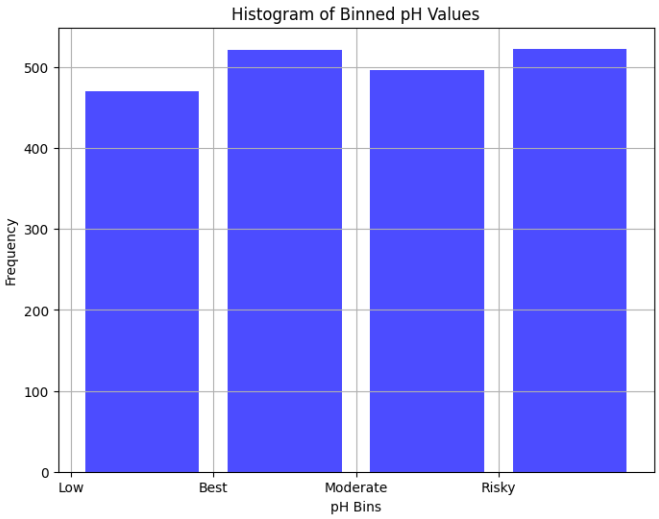

Figure 3.2: Barchart of binned pH values

3.1.2 Normalization

```
# Initialize the MinMaxScaler
scaler = MinMaxScaler(feature_range=(0, 1))

# Select only float64 columns to scale, excluding 'Potability' or other categorical columns
float_cols = df.select_dtypes(include=['float64']).columns
float_cols = float_cols.drop('Potability', errors='ignore')  # Ensure 'Potability' is excluded if present

# Apply MinMax Scaling to the float columns
df[float_cols] = scaler.fit_transform(df[float_cols])

print(df.columns)

# Check if 'pH' column exists before dropping to avoid an error
if 'pH' in df.columns:
    df = df.drop('pH', axis=1)

df = df[[col for col in df.columns if col != 'Potability'] + ['Potability']]

# Save the dataframe to a CSV file
df.to_csv('normalized_water.csv', index=False)
print('Done, the new dataset has been saved.')
✓ 0.0s
```

Figure 3.3: Code for MinMaxScaler

Normalization with MinMaxScaler: MinMax Scaler shrinks the data within the given range,(0 to 1). It transforms data in every feature, except for pH_binned (Figure 3.3).

3.1.3 Interaction Features

```python
# Calculate covariance between pairs of features and create new features
df['Solids_Sulfate'] = df['Solids'].cov(df['Sulfate'])
df['Sulfate_Hardness'] = df['Hardness'].cov(df['Sulfate'])
df['Solids_Organic_Carbon'] = df['Solids'].cov(df['Organic_Carbon'])

# Reorder the columns to ensure 'Potability' is at the end
df = df[[col for col in df.columns if col != 'Potability'] + ['Potability']]

# Save the updated dataframe to a new CSV file
df.to_csv('features_water.csv', index=False)

print('New features created and saved to "features_water.csv".')
```

✓ 0.0s

Figure 3.4: Code for New Features

Interaction features capture relationships between two or more variables. This can be useful when certain combinations of features have a greater influence on the target variable than each feature independently. I tend to apply this on extremely low correlated (independant) features or highly correlated ones (Figure 3.4):

- Solids and Sulfate have the highest negative correlation (-0.15).
- Hardness and Sulfate also show a moderate negative correlation (-0.11).
- Solids and Organic Carbon is low (-0.0038).

3.2 Selection

```python
# List of columns to eliminate
columns_to_drop = ['Trihalomethanes', 'Organic_Carbon']

# Drop the specified columns from the dataframe
df = df.drop(columns=columns_to_drop, axis=1)

# Save the updated dataframe to a new CSV file
df.to_csv('selected_feature_water.csv', index=False)

print('Selected columns removed. The updated dataset has been saved to "selected_feature_water.csv".')
```

[2048] ✓ 0.0s Python

```
Selected columns removed. The updated dataset has been saved to "selected_feature_water.csv".
```

```python
df = pd.read_csv('converted_water.csv')

# List of columns to eliminate
columns_to_drop = ['Trihalomethanes', 'Conductivity']

# Drop the specified columns from the dataframe
df = df.drop(columns=columns_to_drop, axis=1)

# Save the updated dataframe to a new CSV file
df.to_csv('selected_converted_water.csv', index=False)

print('Selected columns removed. The updated dataset has been saved to "selected_converted_water.csv".')
```

[2049] ✓ 0.0s Python

```
Selected columns removed. The updated dataset has been saved to "selected_converted_water.csv".
```

Figure 3.5: Code for Dropping Features

Since none of the features show a strong correlation with Potability, the decision to drop features would be based on redundancy or minimal contribution to the model (Figure 3.5). Some columns to drop:

- **Trihalomethanes:** The correlation of Trihalomethanes with Potability is low (0.009), and its correlation with other features is also low (-0.0051 with Organic_Carbon, 0.0048 with Conductivity). This feature might not contribute much to model performance.
- **Organic_Carbon or Conductivity**: As presented above, the correlation of these two features to target is the same. So, the elimination of either of them may benefit the model by avoiding redundancy or bias. For 'selected_feature_water.csv', I drop Organic_Carbon, while I drop Conductivity on 'selected_converted_water.csv'.

Chapter 4

Training & Results

4.1 Training

```python
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import precision_score, recall_score, f1_score

# List of datasets (replace these paths with the actual paths to your datasets)
dataset_paths = ['converted_water.csv', 'normalized_water.csv', 'features_water.csv', 'selected_feature_water.csv', 'selected_converted_water.csv']

# Initialize an empty list to store results
results = []

# Loop through each dataset
for dataset_path in dataset_paths:
    # Load the dataset
    df = pd.read_csv(dataset_path)

    # Define features (X) and target (y)
    X = df.drop('Potability', axis=1)  # Features
    y = df['Potability']  # Target

    # Split the dataset into 70% training and 30% validation
    X_train, X_val, y_train, y_val = train_test_split(X, y, test_size=0.3, random_state=1)

    # Initialize and train a Decision Tree model
    decision_tree_model = DecisionTreeClassifier(random_state=1)
    decision_tree_model.fit(X_train, y_train)

    # Make predictions on the validation set
    y_pred = decision_tree_model.predict(X_val)

    # Evaluate the model
    # Precision
    precision = precision_score(y_val, y_pred)

    # Recall
    recall = recall_score(y_val, y_pred)

    # F1-Score
    f1 = f1_score(y_val, y_pred)

    # Append the results into a list as a dictionary
    results.append({
        'Dataset': dataset_path,
        'Precision': precision,
        'Recall': recall,
        'F1 Score': f1
    })

# Convert the list of results to a DataFrame
results_df = pd.DataFrame(results)

# Save the results to a CSV file
results_df.to_csv('results_water.csv', index=False)

print('Model evaluation results saved to "results_water.csv".')
```

✓ 0.1s Python

Figure 4.1: Code for Model Training

4.1.1 Dataset Preparation

Input: A list of dataset file paths (`dataset_paths`), which includes datasets:

- `converted_water.csv`: all features without normalisation and without composite features
- `normalized_water.csv`: all features with normalisation and without composite features
- `features_water.csv`: all features with normalisation and containing composite features
- `selected_feature_water.csv`: selected features with normalisation
- `selected_converted_water.csv`: selected features without normalisation

4.1.2 Feature and Target Separation

Steps: The features (**X**) are selected by dropping the `Potability` column, which represents the target variable. The target variable (**y**) is assigned as the `Potability` column.

Output: Two datasets are created:

- `X` contains the features.
- `y` contains the target variable.

4.1.3 Data Splitting (Train/Validation Split)

Steps: The dataset is split into training and validation sets using the `train_test_split` function with a 70% training and 30% validation ratio. The `random_state` parameter is set to 1 to ensure reproducibility of the splits. The **Output** is:

- `X_train`: 70% of the data for training the model.
- `X_val`: 30% of the data for validating the model.
- `y_train`: Target variable for the training data.
- `y_val`: Target variable for the validation data.

4.1.4 Decision Tree Model Initialization, Training, & Prediction

Steps: A **DecisionTreeClassifier** is initialized with `random_state=1` for consistent results. This model will take `X_train` (training features) and `y_train` (training labels) as inputs. The decision tree model is trained using the `.fit()` method on the training data. Predictions are made on the validation set using the trained model's `.predict()` method. The final output will be predicted labels for the validation set (`y_pred`).

4.1.5 Model Evaluation Metrics

The program will then provide evaluation metrics using `sklearn.metrics` library. Some metrics I found useful for this **imbalanced classification task** are:

- **Not using Accuracy:** For imbalanced datasets like `water potability`, accuracy can be misleading. This is because the model could predict the majority class (non-potable) most of the time and still appear to have high accuracy without actually performing well on the minority class (potable).
- **Precision:** A high precision score indicates that when the model predicts a positive class, it is correct most of the time. In other words, it answers the question: "When the model predicts that water is safe, how often is it correct?". In the real world, labeling unsafe water as safe to drink can have serious health consequences. Therefore, ensuring that the positive predictions (potable) are accurate is vital.
- **Recall:** A high recall means the model correctly identifies most of the positive instances in the dataset, minimizing the number of false negatives. In other words, it answers the question: "Out of all the safe water samples, how many did the model correctly classify as safe?". It is important when we do not want to waste any water resources.
- **F1-score:** The F1 Score balances precision and recall, giving an overall picture of the model's performance when both false positives (incorrectly labeling unsafe water as safe) and false negatives (failing to identify safe water) are equally important.

4.1.6 Storing Results

Steps: Each evaluation result is appended to the `results` list as a dictionary. The dictionary contains the dataset name and the evaluation metrics. After that, the list of

results dictionary is converted into a Pandas DataFrame. The DataFrame is saved to a CSV file named `results_water.csv`.

4.2 Results

The table below shows the evaluation results for Precision, Recall, and F1 Score across five datasets. These metrics were computed to assess the performance of a Decision Tree model on the water potability classification task.

Dataset	Precision	Recall	F1 Score
converted_water.csv	0.4756	0.5132	0.4937
normalized_water.csv	0.4958	0.5132	0.5043
features_water.csv	0.4980	0.5351	0.5159
selected_feature_water.csv	0.5232	0.5439	0.5333
selected_converted_water.csv	0.5216	0.5307	0.5261

Table 4.1: Precision, Recall, and F1 Score for each dataset

A plot (Figure 4.2) has also already been made for data visualisation.

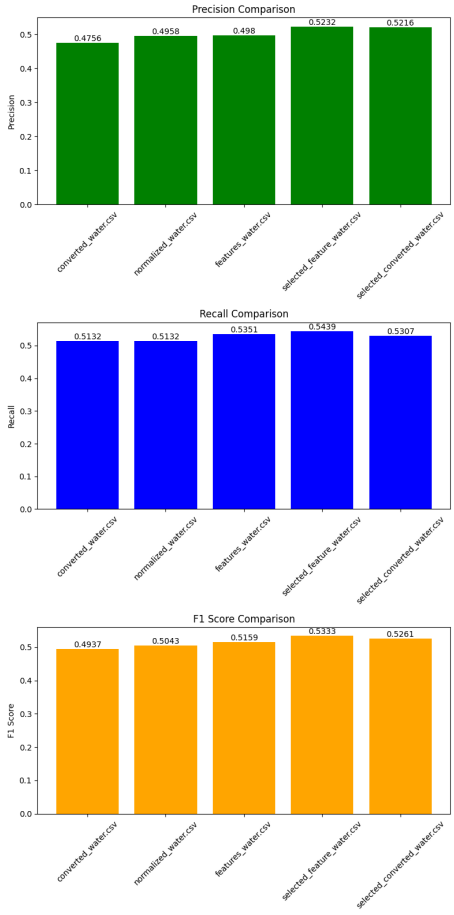

Figure 4.2: Plots of Precision, Recall, and F1-Score across 5 datasets

4.3 Comparison of Datasets

The following analysis compares the performance of the five datasets based on Precision, Recall, and F1 Score:

4.3.1 Precision

- The highest precision is achieved by `selected_feature_water.csv` with a value of **0.5232**, followed closely by `selected_converted_water.csv` at `0.5216`.
- The lowest precision is seen in `converted_water.csv` with **0.4756**, indicating the model struggles with making confident positive predictions on this dataset.

4.3.2 Recall

- `selected_feature_water.csv` also shows the highest recall at **0.5439**, meaning it correctly identifies the most positive samples.
- Both `converted_water.csv` and `normalized_water.csv` have the lowest recall at **0.5132**.

4.3.3 F1 Score

- The highest F1 score is observed in `selected_feature_water.csv` at **0.5333**, representing the best balance between precision and recall.
- `converted_water.csv` has the lowest F1 score at **0.4937**, indicating a weaker balance between precision and recall.

4.4 Conclusion

Datasets that involve feature selection (1st-place: `selected_feature_water.csv`) and (2nd-place: `selected_converted_water.csv`) show improved performance, highlighting the importance of feature selection in achieving better classification outcomes. For each task of feature engineering and selection, the result gets better. The weaker performance of `converted_water.csv` suggests that further data processing and feature refinement are essential to improve classification accuracy in the water potability task.

Chapter 5

Appendix

For marking and reference purposes, I provide a link to my GitHub repository, which contains the portfolio requirements and my source code, and the Dataset.

- **GitHub Repository**: link
- **Kaggle Dataset**: link

Bibliography

C.O.B., O. and Nyiatagher, T. (2009). Physico-chemical quality of shallow well-waters in gboko, benue state, nigeria. *Bulletin of the Chemical Society of Ethiopia*, 23.

Moore, D. R. J. (1998). Ambient water quality guidelines for organic carbon in british columbia. Technical report, Ministry of Environment, Lands and Parks (now Ministry of Water, Land and Air Protection), Victoria, BC. A report submitted to N.K. Nagpal, Ph.D., Contract Manager, Water Quality Section, Water Management Branch.

Organization, W. H. et al. (2003). Total dissolved solids in drinking-water: background document for development of who guidelines for drinking-water quality. Technical report, World Health Organization.

Organization, W. H. et al. (2004a). Monochloramine in drinking-water: background document for development of who guidelines for drinking-water quality. Technical report, World Health Organization.

Organization, W. H. et al. (2004b). Sulfate in drinking-water: background document for development of who guidelines for drinking-water quality. Technical report, World Health Organization.

Organization, W. H. et al. (2004c). Trihalomethanes in drinking-water: background document for development of who guidelines for drinking-water quality. Technical report, World Health Organization.

Organization, W. H. et al. (2007). ph in drinking-water revised background document for development of who guidelines for drinking-water quality. *World Health Organization: Geneva, Switzerland*.

Organization, W. H. et al. (2010). Hardness in drinking-water: background document for development of who guidelines for drinking-water quality. Technical report, World Health Organization.

Organization, W. H. et al. (2017). Water quality and health-review of turbidity: information for regulators and water suppliers. *World Health Organization: Geneva, Switzerland*.