**WEEK-5**

**Classification Analysis of Water Quality for Potability**

Name of the Team Lead: Lakshanya S

Name of the Team Members:

A Bhavya Sri

E Saikiran

J Sinduja

K Nanda Kishore Reddi

R Naveen

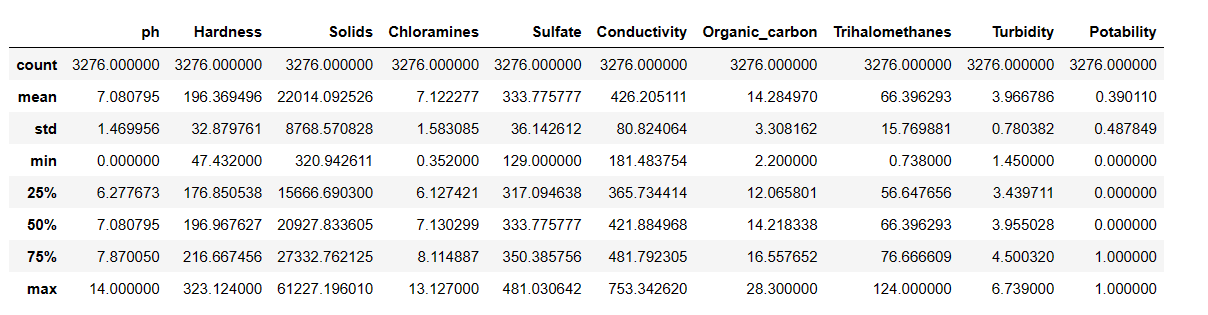
Submission Date:

22-12-2023

**Classification Analysis of Water Quality for Potability**

**1. Initial Exploration**

* Loaded the dataset and conducted an initial exploration using describe() and info().
* We ensured that there were no missing values in the dataset.



**2. Exploratory Data Analysis**

**2.1 Count Plot:**

The visualization, generated through Seaborn's `countplot`, provides a succinct depiction of the water potability distribution in the dataset. As indicated by the plot, the count of non-potable water samples (labeled as 0) ranges approximately from 0 to 2000, whereas the count for potable water samples (labeled as 1) spans approximately from 0 to 1250. This visual representation offers valuable insights into the relative abundance of potable and non-potable water samples.

A graph with a blue and orange rectangle

Description automatically generated

**2.2 Histograms:**

The grid of subplots displays histograms for each numerical feature and the binary 'Potability' variable in the dataset. These histograms provide a concise visual summary of the frequency and distribution of values for each feature, aiding in the identification of patterns and outliers within the data.

A diagram of different types of graphs

Description automatically generated with medium confidence

**2.3 Isolation of Non-Potable and Potable Water Subsets:**

Two subsets, 'NPWater' and 'PotableWater,' have been created through the application of logical filters on the main DataFrame ('df'). The 'NPWater' subset is formed by selecting rows where the 'Potability' column is assigned the value of 0, signifying non-potable water samples. Conversely, the 'PotableWater' subset includes rows where the 'Potability' column is assigned the value of 1, indicating potable water samples. These subsets serve as distinct datasets, allowing for targeted analysis and focused examination of water samples based on their potability classification.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**2.4 Histogram for Non-Potable Water Subset:**

The code generates histograms for each numerical feature in the 'NPWater' subset, providing a visual representation of the distribution of values for non-potable water samples. This visualization aids in understanding the spread and concentration of various water quality parameters within the subset.

A group of blue and white graphs

Description automatically generated

**2.5 Histogram for Potable Water Subset:**

This code generates histograms for each numerical feature in the 'PotableWater' subset, offering a visual summary of the distribution of values for potable water samples. The histograms provide insights into the spread and concentration of different water quality parameters within the subset.

A group of blue and white graphs

Description automatically generated

**3.Regression Model:**

**3.1 Train-Test Data Split:**

The code employs the `train\_test\_split` function to partition the dataset into training and testing sets. Features (`x`) and the corresponding target variable (`y`) are divided into `x\_train`, `x\_test`, `y\_train`, and `y\_test`. With a specified test size of 20% and a fixed random seed of 42, this split facilitates model evaluation on an independent dataset for robust performance assessment.

**3.2 Feature Binning for Categorical Representation:**

The code snippet transforms numerical features in the 'newData' DataFrame, including 'Solids,' 'Chloramines,' 'Sulfate,' 'Conductivity,' 'Organic\_carbon,' 'Trihalomethanes,' and 'Turbidity,' into categorical representations by applying binning. Binning is achieved by defining bins based on the minimum and maximum values of each feature, and categorizing data points into 'Low,' 'Medium,' or 'High' labels. This categorical representation enables a more intuitive understanding of the distribution and impact of these water quality parameters, serving as a pre-processing step for subsequent analyses or modeling.

A screenshot of a computer

Description automatically generated

**3.3 Model Comparison Using PyCaret:**

The code leverages PyCaret's `compare\_models` function to facilitate the comparison of multiple machine learning models. By calling `compare\_models()`, PyCaret automates the process of training and evaluating various classification algorithms on the provided dataset. This step is instrumental in identifying promising models based on performance metrics, allowing for an informed selection of the most suitable algorithm for the water potability prediction task.

A screenshot of a computer screen

Description automatically generated

Summarizing, the above parameters can be represented in a tabular format as follows:

The Evaluation Metrics are:

**Accuracy:** The ratio of correctly predicted instances to the total instances.

**AUC (Area Under the Curve):** The area under the Receiver Operating Characteristic (ROC) curve, which measures the model's ability to distinguish between classes.

**Recall:** The proportion of actual positive instances correctly predicted by the model.

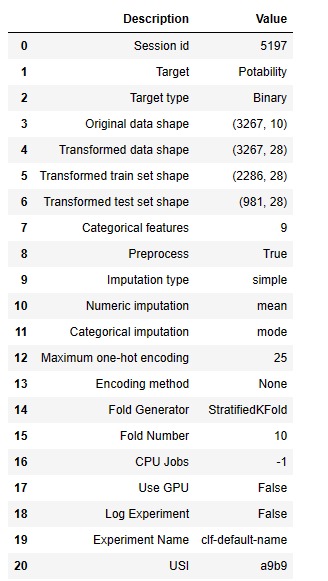
**Precision:** The ratio of correctly predicted positive observations to the total predicted positives.

**F1 Score:** The harmonic mean of precision and recall, providing a balance between the two metrics.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Model | Accuracy | AUC | Recall | Precision | F1 |
| Logistic Regression | 0.61 | 0.56 | 0.09 | 0.52 | 0.15 |
| Decision Tree | 0.56 | 0.54 | 0.22 | 0.40 | 0.28 |
| Random Forest | 0.58 | 0.57 | 0.25 | 0.44 | 0.31 |
| Gradient Boosting | 0.61 | 0.58 | 0.13 | 0.54 | 0.21 |
| K-Nearest Neighbors | 0.57 | 0.55 | 0.36 | 0.44 | 0.39 |

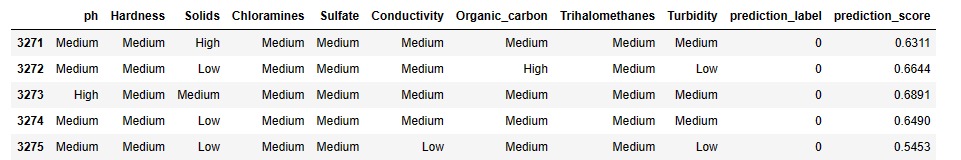
**3.4 Data Preparation for ML:**

The code selects relevant features ('x') and the target variable ('y') from the 'newData' dataset. Utilizing PyCaret's 'setup' function, the machine learning environment is initialized, automating preprocessing tasks like normalization and encoding. Categorical features are explicitly defined, streamlining the dataset for subsequent machine learning model training and evaluation.



**3.5 Model Prediction on New Data:**

The code employs the 'predict\_model' function to make predictions using the previously built best-performing model ('bm') on a subset of new data. Specifically, it predicts the 'Potability' outcome for the last few rows of the 'newData' DataFrame, excluding the target variable. This step allows for a quick assessment of how well the trained model generalizes to unseen data, providing insights into its predictive capabilities.



Here, the prediction label is the output obtained i.e., if the water is potable or not. The value 0 determines that the water sample is not potable and 1 determines that the given water sample is potable.

**4.Conclusion:**

In this project, we delved into the crucial realm of water potability analysis using machine learning techniques. Through comprehensive exploratory data analysis, we gained valuable insights into the distribution of key water quality metrics, addressing parameters like pH, hardness, and chloramines. Leveraging PyCaret, we systematically built and compared multiple classification models, identifying the most promising algorithm for predicting water potability. Our preprocessing steps included handling null values, visualizing data distributions, and feature engineering through numerical feature categorization. The division of the dataset into training and testing sets facilitated robust model evaluation, and the application of PyCaret significantly streamlined the modeling process. Binning certain features further enhanced interpretability. Overall, this project contributes to understanding the factors influencing water potability, offering valuable insights within the broader context of water resources and river systems.