NAAN MUDHALVAN

E2324 - GENERATIVAI

HYDROSAFE: ENHANCING WATER POTABILITY THROUGH PREDICTIVE ANALYTICS

KIRANKUMAR M

NM ID: 048ECB0D9F48C56BF304011FE331B325

MENTORED BY

KEERTHANA R

MADRAS INSTITUTE OF TECHNOLOGY, ANNA UNIVERSITY

PROBLEM STATEMENT:

- In the realm of public health and environmental safety, ensuring access to clean and potable water is paramount. However, conventional methods of water quality assessment often rely on labor-intensive and time-consuming manual inspections, leading to inefficiencies and potential oversights.
- Developing an automated solution for water potability prediction is imperative to streamline the monitoring process, mitigate risks, and safeguard public health. By harnessing advanced analytics and predictive modeling techniques, we aim to create a system capable of accurately assessing water quality, enabling timely interventions, and facilitating informed decision-making for water treatment facilities and public health authorities

OBJECTIVES:

- The objective is to create an automated solution for predicting water potability, using advanced analytics and predictive modeling.
- It aims to streamline water quality monitoring, mitigate risks, and safeguard public health by replacing manual inspections. This system will provide timely insights to identify contaminants, enabling proactive interventions and ensuring access to clean drinking water.

END USERS:

- **Water Quality Testing Laboratories:** Laboratories specializing in water quality analysis can utilize our solution to streamline the testing process and ensure accurate and timely assessments.
- **Industrial Facilities:** Manufacturing plants, refineries, and industrial sites that rely on water for their operations can implement our system to monitor water quality and compliance with environmental regulations.
- **Agricultural Sector:** Farms and agricultural enterprises can benefit from our solution to assess the quality of irrigation water and prevent contamination of crops and livestock.
- **Residential Consumers:** Individuals and households concerned about the quality of their drinking water can access our solution to monitor water safety and make informed decisions about filtration and purification systems.

SOLUTION AND VALUE PROPOSITION:

Solution:

Our solution employs machine learning algorithms, including Random Forest, XGBoost, and neural networks, to analyze water quality data and predict water potability. By training on a diverse dataset containing various water parameters, our model can effectively classify water samples as potable or non-potable based on their characteristics.

Value proposition:

- **Accuracy:** High accuracy rates ensure reliable predictions of water potability.
- **Efficiency:** Automated assessment saves time and resources for water treatment facilities.
- **Adaptability:** Versatile and applicable across different water sources and treatment processes.
- **Early Detection:** Enables proactive measures by detecting water contamination early.
- **Public Health:** Enhances public health outcomes by ensuring safe drinking water and reducing waterborne diseases.

THE WOW IN YOUR SOLUTION:

Our solution excels in employing cutting-edge machine learning techniques to predict water potability with exceptional accuracy and efficiency. Here's why it stands out

- **Advanced Technology:** Leveraging state-of-the-art algorithms and neural networks, our model delivers precise predictions of water quality.
- **Robust Data Analysis:** Thorough data preprocessing ensures dataset integrity, enabling reliable predictions across diverse water sources.
- **Adaptability:** Our solution seamlessly adjusts to different treatment processes, making it versatile and applicable in various contexts.
- **Real-Time Monitoring:** With real-time analysis capabilities, our solution enables proactive measures to address water contamination swiftly.
- **Public Health Impact:** By ensuring safe drinking water and reducing waterborne diseases, our solution significantly improves public health outcomes.

MODELING:

Data Preparation:

- Utilize a comprehensive dataset containing water quality parameters and corresponding potability labels.
- Perform data preprocessing, including handling missing values, scaling features, and splitting data into training and validation sets.

Model Architecture:

- Construct a predictive model using machine learning algorithms such as Random Forest, XGBoost, and neural networks.
- Implement ensemble methods to combine the strengths of multiple models and improve predictive performance.

Training and Evaluation:

- Compile each model with appropriate hyperparameters, optimizer, and loss function.
- Train the models on the training data and evaluate their performance using metrics like accuracy, precision, recall, and F1-score.

- Fine-tune the models using techniques like grid search or randomized search to optimize performance further.

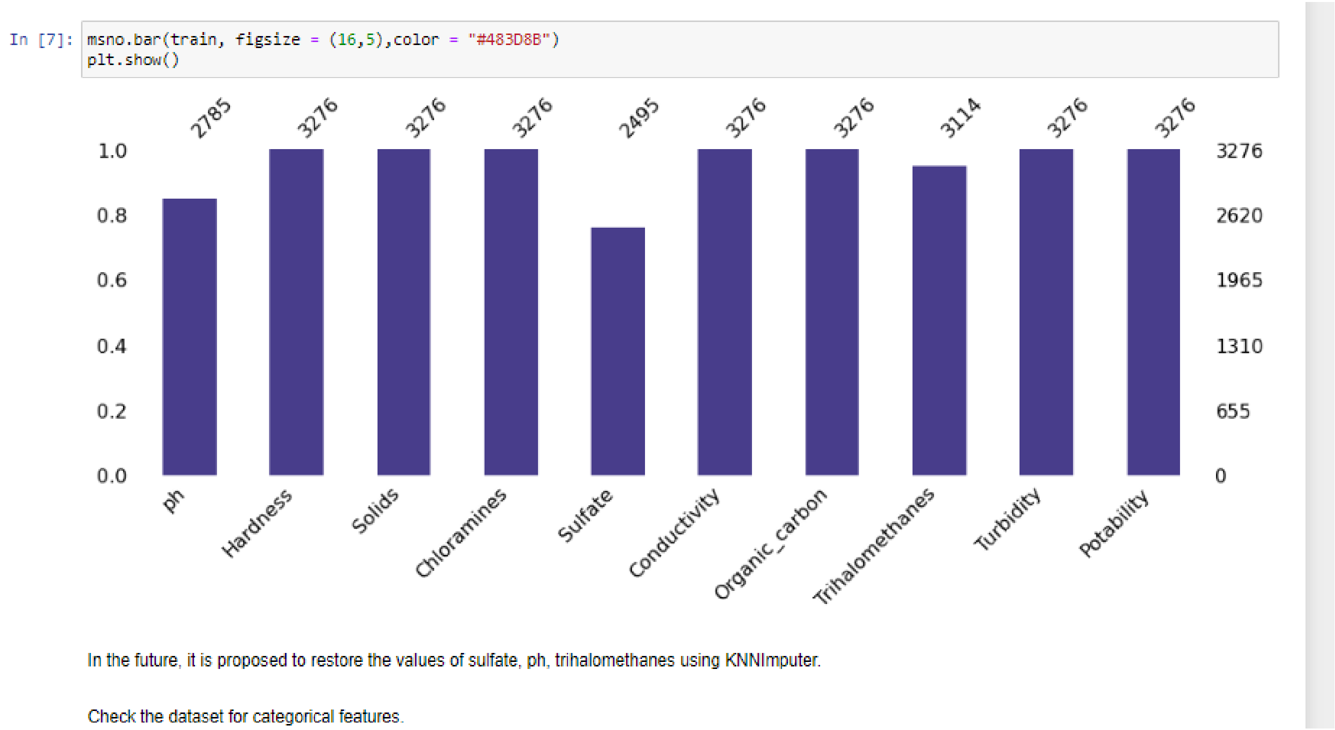
Predictive System Development:

- Develop a robust system for predicting water potability based on input water quality parameters.
- Implement preprocessing steps to handle incoming data and prepare it for prediction.
- Deploy the predictive system to enable real-time monitoring of water quality and potability.

Technology Used:

- TensorFlow and Keras for implementing neural networks and deep learning models.
- Scikit-learn for implementing machine learning algorithms and model evaluation.
- NumPy and pandas for data manipulation and handling.
- Matplotlib and Seaborn for data visualization and result interpretation.

RESULT:

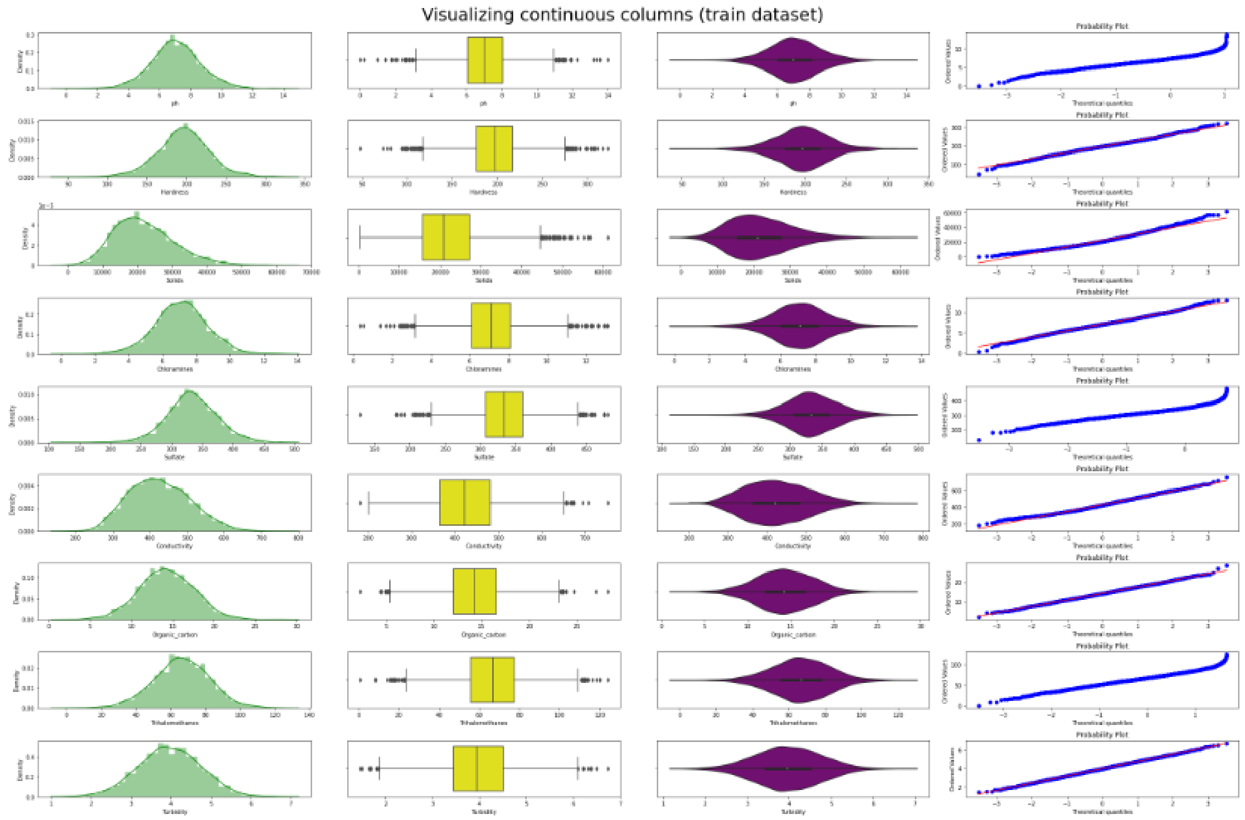


```
In [7]: msno.bar(train, figsize = (16,5),color = "#483D8B")
        plt.show()
```

In the future, it is proposed to restore the values of sulfate, ph, trihalomethanes using KNNImputer.

Check the dataset for categorical features.

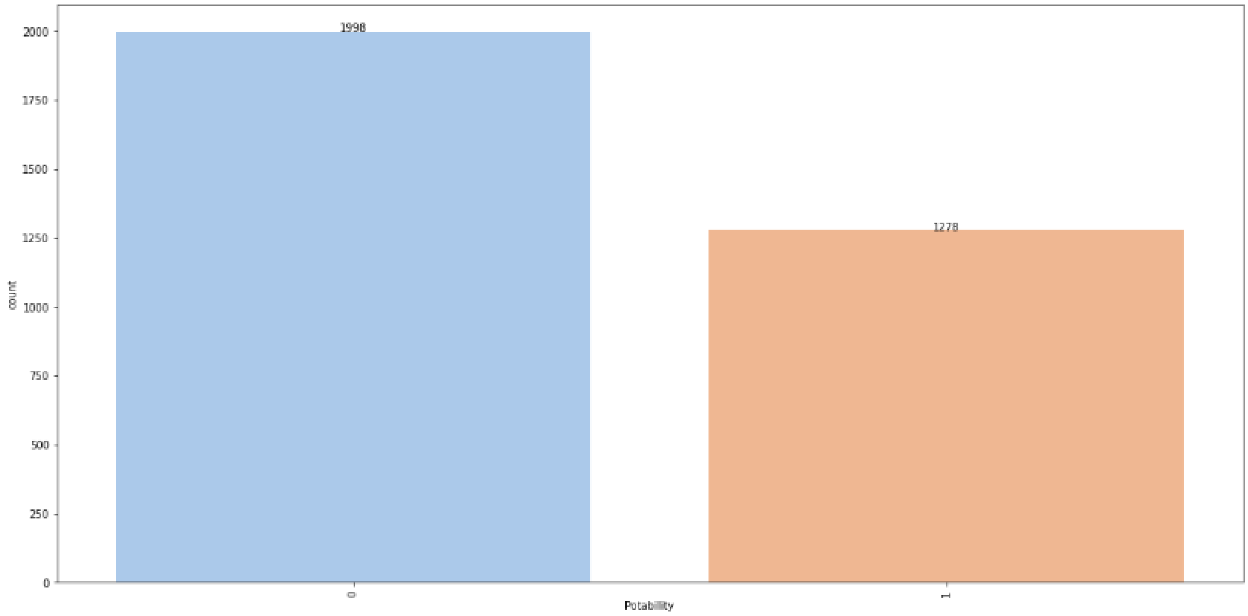
```
Out[11]: Text(0.5, 0.98, 'Visualizing continuous columns (train dataset)')
```



After analyzing these graphs, the following hypotheses can be made:

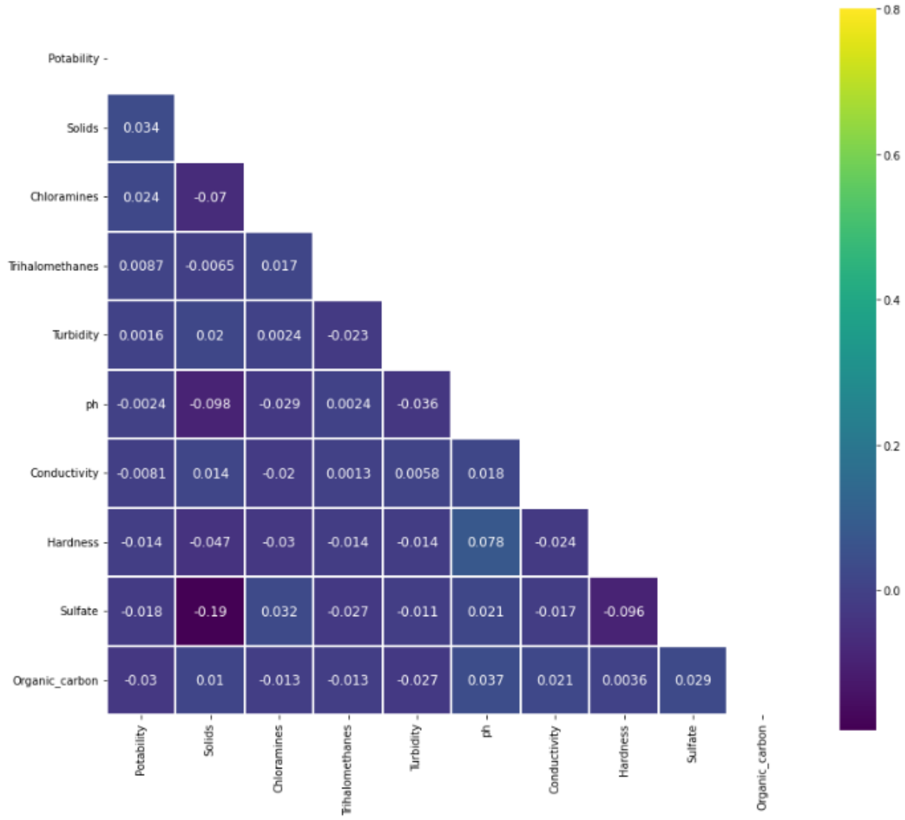
- Most features are distributed according to the normal distribution law.
- There are minor outliers for some features.

Let's look at the distribution of the target feature.

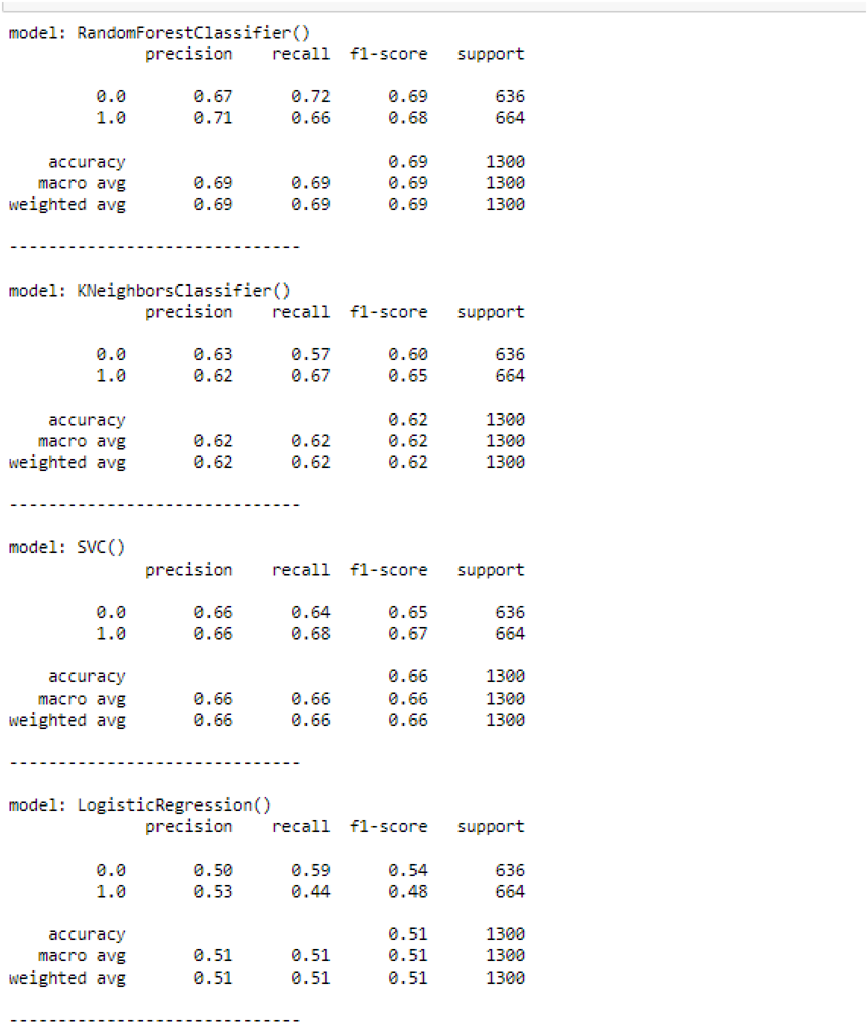


There is a clear class imbalance.

Let's fill in the gaps of information, after which we will begin to conduct statistical tests.



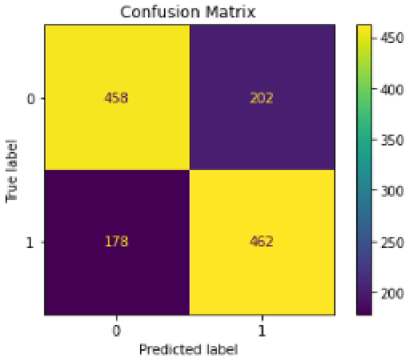
Great, there are no mutually correlated signs, outliers can be removed, and class imbalance can also be eliminated.



```
model:  RandomForestClassifier()
              precision    recall  f1-score   support

         0.0       0.67      0.72      0.69       636
         1.0       0.71      0.66      0.68       664

    accuracy                           0.69      1300
   macro avg       0.69      0.69      0.69      1300
weighted avg       0.69      0.69      0.69      1300

------------------------------

model:  KNeighborsClassifier()
              precision    recall  f1-score   support

         0.0       0.63      0.57      0.60       636
         1.0       0.62      0.67      0.65       664

    accuracy                           0.62      1300
   macro avg       0.62      0.62      0.62      1300
weighted avg       0.62      0.62      0.62      1300

------------------------------

model:  SVC()
              precision    recall  f1-score   support

         0.0       0.66      0.64      0.65       636
         1.0       0.66      0.68      0.67       664

    accuracy                           0.66      1300
   macro avg       0.66      0.66      0.66      1300
weighted avg       0.66      0.66      0.66      1300

------------------------------

model:  LogisticRegression()
              precision    recall  f1-score   support

         0.0       0.50      0.59      0.54       636
         1.0       0.53      0.44      0.48       664

    accuracy                           0.51      1300
   macro avg       0.51      0.51      0.51      1300
weighted avg       0.51      0.51      0.51      1300

------------------------------
```

In [50]:

```
cm = confusion_matrix(best_clf2.predict(X_test),y_test)
disp = ConfusionMatrixDisplay(cm, display_labels=["0","1"])
disp.plot()
plt.title("Confusion Matrix")
plt.show()
```



In [51]:

```
feature_importances=best_clf2.feature_importances_
feature_importances_df=pd.DataFrame({'features':list(X_train), 'feature_importances':feature_importances})
feature_importances_df=pd.DataFrame({'features':list(X_train), 'feature_importances':feature_importances})
feature_importances_df.sort_values('feature_importances',ascending=False)
```

Out[51]:

	features	feature_importances
0	ph	0.137891
4	Sulfate	0.125809
2	Solids	0.118570
3	Chloramines	0.115210
1	Hardness	0.110647
6	Organic_carbon	0.100476
5	Conductivity	0.100368
7	Trihalomethanes	0.096037
8	Turbidity	0.094992