Essay / Assignment Title: Water Potability

Programme title: M.Sc Data Analytics

Name: Ismail Abdul Maroof(Q1066461)

Year:2024

Table of Contents

Statement of compliance with academic ethics and the avoidance of plagiarism

I honestly declare that this dissertation is entirely my own work and none of its part has been copied from printed or electronic sources, translated from foreign sources and reproduced from essays of other researchers or students. Wherever I have been based on ideas or other people texts I clearly declare it through the good use of references following academic ethics.

(In the case that is proved that part of the essay does not constitute an original work, but a copy of an already published essay or from another source, the student will be expelled permanently from the postgraduate program).

Name and Surname (Capital letters):

ISMAIL ABDUL MAROOF

Date: 10/06/2024

Chapter 1: INTRODUCTION:

Predictive analytics and machine learning have a huge role in data-driven decision-making

These are the following:

1) Predictive analytics and machine learning help in analyzing large data so that they can identify the long-lasting patterns that cannot be quickly predicted by Human analytics and it also gives more precise and accurate analysis.

2) Predictive analytics and machine learning process large data sets quickly and give more efficient results.

3) Predictive analytics and machine learning also help to detect the fraudulent and risks.

Problem:

1)As predictive analytics and machine learning help in processing large data and give result significant so we selected the major problem regarding drinkable water, In this scenario we will identify that is this water is potable or not which means we have to check whether it is drinkable or not.

The business problem we are analyzing things based on portability. With the help of Predictive analytics and machine learning, we are making a model in which we aim to predict whether the water is potable (drinkable for humans) or not. This model is very useful for companies and other firms like water authorities

This model will help those who are trying to filter the water and want to see which element should be removed so the water becomes drinkable. The main reason for this model is that there would be access to drinkable and clean water for all the people around the world.

Chapter 2: Problem Formulation

The specific problem we aim to solve using predictive analytics and Machine Learning is that the water is drinkable

or not based on different parameters, it is only indicated by water potability whether the water is safe for human

consumption or not by denoting by 1 which means it is drinkable and 0 indicates it is not drinkable for humans. This is

analyzed by multiple parameters such as pH value, hardness, total dissolved solids (TDS), chloramines, sulfate, conductivity,

organic carbon, trihalomethanes, and turbidity, which collectively influence water quality.

Solving this problem is essential for public health to get clean water. So it can try to stop illnesses from being waterborne. Water

treatment can get quick results that this water can be consumed by a human or not. Through this, they can identify which chemical treatments

and filtration are needed. It will reduce the time and cost when the water is analyzed by the model. Water management authorities can easily

make better decisions regarding water treatment and where should be invested. predictive analytics helps in data-driven decision helps in making

precise and better decisions.

In a business context companies providing water supply build trust with the customers by providing safe and clean drinkable water.

The predictive analytics model ensures high standards of water quality. The machine learning model will help reduce manual testing

and monitoring by automating the complexity of water quality.

Predictive analytics can easily analyze large volumes of data from different water sources. So that the business would be responsive to new challenges

in water quality management.

Chapter 3: Data Collection and Preparation

Data Collection

The Data set of water potability that water is drinkable or not drinkable water, this data set is from Kaggle to design and implement on a machine learning model

that tells whether this water is suitable for humans or not. It contains 10 columns, of which 9 columns are independent features, and one is the targeted value on which we are performing the predictions.

This dataset is suitable for predicting water potability as it includes all the water parameters that are necessary. These parameters include pH value, hardness, solids, chloramines,

sulfate, conductivity, organic carbon, trihalomethanes, and turbidity, along with potability. By using the Pandas library we are importing the dataset of water potability.

Stored data using Pandas and created a data-frame 'df'. After that we used df.head() to check how the data looks like, I used df.head() to check the first five rows of data.

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import matplotlib_inline
import sklearn
import numpy as np
from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import accuracy_score, classification_report, confusion_matrix,roc_curve, auc


from google.colab import files
uploaded = files.upload()
```

```
Choose Files No file chosen          Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.
Saving water_potability.csv to water_potability (1).csv
```

```
df = pd.read_csv("water_potability.csv")
print(df)
```

```
            ph    Hardness        Solids  Chloramines     Sulfate  \
0          NaN  204.890455  20791.318981     7.300212  368.516441
1     3.716080  129.422921  18630.057858     6.635246         NaN
2     8.099124  224.236259  19909.541732     9.275884         NaN
3     8.316766  214.373394  22018.417441     8.059332  356.886136
4     9.092223  181.101509  17978.986339     6.546600  310.135738
...        ...         ...           ...          ...         ...
3271  4.668102  193.681735  47580.991603     7.166639  359.948574
3272  7.808856  193.553212  17329.802160     8.061362         NaN
3273  9.419510  175.762646  33155.578218     7.350233         NaN
3274  5.126763  230.603758  11983.869376     6.303357         NaN
3275  7.874671  195.102299  17404.177061     7.509306         NaN

      Conductivity  Organic_carbon  Trihalomethanes  Turbidity  Potability
0       564.308654       10.379783        86.990970   2.963135           0
1       592.885359       15.180013        56.329076   4.500656           0
2       418.606213       16.868637        66.420093   3.055934           0
3       363.266516       18.436524       100.341674   4.628771           0
4       398.410813       11.558279        31.997993   4.075075           0
...            ...             ...              ...        ...         ...
3271    526.424171       13.894419        66.687695   4.435821           1
3272    392.449580       19.903225              NaN   2.798243           1
3273    432.044783       11.039070        69.845400   3.298875           1
3274    402.883113       11.168946        77.488213   4.708658           1
3275    327.459760       16.140368        78.698446   2.309149           1

[3276 rows x 10 columns]
```

Data Processing

Before analysis of data, it is essential to process the data. Data processing is an essential process in machine learning predictive analytics. Because there are so many duplicate, null, and missing values and issues in rows or columns. So data processing involves data cleaning and preparation for the next step.

In this process, we have to check whether there is any data that is going to cause trouble in analysis in further steps so we have to make sure there are no missing values.

```
df.head()
```

	ph	Hardness	Solids	Chloramines	Sulfate	Conductivity	Organic_car
0	NaN	204.890455	20791.318981	7.300212	368.516441	564.308654	10.379
1	3.716080	129.422921	18630.057858	6.635246	NaN	592.885359	15.180
2	8.099124	224.236259	19909.541732	9.275884	NaN	418.606213	16.868
3	8.316766	214.373394	22018.417441	8.059332	356.886136	363.266516	18.436
4	9.092223	181.101509	17978.986339	6.546600	310.135738	398.410813	11.558

```
df.isna().sum()
```

```
ph                 491
Hardness             0
Solids               0
Chloramines          0
Sulfate            781
Conductivity         0
Organic_carbon       0
Trihalomethanes    162
Turbidity            0
Potability           0
dtype: int64
```

```
df.dropna(inplace=True)
```

Missing values

we have checked whether there is any missing value by using isna() and sum(). If there is a missing value you have to find a solution for it by taking the median or you can remove the rows of missing values it depends on the situation.

for this data set, we have removed the rows. We removed it by using df.dropna(inplace=True). Then we again verified that is there any missing values in the data set by using isna() and sum().

Now our dataset have no dublicate or missing value and it's ready for further steps.

```
[ ] df.dropna(inplace=True)
```

```
[ ] df.isna().sum()
```

```
ph                 0
Hardness           0
Solids             0
Chloramines        0
Sulfate            0
Conductivity       0
Organic_carbon     0
Trihalomethanes    0
Turbidity          0
Potability         0
dtype: int64
```

Chapter 4: Exploratory Data Analysis

Exploratory Data Analysis is an essential part of Data Analytics in which we understand the data patterns and relationships. In this, we conducted the EDA on

the water potability dataset to gain insights into water quality metrics and their impact on the potability of water.

Target value:

Water potability is the targeted value with non-potable it's (0) and for potable its (1).

descriptive stats provide an overview of the central tendency, dispersion, and shape of the distribution of each feature df.describe().T.

```
df.describe().T
```

	count	mean	std	min	25%	50%	75%	max
ph	2011.0	7.085990	1.573337	0.227499	6.089723	7.027297	8.052969	14.000000
Hardness	2011.0	195.968072	32.635085	73.492234	176.744938	197.191839	216.441070	317.338124
Solids	2011.0	21917.441374	8642.239815	320.942611	15615.665390	20933.512750	27182.587067	56488.672413
Chloramines	2011.0	7.134338	1.584820	1.390871	6.138895	7.143907	8.109726	13.127000
Sulfate	2011.0	333.224672	41.205172	129.000000	307.632511	332.232177	359.330555	481.030642
Conductivity	2011.0	426.526409	80.712572	201.619737	366.680307	423.455906	482.373169	753.342620
Organic_carbon	2011.0	14.357709	3.324959	2.200000	12.124105	14.322019	16.683049	27.006707
Trihalomethanes	2011.0	66.400859	16.077109	8.577013	55.952664	66.542198	77.291925	124.000000
Turbidity	2011.0	3.969729	0.780346	1.450000	3.442915	3.968177	4.514175	6.494749
Potability	2011.0	0.403282	0.490678	0.000000	0.000000	0.000000	1.000000	1.000000

pH: The mean value of pH which is 7 indicates that most samples are neutral.

Hardness: this indicated the varying levels of calcium and magnesium salts

Solids: A significant difference is dissolved solids across samples.

Other metrics like Chloramines, Sulfate, and Conductivity also show variability, which can influence water potability.

Correlation Matrix

In correlation, the method helps in identifying the relationship between metrics

solids and conductivity show a strong correlation

other show low correlation

Distribution of pH level

the pH levels are mostly within the recommended range of 6.5 to 8.5, but there are some outliers.

• Highlight any insights or patterns discovered during this phase.

Portability: The dataset is slightly imbalanced as it shows a high number of nonpotable (0) samples as compared to potable (1) samples.

This impact also shows that more water samples are considered unsafe to drink water.

Relation between pH and hardness

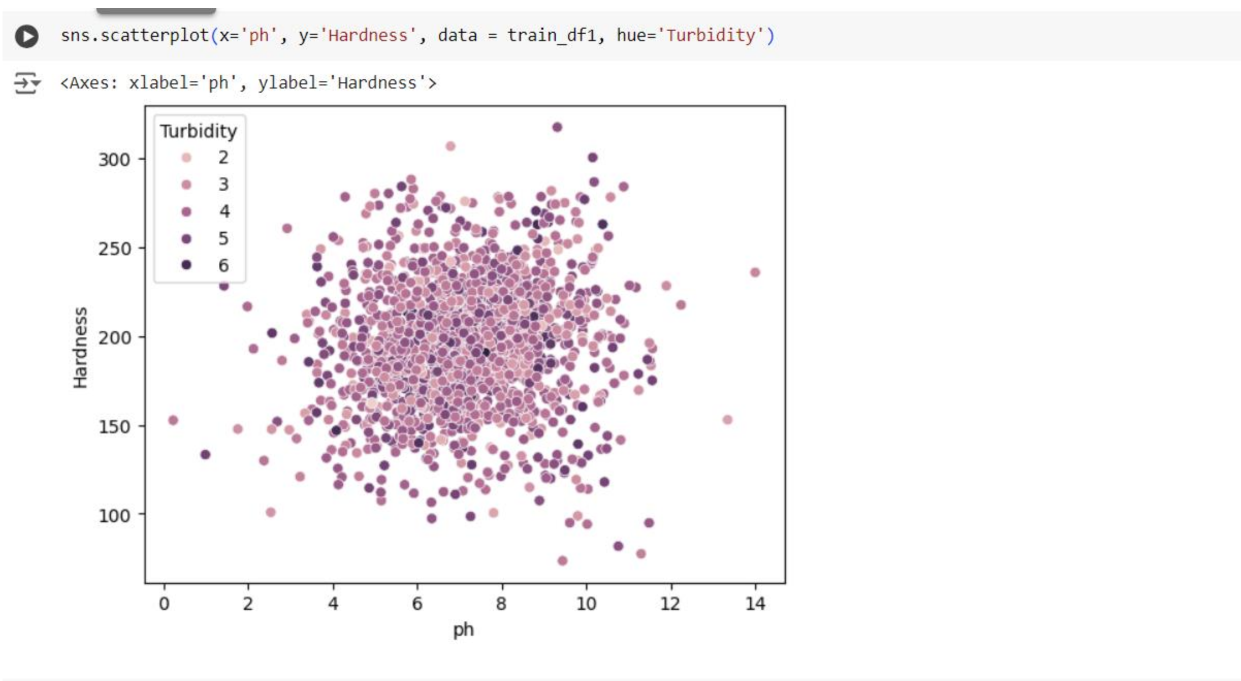
insight: There is a notable relationship between pH and hardness which can impact the potability of water

pattern: scatter plot shows that certain ranges of pH and hardness are more likely associated with potable and nonpotable water.

sns.scatterplot(x='pH', y='Hardness', data=df, hue='Potability')

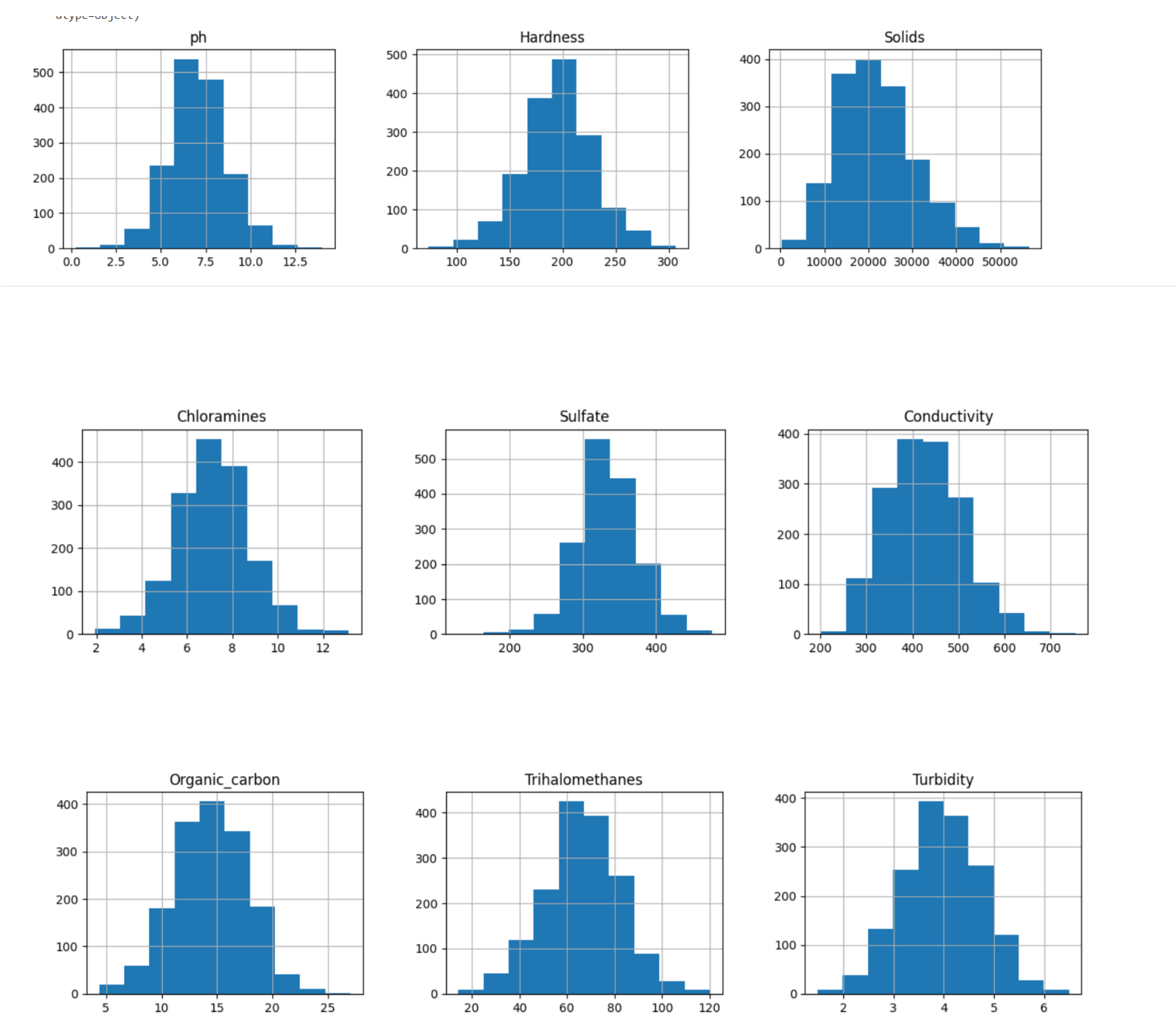
plt.title('Scatter Plot of pH vs Hardness')

plt.show()



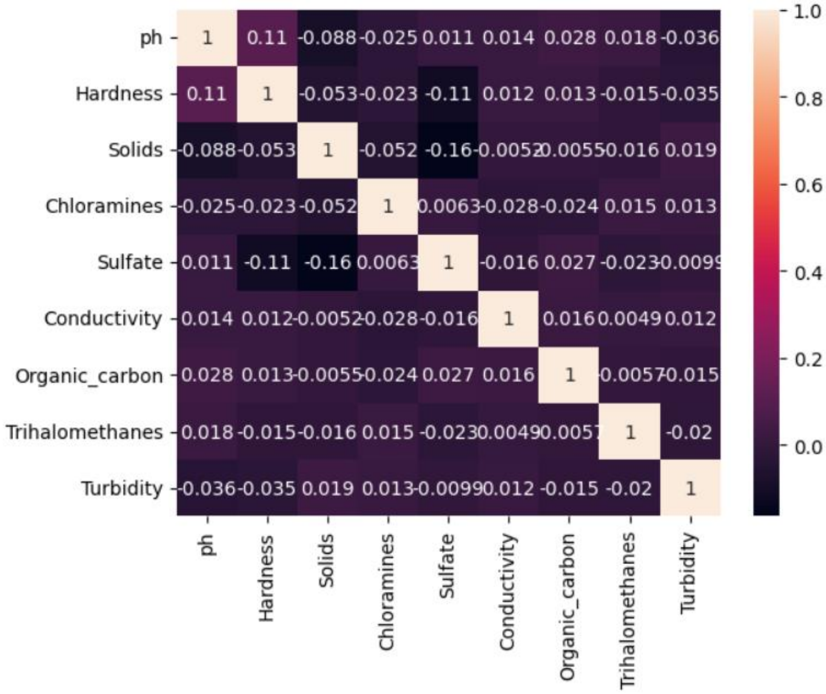
```
train_df.hist(figsize=(15,15))
```

```
array([[<Axes: title={'center': 'ph'}>,
        <Axes: title={'center': 'Hardness'}>,
        <Axes: title={'center': 'Solids'}>],
       [<Axes: title={'center': 'Chloramines'}>,
        <Axes: title={'center': 'Sulfate'}>,
        <Axes: title={'center': 'Conductivity'}>],
       [<Axes: title={'center': 'Organic_carbon'}>,
        <Axes: title={'center': 'Trihalomethanes'}>,
        <Axes: title={'center': 'Turbidity'}>],
       [<Axes: title={'center': 'Potability'}>, <Axes: >, <Axes: >]],
      dtype=object)
```



```python
sns.heatmap(train_df1.corr(),annot =True)
```

```
<Axes: >
```



Summary

The dataset has more nonpotable water in it.

The solid and conductivity have a strong correlation.

The visibility of clusters and in pair and scatter plots suggest certain combinations in the prediction of water potability.

Chapter 5: Model Selection and Implementation

Machine Learning Models and Algorithms Selection:

For this model, Our main goal is to predict the potability of water on the basis of different attributes. To achieve this aim, the Random Forest is selected to perform as machine learning model.

Random Forest classifier:

A random forest classifier is used for solving the numeric target value and classification. It improves the accuracy and overfitting risk. It also helps in understanding the participation

of each feature in prediction. This particularly helps in finding the attributes that affect the potability. By recognizing important features it can improve the model's performance and

speed as well. It handles the missing values by using the median or mode of the dataset. It's less sensitive as compared to other algorithm.

The Random forest can handle both numerical and categorical data, making it easier for every dataset.

Implementation:

To get your model ready for Random Forest Classifier, These are the following steps were taken:

Data loading, preprocessing, model training, evaluation, and feature importance analysis.

Data loading and preprocessing:

First, we load the dataset and handle any missing values.

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import matplotlib_inline
import sklearn
import numpy as np
from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import accuracy_score, classification_report, confusion_matrix,roc_curve, auc


from google.colab import files
uploaded = files.upload()
```

```
Choose Files No file chosen     Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to
enable.
Saving water_potability.csv to water_potability (1).csv
```

```
df = pd.read_csv("water_potability.csv")
print(df)
```

```
            ph    Hardness        Solids  Chloramines     Sulfate  \
0          NaN  204.890455  20791.318981     7.300212  368.516441
1     3.716080  129.422921  18630.057858     6.635246         NaN
2     8.099124  224.236259  19909.541732     9.275884         NaN
3     8.316766  214.373394  22018.417441     8.059332  356.886136
4     9.092223  181.101509  17978.986339     6.546600  310.135738
...        ...         ...           ...          ...         ...
3271  4.668102  193.681735  47580.991603     7.166639  359.948574
3272  7.808856  193.553212  17329.802160     8.061362         NaN
3273  9.419510  175.762646  33155.578218     7.350233         NaN
3274  5.126763  230.603758  11983.869376     6.303357         NaN
3275  7.874671  195.102299  17404.177061     7.509306         NaN

      Conductivity  Organic_carbon  Trihalomethanes  Turbidity  Potability
0       564.308654       10.379783        86.990970   2.963135           0
1       592.885359       15.180013        56.329076   4.500656           0
2       418.606213       16.868637        66.420093   3.055934           0
3       363.266516       18.436524       100.341674   4.628771           0
4       398.410813       11.558279        31.997993   4.075075           0
...            ...             ...              ...        ...         ...
3271    526.424171       13.894419        66.687695   4.435821           1
3272    392.449580       19.903225              NaN   2.798243           1
3273    432.044783       11.039070        69.845400   3.298875           1
3274    402.883113       11.168946        77.488213   4.708658           1
3275    327.459760       16.140368        78.698446   2.309149           1

[3276 rows x 10 columns]
```

```
df.head()
```

	ph	Hardness	Solids	Chloramines	Sulfate	Conductivity	Organic_carl
0	NaN	204.890455	20791.318981	7.300212	368.516441	564.308654	10.3797
1	3.716080	129.422921	18630.057858	6.635246	NaN	592.885359	15.180(
2	8.099124	224.236259	19909.541732	9.275884	NaN	418.606213	16.868(
3	8.316766	214.373394	22018.417441	8.059332	356.886136	363.266516	18.436
4	9.092223	181.101509	17978.986339	6.546600	310.135738	398.410813	11.558

```
df.isna().sum()
```

```
ph                 491
Hardness             0
Solids               0
Chloramines          0
Sulfate            781
Conductivity         0
Organic_carbon       0
Trihalomethanes    162
Turbidity            0
Potability           0
dtype: int64
```

```
df.dropna(inplace=True)
```

```
[ ] df.dropna(inplace=True)
```

```
[ ] df.isna().sum()
```

```
ph                 0
Hardness           0
Solids             0
Chloramines        0
Sulfate            0
Conductivity       0
Organic_carbon     0
Trihalomethanes    0
Turbidity          0
Potability         0
dtype: int64
```

For loading the dataset we have chosen an Excel file as the data set is from Kaggle with the help of the pandas library, we imported the Excel file. then for missing values, we used df.isna().sum()

to check if there were any missing values and after that, we removed them by using dropna(inplace=True)

before proceeding with the model training, it's essential to understand the data distribution and relation between attributes

```
from sklearn.model_selection import train_test_split
X= df.drop('Potability', axis=1)
y= df['Potability']
```

```
X_train,X_test,y_train,y_test = train_test_split(X,y, test_size = 0.2)
```

```
train_df =X_train.join(y_train)
```

The histogram is used to understand the distribution of the features.

Heatmap is generated to see how they are related to each other. This helps in identifying the multicollinearity, which can be addressed later.

Model Training:

After that, we initialize and train the Random forest Classifier on the training data.

```python
rf = RandomForestClassifier()
rf.fit(X_train, y_train)
y_pred_rf = rf.predict(X_test)
```

```python
[ ] X_train,X_test, y_train, y_test =train_test_split(X,y, test_size=0.2)
```

```python
[ ] model = RandomForestClassifier()
```

```python
[ ] model.fit(X_train,y_train)
```

```
▾ RandomForestClassifier
RandomForestClassifier()
```

Conclusion:

In this summary, we will discuss about the Random Forest Classifier for predicting water potability.

Data Loading and Processing: Handling missing values and splitting them to train and testing set.

Exploratory Data Analysis: Understanding data and the relation between attributes.

Feature selection: From the Random forest model to identify key features.

Chapter 6: Model Evaluation

The Random Forest Classifier was chosen to predict the potability of water based on various metrics. The model's performance was evaluated using several metrics:

accuracy, precision, recall, and F1-score. These provide a complete understanding of distinguishing between potable and nonpotable water.

Evaluation metric:

Evaluation metrics are crucial in evaluating the performance of a classification model. Each metrics give different results than how well the model is doing.

Accuracy:

The portion of true results (both negative and positive) among the total number of cases examined.

Precision:

In this proportionate, all depends on the true positive proportion among all positive results predicted by the model.

Recall:

the proportion of true positive among all actual positive cases.

F1-score:

The harmonic mean of precision and recall provides the balance between the two.

Model performance:

after training the Random Forest classifier, the model was evaluated on the test data. As shown in the code

```python
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score,confusion_matrix
```

```python
# Make predictions

# Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)

print(f"Random Forest - Accuracy: {accuracy:.4f}, Precision: {precision:.4f}, Recall: {recall:.4f}, F1 Score: {f1:.4f}")
```

```
Random Forest - Accuracy: 0.6625, Precision: 0.7100, Recall: 0.3989, F1 Score: 0.5108
```

Here are the values of

Accuracy: 0.6625

Precision: 0.7100

Recall: 0.3989

F1-score:0.5108

These results indicate the following:

1)This model correctly predicts the potability of water about 70% of the time.

2)All the samples predicted as potable, about 67% are actual potable.

3)It actually identifies about 50% of actual potable water samples.

4) The F1-score of 0.5108 suggests a moderate balance between precision and recall.

Implications of Model Performance on Solving the Business Problem

Random forest Classifier helps predict the potability of water for solving the business problem of ensuring access to safe drinking water. With different aspects including

health and safety and cost efficiency.

Health and safety:

Doing business is the initial step to ensuring that water is safe for consumption for public health and safety. The Random Forest Model, Helps in getting an accuracy of approximately 70%, predicting the potability of water. By this level, we can say that this model can correctly predict potable and nonpotable water in 70% of cases.

Impact:

Detection:

This model can easily identify that water is potentially unsafe. This helps timely intervention from health hazards that are associated with unsafe water.

Preventive:

By getting reports, authorities can refine the water by taking some action, such as boiling or purifying.

Cost Efficiency:

By using this model you can save costs from extensive tests of water which are costly, this model can lead towards cost saving.

Conclusion:

Random forest model performance in predicting accuracy, reliable prediction, model help addressing critical business for ensuring safe drinking water.

Chapter 7: Conclusion and Recommendations

In this project, we have focused on predicting the potability of water using different water attributes from the datasets. Key water

quality parameters included pH value, hardness, Solids, chloramines, sulfate, conductivity, organic carbon, trihalomethanes, and turbidity.

The main goal is to get the result that water is potable or not potable.

Data preprocessing:

The data sets contain 3,276 records and each represents different water attributes.

the missing values were found and incomplete records were found, resulting in clean data.

Model Selection and Performance:

The Random forest performance was superior:

Random Forest Model Performance:

Accuracy: 0.6625

Precision: 0.7100

Recall: 0.3989

F1-score:0.5108

Feature Importance:

The model's important features for predicting the water potability predictions include pH, solids, and sulfate are critical parameters for water quality.

Visualization and correlation:

Exploratory Data Analysis revealed correlations between different water qualities.

Visualizations such as heatmaps and scatter plots helped in understanding the relationships between variables and their impact on water potability.

Recommendations based on your analysis for addressing the identified business problem.

Enhance Data Collection and Quality:

Updating the data regularly to ensure water quality measurements to ensure the model remains accurate and relevant.

Implement data quality checks to reduce missing values.

Use advanced methods to get datasets to reduce human error.

Chapter 8: Project Reflection

Challenges encountered:

The missing values affect the result and model accuracy.

Identifying and getting useful data from raw datasets is complex. Some have nonlinear relationships with water potability which are difficult to capture.

Selection of the right model.

The models are hard to interpret and to explain stakeholders.

lessons learned:

Making data high-quality is difficult for building a predictive model. Preprocessing steps like handling missing values, scaling features, and ensuring data integrity

can easily create an impact on model performance. Exploratory Data Analysis is very useful for understanding the data through visualization, also we can see the relationship between

the datasets.

Potential Improvements and Additional Steps:

If revisiting the project, implementing more better model such as K-Nearest Neighbors(KNN), will help it handle missing data better. We can further increase the effectiveness and

applicability of predictive analytics solutions for ensuring safe drinking.

So this leads to more accurate, reliable, and impactful outcomes in public health and water quality management.

Bibliography

Breiman, 2001. Machine Learning. In: *Random Forests.* s.l.:s.n., pp. 5-32.

CDC, 2022. *Centers for Disease Control and Prevention.* [Online]
Available at: https://www.cdc.gov/healthywater/drinking/index.html
[Accessed 2024].

Matplotlib, 2022. *Visualization with Python.* [Online]
Available at: https://matplotlib.org/
[Accessed 2024].

Organization, W. H., 2020. *WHO.* [Online]
Available at: https://www.who.int/publications/i/item/9789241549950
[Accessed 2024].

Seaborn, 2022. *Statistical Data Visualization.* [Online]
Available at: https://seaborn.pydata.org/
[Accessed 2024].

Tharmalingam, L., 2023. *Water Quality and Potability.* [Online]
Available at: https://www.kaggle.com/datasets/uom190346a/water-quality-and-potability/discussion/441751