NAME-Uday Kumar S V

USN-ENG21CS0451

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

#dataset
csv_file='/content/water_potability - water_potability.csv'
df_water=pd.read_csv(csv_file)
df_water.head()
```

	ph	Hardness	Solids	Chloramines	Sulfate	Conductivity	Organic_c
0	NaN	204.890456	20791.31898	7.300212	368.516441	564.308654	10.3
1	3.716080	129.422921	18630.05786	6.635246	NaN	592.885359	15.1
2	8.099124	224.236259	19909.54173	9.275884	NaN	418.606213	16.8
3	8.316766	214.373394	22018.41744	8.059332	356.886136	363.266516	18.4
4	9.092223	181.101509	17978.98634	6.546600	310.135738	398.410813	11.5

Double-click (or enter) to edit

```
#  Apply the 'info()' function on the 'df_play' DataFrame.
df_water.info()
```

```
<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
```

```
#  Encode the categorical values

from sklearn.preprocessing import LabelEncoder

label = LabelEncoder()
for column in df_water.columns:
  df_water[column] = label.fit_transform(df_water[column])

df_water
```

	ph	Hardness	Solids	Chloramines	Sulfate	Conductivity	Organic_carbon	Tri
0	2785	1997	1619	1791	2009	3104	369	
1	49	86	1283	1228	2495	3195	2001	
2	2107	2676	1466	2997	2495	1589	2549	
3	2233	2373	1794	2410	1819	782	2931	
4	2509	974	1175	1151	663	1278	663	
...	...	...	...	...	...	...	...	
3271	162	1510	3249	1674	1870	2892	1488	
3272	1924	1503	1068	2411	2495	1189	3138	
3273	2581	789	2917	1834	2495	1792	518	
3274	283	2837	354	960	2495	1361	561	
3275	1969	1561	1079	1976	2495	353	2333	

3276 rows × 10 columns

```
#  Create separate DataFrames for feature and target

features_df = df_water.drop('Potability', axis = 1)
target_df = df_water['Potability']

print(features_df.shape)
print(target_df.shape)
```

```
(3276, 9)
(3276,)
```

```
#  Import train_test_split function
from sklearn.model_selection import train_test_split

# Split dataset into training set and test set
X_train, X_test, y_train, y_test = train_test_split(features_df, target_df, test_size = 0.3,
                                                    random_state = 2)

# Print the shape of train and test sets.
print("Shape of X_train:", X_train.shape)
print("Shape of X_test:", X_test.shape)
print("Shape of y_train:", y_train.shape)
print("Shape of y_test:", y_test.shape)
```

```
Shape of X_train: (2293, 9)
Shape of X_test: (983, 9)
Shape of y_train: (2293,)
Shape of y_test: (983,)
```

```
# Implement Naive Bayes Classifier

# Import the required library
from sklearn.naive_bayes import MultinomialNB

# Model the NB Classifier
nb_clf = MultinomialNB()
nb_clf.fit(X_train, y_train)

# Predict the train and test sets
y_train_predict_nb = nb_clf.predict(X_train)
y_test_predict_nb = nb_clf.predict(X_test)

# Evaluate the accuracy scores
print('Accuracy on the training set: {:.2f}'.format(nb_clf.score(X_train, y_train)))
print('Accuracy on the test set: {:.2f}'.format(nb_clf.score(X_test, y_test)))
```

```
Accuracy on the training set: 0.53
Accuracy on the test set: 0.49
```