PROJECT TITILE:

WATER POTABILITY PREDICTION

Name: Rahul Rakesh Turupada

Registration Number: 12018791

Section: K20CH

Roll Number: RK20CHB55

Course code: INT353

About Dataset

Context: Why I have taken this dataset

Water is the most significant resource of life, crucial for supporting the life of most existing creatures and human beings. Living organisms need water with enough quality to continue their lives. There are certain limits of pollution that water species can tolerate. Exceeding these limits affects the existence of these creatures and threatens their lives.

As per the United Nations report, about 1.5 million people die each year because of contaminated water-driven diseases. In developing countries, it is announced that 80% of health problems are caused by contaminated water. Five million deaths and 2.5 billion illnesses are reported annually. Such a mortality rate is higher than deaths resulting from accidents, crimes, and terrorist attacks.

Access to safe drinking-water is essential to health, a basic human right and a component of effective policy for health protection. This is important as a health and development issue at a national, regional, and local level. In some regions, it has been shown that investments in water supply and sanitation can yield a net economic benefit, since the reductions in adverse health effects and health care costs outweigh the costs of undertaking the interventions

Potable water is a threatened resource for the developing world.

So, it is important to check the quality of water before drinking.

Content:

The water_potability.csv file contains water quality metrics for 3276 different water bodies.

1. pH value:
PH is an important parameter in evaluating the acid-base balance of water. It is also the indicator of acidic or alkaline condition of water status. WHO has recommended maximum permissible limit of pH from 6.5 to 8.5. The current investigation ranges were 6.52-6.83 which are in the range of WHO standards.

2. Hardness:
Hardness is mainly caused by calcium and magnesium salts. These salts are dissolved from geologic deposits through which water travels. The length of time water is in contact with hardness producing material helps determine how much hardness there is in raw water. Hardness was originally defined as the capacity of water to precipitate soap caused by Calcium and Magnesium.

3. Solids (Total dissolved solids - TDS):
Water can dissolve a wide range of inorganic and some organic minerals or salts such as potassium, calcium, sodium, bicarbonates, chlorides, magnesium, sulphates etc. These minerals produced un-wanted taste and diluted color in appearance of water. This is the important parameter for the use of water. The water with high TDS value indicates that water is highly mineralized. Desirable limit for TDS is 500 mg/l and maximum limit is 1000 mg/l which prescribed for drinking purpose.

4. Chloramines:
Chlorine and chloramine are the major disinfectants used in public water systems. Chloramines are most formed when ammonia is added to chlorine to treat drinking water. Chlorine levels up to 4 milligrams per liter (mg/L or 4 parts per million (ppm)) are considered safe in drinking water.

5. Sulfate:
Sulphates are naturally occurring substances that are found in minerals, soil, and rocks. They are present in ambient air, groundwater, plants, and food. The principal commercial use of sulphate is in the chemical industry. Sulfate concentration in seawater is about 2,700 milligrams per liter (mg/L). It ranges from 3 to 30 mg/L in most freshwater supplies, although much higher concentrations (1000 mg/L) are found in some geographic locations.

6. Conductivity:
Pure water is not a good conductor of electric current rather a good insulator. Increase in ions concentration enhances the electrical conductivity of water. Generally, the number of dissolved solids in water determines the electrical conductivity. Electrical conductivity (EC) measures the ionic process of a solution that enables it to transmit current. According to WHO standards, EC value should not exceed 400 micro s/cm.

7. Organic carbon:
Total Organic Carbon (TOC) in source waters comes from decaying natural organic matter (NOM) as well as synthetic sources. TOC is a measure of the total amount of carbon in organic compounds in pure water. According to US EPA < 2 mg/L as TOC in treated / drinking water, and < 4 mg/Lit in source water which is use for treatment.

8. Trihalomethanes:
THMs are chemicals which may be found in water treated with chlorine. The concentration of THMs in drinking water varies according to the level of organic material in the water, the amount of chlorine required to treat the water, and the temperature of the water that is being treated. THM levels up to 80 ppm is considered safe in drinking water.

9. Turbidity:
The turbidity of water depends on the quantity of solid matter present in the suspended state. It is a measure of light emitting properties of water and the test is used to indicate the quality of waste discharge with respect to colloidal matter. The mean turbidity value obtained for Wonda Genet Campus (0.98 NTU) is lower than the WHO recommended value of 5.00 NTU.

10. Potability:
Indicates if water is safe for human consumption where 1 means Potable and 0 means Not potable.

Initial Analysis:

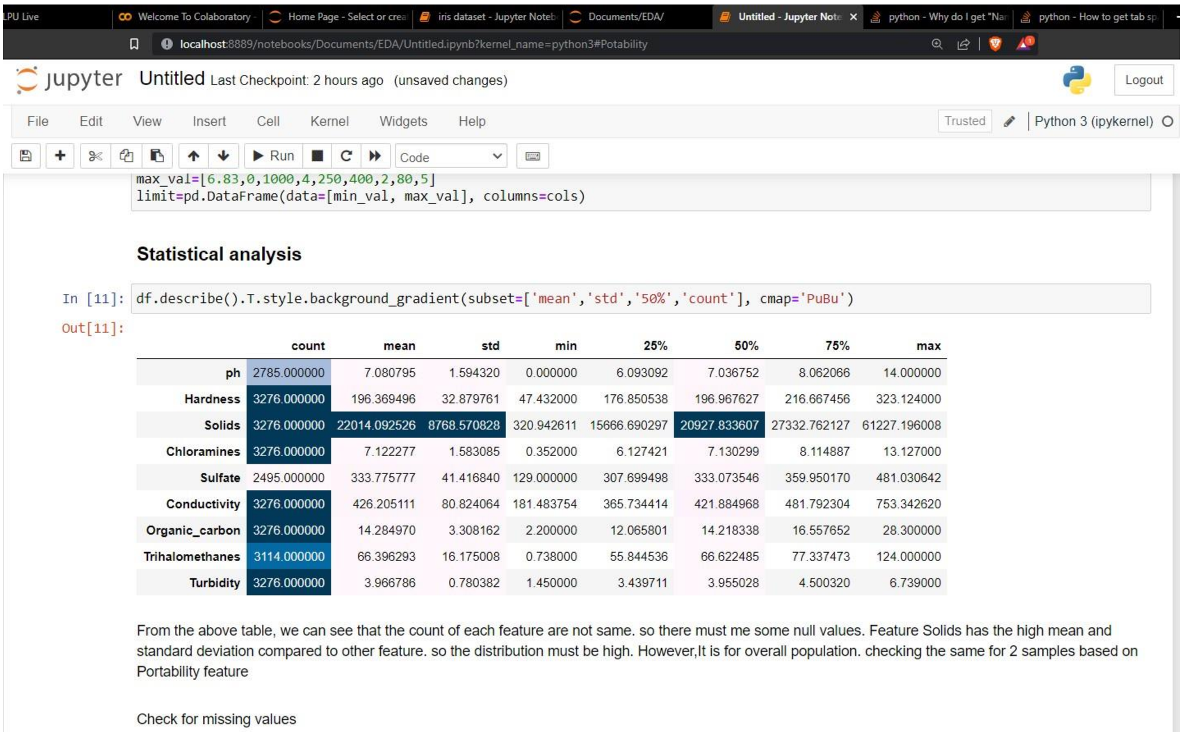
- There are 9 Independent features and 1 dependent features. Details for each feature are provided in the above. Main objective is to use the Potability feature as target feature for classification problem.
- Except Target feature, other features are float and continuous value. we can convert the Portability into Categorizing feature.

Dataset info:

```
<class 'pandas. core. frame. Data
Frame'>Range Index: 3276 entries, 0 to
3275 Data columns (total 10 columns):
 #   Column           Non-Null Count D type
---  ----------       --------------  --------
 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
```

Aim of the project:

1. To Classify the water quality whether it is potable or not based on the features provided.
2. To find if there is any Correlation between different columns and its effect on the potability of water.
3. Major causes for the non-potability of water.



jupyter Untitled Last Checkpoint: 2 hours ago (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help

```
max_val=[6.83,0,1000,4,250,400,2,80,5]
limit=pd.DataFrame(data=[min_val, max_val], columns=cols)
```

Statistical analysis

In [11]:
```
df.describe().T.style.background_gradient(subset=['mean','std','50%','count'], cmap='PuBu')
```

Out[11]:

	count	mean	std	min	25%	50%	75%	max
ph	2785.000000	7.080795	1.594320	0.000000	6.093092	7.036752	8.062066	14.000000
Hardness	3276.000000	196.369496	32.879761	47.432000	176.850538	196.967627	216.667456	323.124000
Solids	3276.000000	22014.092526	8768.570828	320.942611	15666.690297	20927.833607	27332.762127	61227.196008
Chloramines	3276.000000	7.122277	1.583085	0.352000	6.127421	7.130299	8.114887	13.127000
Sulfate	2495.000000	333.775777	41.416840	129.000000	307.699498	333.073546	359.950170	481.030642
Conductivity	3276.000000	426.205111	80.824064	181.483754	365.734414	421.884968	481.792304	753.342620
Organic_carbon	3276.000000	14.284970	3.308162	2.200000	12.065801	14.218338	16.557652	28.300000
Trihalomethanes	3114.000000	66.396293	16.175008	0.738000	55.844536	66.622485	77.337473	124.000000
Turbidity	3276.000000	3.966786	0.780382	1.450000	3.439711	3.955028	4.500320	6.739000

From the above table, we can see that the count of each feature are not same. so there must me some null values. Feature Solids has the high mean and standard deviation compared to other feature. so the distribution must be high. However,It is for overall population. checking the same for 2 samples based on Portability feature

Check for missing values

- From the above table, we can see that the count of each feature are not same. so there must me some null values.
  Feature Solids has the high mean and standard deviation comparted to other feature. so the distribution must be high.
  However, the above description is for overall population.

- Features ph, Sulphate and Trihalomethanes are having null values.

- Since the missing values are on both classes (Potability 1 & 0), we can replace it with population mean. so, we will replace the Nan values bases on sample mean from both classes.

Features ph, Sulfate and Trihalomethanes are having null values.

In [14]:
```
df[df['Sulfate'].isnull()]
df[df['ph'].isnull()]
df[df['Trihalomethanes'].isnull()]
```

Out[14]:

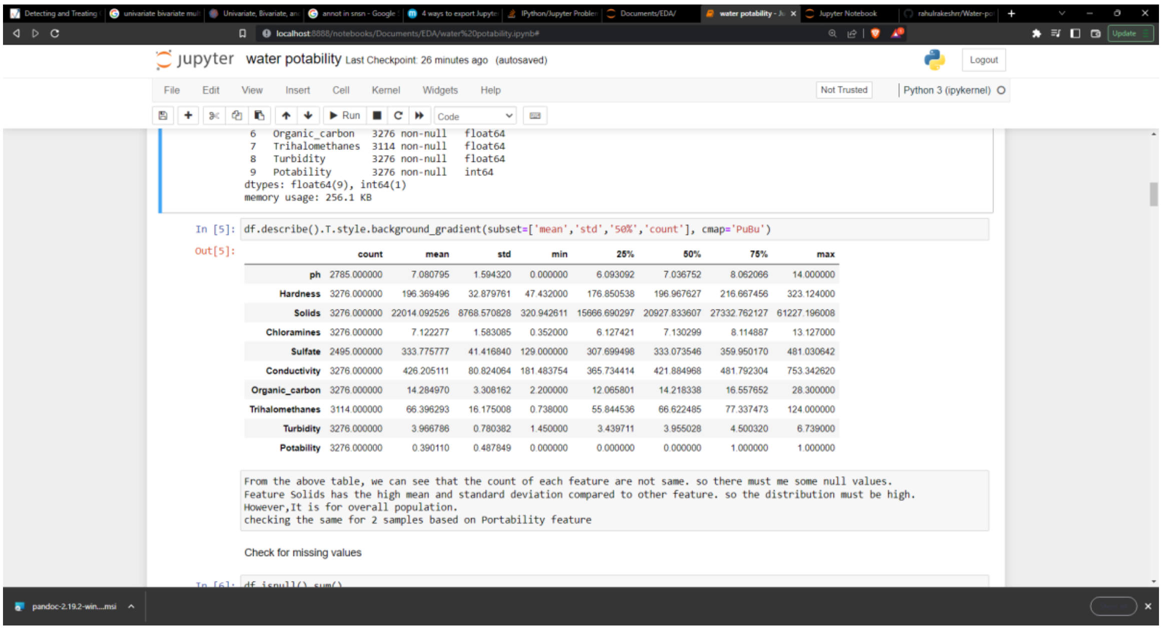
	ph	Hardness	Solids	Chloramines	Sulfate	Conductivity	Organic_carbon	Trihalomethanes	Turbidity	Potability
62	NaN	229.485694	35729.692709	8.810843	384.943779	296.397547	16.927092	NaN	3.855602	0
81	5.519126	168.728583	12531.601921	7.730723	NaN	443.570372	18.099078	NaN	3.758996	0
110	9.286155	222.661551	12311.268366	7.289866	332.239359	353.740100	14.171763	NaN	5.239982	0
118	7.397413	122.541040	8855.114121	6.888689	241.607532	489.851600	13.365906	NaN	3.149158	0
119	7.812804	196.583886	42550.841816	7.334648	NaN	442.545775	14.666917	NaN	6.204846	0
...	...	...	...	...	...	...	...	...	...	...
3174	6.698154	198.286268	34675.862845	6.263602	360.232834	430.935009	12.176678	NaN	3.758180	1
3185	6.110022	234.800957	16663.539074	5.984536	348.055211	437.892115	10.059523	NaN	2.817780	1
3219	6.417716	209.702425	31974.481631	7.263425	321.382124	289.450118	11.369071	NaN	4.210327	1
3259	9.271355	181.259617	16540.979048	7.022499	309.238865	487.692788	13.228441	NaN	4.333953	1
3272	7.808856	193.553212	17329.802160	8.061362	NaN	392.449580	19.903225	NaN	2.798243	1

162 rows × 10 columns

Since the missing values are on both classess (Potability 1 & 0), we can replace it with population mean. so, we will replace the Nan values bases on sample mean from both classes

- # Data Cleaning:

There are 9 Independent features and 1 dependent features. Details for each features are provided in the above top. objective is to use the Potability feature as target feature for classification problem.



```
 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
```

In [5]:
```
df.describe().T.style.background_gradient(subset=['mean','std','50%','count'], cmap='PuBu')
```

Out[5]:

	count	mean	std	min	25%	50%	75%	max
ph	2785.000000	7.080795	1.594320	0.000000	6.093092	7.036752	8.062066	14.000000
Hardness	3276.000000	196.369496	32.879761	47.432000	176.850538	196.967627	216.667456	323.124000
Solids	3276.000000	22014.092526	8768.570828	320.942611	15666.690297	20927.833607	27332.762127	61227.196008
Chloramines	3276.000000	7.122277	1.583085	0.352000	6.127421	7.130299	8.114887	13.127000
Sulfate	2495.000000	333.775777	41.416840	129.000000	307.699498	333.073546	359.950170	481.030642
Conductivity	3276.000000	426.205111	80.824064	181.483754	365.734414	421.884968	481.792304	753.342620
Organic_carbon	3276.000000	14.284970	3.308162	2.200000	12.065801	14.218338	16.557652	28.300000
Trihalomethanes	3114.000000	66.396293	16.175008	0.738000	55.844536	66.622485	77.337473	124.000000
Turbidity	3276.000000	3.966786	0.780382	1.450000	3.439711	3.955028	4.500320	6.739000
Potability	3276.000000	0.390110	0.487849	0.000000	0.000000	0.000000	1.000000	1.000000

```
From the above table, we can see that the count of each feature are not same. so there must me some null values.
Feature Solids has the high mean and standard deviation compared to other feature. so the distribution must be high.
However,It is for overall population.
checking the same for 2 samples based on Portability feature
```

Check for missing values

jupyter water potability Last Checkpoint: 26 minutes ago (autosaved)

```
However,It is for overall population.
checking the same for 2 samples based on Portability feature
```

Check for missing values

In [6]:
```
df.isnull().sum()
```

Out[6]:
```
ph                 491
Hardness             0
Solids               0
Chloramines          0
Sulfate            781
Conductivity         0
Organic_carbon       0
Trihalomethanes    162
Turbidity            0
Potability           0
dtype: int64
```

Features ph, Sulfate and Trihalomethanes are having null values. we shouldn't remove all null data because if we remove then most of our data will be lost.

Filling null values

Since the missing values are on both classess (Potability 1 & 0), we can replace it with population mean. so, we will replace the Nan values bases on sample mean from both classes

In [7]:
```
df.fillna(df.mean(),inplace=True)
```

In [8]:
```
df
```

Out[8]:

	ph	Hardness	Solids	Chloramines	Sulfate	Conductivity	Organic_carbon	Trihalomethanes	Turbidity	Potability

jupyter water potability Last Checkpoint: 26 minutes ago (autosaved)

Since the missing values are on both classess (Potability 1 & 0), we can replace it with population mean. so, we will replace the Nan values bases on sample mean from both classes

In [7]:
```
df.fillna(df.mean(),inplace=True)
```

In [8]:
```
df
```


Out[8]:

	ph	Hardness	Solids	Chloramines	Sulfate	Conductivity	Organic_carbon	Trihalomethanes	Turbidity	Potability
0	7.080795	204.890455	20791.318981	7.300212	368.516441	564.308654	10.379783	86.990970	2.963135	0
1	3.716080	129.422921	18630.057858	6.635246	333.775777	592.885359	15.180013	56.329076	4.500656	0
2	8.099124	224.236259	19909.541732	9.275884	333.775777	418.606213	16.868637	66.420093	3.055934	0
3	8.316766	214.373394	22018.417441	8.059332	356.886136	363.266516	18.436524	100.341674	4.628771	0
4	9.092223	181.101509	17978.986339	6.546600	310.135738	398.410813	11.558279	31.997993	4.075075	0
...	...	...	...	...	...	...	...	...	...	...
3271	4.668102	193.681735	47580.991603	7.166639	359.948574	526.424171	13.894419	66.687695	4.435821	1
3272	7.808856	193.553212	17329.802160	8.061362	333.775777	392.449580	19.903225	66.396293	2.798243	1
3273	9.419510	175.762646	33155.578218	7.350233	333.775777	432.044783	11.039070	69.845400	3.298875	1
3274	5.126763	230.603758	11983.869376	6.303357	333.775777	402.883113	11.168946	77.488213	4.708658	1
3275	7.874671	195.102299	17404.177061	7.509306	333.775777	327.459760	16.140368	78.698446	2.309149	1

3276 rows × 10 columns

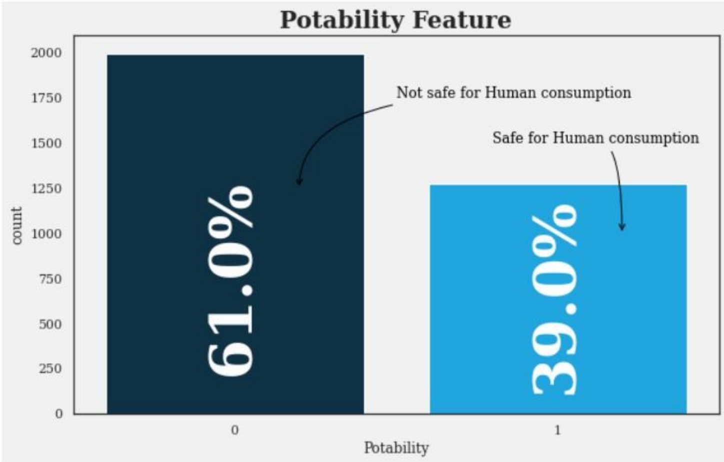
Exploratory Data Analysis

In [15]:
```
df.describe()
```

Out[15]:

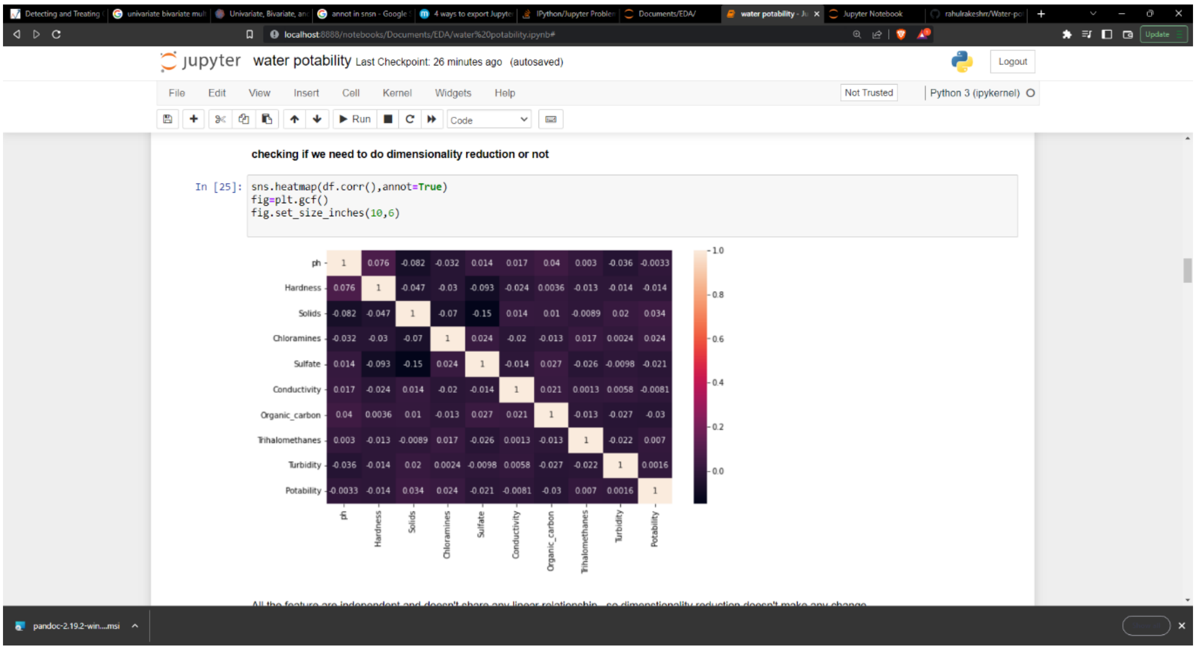
- # Exploratory Data Analysis

Out[31]: Text(0.8, 1500, 'Safe for Human consumption')



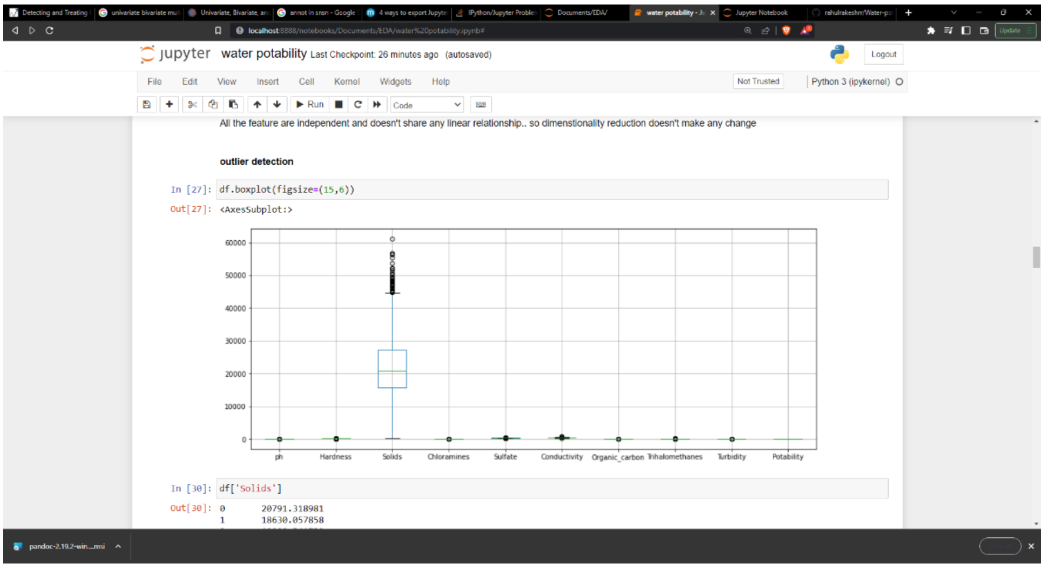
- There is imbalance in the Target variable. which should be considered for modelling.

  - **checking if we need to do dimensionality reduction or not**



- All the feature are independent and doesn't share any linear relationship. so dimensionality reduction doesn't make any change

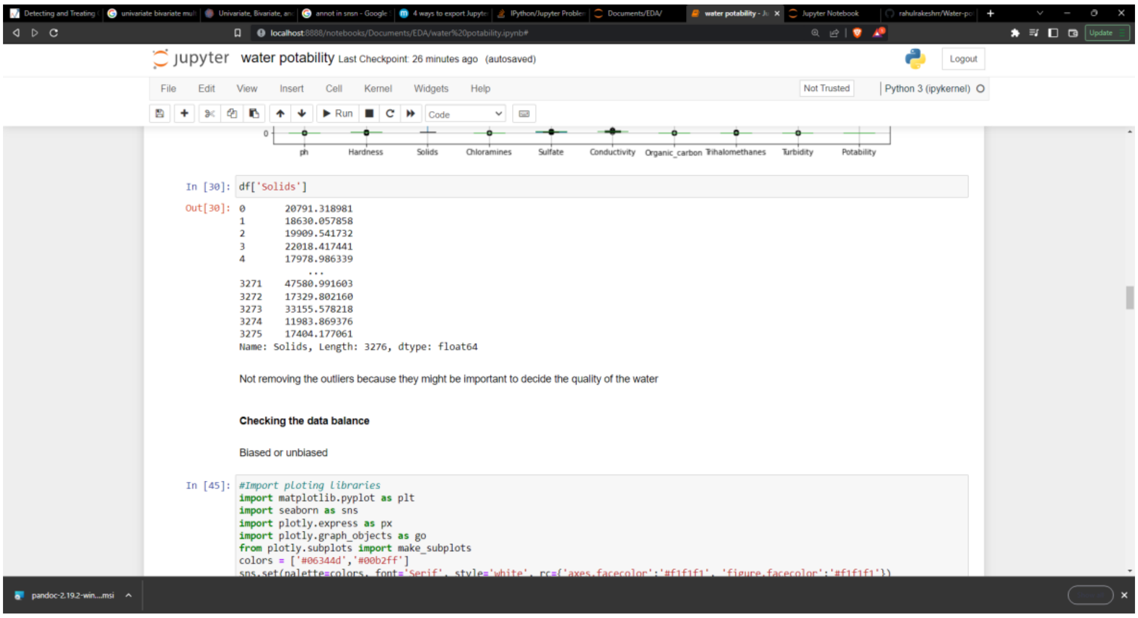
Outlier detection:

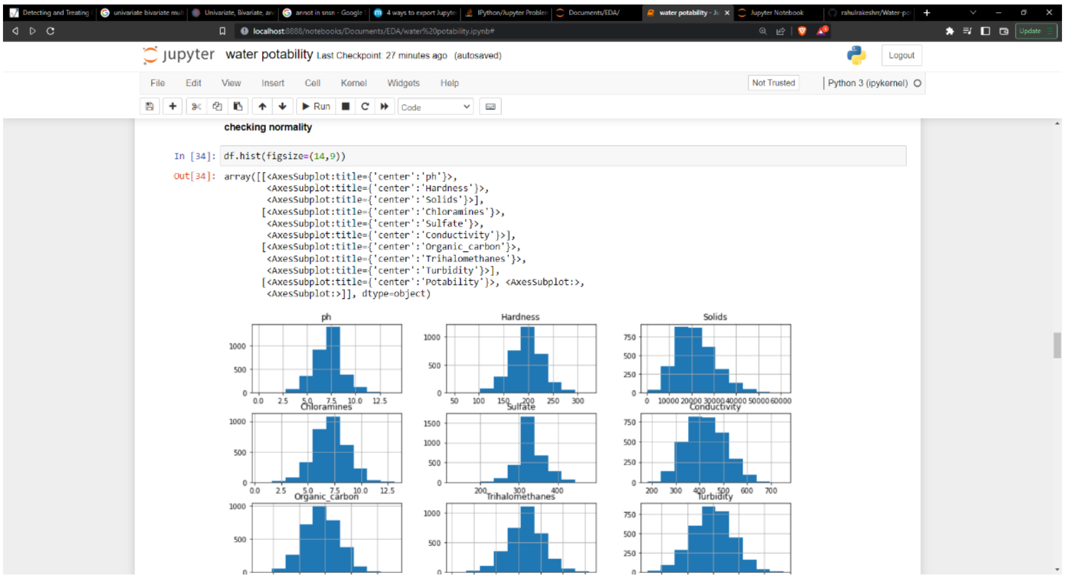


- Not removing the outliers because there are lot of outliers if removed effects the data and they might be important to decide the quality of the water

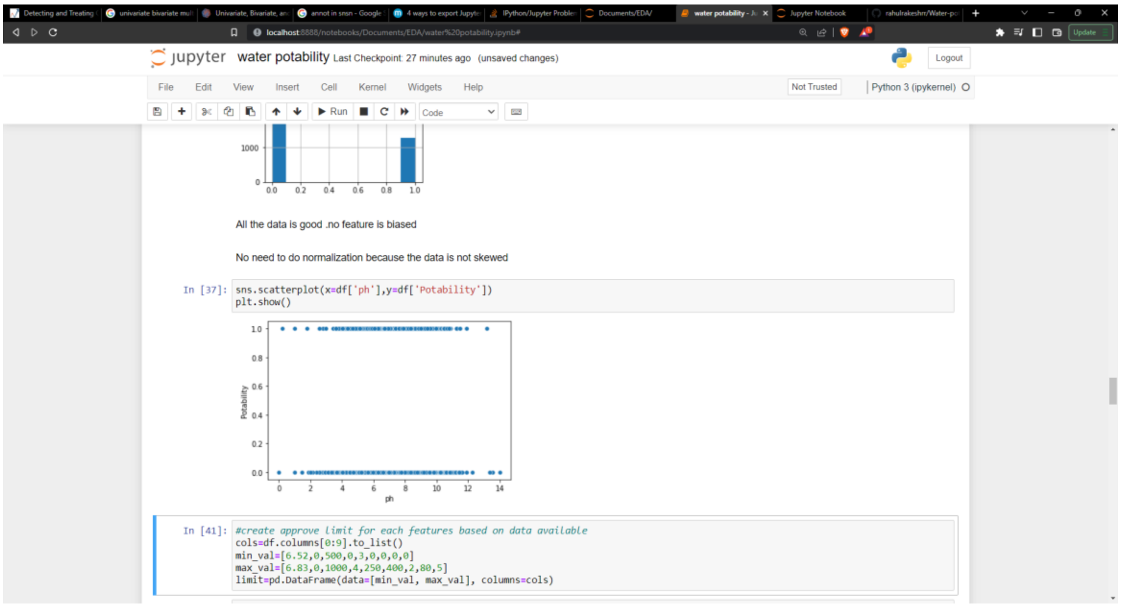
Checking the data balance:

Biased or unbiased

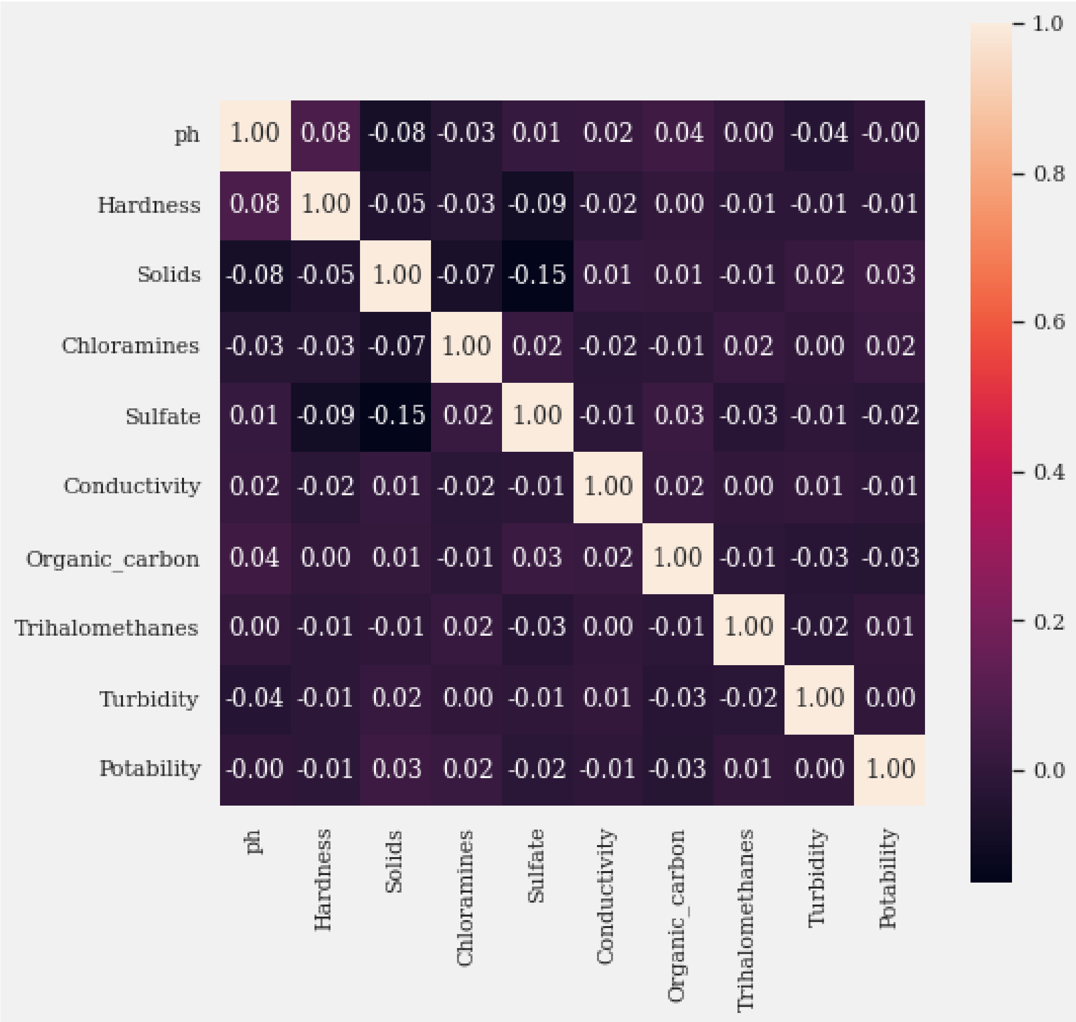


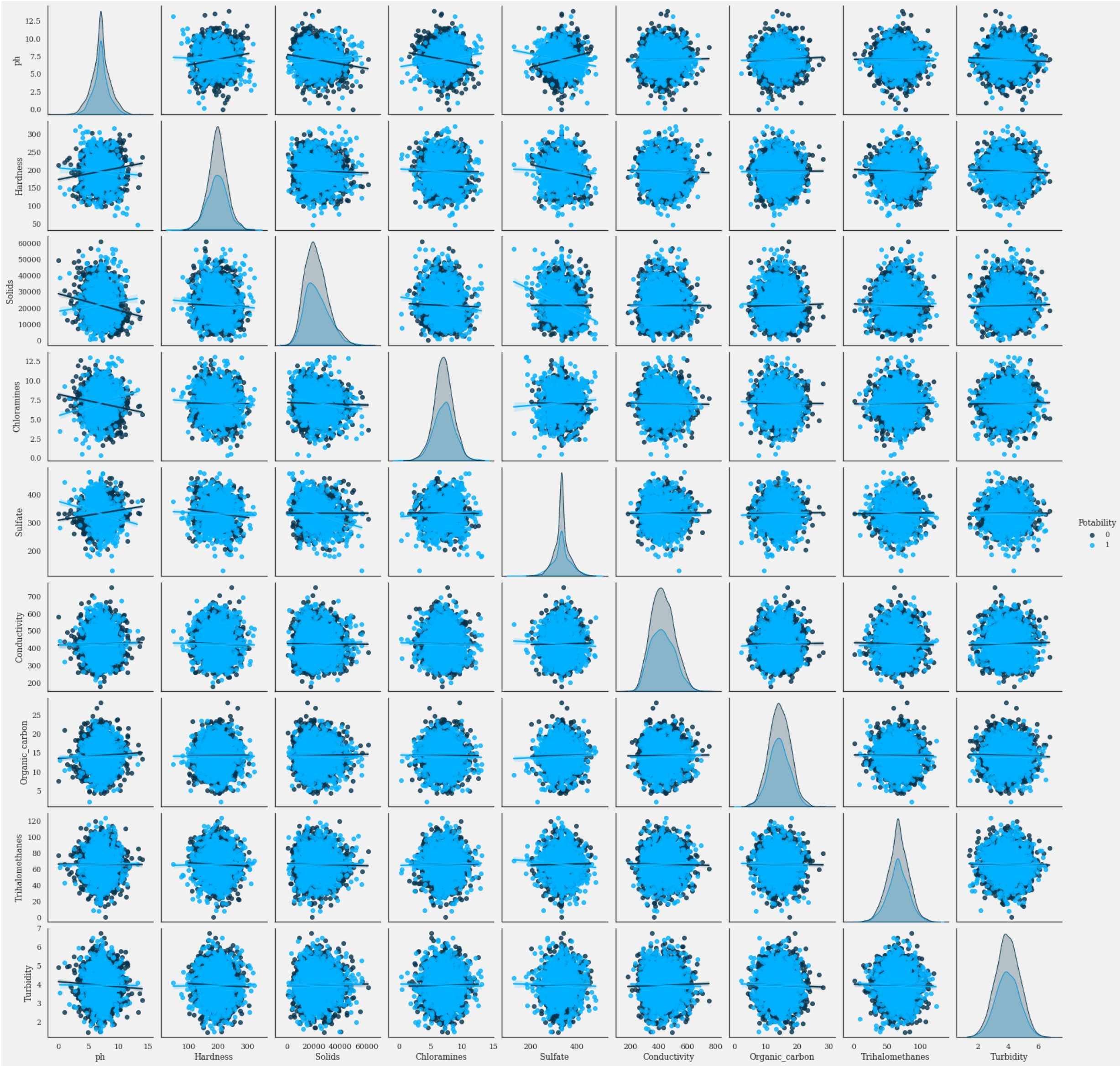


- All the data is good .no feature is biased
- No need to do normalization because the data is not skewed

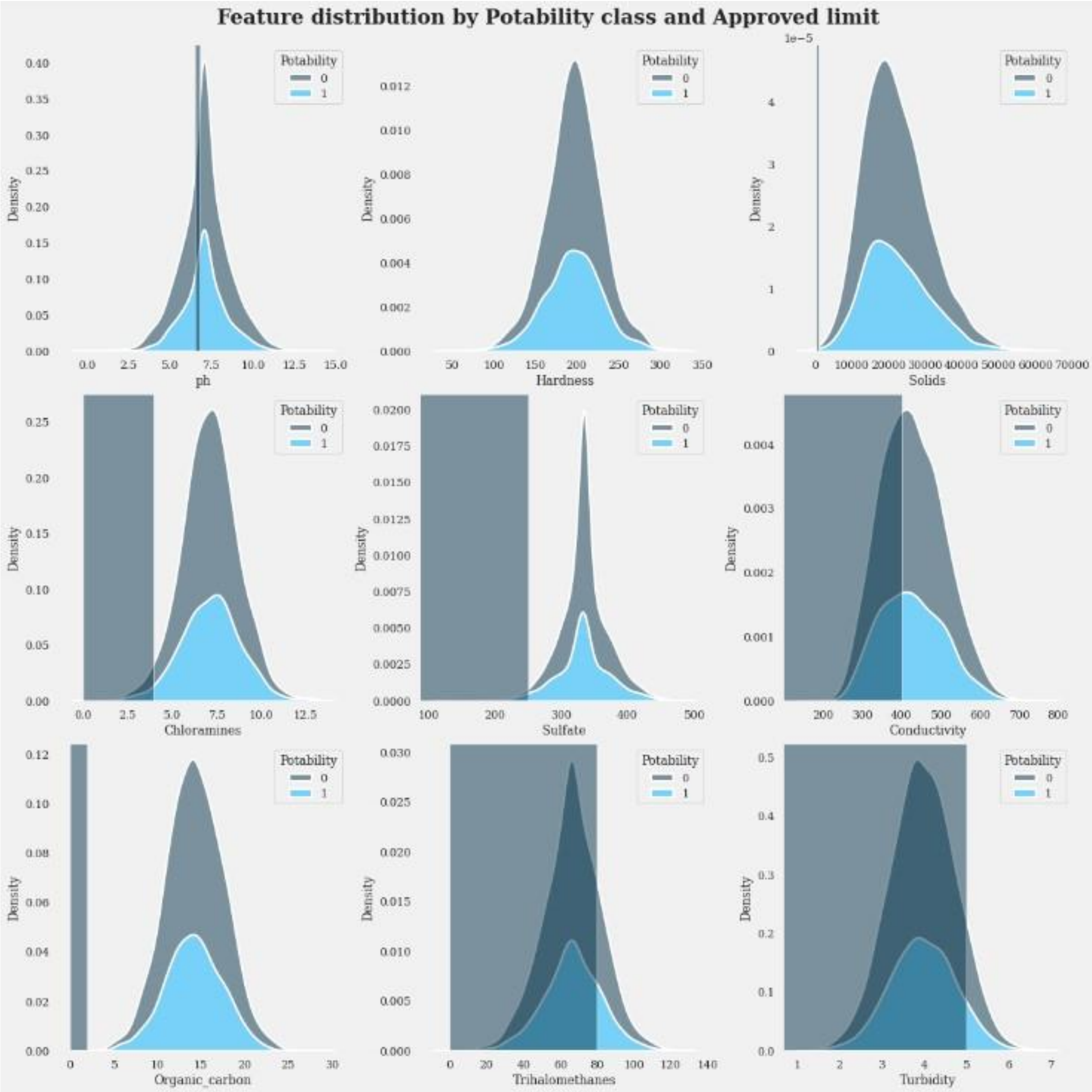


Univariate and Multivariate Analysis:





- Both Co-relation matrix & Pair plot says that there is no linear relationship between the features that can explain the target variable. So, Linear model may not work on this problem. we need to try with probability-based models



Based on the approved limit, we can clearly see the difference in the water classification.

Distribution of non potable water is high on conductivity compared to potable water. same applicable to Turbidity, Trihalomethanes. But, Ph value, Chloramines, Sulfate, Organic carbon presence doesn't show significant difference. I hope the hypothetical testing can help us here.

- Based on the approved limit, we can clearly see the difference in the water classification.
- Ex: distribution of non-potable water is high on conductivity compared to potable water. same applicable to Turbidity, Trihalomethanes. But, Ph value, Chloramines, Sulphate, Organic carbon presence.

Machine Learning Models

Partitioning

In [54]:
```
x = df.drop('Potability',axis=1) #input data
```

In [61]:
```
y = df['Potability']
```

In [62]:
```
from sklearn.model_selection import train_test_split
```

In [99]:
```
x_train , x_test , y_train , y_test =train_test_split(x,y,test_size=0.1,shuffle=True,random_state=None)
```

Model Training

Decision Tree

In [100]:
```
from sklearn.tree import DecisionTreeClassifier
dtc = DecisionTreeClassifier()
```

In [101]:
```
dtc.fit(x_train,y_train) # model training
```

Out[101]: DecisionTreeClassifier()

In [102]:
```
x_test
```

Out[102]:

	ph	Hardness	Solids	Chloramines	Sulfate	Conductivity	Organic_carbon	Trihalomethanes	Turbidity
2649	5.735359	149.171690	28628.151856	5.073074	333.775777	465.554894	12.201950	65.909850	4.085055
914	5.791154	190.431679	20288.235569	6.298366	347.041020	544.857481	17.706838	84.388191	4.104835
144	4.965774	220.848639	21802.744898	6.250578	311.598136	386.413332	18.035238	53.908177	3.421767
1865	8.822610	216.432576	27109.239652	7.322163	333.775777	384.018547	15.955491	78.828883	4.415596
261	4.815767	217.687093	16392.135069	7.461170	278.742296	481.480828	15.517296	77.693374	4.375224

In [106]:
```
accuracy_score(y_prediction,y_test)
```

Out[106]: 0.5975609756097561

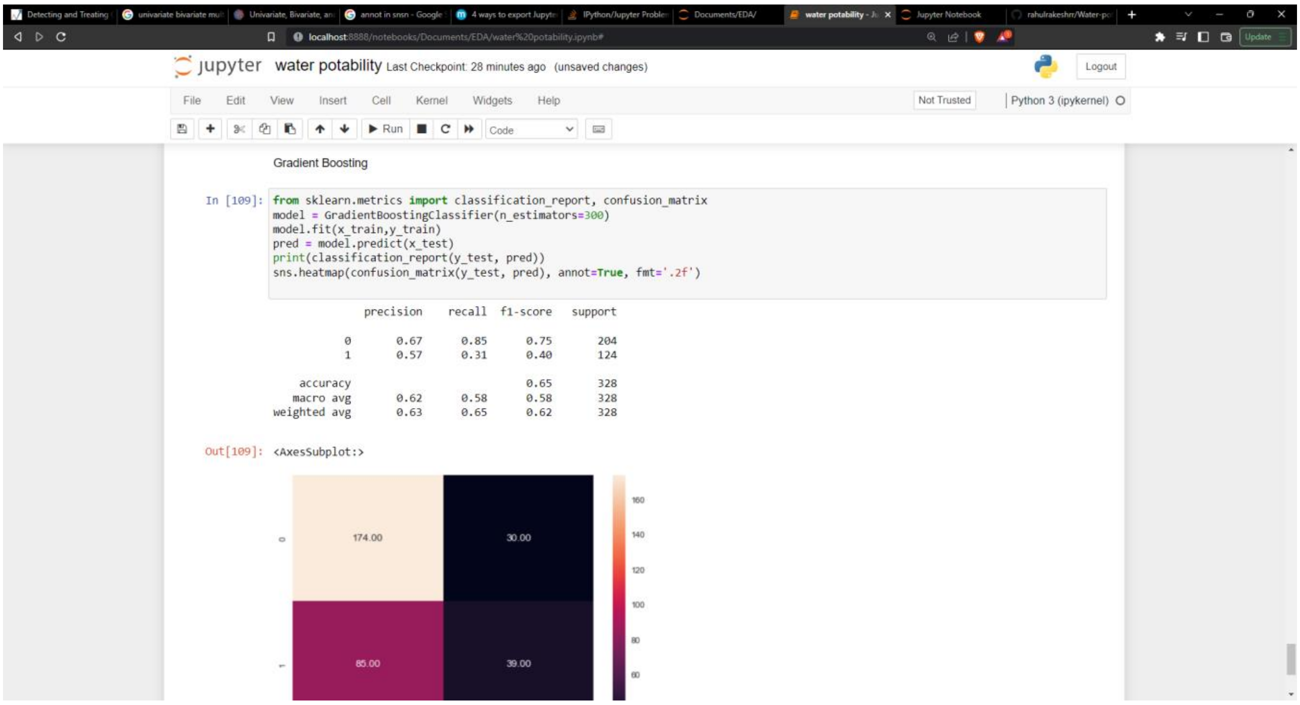
In [107]:
```
from yellowbrick.classifier import ROCAUC
from sklearn.ensemble import AdaBoostClassifier, BaggingClassifier, GradientBoostingClassifier
from sklearn.tree import DecisionTreeClassifier, ExtraTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import cross_val_score
mod = []
cv_score=[]

model =[AdaBoostClassifier(), BaggingClassifier(), GradientBoostingClassifier(),
        DecisionTreeClassifier(), ExtraTreeClassifier(), KNeighborsClassifier()]

for m in model:
    cv_score.append(cross_val_score(m, x_train, y_train, scoring='accuracy', cv=5).mean())
    mod.append(m)

model_df=pd.DataFrame(columns=['model','cv_score'])
model_df['model']=mod
model_df['cv_score']=cv_score
model_df.sort_values(by=['cv_score'], ascending=True).style.background_gradient(subset=['cv_score'])
```

Out[107]:

	model	cv_score
5	KNeighborsClassifier()	0.548186
4	ExtraTreeClassifier()	0.567843
3	DecisionTreeClassifier()	0.592931
0	AdaBoostClassifier()	0.616009
1	BaggingClassifier()	0.654343
2	GradientBoostingClassifier()	0.656715

From the above basic modeling techinique, average cv score for GradientBoosting & BaggingClassifier is high compared to other models.

- From the above basic modeling technique, average cv score for Gradient Boosting & Bagging Classifier is high compared to other models



- I hope this is good model for initial analysis. further fine tuning and outlier handling might help for more accuracy.

Conclusion

- **Challenges –**
  It was difficult to identify the right feature to predict the right value. As there is no good correlation between features.

  The most challenging part is finding the best model.

- **It can be observed that the model which yields the most accurate result is Gradient boosting**

- **if Ph value is high that makes the water not potable, other features don't matter. If one feature makes water non-drinkable, other features shouldn't matter. Maybe the features are just indicators for the interaction of other substances.**

References:
https://www.kaggle.com/datasets/adityakadiwal/water-potability

GitHub:
https://github.com/rahulrakeshrr/Water-potability