Open in Colab

(https://colab.research.google.com/github/mohameddhameem/IBM-Machine-Learning/blob/master/Exploratory%20Data%20Analysis%20for%20Machine%20Learning/Final_Assignment.i

Summary - Exploratory Data Analysis for Machine Learning Assesement

Access to Safe drinking water is eassently a global issue. The World Health Organization (WHO) estimates that half of all people in the world are affected by the lack of safe drinking water. With this assesment, we will explore the data and look for patterns in the data to analyze if the given data is a good indicator of safe drinking water.

In [20]:

```
# Import all required libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from scipy import stats
np.warnings.filterwarnings('ignore', category=np.VisibleDeprecationWarning)
#sns.set_context('notebook')
```

Data Set

The dataset is downloaded from kaggle.com and is available for download at:

https://www.kaggle.com/adityakadiwal/water-potability (https://www.kaggle.com/adityakadiwal/water-potability)

EDA - Exploratory Data Analysis

In this section we will explore the data and look for patterns in the data to analyze if the given data is a good indicator of safe drinking water.

1) Describe the data

2) Visualize the data

3) Identify the missing values and fill them

4) Identify the outliers and remove them

5) Identify the categorical variables and encode them (if any)

6) Identify the numerical variables and perform basic statistical analysis

In [2]:

```
# File is stored in github repository for easiness of access
INPUT_FILE_PATH = "https://raw.githubusercontent.com/mohameddhameem/IBM-Machine-Learning/master/Exploratory%20Data%20Analysis%20for%20Machine%20Learning/water_potability.csv"
```

In [3]:

```
# Read the csv file from the url
df = pd.read_csv(INPUT_FILE_PATH)
```

In [4]:

```
# Print the first 5 rows of the dataframe
display(df.head())
```

	ph	Hardness	Solids	Chloramines	Sulfate	Conductivity	Organic_carbor
0	NaN	204.890455	20791.318981	7.300212	368.516441	564.308654	10.37978
1	3.716080	129.422921	18630.057858	6.635246	NaN	592.885359	15.18001
2	8.099124	224.236259	19909.541732	9.275884	NaN	418.606213	16.86863
3	8.316766	214.373394	22018.417441	8.059332	356.886136	363.266516	18.43652
4	9.092223	181.101509	17978.986339	6.546600	310.135738	398.410813	11.55827

More information about the data

ph - 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.

Hardness - Hardness is a measure of the physical properties of the water. It is a measure of the ability of the water to support the roots and the leaves. The lower the hardness, the more support the roots and leaves can have.

Solids (Total dissolved solids - TDS) - TDS is a measure of the solids in the 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.

Chloramines - Chlorine and chloramine are the major disinfectants used in public water systems. Chloramines are most commonly 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.

Sulfate - Sulfate is a common disinfectant used in public water systems. Sulfate levels up to 2 milligrams per liter (mg/L or 2 parts per million (ppm)) are considered safe in drinking water.

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

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.

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.

Turbidity - Turbidity is a measure of the water's ability to absorb particulate matter. The lower the turbidity, the more it can absorb particulate matter.

Potability (Target variable) - Indicates if water is safe for human consumption where 1 means Potable and 0 means Not potable.

In [5]:

```
# datatypes of the columns
display(df.dtypes)
```

```
ph                 float64
Hardness           float64
Solids             float64
Chloramines        float64
Sulfate            float64
Conductivity       float64
Organic_carbon     float64
Trihalomethanes    float64
Turbidity          float64
Potability           int64
dtype: object
```

In [6]:

```
# Describe the data
df.describe()
```

Out[6]:

	ph	Hardness	Solids	Chloramines	Sulfate	Conductivity	Orga
count	2785.000000	3276.000000	3276.000000	3276.000000	2495.000000	3276.000000	3
mean	7.080795	196.369496	22014.092526	7.122277	333.775777	426.205111	
std	1.594320	32.879761	8768.570828	1.583085	41.416840	80.824064	
min	0.000000	47.432000	320.942611	0.352000	129.000000	181.483754	
25%	6.093092	176.850538	15666.690297	6.127421	307.699498	365.734414	
50%	7.036752	196.967627	20927.833607	7.130299	333.073546	421.884968	
75%	8.062066	216.667456	27332.762127	8.114887	359.950170	481.792304	
max	14.000000	323.124000	61227.196008	13.127000	481.030642	753.342620	

In [7]:

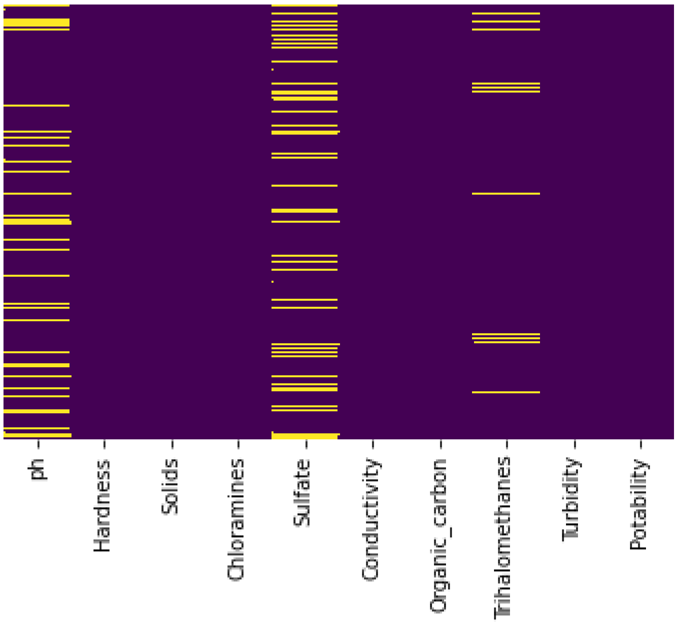
```
# Check if there are any null columns
df.isnull().sum()
```

Out[7]:

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

In [8]:

```
# Lets try to plot misisng values
sns.heatmap(df.isnull(),yticklabels=False,cbar=False,cmap='viridis')
plt.show()
```

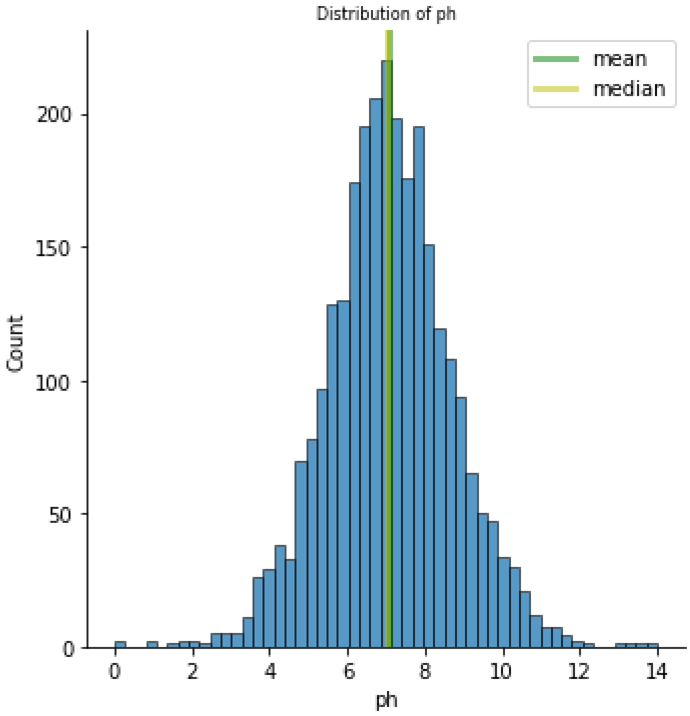


Analyze ph column

In [9]:

```
# for  ph column
# set the histogram, mean and median
sns.displot(df["ph"], kde=False)
plt.axvline(x=df.ph.mean(), linewidth=3, color='g', label="mean", alpha=0.5)
plt.axvline(x=df.ph.median(), linewidth=3, color='y', label="median", alpha=0.5)

plt.xlabel("ph")
plt.ylabel("Count")
plt.title("Distribution of ph", size=8)
plt.legend(["mean", "median"])
plt.show()
```



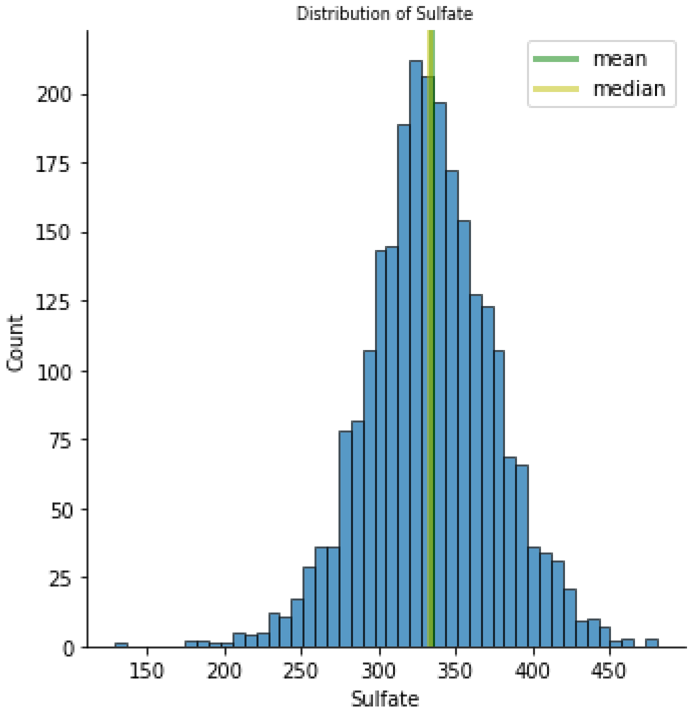
Based on the above data, we can impute ph with either mean or median. There is no skweness in the data.

Analyze Sulfate column

In [10]:

```
sns.displot(df["Sulfate"], kde=False)
plt.axvline(x=df.Sulfate.mean(), linewidth=3, color='g', label="mean", alpha=0.5
)
plt.axvline(x=df.Sulfate.median(), linewidth=3, color='y', label="median", alpha
=0.5)

plt.xlabel("Sulfate")
plt.ylabel("Count")
plt.title("Distribution of Sulfate", size=8)
plt.legend(["mean", "median"])
plt.show()
```



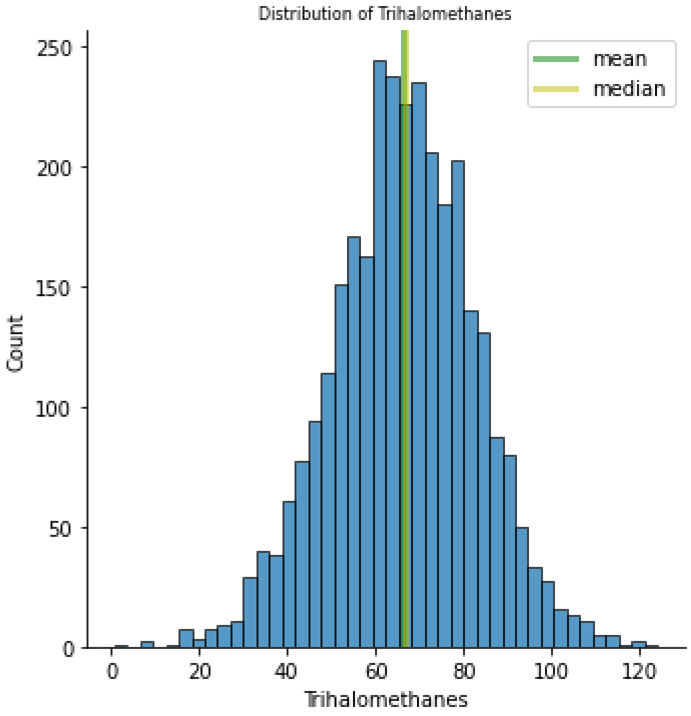
Based on the above data, we can impute Sulphate with either mean or median.

Analyze Trihalomethanes column

In [11]:

```
sns.displot(df["Trihalomethanes"], kde=False)
plt.axvline(x=df.Trihalomethanes.mean(), linewidth=3, color='g', label="mean", alpha=0.5)
plt.axvline(x=df.Trihalomethanes.median(), linewidth=3, color='y', label="median", alpha=0.5)

plt.xlabel("Trihalomethanes")
plt.ylabel("Count")
plt.title("Distribution of Trihalomethanes", size=8)
plt.legend(["mean", "median"])
plt.show()
```



Based on the above data, we can impute Trihalomethanes with either mean or median.

Missing Value imputation

Missing values in ph column

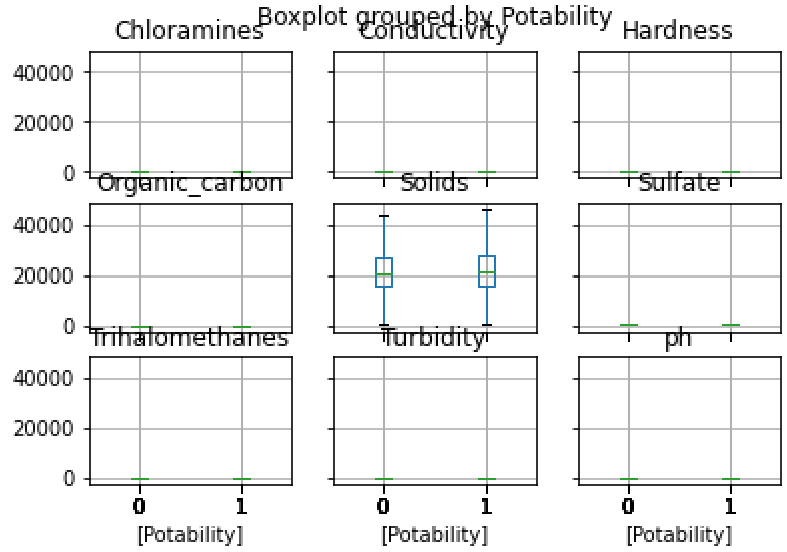
In [12]:

```
# impute missing values with mean
df["ph"] = df["ph"].fillna(df["ph"].mean())
df["Sulfate"] = df["Sulfate"].fillna(df["Sulfate"].mean())
df["Trihalomethanes"] = df["Trihalomethanes"].fillna(df["Trihalomethanes"].mean())
```

Identify outliers in the data

In [13]:

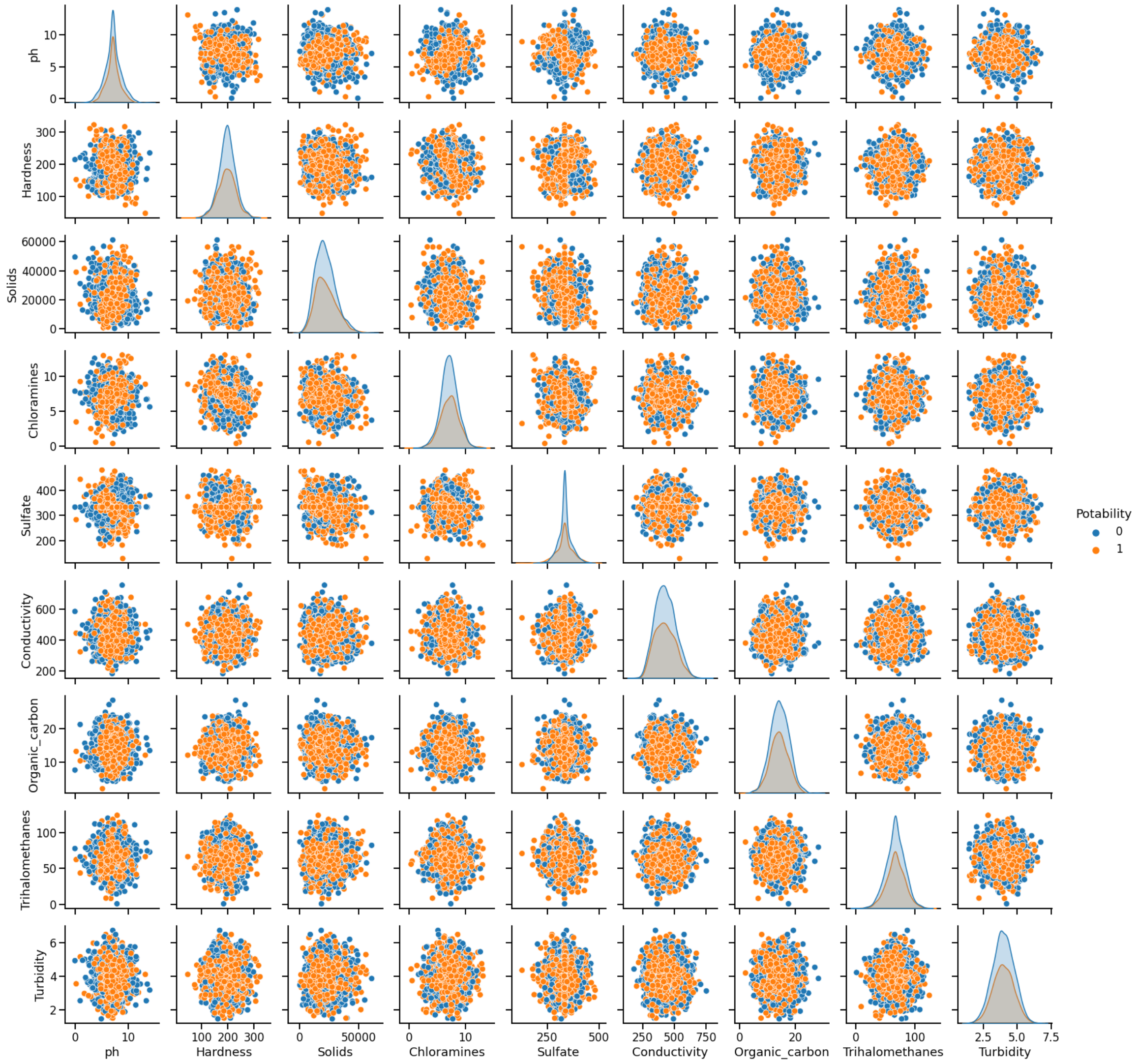
```
# check outliers
df.boxplot(by="Potability", showfliers=False)
plt.show()
```



Identify corrleation between variables

In [14]:

```
sns.set_context('talk')
sns.pairplot(df, hue='Potability')
plt.show()
```



There are no categorical variables in the dataset.

Identify skweness in the data

In [15]:

```
# identify skewness
mask = df.dtypes == np.float64
float_cols = df.columns[mask]

skew_limit = 0.75 # define a limit above which we will log transform
skew_vals = df[float_cols].skew()
# Showing the skewed columns
skew_cols = (skew_vals
             .sort_values(ascending=False)
             .to_frame()
             .rename(columns={0:'Skew'})
             .query('abs(Skew) > {}'.format(skew_limit)))

print('Number of skewed columns :', skew_cols.shape[0])
skew_cols
```

```
Number of skewed columns : 0
```

Out[15]:

Skew

There are no skew in our data :)

Lets see the distribution of Potability

In [31]:

```
df.Potability.value_counts()
```

Out[31]:

```
0    1998
1    1278
Name: Potability, dtype: int64
```

Feature Transformation

In [16]:

```
# print the dataframe head
df.head()
```

Out[16]:

	ph	Hardness	Solids	Chloramines	Sulfate	Conductivity	Organic_carbo
0	7.080795	204.890455	20791.318981	7.300212	368.516441	564.308654	10.37978
1	3.716080	129.422921	18630.057858	6.635246	333.775777	592.885359	15.18001
2	8.099124	224.236259	19909.541732	9.275884	333.775777	418.606213	16.86863
3	8.316766	214.373394	22018.417441	8.059332	356.886136	363.266516	18.43652
4	9.092223	181.101509	17978.986339	6.546600	310.135738	398.410813	11.55827

In [17]:

```
# Feature transformation
# scale the numeric columns
from sklearn.preprocessing import RobustScaler
scalar = RobustScaler()
df[float_cols] = scalar.fit_transform(df[float_cols])
```

In [18]:

```
# After transformation print the dataframe head
df.head()
```

Out[18]:

	ph	Hardness	Solids	Chloramines	Sulfate	Conductivity	Organic_carbon	Trih
0	0.000000	0.198981	-0.011702	0.085492	1.043542	1.227178	-0.854560	
1	-2.113014	-1.696382	-0.196962	-0.249088	0.000000	1.473406	0.214093	
2	0.639503	0.684850	-0.087287	1.079558	0.000000	-0.028251	0.590024	
3	0.776180	0.437145	0.093483	0.467446	0.694190	-0.505079	0.939076	
4	1.263161	-0.398477	-0.252771	-0.293690	-0.710100	-0.202262	-0.592197	

Save the cleaned data

In [30]:

```
df.to_csv("cleaned_data.csv", index=False)
```

Hypothesis Testing

We define a hypothesis to test in our data set

Hypothesis 1:

Null: Increase in pH is associated with increase in Solids

Alternate : No relataion between ph and Solids

In [29]:

```
hle = np.array(df.ph)
ylwd = np.array(df.Solids)
stats.ttest_ind(hle, ylwd, equal_var = False)
```

Out[29]:

```
Ttest_indResult(statistic=-4.476932191647608, pvalue=7.7059403066192
21e-06)
```

the p value is less than 0.05 , so we are rejecting the null hypothesis at 5% significance level.

Next Step in analyzing the data

1. We have tested only with 1 hypothesis and eventually we can test with more hypotheseis
2. We can try to bin the data and test if binning gives same result in modeling

Quality of data

As per our understanding the data is clean and only minimal clean up is required for analysis. The cleaned data can be considered for further analysis and model building

Key findings

1. The dataset is mostly clean
2. We could see very few section of values exceeding the WHO mandated levels
3. The dataset is balanced interms of target Potability