Water quality Analysis Using Data Analytics With Cognos

BATCH MEMBERS

2021115021 – T Bhuvanaradja

2021115022 – D Cheran

2021115024 – C N Deepakraaj

2021115026 – S Dhanushkumar

2021115334 – R Vinoth kumar

2021115303 - J Jobson Charles

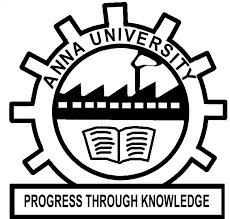
***In partial fulfillment for the award of the degree***

***Of***

**BACHELOR OF TECHNOLOGY**

***In***

**INFORMATION TECHNOLOGY**



**DEPARTMENT OF INFORMATION SCIENCE AND TECHNOLOGY**

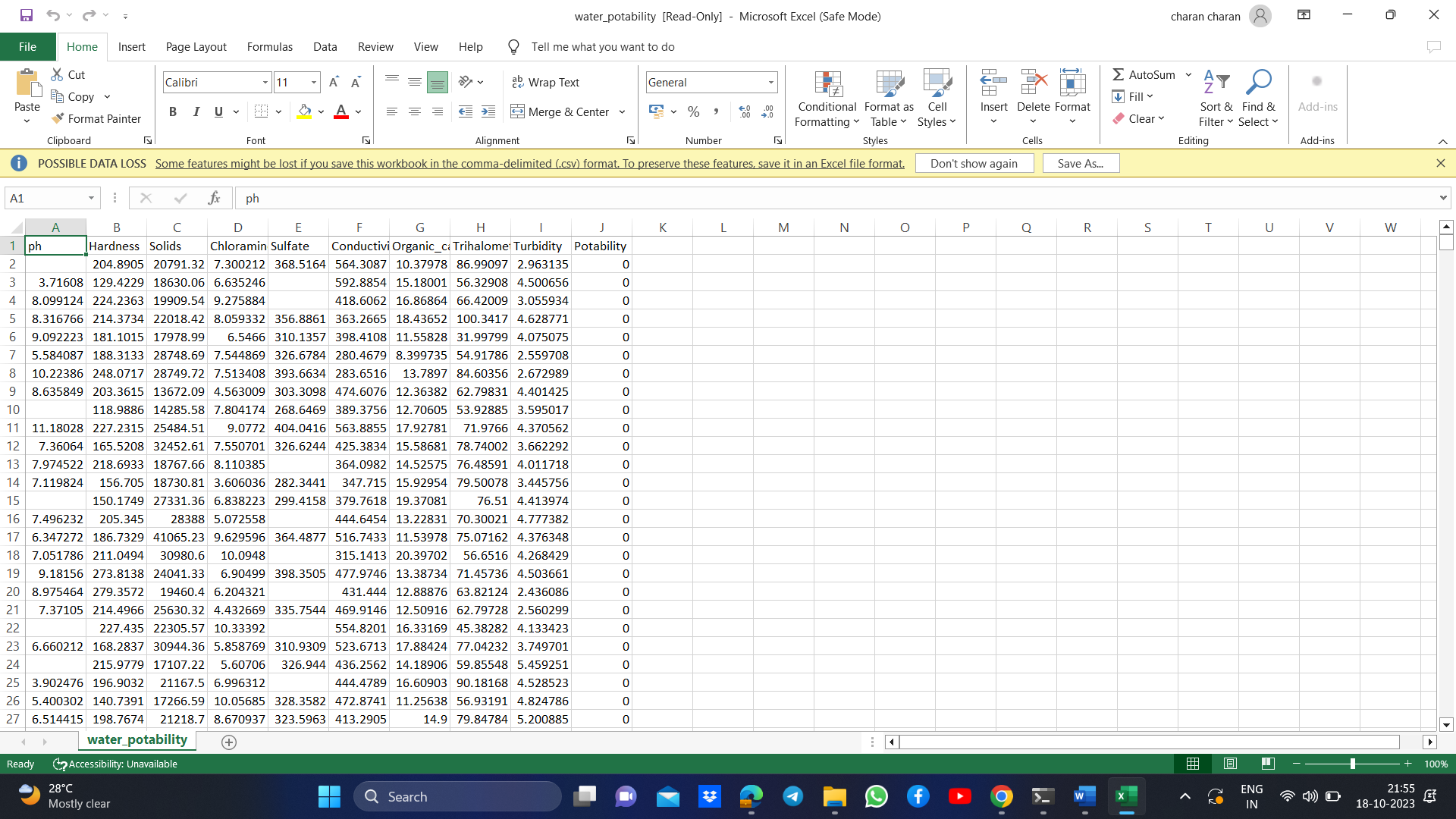
**COLLEGE OF ENGINEERING GUINDY**

**ANNA UNIVERSITY, CHENNAI 600025**

**OCTOBER 2023**

We'll explore how to import essential libraries, load the housing dataset, and perform critical preprocessing steps. Data preprocessing is crucial as it helps clean, format, and prepare the data for further analysis. This includes handling missing values, encoding categorical variables, and ensuring that the data is appropriately scaled.

Given data set:



Necessary step to follow:

1.Import libaries

Start by importing the necessary libraries:

Program:

import pandas as pd import numpy as np

import seaborn as sb

import matplotlib.pyplot as plt

import plotly.express as px

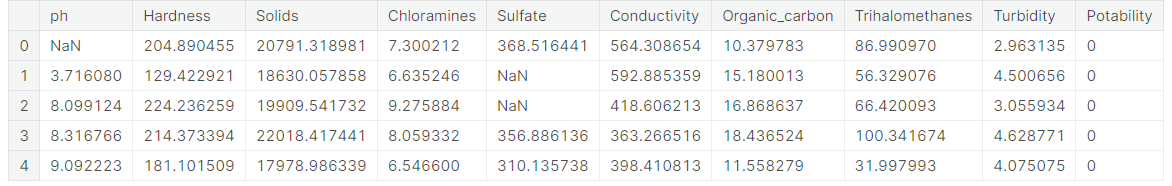
2. Read Dataset

Load your dataset into a Pandas DataFrame. You can typically find water potability datasets in CSV format, but you can adapt this code to other formats as needed.

df = pd.read\_csv('C:/Users/HP/OneDrive/Documents/water quality analysis/water\_potability.csv')

df.head()

output:

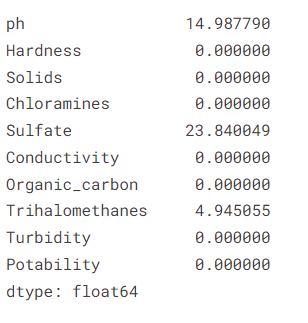


3. Exploratory Data Analysis (EDA)

 Perform EDA to understand your data better. This includes checking for missing values, exploring the data's statistics, andvisualizing it to identify patterns.

(df.isnull().sum()/df.shape[0])\*100

Output:

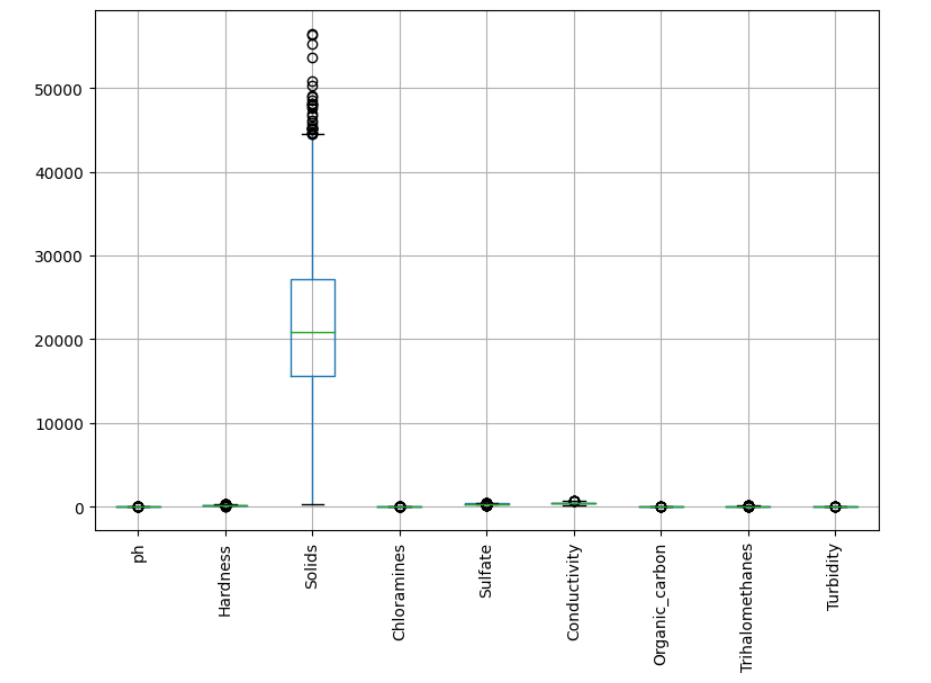


df.iloc[:,0:-1].boxplot()

plt.xticks(rotation=90)

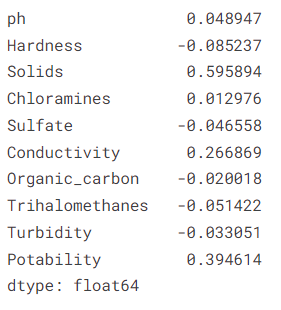
plt.show()

Output:



df.skew()

Output:

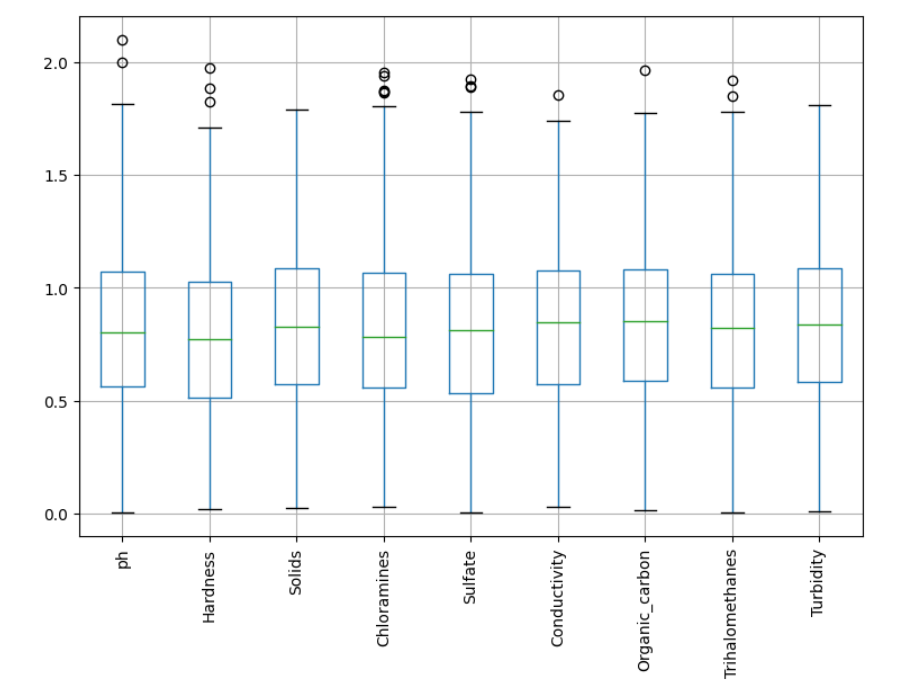


np.sqrt(df.iloc[:,0:-1]).boxplot()

plt.xticks(rotation=90)

plt.show()

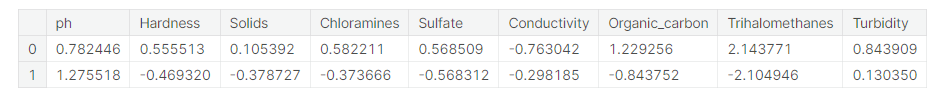
Output:



x = df[cols]

x.head(2)

Output:



from sklearn.cluster import KMeans

max\_range=7

wcss = []

for k **in** range(1,max\_range):

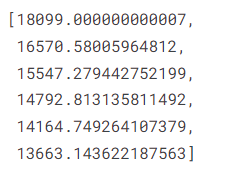
kmeans = KMeans(n\_clusters=k,random\_state=42)

model = kmeans.fit(x)

wcss.append(kmeans.inertia\_)

wcss

Output:

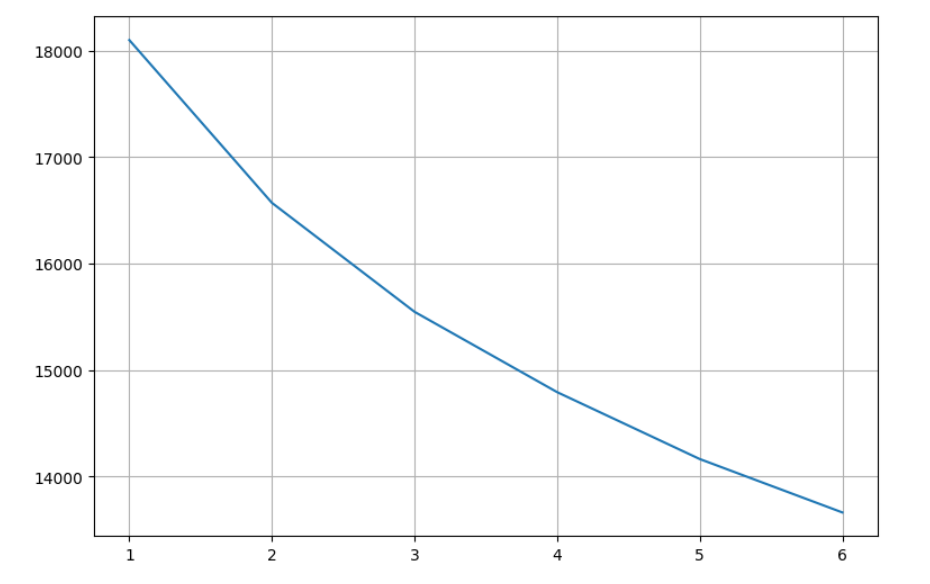


plt.plot(range(1,max\_range),wcss)

plt.grid()

plt.show()

Output:



kmeans = KMeans(n\_clusters=2,random\_state=42)

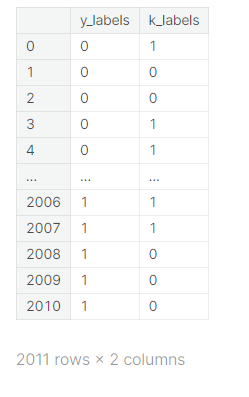
model = kmeans.fit(x)

model.labels\_

compare=pd.DataFrame({'y\_labels':df['Potability'],'k\_labels':model.labels\_})

compare

Output:



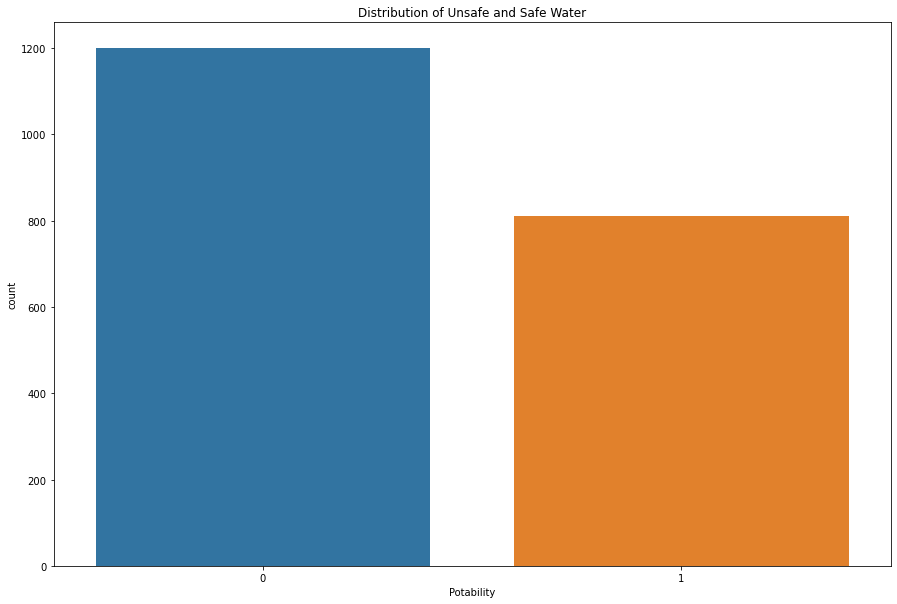
plt.figure(figsize=(15, 10))

sns.countplot(data.Potability)

plt.title("Distribution of Unsafe and Safe Water")

plt.show()

Output:



import plotly.express as px

data = data

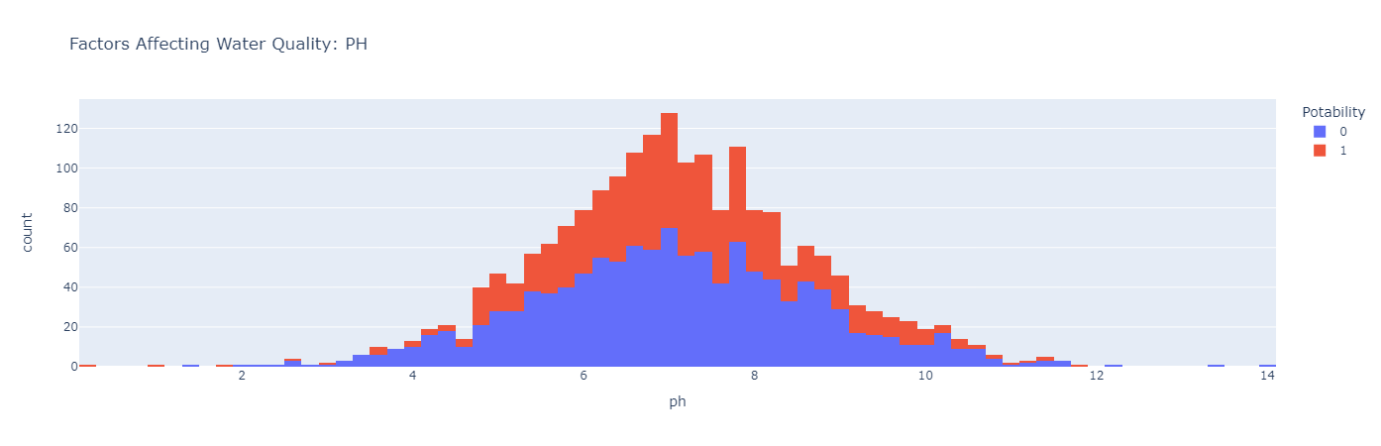
figure = px.histogram(data, x = "ph",

color = "Potability",

title= "Factors Affecting Water Quality: PH")

figure.show()

Output:



import plotly.express as px

data = data

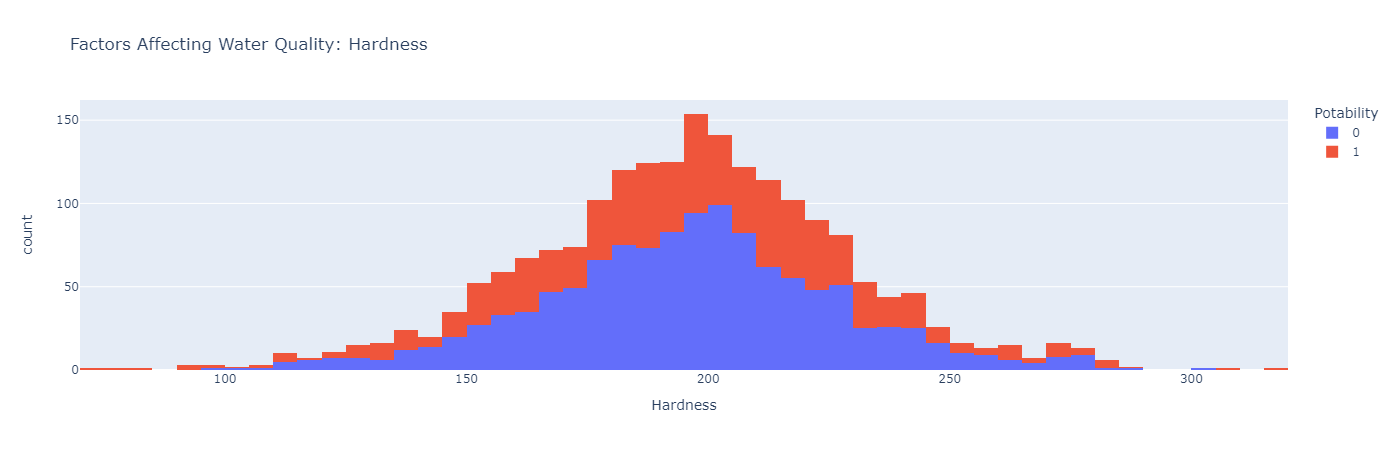
figure = px.histogram(data, x = "Hardness",

color = "Potability",

title= "Factors Affecting Water Quality: Hardness")

figure.show()

Output:



import plotly.express as px

data = data

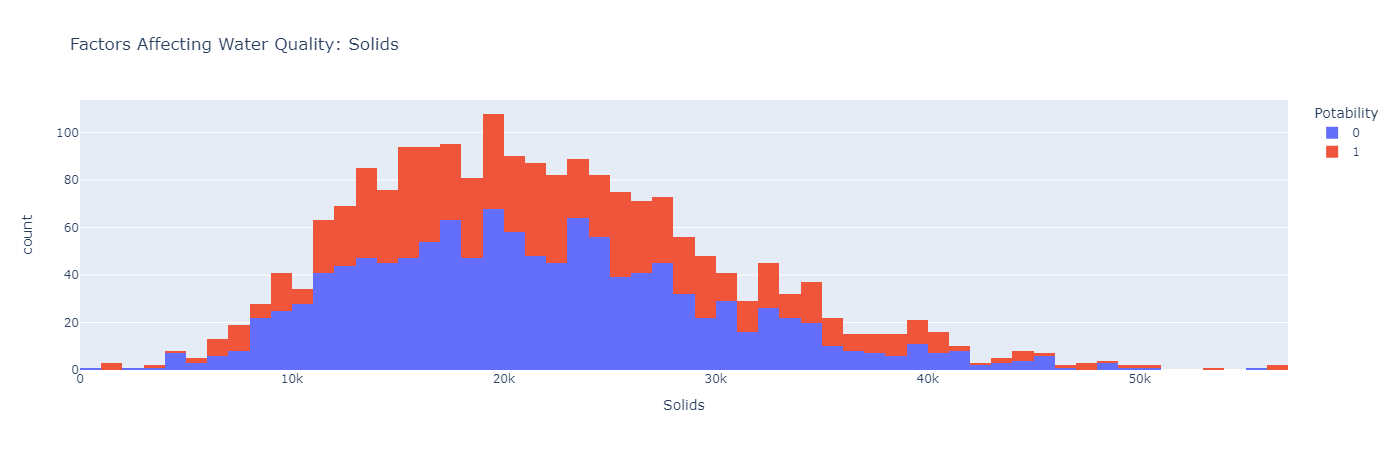
figure = px.histogram(data, x = "Solids",

color = "Potability",

title= "Factors Affecting Water Quality: Solids")

figure.show()

Output:



import plotly.express as px

data = data

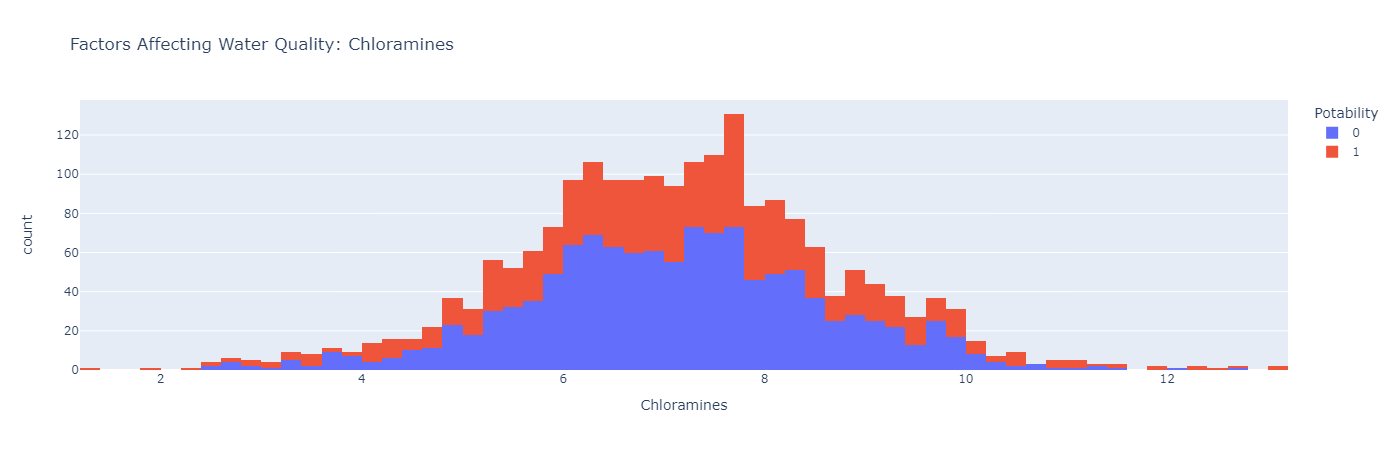
figure = px.histogram(data, x = "Chloramines",

color = "Potability",

title= "Factors Affecting Water Quality: Chloramines")

figure.show()

Output:



import plotly.express as px

data = data

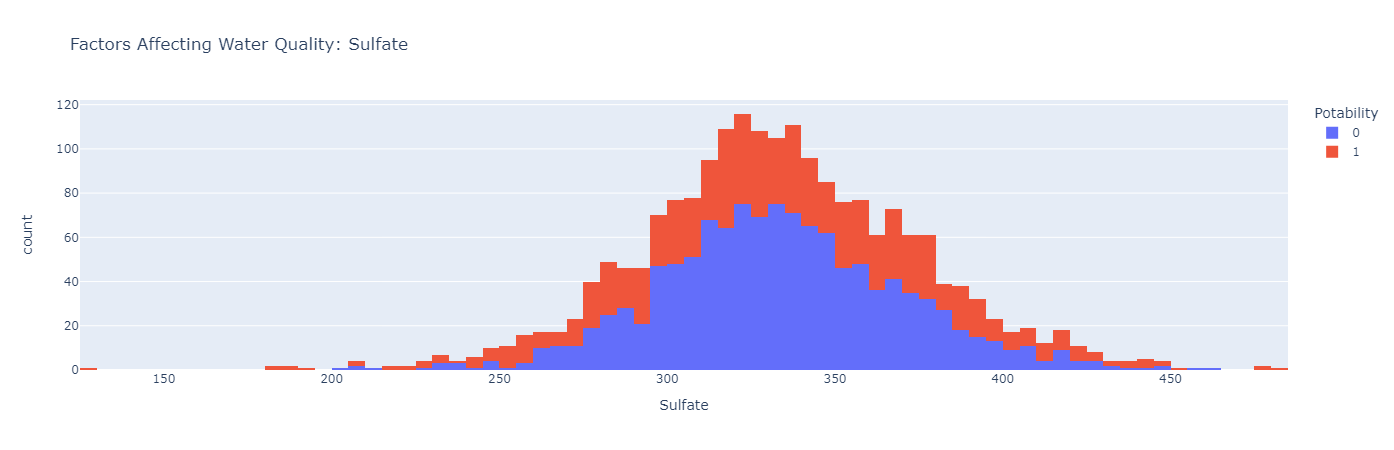
figure = px.histogram(data, x = "Sulfate",

color = "Potability",

title= "Factors Affecting Water Quality: Sulfate")

figure.show()

Output:



import plotly.express as px

data = data

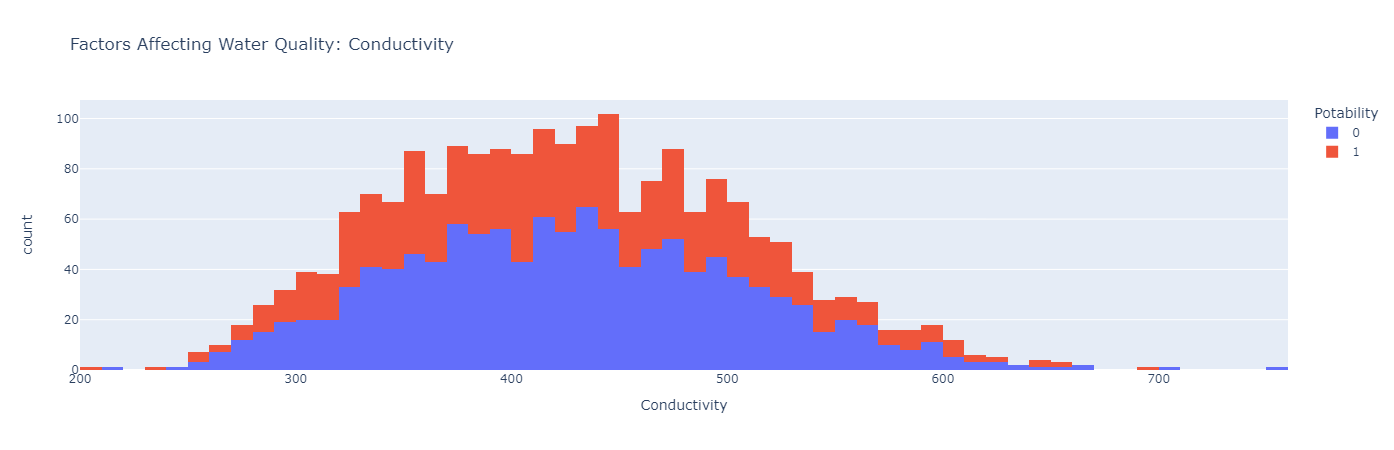
figure = px.histogram(data, x = "Conductivity",

color = "Potability",

title= "Factors Affecting Water Quality: Conductivity")

figure.show()

Output:



import plotly.express as px

data = data

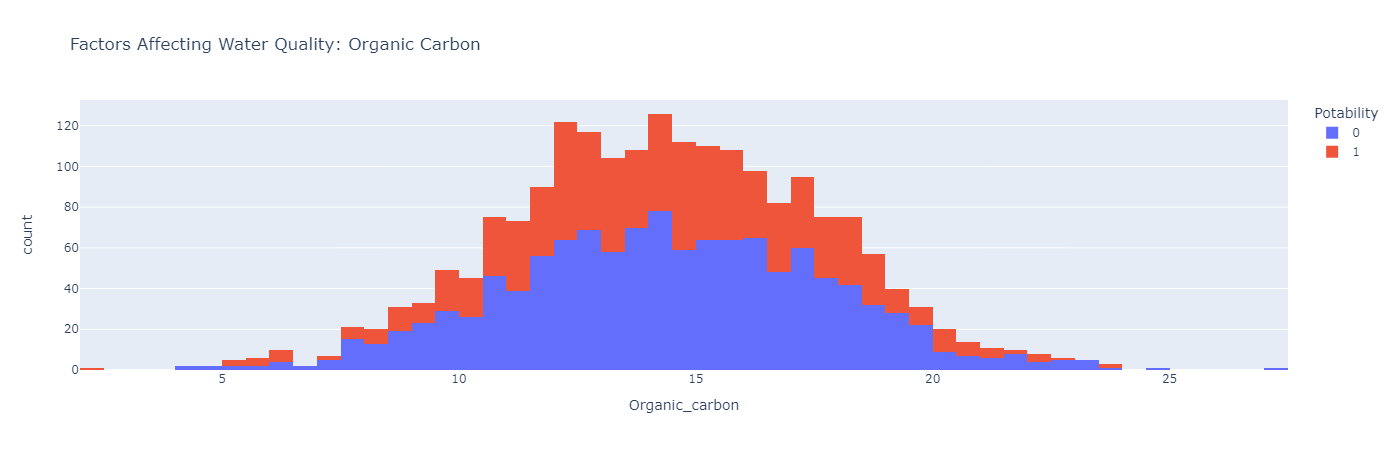
figure = px.histogram(data, x = "Organic\_carbon",

color = "Potability",

title= "Factors Affecting Water Quality: Organic Carbon")

figure.show()

Output:



import plotly.express as px

data = data

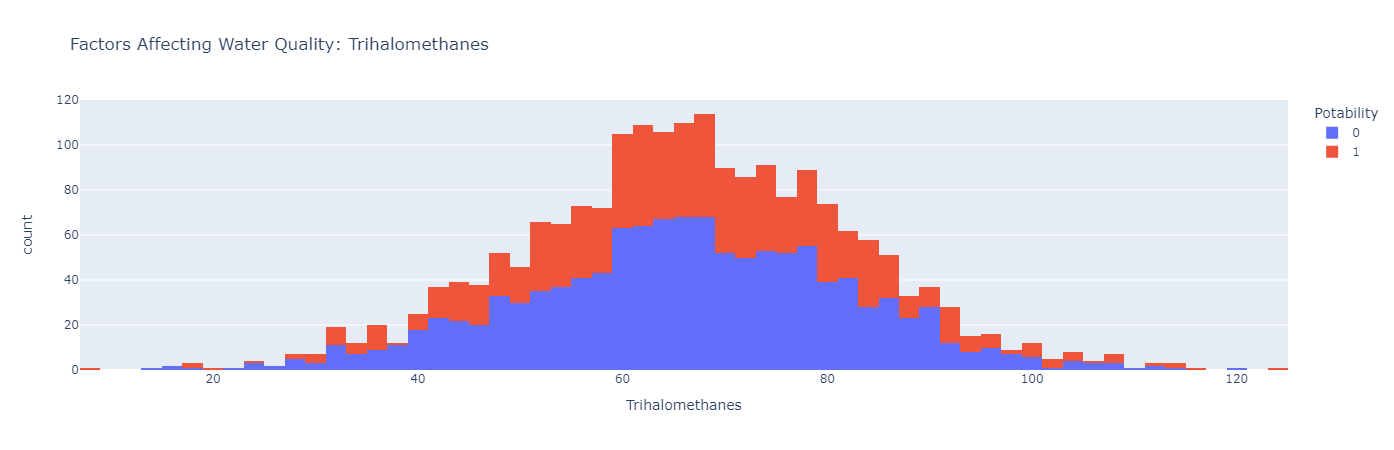
figure = px.histogram(data, x = "Trihalomethanes",

color = "Potability",

title= "Factors Affecting Water Quality: Trihalomethanes")

figure.show()

Output:



import plotly.express as px

data = data

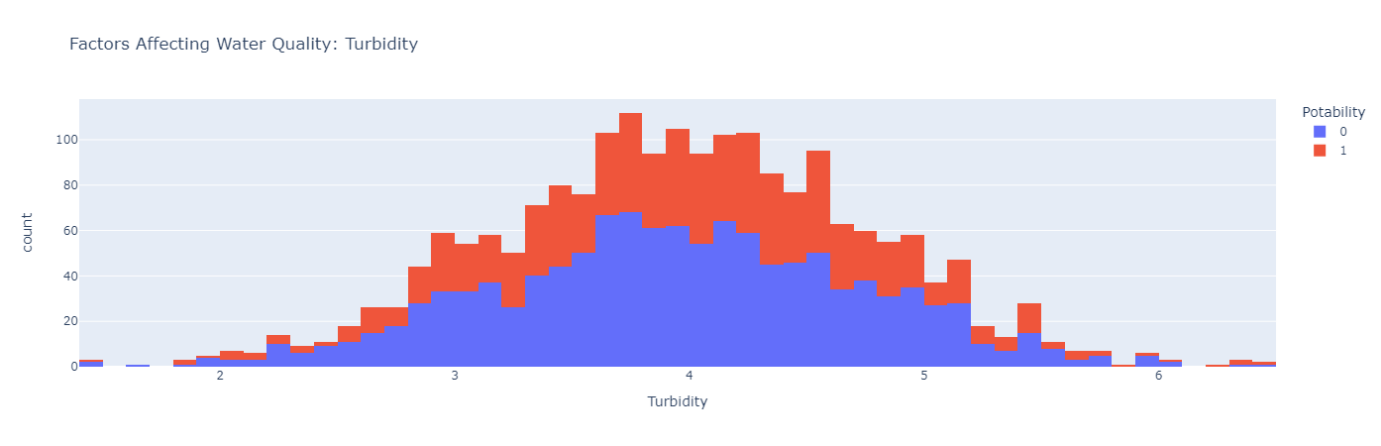
figure = px.histogram(data, x = "Turbidity",

color = "Potability",

title= "Factors Affecting Water Quality: Turbidity")

figure.show()

Output:



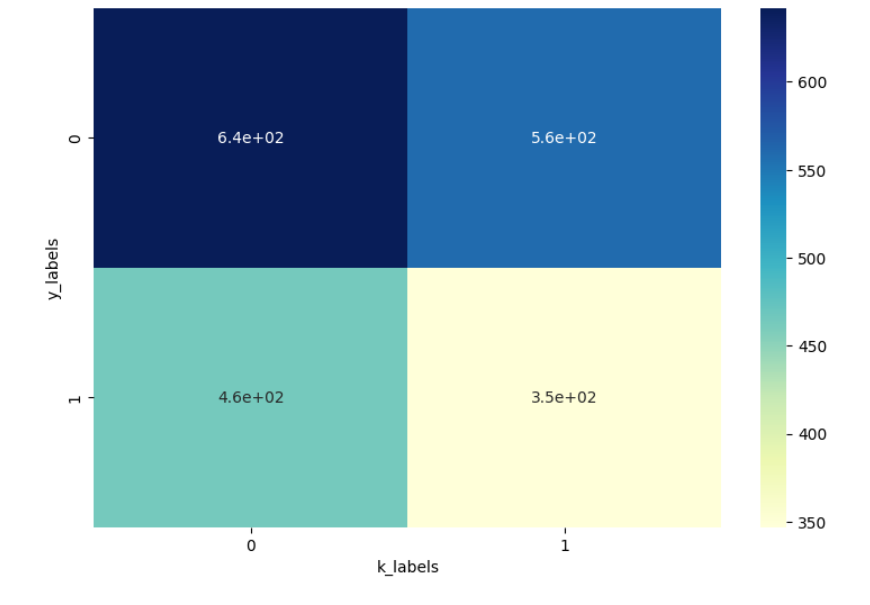
4. Split the Data

 Split your dataset into training and testing sets. This helps you evaluateyour model's performance later.

sns.heatmap(pd.crosstab(compare['y\_labels'],compare['k\_labels']),annot=True,cmap='YlGnBu')

plt.show()

Output:



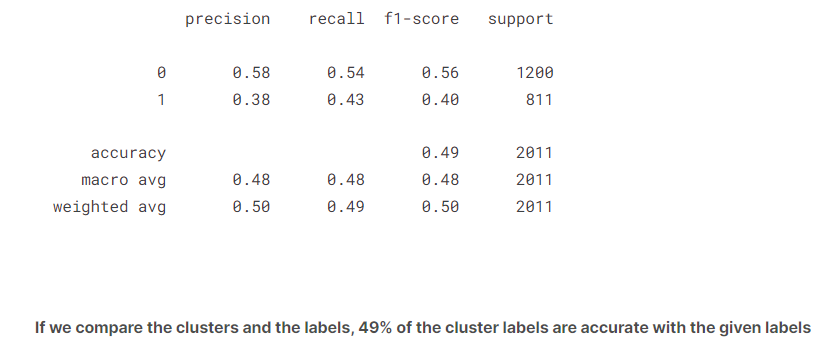
from sklearn.metrics import accuracy\_score, precision\_score, recall\_score, f1\_score,classification\_report

accuracy\_score(compare['y\_labels'],compare['k\_labels'])

recall\_score(compare['y\_labels'],compare['k\_labels'])

print(classification\_report(compare['y\_labels'],compare['k\_labels']))

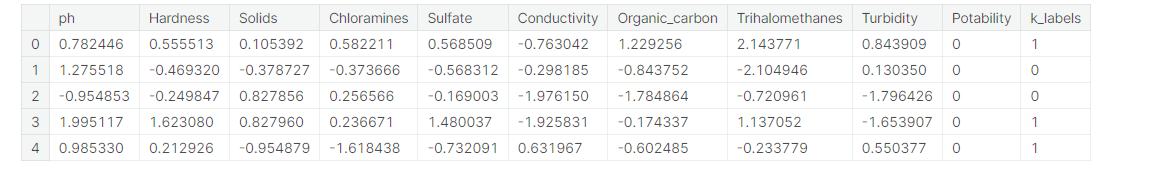
Output:



df['k\_labels'] = pd.Series(model.labels\_)

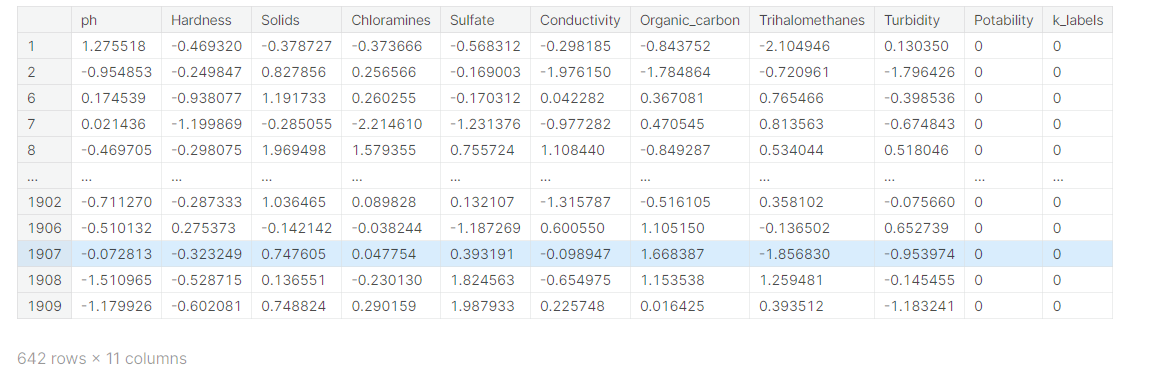
df.head()

Output:



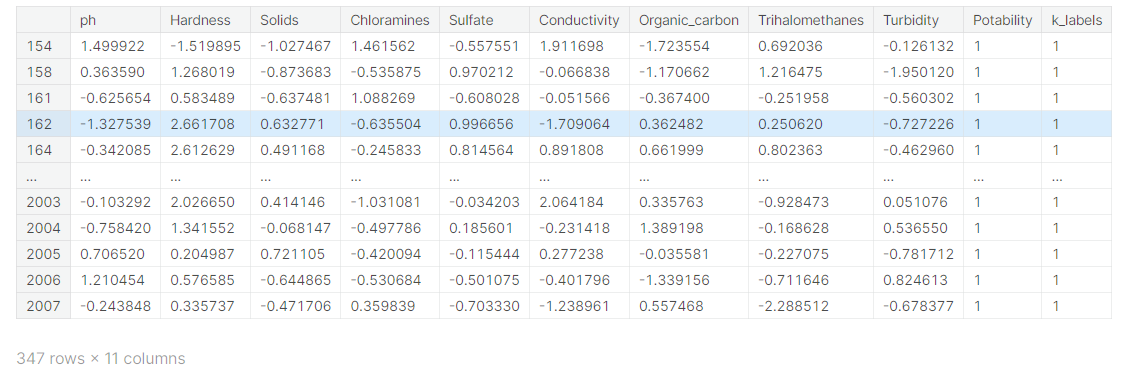
df[((df['Potability']==0)&(df['k\_labels']==0))]

Output:



df[((df['Potability']==1)&(df['k\_labels']==1))]

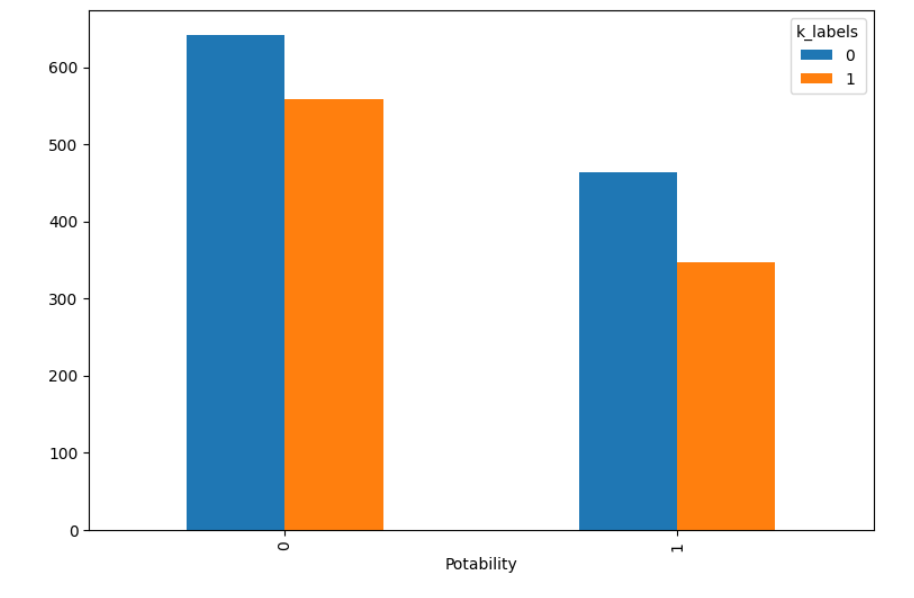
Output:



pd.crosstab(df['Potability'],df['k\_labels']).plot(kind='bar')

plt.show()

Output:



5. Final Result

a=(642/df.shape[0])\*100

b=(347/df.shape[0])\*100

print("We can see that the percentage of POTABLE water samples as potable and equally clustered are: **{}**%".format(round(a),2))

print("-"\*100)

print("We can see that the percentage of NON-POTABLE water samples as potable and equally clustered are: **{}**%".format(round(b),2))

Output:

