Problem definition:

Predicting water quality parameters such as pollutants levels at different points along a river over a specific period of time. The main goal is to develop a model that can accurately forecast these parameters based on various environmental factors, allowing for proactive management and mitigation of water pollution.

Literature Survey:

The momentum concentrate on manages computer based intelligence models utilized for stream WQ displaying. Various models have been planned and applied throughout the long term. Artificial intelligence models have been partitioned into six developments, specifically, ANN model, bit based model, fluffy based models, corresponding models, half breed models that join designs of more than one model or procedures and other metaheuristic models that come from various model characterizations. The initial three classes are the traditional computer based intelligence models and their better forms every now and again named similar by past scientists. In any case, reciprocal models assess the conceivable utilization of wavelet with the blend of different models consequently, they have been classified in a different gathering, which permits the perusers to all the more likely comprehend the near examination of its presentation and viability to manage waterway WQ information when combined with artificial intelligence models. The fifth class was made considering the investigations which managed the nature-propelled calculation for waterway WQ demonstrating, as these calculations conduct depend on a comparative idea which makes them reasonable for a similar class. Subsequently, it groups any remaining metaheuristic models which fall under no past classifications. In any case, in future, these different models can be additionally separated according to their engineering or idea when there is an impressive expansion in the examination articles which is likewise a current hole and future perspectives.

Data Collection & Preparation:

The dataset utilized in this study is gathered from certain verifiable areas in India. It contained 1679 examples from various Indian states during the period from 2005 to 2014. The dataset has 7 huge boundaries, in particular, broke up oxygen (DO), pH, conductivity, organic oxygen interest (Body), nitrate, waste coliform, and complete coliform. Information was gathered by the Indian government to guarantee the nature of the provided drinking water. Information Preprocessing. The handling stage is vital in information examination to further develop the information quality. In this stage, the WQI has been determined from the most huge boundaries of the dataset. Then, at that point, water tests have been arranged based on the WQI values. For getting predominant precision, the z-score strategy has been utilized as an information standardization strategy. Water Quality Record Estimation. To gauge water quality, WQI is utilized to be determined utilizing different boundaries that essentially influence WQ [40-42].

The WQI has been calculated using the following formula:

WQI = ∑N i=1qi × wi ∑N i=1wi , ð1Þ

where: N is the total number of parameters included in the WQI calculations qi is the quality rating scale for each parameter i calculated below, and wi is the unit weight for each parameter .

qi = 100 × Vi − VIdeal Si − VIdeal , ð2Þ

where: Vi is the measured value of parameter i in the tested water samples VIdeal is the ideal value of parameter i in pure water (0 for all parameters except DO = 14:6 mg/l and pH = 7:0), and Si is the recommended standard value of parameter i .wi = K Si , ð3Þ where K is the proportionality constant that can be calculated as follows: K = 1 ∑N i=1Si , ð4Þ .Z-Score Normalization Method. Normalization is a way to simplify calculations. It is a dimensional expression transformed into a nondimensional expression and becomes a scalar. Z-score normalization (or normalization score) is a normalization method used to normalize parameters by using the mean (μ) and standard deviation (σ) values of the tested data. It can be calculated as follows:

Z-score = ð Þ x − μ σ ,

```
import numpy as np
import pandas as pd
import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))
main_df = pd.read_csv("/kaggle/input/water-potability/water_potability.
csv")
df = main_df.copy()
df.head()
```

	ph	hardness	so	Sulfate	Conductivity	Organic_carbon	Trihalomethanes	Turbidity	Potability	
0	NaN	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	NaN	592.885359	15.180013	56.329076	4.500656	0
2	8.099124	224.236259	19909.541732	9.275884	NaN	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

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
import plotly.express as px
import warnings
warnings.filterwarnings('ignore')
print(df.shape)
```

```
print(df.shape)
```

```
(3276, 10)
```

```
print(df.columns)
```

```
Index(['ph', 'Hardness', 'Solids', 'Chloramines', 'Sulfate', 'Conductiv
ity',
       'Organic_carbon', 'Trihalomethanes', 'Turbidity', 'Potability'],
      dtype='object')
```

```
df.describe()
```

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

```
df.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
```

```python
print(df.nunique())
```

```
ph                 2785
Hardness           3276
Solids             3276
Chloramines        3276
Sulfate            2495
Conductivity       3276
Organic_carbon     3276
Trihalomethanes    3114
Turbidity          3276
Potability            2
dtype: int64
```

```python
print(df.isnull().sum())
```

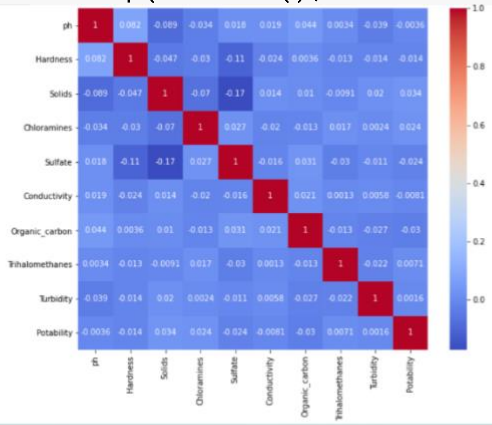
```
ph                 491
Hardness             0
Solids               0
Chloramines          0
Sulfate            781
Conductivity         0
Organic_carbon       0
Trihalomethanes    162
Turbidity            0
Potability           0
dtype: int64
```

```python
df.dtypes
```

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

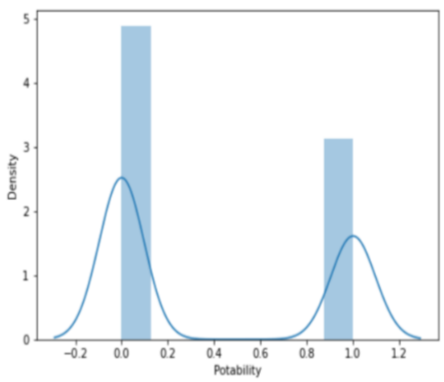
```
dtype: object
```

```python
plt.figure(figsize=(10, 8))
sns.heatmap(df.corr(), annot= True, cmap='coolwarm')
```



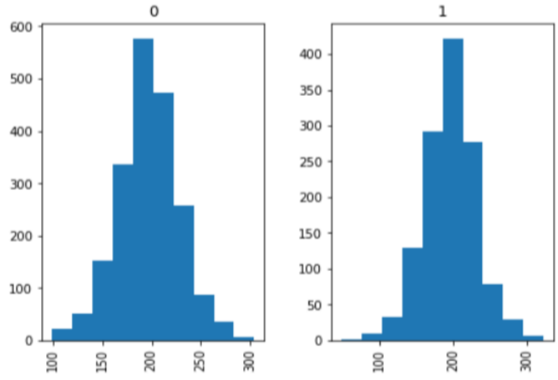

```python
corr = df.corr()
c1 = corr.abs().unstack()
c1.sort_values(ascending = False)[12:24:2]
```

```
Hardness   Sulfate            0.106923
ph         Solids             0.089288
Hardness   ph                 0.082096
Solids     Chloramines        0.070148
Hardness   Solids             0.046899
ph         Organic_carbon     0.043503
dtype: float64
```




```python
df.hist(column='Hardness', by='Potability')
```

```python
df.nunique()
df.nunique()
```




```
df.nunique()
```

```
ph                 2785
Hardness           3276
Solids             3276
Chloramines        3276
Sulfate            2495
Conductivity       3276
Organic_carbon     3276
Trihalomethanes    3114
Turbidity          3276
Potability            2
dtype: int64
```

```
lg = accuracy_score(y_test, pred_lg)
print(lg)
```

```
0.6284658040665434
```