Water quality analysis using ML Algorithm

Analyzing water quality using Machine Learning (ML) algorithms involves leveraging computational techniques to process and interpret water quality data, predict parameters, and detect anomalies. Here's a step-by-step approach to implementing water quality analysis using ML algorithms:

Machine Learning Algorithms: that we are Going to use

1. Logistic Regression

2. Decision Tree

3. Random Forest

4. K-Nearest Neighbours

5. Support Vector Machine

whole Data set Required

Data Preprocessing:

Handle missing values, outliers, and any inconsistencies in the dataset.
Perform feature scaling, normalization, or transformation as needed.

Feature Selection:

Identify relevant features that contribute to the water quality analysis using techniques like correlation analysis, feature importance, or domain knowledge.

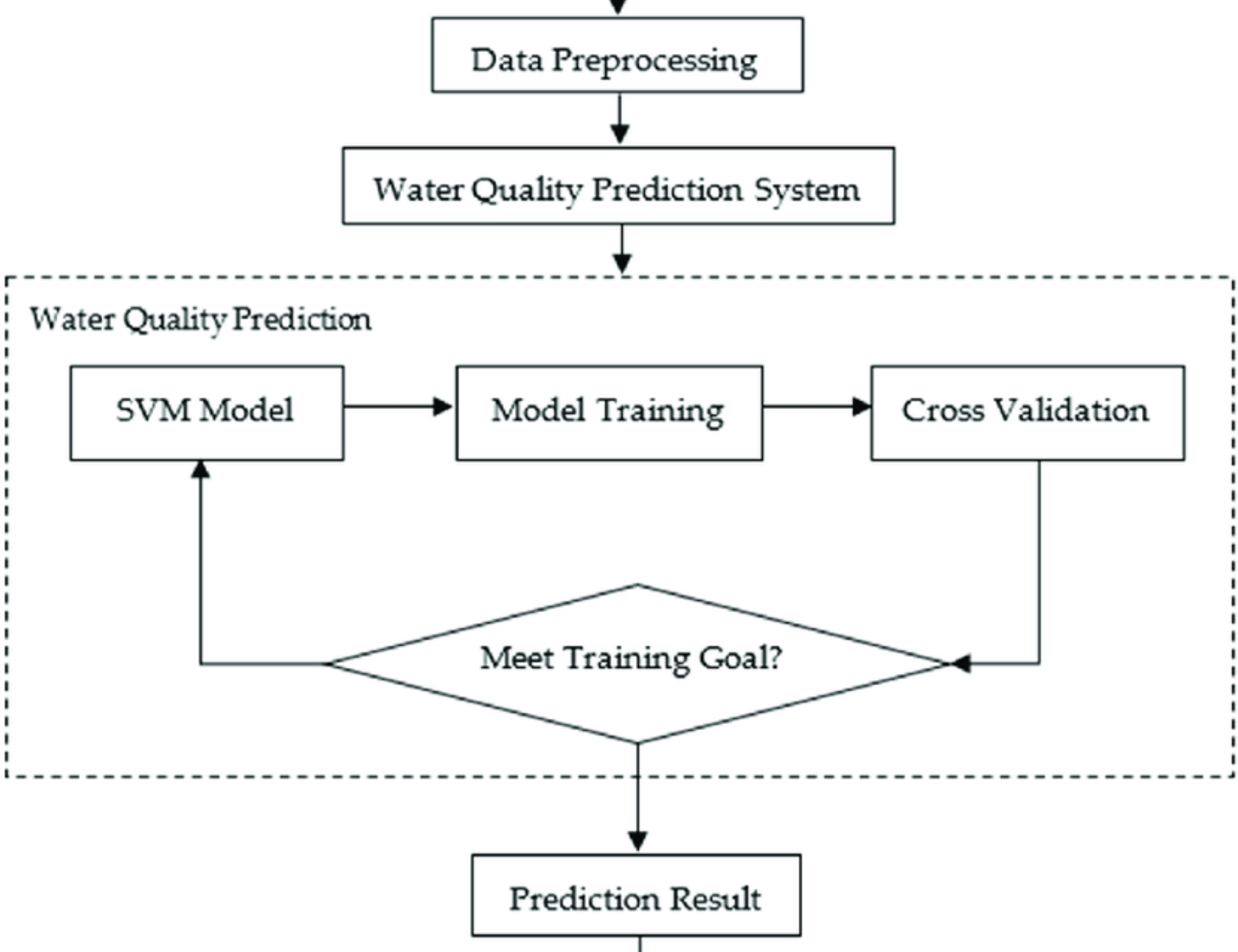
Model Training:

Split the dataset into training and testing sets.
Train the selected ML models using the training set and validate them using the testing set.

Hyperparameter Tuning

- Optimize model performance by tuning hyperparameters using techniques like grid search or random search.

Copper (mg/l)	Zinc (mg/l)	Lead (mg/l)	Nickel (mg/l)	Total Microbial count cfu/ml	E.coli /ml	Total coliform bacteria (MPN)/ml
0.05	5	0.01	0.02	--	--	--
1.5	15	no relaxation	no relaxation	20	--	--
0.003	0.014	0.0014	0.0176	108	71	121
0.0012	**0.0098**	**0**	**0.0139**	**14**	**0**	**0**
0.038	0.028	0.0042	0.0253	140	98	167
0.0015	**0.0022**	**0**	**0.0169**	**16**	**0**	**0**
0.001	0.006	0.0002	0.0176	22	25	2
0.0008	0.005	0	0.0158	19	34	9.6


Data Preprocessing
Water Quality Prediction System
Water Quality Prediction
SVM Model
Model Training
Cross Validation
Meet Training Goal?
Prediction Result

Rishwant/Water-Quality-A | Water quality analysis/ | Water quality analysis(tea

localhost:8888/notebooks/Water%20quality%20analysis/Water%20quality%20analysis(team).ipynb

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 (ipykernel)

Run Code

In [1]:
```
pwd
```

Out[1]:
```
'C:\\Users\\Roshan\\Water quality analysis'
```

In [49]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import plotly.express as px
import matplotlib.pyplot as plt

#sklearn
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split

from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
```

In [3]:
```
df = pd.read_csv("C:\\Users\\Roshan\\Water quality analysis\\water_potability.csv")
df
```

Out[3]:

	ph	Hardness	Solids	Chloramines	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
...	...	...	...	...	...	...	...	...	...	...
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	NaN	392.449580	19.903225	NaN	2.798243	1
3273	9.419510	175.762646	33155.578218	7.350233	NaN	432.044783	11.039070	69.845400	3.298875	1
3274	5.126763	230.603758	11983.869376	6.303357	NaN	402.883113	11.168946	77.488213	4.708658	1
3275	7.874671	195.102299	17404.177061	7.509306	NaN	327.459760	16.140368	78.698446	2.309149	1

3276 rows × 10 columns

Rishwant/Water-Quality-A | Water quality analysis/ | Water quality analysis(tea

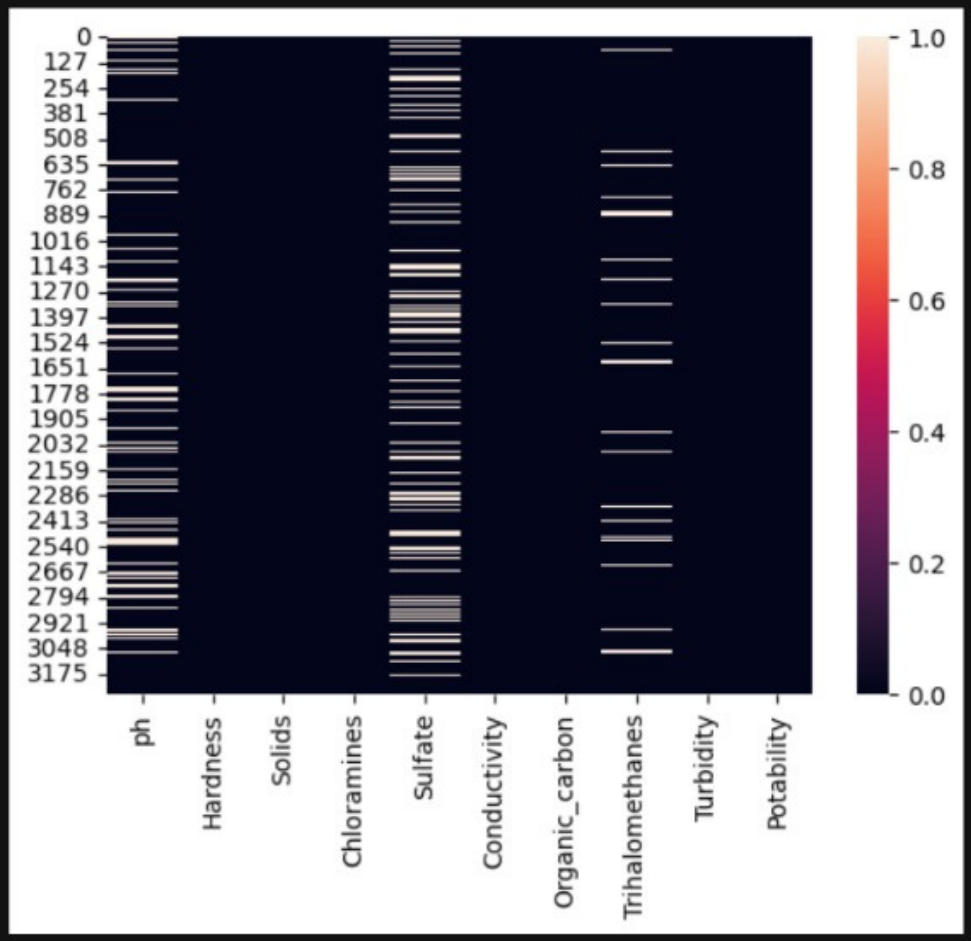
localhost:8888/notebooks/Water%20quality%20analysis/Water%20quality%20analysis(team).ipynb

File Edit View Insert Cell Kernel Widgets Help | Not Trusted | Python 3 (ipykernel)

Run | Code

In [8]:
```
sns.heatmap(df.isnull())
```

Out[8]: <Axes: >



In [9]:
```
plt.figure(figsize=(12,8))
sns.heatmap(df.isnull())
```

Continuous Monitoring and Improvement

Monitor the model's performance and gather new data to continually improve the model's accuracy and relevance.