

Project Development Phase Model Performance Test

Date	17 November 2022
Team ID	PNT2022TMID52625
Project Name	EFFICIENT WATER QUALITY ANALYSIS AND PREDICTION USING MACHINE LEARNING.
Maximum Marks	10 Marks




Model Performance Testing:

Project team shall fill the following information in model performance testing template.

S. No	Parameter	Values	Screenshot
1.	Metrics	Regression Model: MAE - 7.606715556111616, MSE - 91.08982449225519, R2 score - 0.6032158207807568	Screenshot 1
2.	Tune the Model	Hyper parameter Tuning - NIL Validation Method – Split Sample/ Data Validation.	Screenshot 2

Screenshot 1:

The screenshot shows a Jupyter Notebook interface with a dark theme. The top bar includes the IBM logo, the file name 'IBM_project.ipynb', and a star icon. Below this is a menu bar with 'File', 'Edit', 'View', 'Insert', 'Runtime', 'Tools', and 'Help'. On the right side of the top bar are links for 'Comment', 'Share', and a settings icon. The left sidebar contains icons for 'iii', 'x', and a folder icon. The main area is divided into two sections. The top section, labeled 'y_test', displays a preview of a NumPy array with 10 rows of data. The bottom section, labeled '[46]', shows the execution of Python code that imports metrics from sklearn and prints the Mean square error, Mean absolute error, and R_2 score. The output of the code is displayed below the code cell.

RAM  Disk  Editing 

356

y_test

1747	77.72
489	50.74
1575	82.76
149	71.16
104	82.94
...	
1324	66.52
950	44.58
440	67.06
1437	67.06
452	67.06

Name: wqi, Length: 356, dtype: float64

```
[46] from sklearn.metrics import r2_score
from sklearn.metrics import mean_absolute_error, mean_squared_error
print("Mean square error: ", mean_squared_error(y_test, a))
print("Mean absolute error: ", mean_absolute_error(y_test, a))
print("R_2 score: ", r2_score(y_test, a))
```

Mean square error: 91.08982449225519
Mean absolute error: 7.606715556111616
R_2 score: 0.6032158207807568

Screenshot 2:

Splitting the data into dependent and independent variables

```
[12] df=pd.concat([station,location,state,do,ph,co,bod,na,tc,year],axis=1)
df.columns = ['station','location','state','do','ph','co','bod','na','tc','year'] Python
```

[13]

```
df.head() Python
```

...

```
[14] df=df.rename(columns={'D.O. (mg/l)': 'do'})
df=df.rename(columns={'CONDUCTIVITY (μmhos/cm)': 'co'})
df=df.rename(columns={'B.O.D. (mg/l)': 'bod'})
df=df.rename(columns={'NITRATENAN N+ NITRITENANN (mg/l)': 'na'})
df=df.rename(columns={'TOTAL COLIFORM (MPN/100ml)Mean': 'tc'})
df=df.rename(columns={'STATION CODE': 'station'})
df=df.rename(columns={'LOCATIONS': 'location'})
df=df.rename(columns={'STATE': 'state'})
df=df.rename(columns={'PH': 'ph'})
df_new=df Python
```