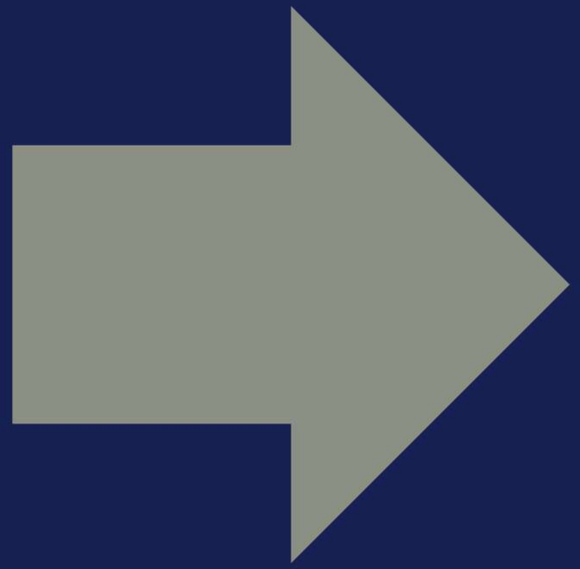
WATER QUALITY ANALYSIS

IBM COGNOS



Ph

IBMid - Sign in or create a | * Water quality analysis

ap1.ca.analytics.ibm.com/bi/

Polotno Studio - Ma...

IBM Cognos Analytics | * Water quality analysis

Edit | Analytics | Filters | Fields | Properties

Selected sources /

water_potability.csv

Search

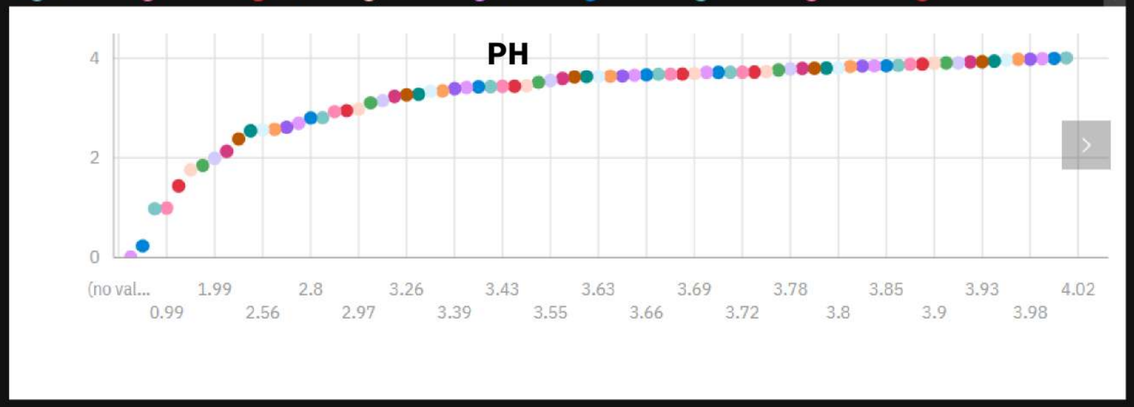
- Navigation paths
- water_potability.csv
  - ph
  - Hardness
  - Solids
  - Chloramines
  - Sulfate
  - Conductivity
  - Organic_carbon
  - Trihalomethanes
  - Turbidity
  - Potability

Ph | Hardness

ph by ph colored by ph

ph

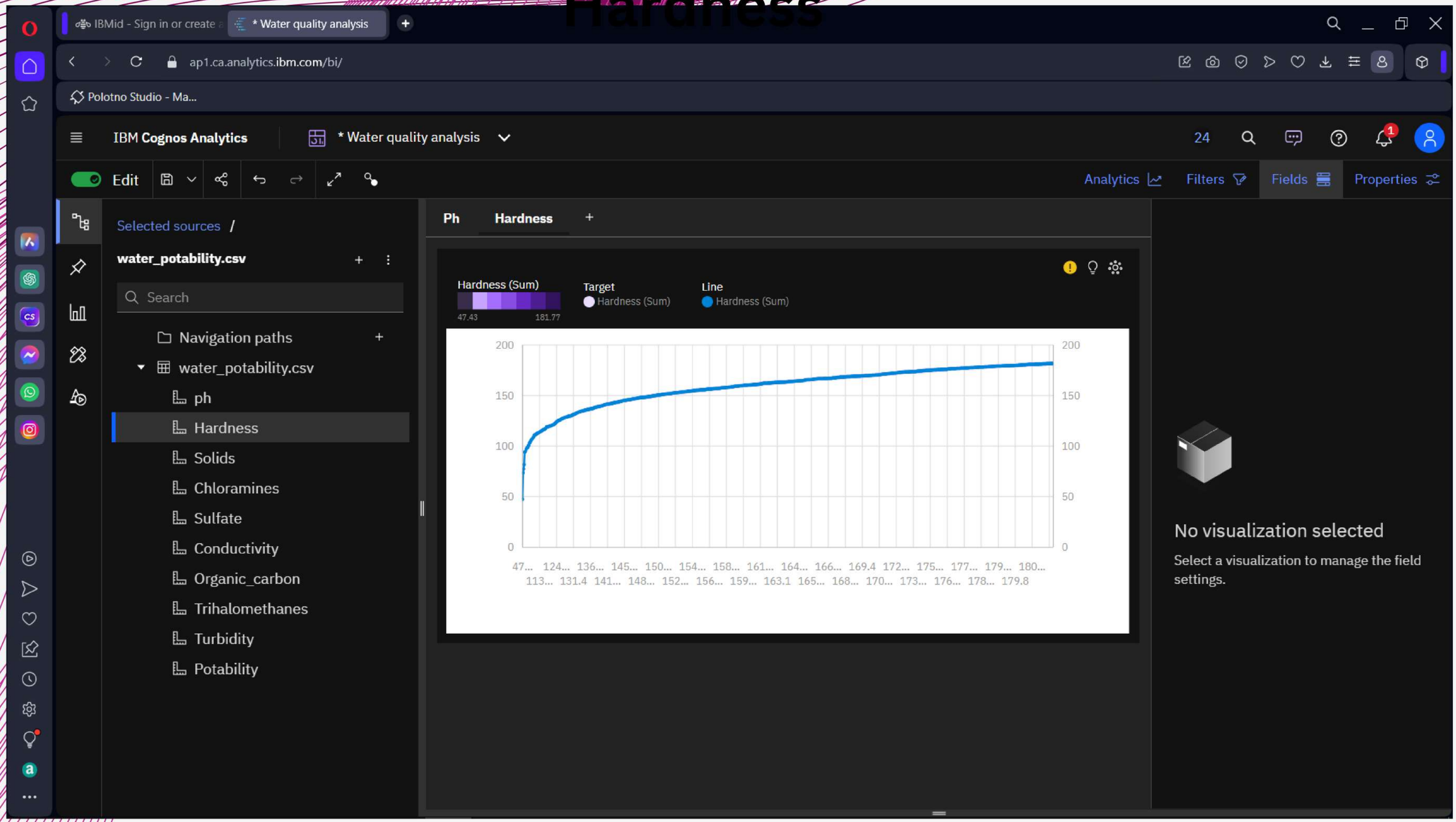
(no value) 0 0.23 0.98 0.99 1.43 1.76 1.84 1.99
2.13 2.38 2.54 2.56 2.57 2.61 2.69 2.8 2.8
2.93 2.95 2.97 3.1 3.15 3.23 3.26 3.27 3.34
3.34 3.39 3.41 3.42 3.43 3.43 3.43 3.45 3.51
3.55 3.59 3.62 3.63 3.63 3.64 3.64 3.66 3.66
3.68 3.68 3.68 3.69 3.72 3.72 3.72 3.72 3.72



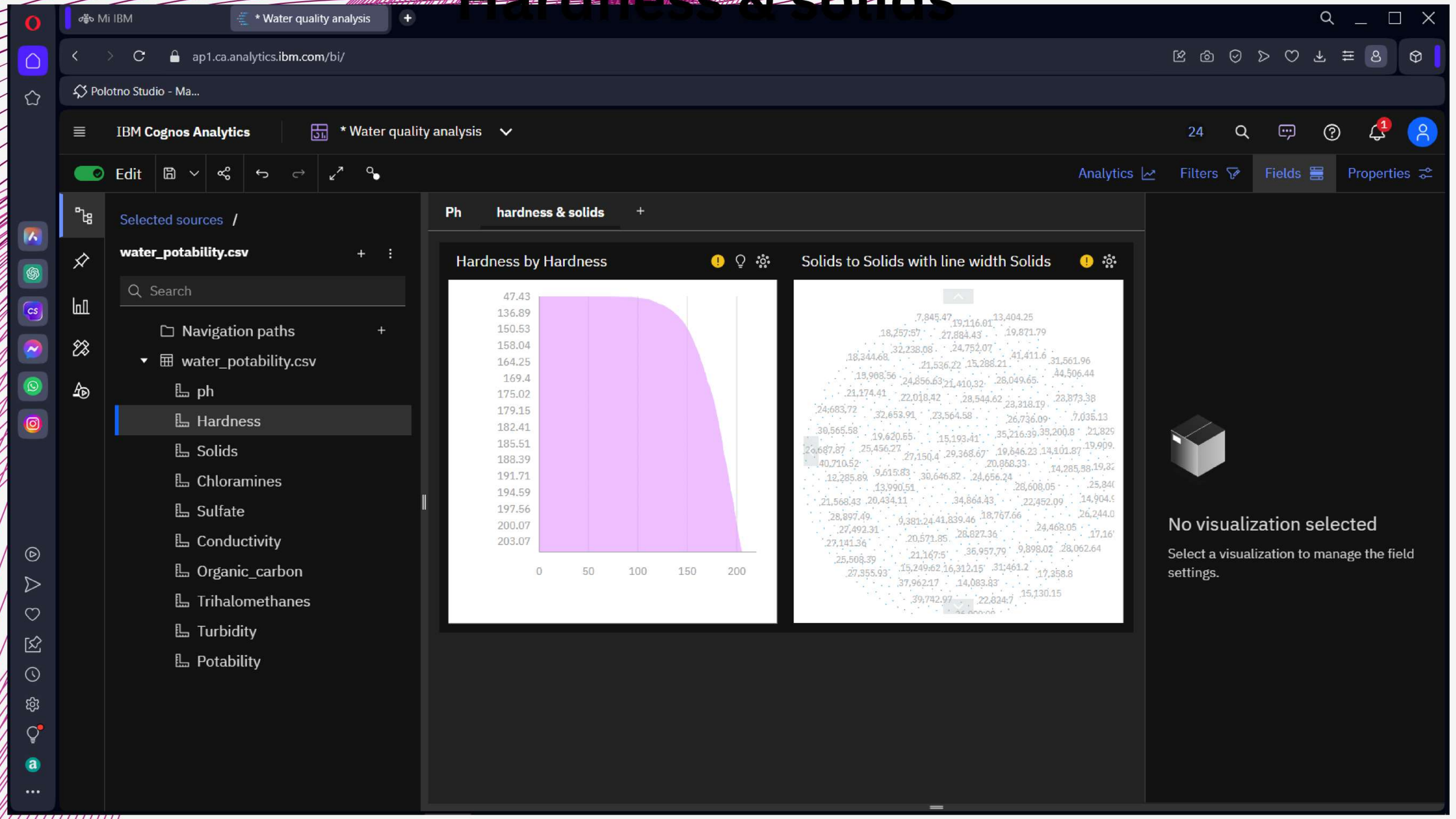
No visualization selected

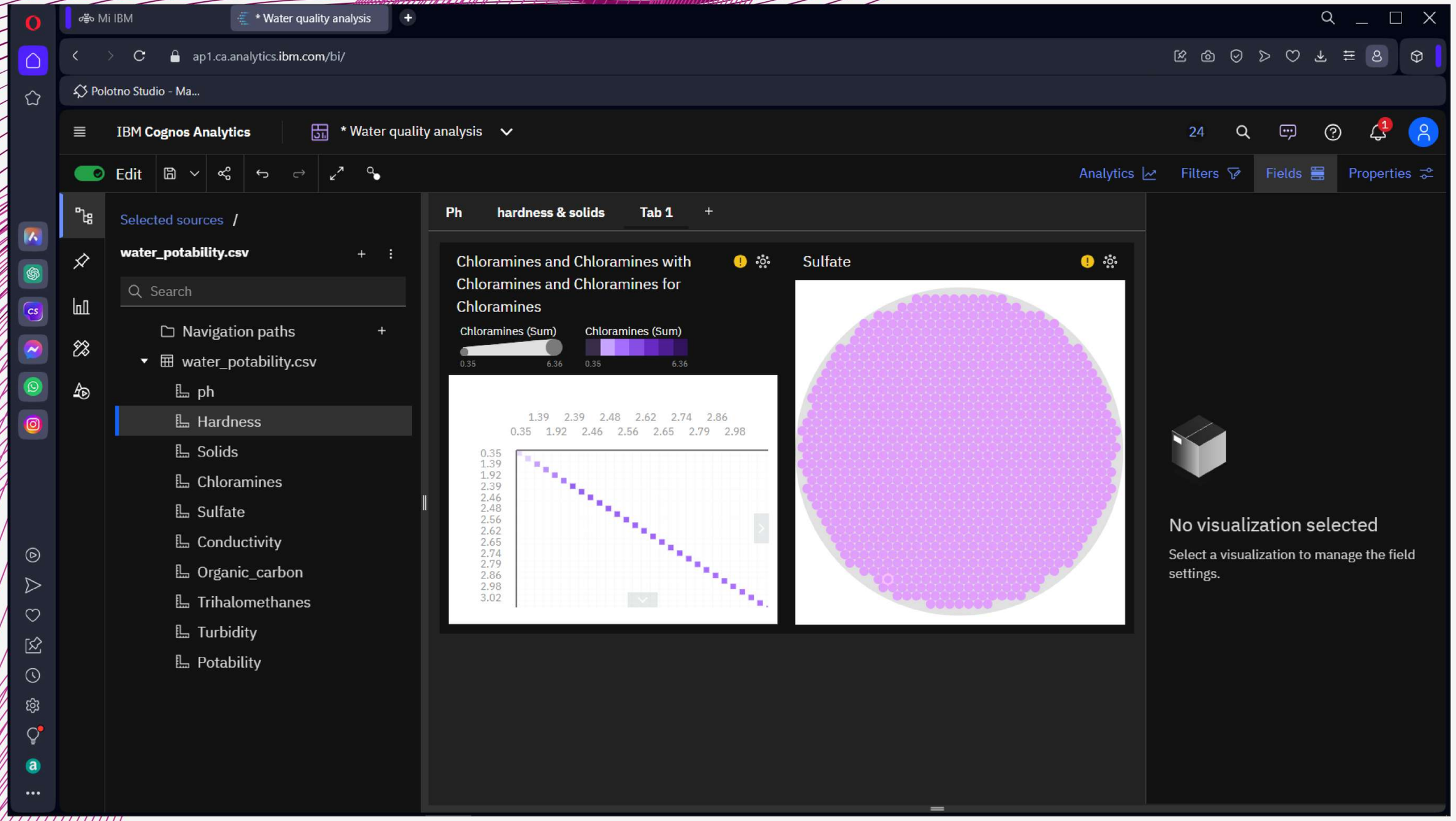
Select a visualization to manage the field settings.

Hardness

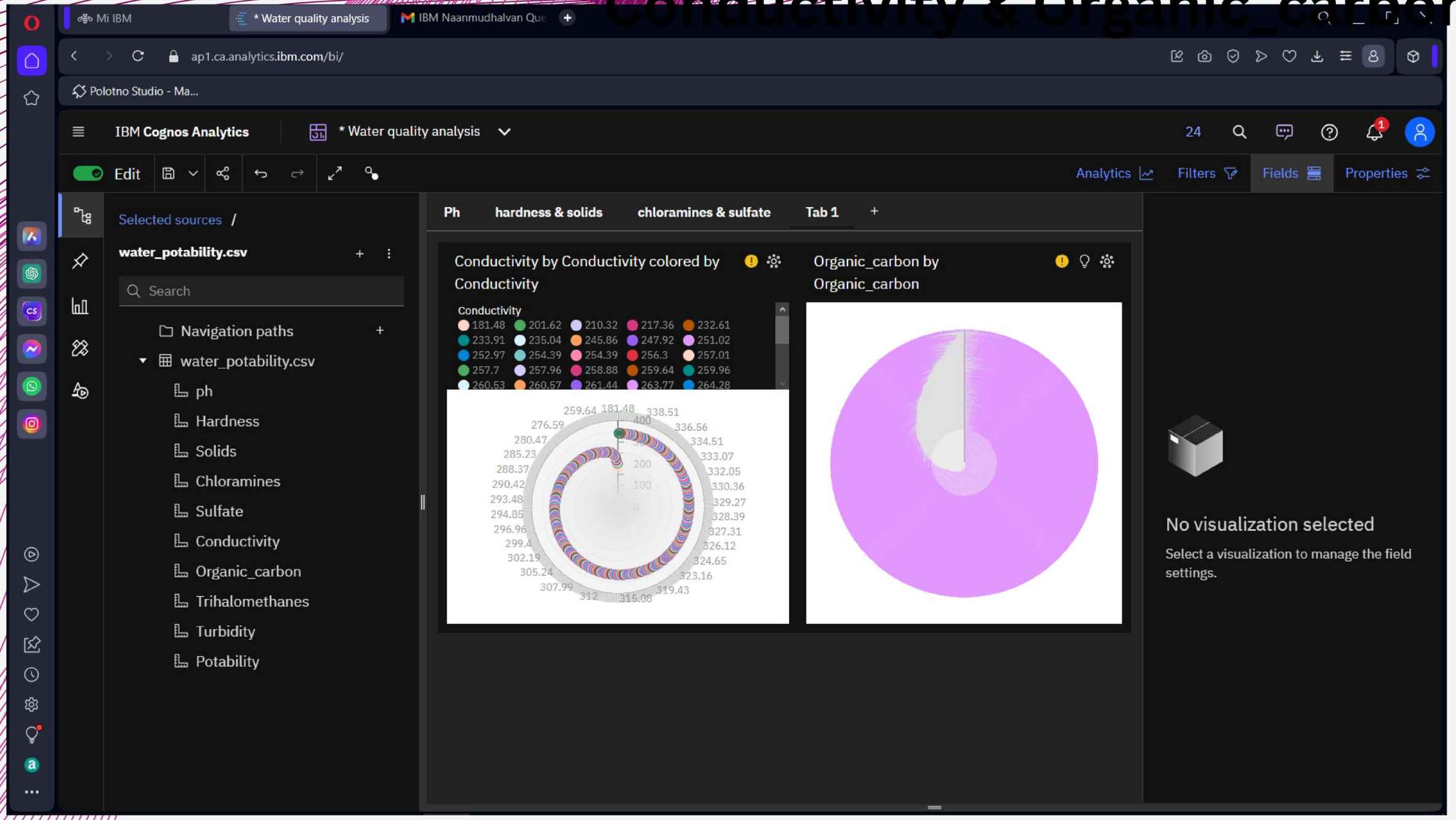

IBMid - Sign in or create
* Water quality analysis
ap1.ca.analytics.ibm.com/bi/
Polotno Studio - Ma...
IBM Cognos Analytics
* Water quality analysis
Edit
Analytics
Filters
Fields
Properties
Selected sources /
water_potability.csv
Search
Navigation paths
water_potability.csv
ph
Hardness
Solids
Chloramines
Sulfate
Conductivity
Organic_carbon
Trihalomethanes
Turbidity
Potability
Ph
Hardness
Hardness (Sum)
47.43
181.77
Target
Hardness (Sum)
Line
Hardness (Sum)
200
150
100
50
0
No visualization selected
Select a visualization to manage the field settings.

Hardness & solids

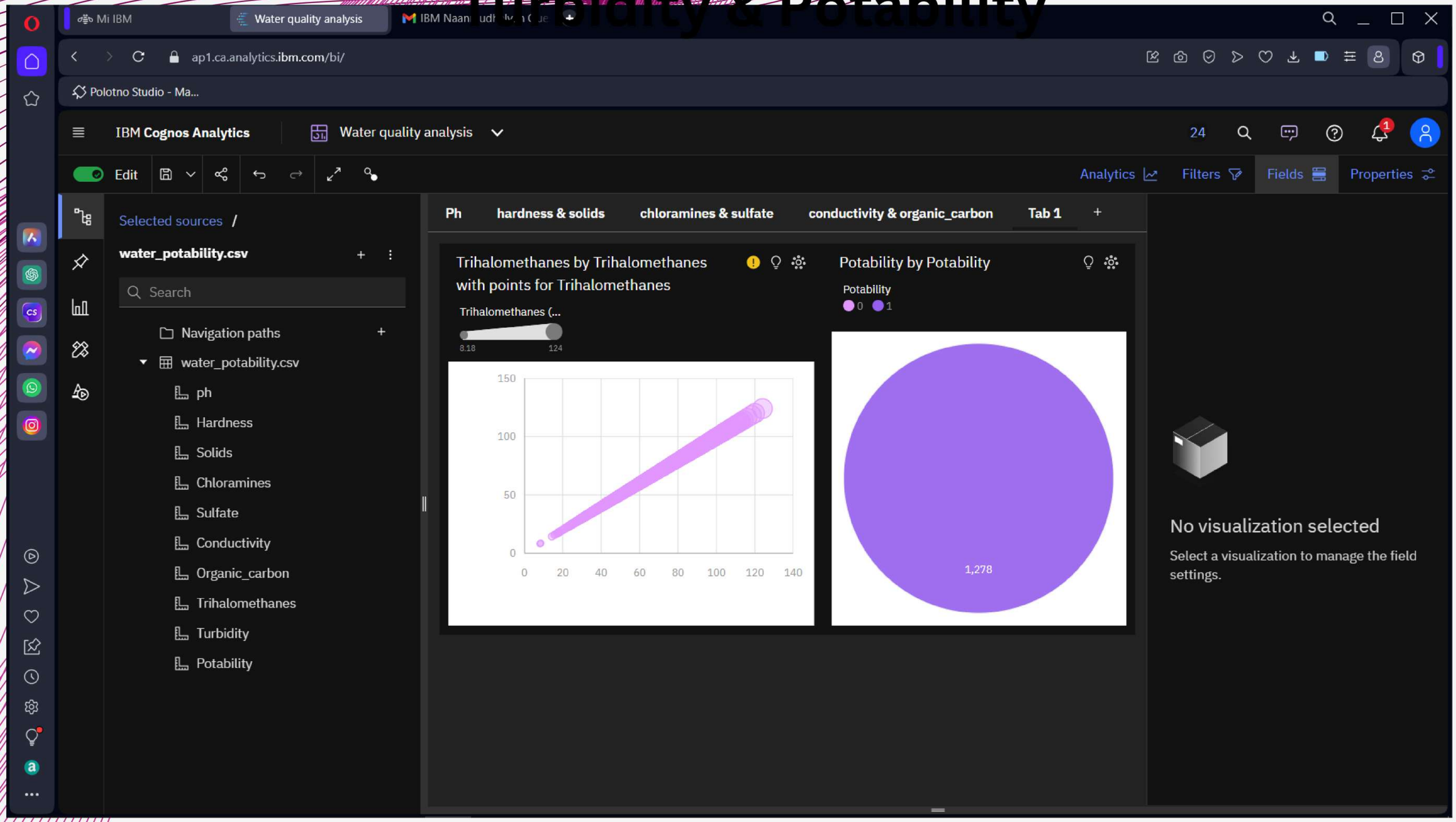


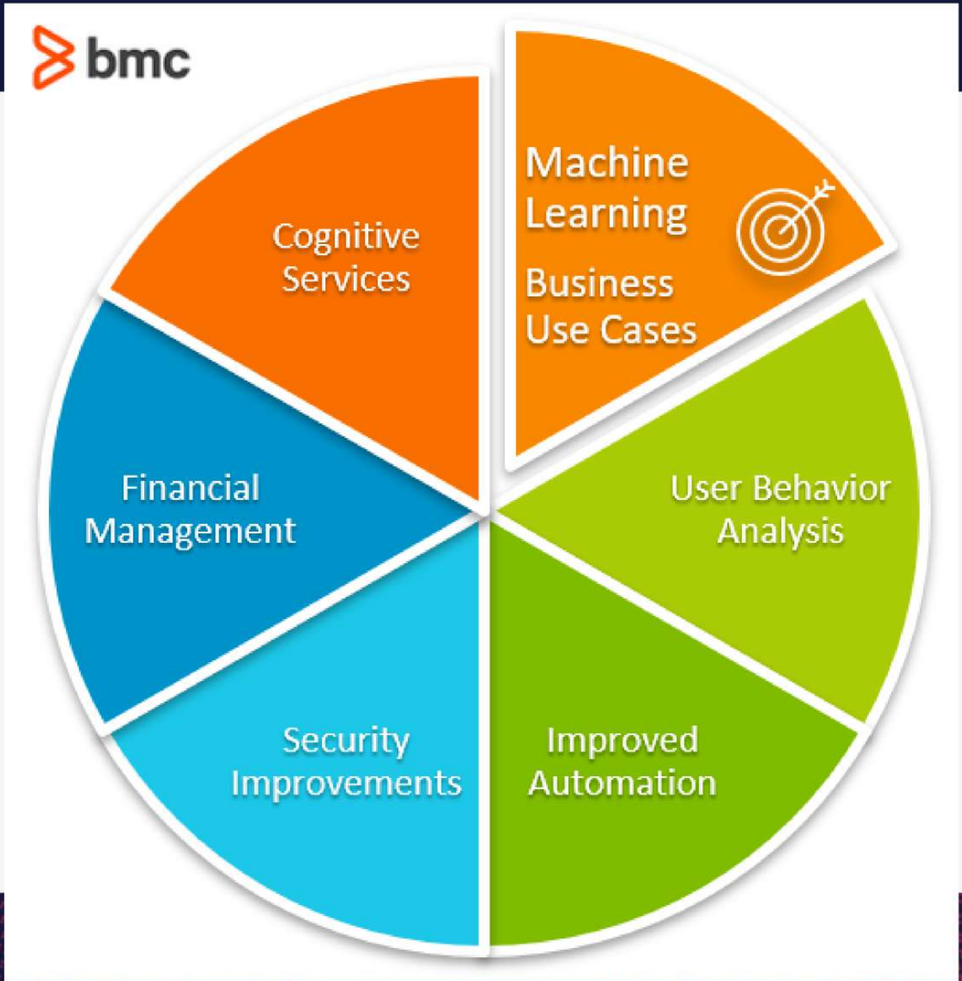

Mi IBM
* Water quality analysis
ap1.ca.analytics.ibm.com/bi/
Polotno Studio - Ma...
IBM Cognos Analytics
* Water quality analysis
24
Edit
Analytics
Filters
Fields
Properties
Selected sources /
water_potability.csv
Search
Navigation paths
water_potability.csv
ph
Hardness
Solids
Chloramines
Sulfate
Conductivity
Organic_carbon
Trihalomethanes
Turbidity
Potability
Ph
hardness & solids
Tab 1
Chloramines and Chloramines with Chloramines and Chloramines for Chloramines
Chloramines (Sum)
0.35
6.36
Chloramines (Sum)
0.35
6.36
1.39 2.39 2.48 2.62 2.74 2.86
0.35 1.92 2.46 2.56 2.65 2.79 2.98
0.35
1.39
1.92
2.39
2.46
2.48
2.56
2.62
2.65
2.74
2.79
2.86
2.98
3.02
Sulfate
No visualization selected
Select a visualization to manage the field settings.

Conductivity & Organic_carbon



Turbidity & Potability





IMPLEMENTATION

Analyzing water quality using MachineLearning (ML) algorithms involvesleveraging computational techniquesto process and interpret water qualitydata, predict parameters, and detectanomalies. Here's a step-by-stepapproach to implementing waterquality analysis using ML algorithms:

MACHINE LEARNING ALGORITHMS: THAT WE ARE GOING TO USE

- Logistic Regressin
- Decision Tree
- Random Forest
- K-Nearest Neighbours
- Suppor Vector Machine

WHOLE DATA SET REQUIRED

Data Preprocessing:

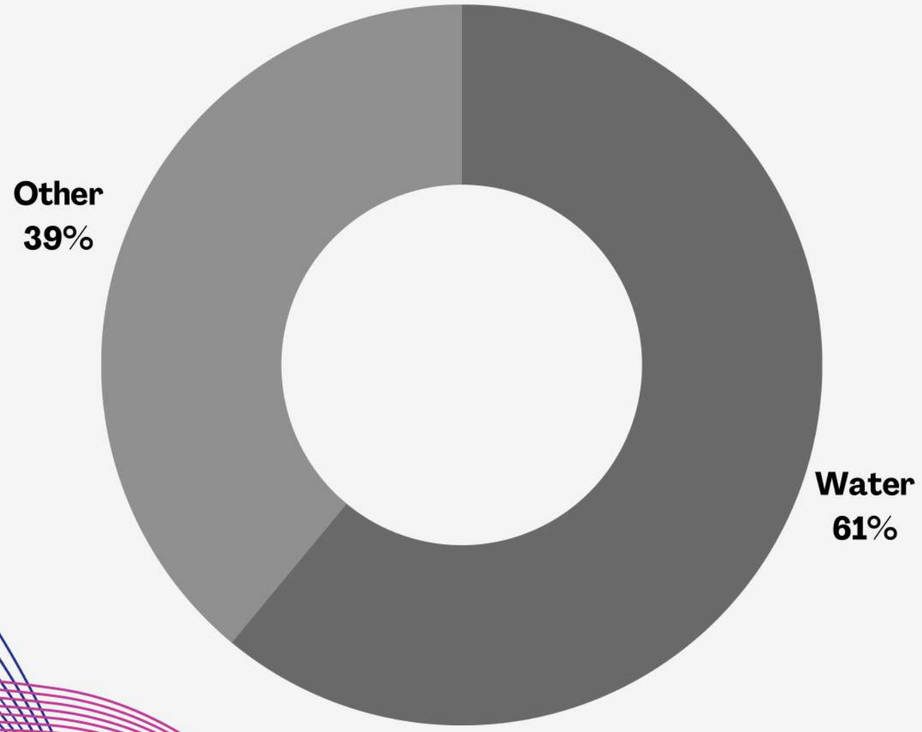
Handle missing values,outliers, and anyinconsistencies in thedataset.
Perform feature scaling,normalization, ortransformation as needed.

Feature Selection:

Identify relevant featuresthat contribute to the waterquality analysis usingtechniques like correlationanalysis, featureimportance, or domainknowledge.

Model Training:

Split the dataset intotraining and testing sets.
Train the selected MLmodels using the trainingset and validate them usingthe testing set.

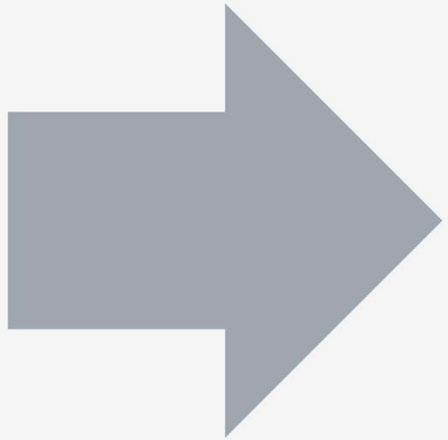


POTABILITY OF WATER

Machine Learning project built to practice and improve coding and deployment skills using python, scikit-learn, jupyter-notebooks, and some visualization packages.

.ipynb

PROGRAMS



```python
# y_test = 1 , 0
pred_lr
```

[59]

```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0], dtype=int64)
```

```python
# accuracy score
```

File Edit Selection View Go Run Terminal Help

Search

File 2.ipynb | Water quality analysis(team).ipynb

C: > Users > Roshan > Water quality analysis > File 2.ipynb > pwd

+ Code + Markdown | Run All | Clear All Outputs | Outline ... | Select Kernel

```python
# accuracy score
accuracy_score_lg = accuracy_score(y_test,pred_lr)
accuracy_score
```

[60] Python

```
<function sklearn.metrics._classification.accuracy_score(y_true, y_pred, *, normalize=True, sample_weight=None)>
```

```python
# accuracy score
accuracy_score_lg = accuracy_score(y_test,pred_lr)
accuracy_score_lg
```

```
(function) def accuracy_score(
    y_true: MatrixLike,
    y_pred: MatrixLike,
    *,
    normalize: bool = True,
    sample_weight: Any | None = None
) -> Float
```

[61] Python

```
0.6097560975609756
```

```python
# accuracy score
accuracy_score_lr = accuracy_score(y_test,pred_lr)
accuracy_score_lr
```

[63] Python

```
0.6097560975609756
```

Decision Tree Classifier

```python
from sklearn.tree import DecisionTreeClassifier

# creating the model object
model_dt = DecisionTreeClassifier(max_depth = 4)
```

[64] Python

File Edit Selection View Go Run Terminal Help

Search

File 2.ipynb | Water quality analysis(team).ipynb

C: > Users > Roshan > Water quality analysis > File 2.ipynb > pwd

+ Code + Markdown | Run All | Clear All Outputs | Outline ··· Select Kernel

```
# accuracy score
accuracy_score_lg = accuracy_score(y_test,pred_lr)
accuracy_score
```

[60] Python

```
<function sklearn.metrics._classification.accuracy_score(y_true, y_pred, *, normalize=True, sample_weight=None)>
```

```
# accuracy score
accuracy_score_lg = accuracy_score(y_test,pred_lr)
accuracy_score_lg
```

[61] Python

```
0.6097560975609756
```

```
# accuracy score
accuracy_score_lr = accuracy_score(y_test,pred_lr)
accuracy_score_lr
```

[63] Python

```
0.6097560975609756
```

+ Code + Markdown

Decision Tree Classifier

```
from sklearn.tree import DecisionTreeClassifier

# creating the model object
model_dt = DecisionTreeClassifier(max_depth = 4)
```

[64] Python

File Edit Selection View Go Run Terminal Help

Search

File 2.ipynb | Water quality analysis(team).ipynb

C: > Users > Roshan > Water quality analysis > File 2.ipynb > pwd

+ Code + Markdown | Run All | Clear All Outputs | Outline ... Select Kernel

Decision Tree Classifier

```
from sklearn.tree import DecisionTreeClassifier

# creating the model object
model_dt = DecisionTreeClassifier(max_depth = 4)
```

[64] Python

```
# Training of Decision Tree
model_dt.fit(x_train,y_train)
```

[65] Python

```
DecisionTreeClassifier
DecisionTreeClassifier(max_depth=4)
```

```
# Making prediction using Decision Tree
pred_dt = model_dt.predict(x_test)
```

[66] Python

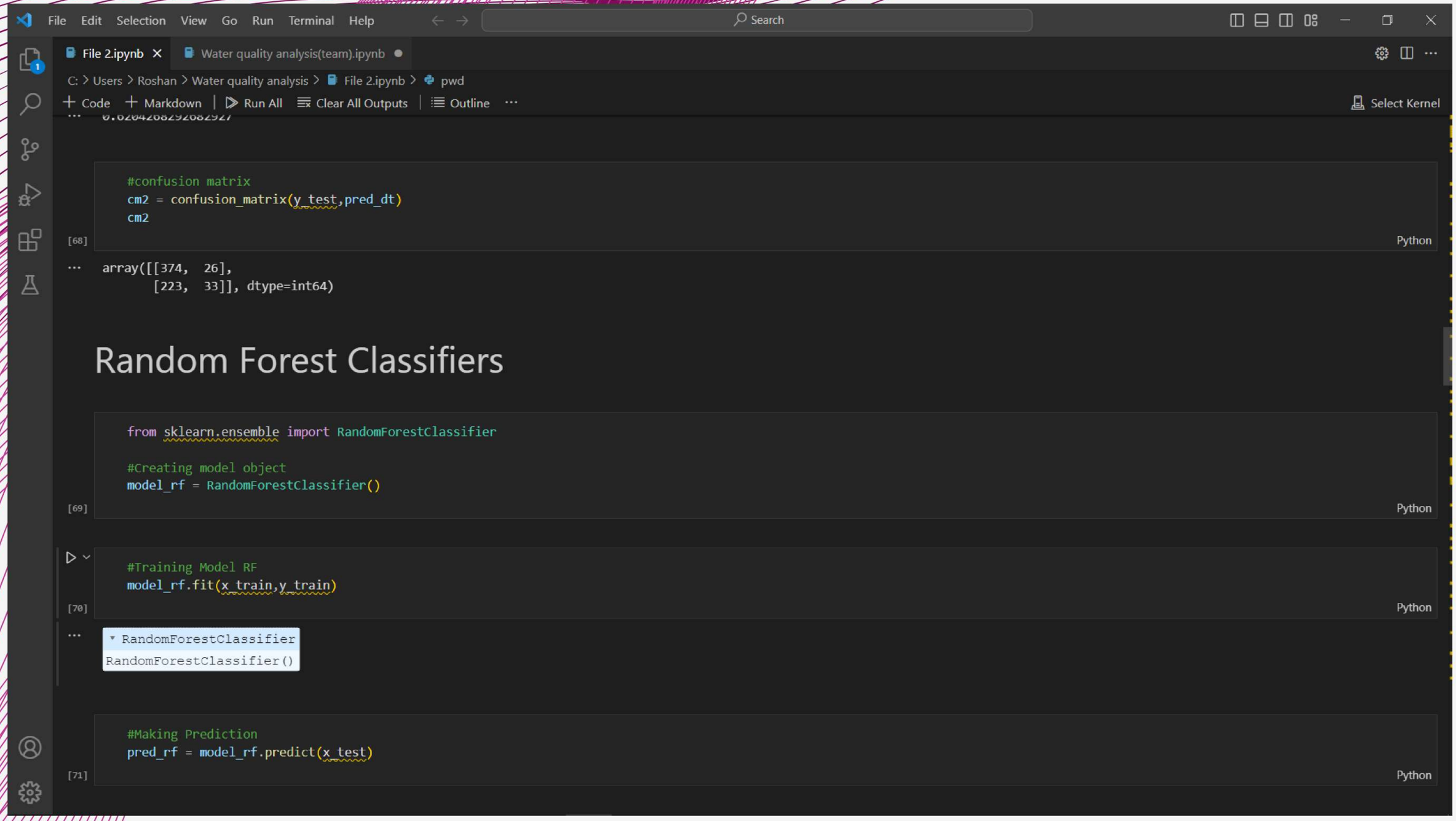
+ Code + Markdown

```
accuracy_score_dt = accuracy_score(y_test,pred_dt)
accuracy_score_dt
```

[67] Python

```
0.6204268292682927
```

```
#confusion matrix
cm2 = confusion_matrix(y_test,pred_dt)
cm2
```


File Edit Selection View Go Run Terminal Help
Search
File 2.ipynb
Water quality analysis(team).ipynb
C: > Users > Roshan > Water quality analysis > File 2.ipynb > pwd
Code Markdown Run All Clear All Outputs Outline
Select Kernel
0.6204268292682927
#confusion matrix
cm2 = confusion_matrix(y_test,pred_dt)
cm2
[68]
Python
array([[374, 26],
[223, 33]], dtype=int64)
Random Forest Classifiers
from sklearn.ensemble import RandomForestClassifier
#Creating model object
model_rf = RandomForestClassifier()
[69]
Python
#Training Model RF
model_rf.fit(x_train,y_train)
[70]
Python
RandomForestClassifier
RandomForestClassifier()
#Making Prediction
pred_rf = model_rf.predict(x_test)
[71]
Python

File Edit Selection View Go Run Terminal Help

Search

File 2.ipynb | Water quality analysis(team).ipynb

C: > Users > Roshan > Water quality analysis > File 2.ipynb > pwd

+ Code + Markdown | Run All | Clear All Outputs | Outline ... | Select Kernel

```
accuracy_score_rf = accuracy_score(y_test,pred_rf)
accuracy_score_rf*100
```

[73] Python

```
65.39634146341463
```

```
cm3 = confusion_matrix(y_test,pred_rf)
cm3
```

[74] Python

```
array([[348,  52],
       [175,  81]], dtype=int64)
```

+ Code + Markdown

KNN -- K-Neighbours

```
from sklearn.neighbors import KNeighborsClassifier

#Creating Model object
model_knn = KNeighborsClassifier()
```

[76] Python

```
for i  in range(4,11):
    model_knn = KNeighborsClassifier(n_neighbors=i)
    model_knn.fit(x_train,y_train)
    pred_knn = model_knn.predict(x_test)
    accuracy_score_knn = accuracy_score(y_test,pred_knn)
    print(i,accuracy_score_knn)
```

[78] Python

```
4 0.6524390243902439
```

File Edit Selection View Go Run Terminal Help

Search

File 2.ipynb | Water quality analysis(team).ipynb

C: > Users > Roshan > Water quality analysis > File 2.ipynb > pwd

+ Code + Markdown | Run All | Clear All Outputs | Outline ... Select Kernel

[76] Python

```
for i  in range(4,11):
    model_knn = KNeighborsClassifier(n_neighbors=i)
    model_knn.fit(x_train,y_train)
    pred_knn = model_knn.predict(x_test)
    accuracy_score_knn = accuracy_score(y_test,pred_knn)
    print(i,accuracy_score_knn)
```

[78] Python

```
4 0.6524390243902439
5 0.6341463414634146
6 0.6371951219512195
7 0.6189024390243902
8 0.6463414634146342
9 0.6341463414634146
10 0.6432926829268293
```

```
model_knn = KNeighborsClassifier(n_neighbors=i)
model_knn.fit(x_train,y_train)
pred_knn = model_knn.predict(x_test)
accuracy_score_knn = accuracy_score(y_test,pred_knn)
print(accuracy_score_knn*100)
```

[79] Python

```
64.32926829268293
```

SVM

```
from sklearn.svm import SVC

#Creating object of Model
```

File Edit Selection View Go Run Terminal Help

Search

File 2.ipynb | Water quality analysis(team).ipynb

C: > Users > Roshan > Water quality analysis > File 2.ipynb > pwd

+ Code + Markdown | Run All | Clear All Outputs | Outline ··· Select Kernel

SVM

```
from sklearn.svm import SVC

#Creating object of Model
model_svm = SVC(kernel="rbf")
```

[81] Python

```
#Model training
model_svm.fit(x_train,y_train)
```

[82] Python

··· SVC
SVC()

```
#Make prediction
pred_svm = model_svm.predict(x_test)
```

[83] Python

```
accuracy_score_svm = accuracy_score(y_test,pred_svm)
accuracy_score_svm*100
```

[84] Python

··· 67.53048780487805

AdaBoost Classifiers

File Edit Selection View Go Run Terminal Help

Search

File 2.ipynb | Water quality analysis(team).ipynb

C: > Users > Roshan > Water quality analysis > File 2.ipynb > pwd

+ Code + Markdown | Run All | Clear All Outputs | Outline ... Select Kernel

AdaBoost Classifiers

```
from sklearn.ensemble import AdaBoostClassifier

#Making object of model
model_ada = AdaBoostClassifier(n_estimators=200,learning_rate=0.03)
```

[118] Python

```
#Training the model
model_ada.fit(x_train,y_train)
```

[119] Python

```
AdaBoostClassifier(learning_rate=0.03, n_estimators=200)
```

```
#Making prediction
pred_ada = model_ada.predict(x_test)
```

[120] Python

```
#accuracy check
accuracy_score_ada = accuracy_score(y_test,pred_ada)
accuracy_score_ada*100
```

[121] Python

```
61.28048780487805
```

Python

File Edit Selection View Go Run Terminal Help

Search

File 2.ipynb | Water quality analysis(team).ipynb

C: > Users > Roshan > Water quality analysis > File 2.ipynb > pwd

+ Code + Markdown | Run All | Clear All Outputs | Outline ... | Select Kernel

XGBoosT

```
from xgboost import XGBClassifier

# Create model
model_xgb = XGBClassifier(n_estimators=200,learning_rate=0.03)
```

[131] Python

```
#Training Model
model_xgb.fit(x_train,y_train)
```

[132] Python

```
XGBClassifier(base_score=None, booster=None, callbacks=None,
              colsample_bylevel=None, colsample_bynode=None,
              colsample_bytree=None, device=None, early_stopping_rounds=None,
              enable_categorical=False, eval_metric=None, feature_types=None,
              gamma=None, grow_policy=None, importance_type=None,
              interaction_constraints=None, learning_rate=0.03, max_bin=None,
              max_cat_threshold=None, max_cat_to_onehot=None,
              max_delta_step=None, max_depth=None, max_leaves=None,
              min_child_weight=None, missing=nan, monotone_constraints=None,
              multi_strategy=None, n_estimators=200, n_jobs=None,
              num_parallel_tree=None, random_state=None, ...)
```

```
#Prediction
pred_xgb = model_xgb.predict(x_test)
```

[133] Python

```
#accuracy
accuracy_score_xgb = accuracy_score(y_test,pred_xgb)
accuracy_score_xgb*100
```

File Edit Selection View Go Run Terminal Help

Search

File 2.ipynb | Water quality analysis(team).ipynb

C: > Users > Roshan > Water quality analysis > File 2.ipynb > pwd

+ Code + Markdown | Run All | Clear All Outputs | Outline ... | Select Kernel

```
accuracy_score_xgb = accuracy_score(y_test,pred_xgb)
accuracy_score_xgb*100
```

[134] Python

```
64.48170731707317
```

```
models = pd.DataFrame({
    "Model": ["Logistic Regression",
              "Decision Tree",
              "Random Forest",
              "KNN",
              "SVM",
              "AdaBoost",
              "XGBoosT"] ,

    "Accuracy Score" : [accuracy_score_lr,accuracy_score_dt,accuracy_score_rf,accuracy_score_knn,accuracy_score_svm,accuracy_score_ada,accuracy_score_xgb]
})
```

[142] Python

```
models
```

[143] Python

	Model	Accuracy Score
0	Logistic Regression	0.609756
1	Decision Tree	0.620427
2	Random Forest	0.653963
3	KNN	0.643293
4	SVM	0.675305
5	AdaBoost	0.612805
6	XGBoosT	0.644817

File Edit Selection View Go Run Terminal Help

Search

File 2.ipynb | Water quality analysis(team).ipynb

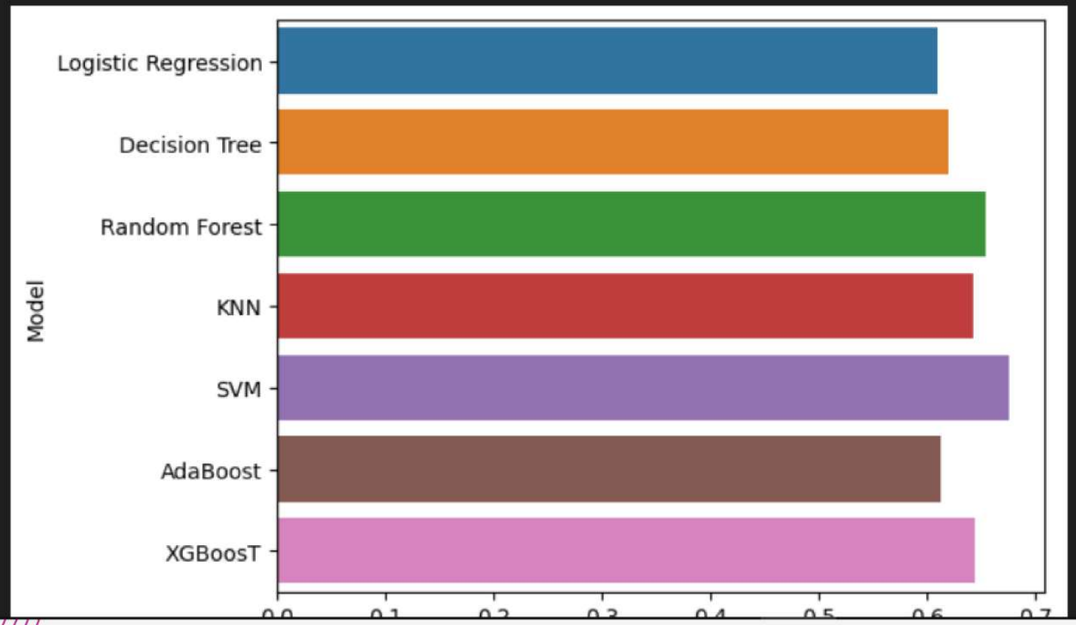
C: > Users > Roshan > Water quality analysis > File 2.ipynb > pwd

+ Code + Markdown | Run All | Clear All Outputs | Outline ... | Select Kernel

```
sns.barplot(x="Accuracy Score" ,y= "Model",data=models)
models.sort_values(by="Accuracy Score",ascending= False)
```

[145] Python

	Model	Accuracy Score
4	SVM	0.675305
2	Random Forest	0.653963
6	XGBoosT	0.644817
3	KNN	0.643293
1	Decision Tree	0.620427
5	AdaBoost	0.612805
0	Logistic Regression	0.609756



File Edit Selection View Go Run Terminal Help

Search

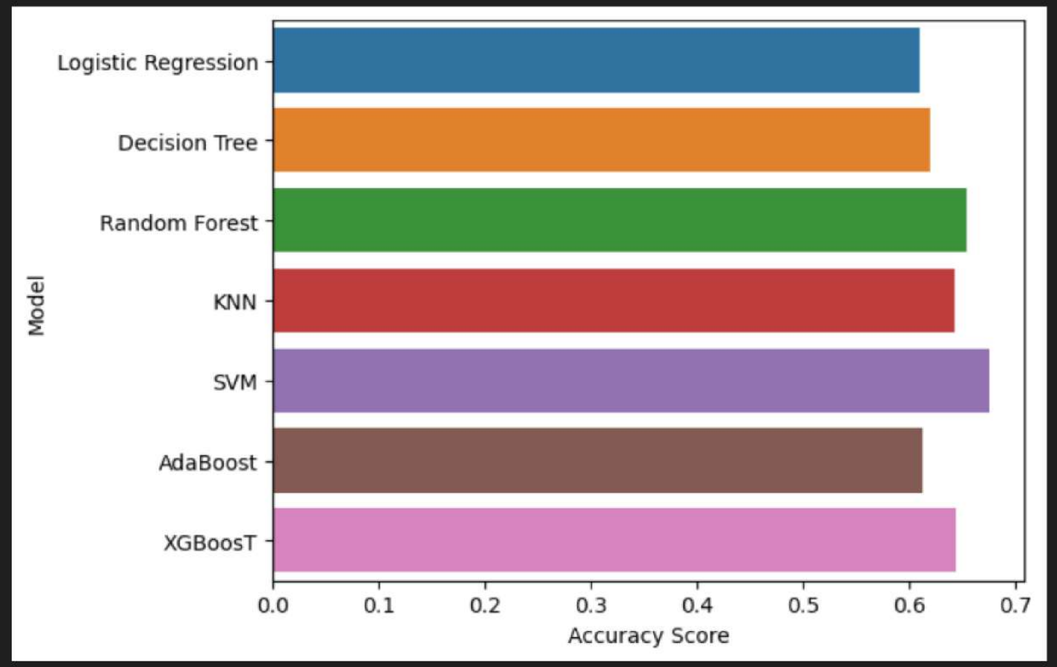
File 2.ipynb | Water quality analysis(team).ipynb

C: > Users > Roshan > Water quality analysis > File 2.ipynb > pwd

+ Code + Markdown | Run All | Clear All Outputs | Outline ...

Select Kernel

	Model	Accuracy Score
4	SVM	0.675305
2	Random Forest	0.653963
6	XGBoosT	0.644817
3	KNN	0.643293
1	Decision Tree	0.620427
5	AdaBoost	0.612805
0	Logistic Regression	0.609756




```
# End
```

All other documents are submitted through Github

Team members:

Roshan.R
Ashwin.N
Rishwanth.S
Samual Rithick.R
Mariya Fernandez.F