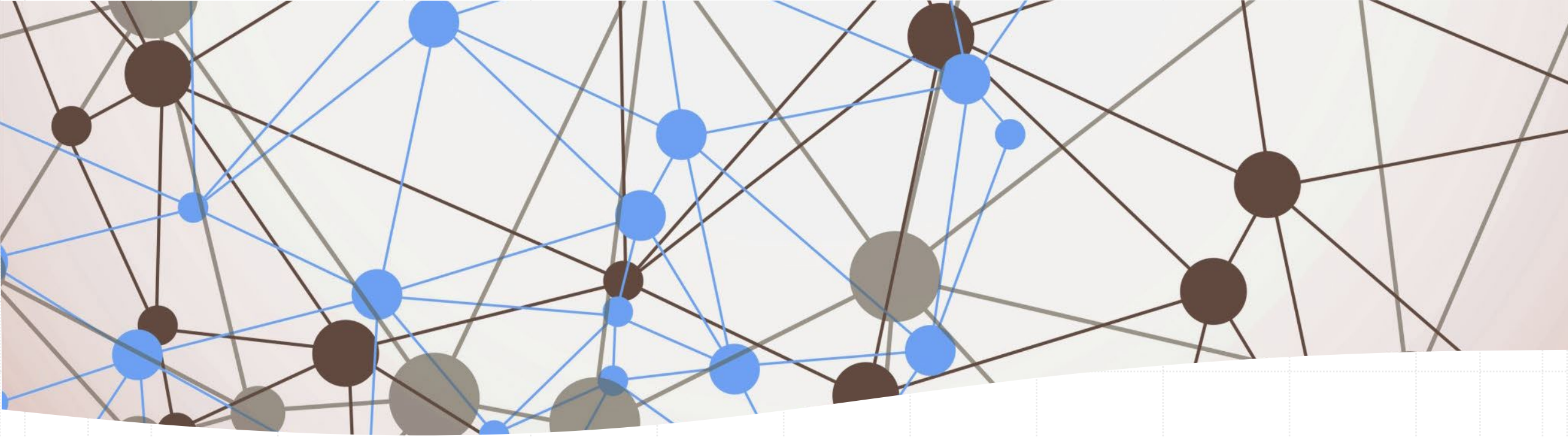
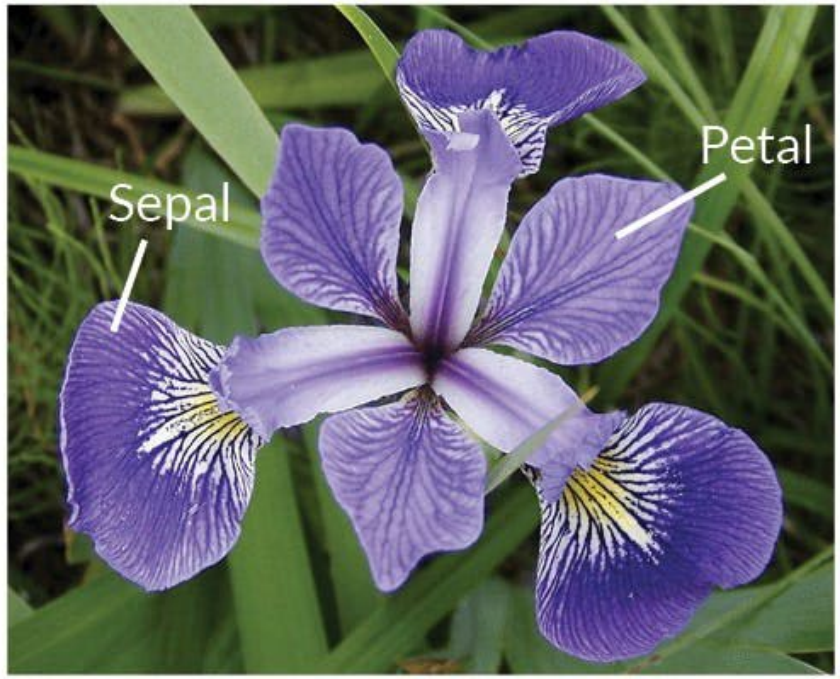


Machine Learning with Python

Python

- Python is one of the most popular programming language for machine learning.
- Ease of learning and use.
- Rich ecosystem of libraries that make ML development easier and faster:
  - **NumPy, Pandas** – for data manipulation
  - **scikit-learn** – for traditional ML algorithms
  - **TensorFlow, PyTorch, Keras** – for deep learning
  - **Matplotlib, Seaborn, Plotly** – for visualization

Iris Clustering and Classification



Iris Versicolor

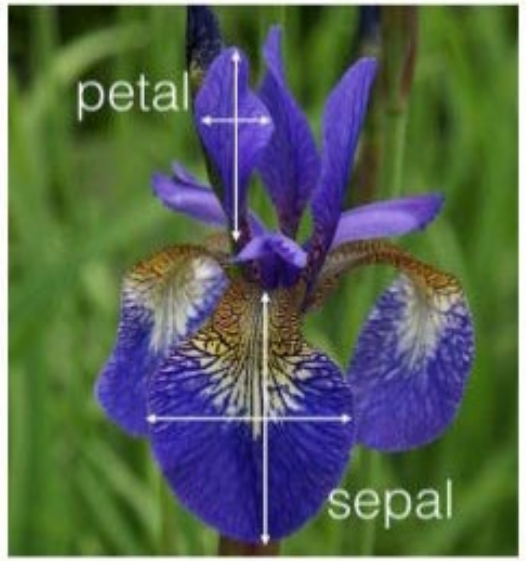
Iris Setosa

Iris Virginica

Iris Clustering and Classification

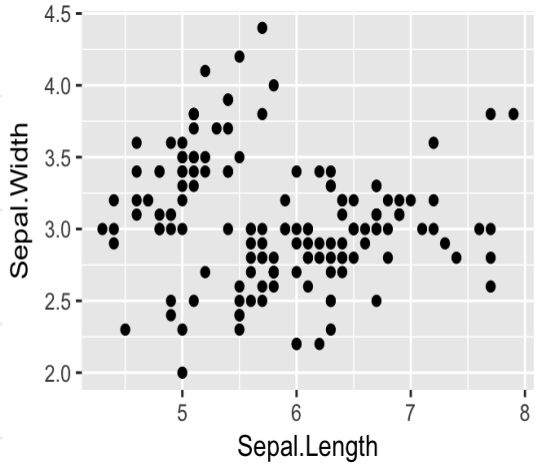
Feature Information:

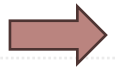
- sepal length in cm
- sepal width in cm
- petal length in cm
- petal width in cm
- class:
  - Iris Setosa
  - Iris Versicolour
  - Iris Virginica

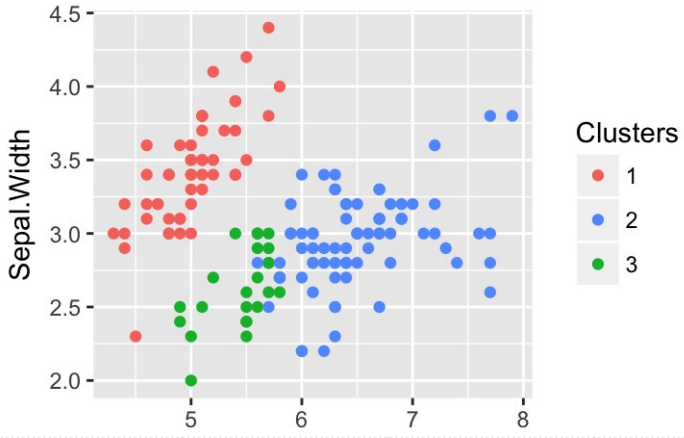


Training / test data

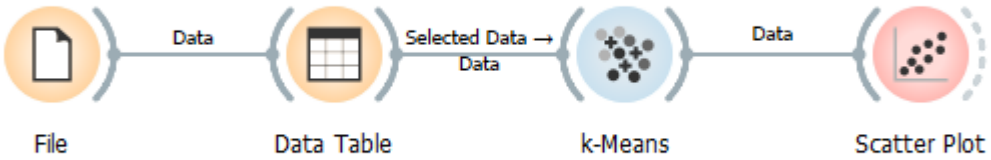
Features				Labels
Sepal length	Sepal width	Petal length	Petal width	Species
5.1	3.5	1.4	0.2	Iris setosa
4.9	3.0	1.4	0.2	Iris setosa
7.0	3.2	4.7	1.4	Iris versicolor
6.4	3.2	4.5	1.5	Iris versicolor
6.3	3.3	6.0	2.5	Iris virginica
5.8	3.3	6.0	2.5	Iris virginica


Sepal.Width
Sepal.Length
4.5
4.0
3.5
3.0
2.5
2.0
5
6
7
8




Sepal.Width
4.5
4.0
3.5
3.0
2.5
2.0
5
6
7
8
Clusters
1
2
3

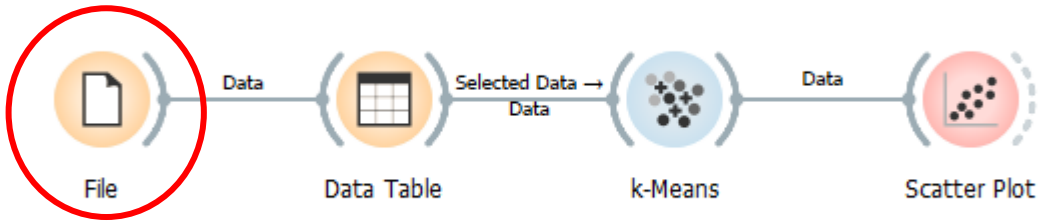
Clustering


Data
Selected Data
Data
Data
File
Data Table
k-Means
Scatter Plot

Import Libraries

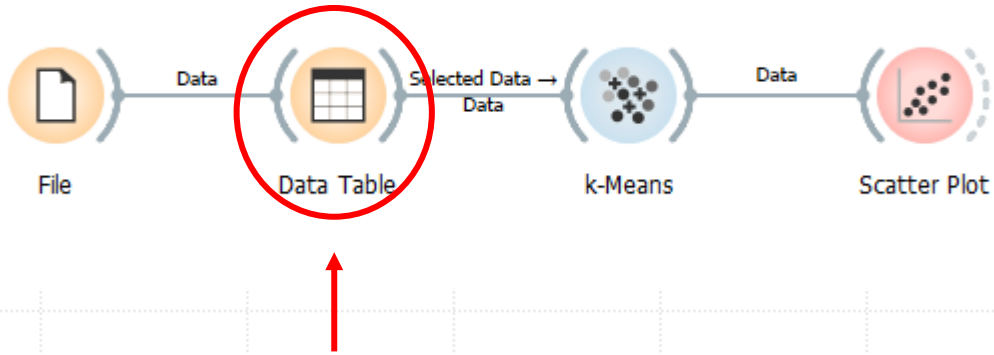
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.datasets import load_iris
from sklearn.cluster import KMeans
```

Load Data



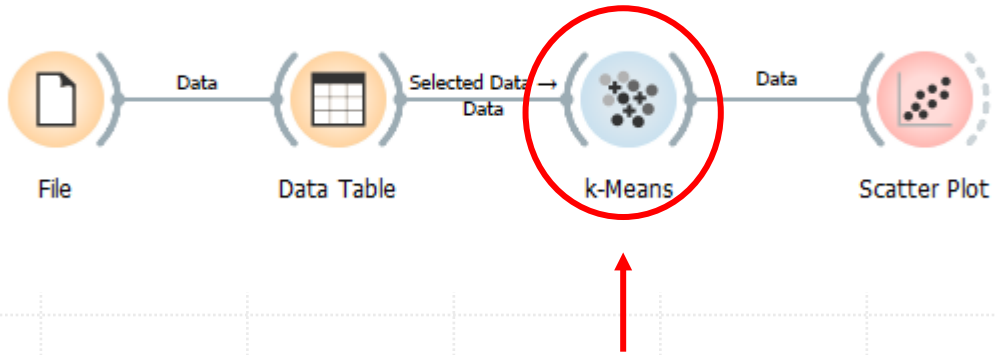

```
iris = load_iris()
X = iris.data
```

View Data (Optional)



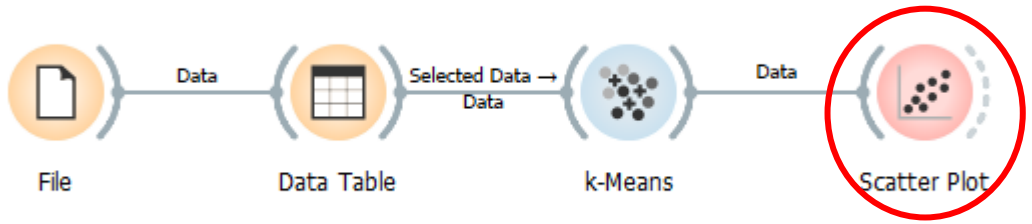

```
df = pd.DataFrame(X, columns=iris.feature_names)
df
```

Clustering with k-Means

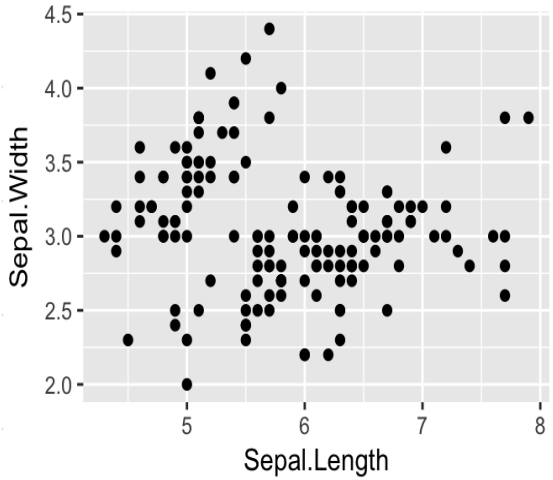


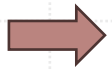

```
kmeans = KMeans(n_clusters=3, random_state=42)
kmeans.fit(X)
cluster_labels = kmeans.labels_
```

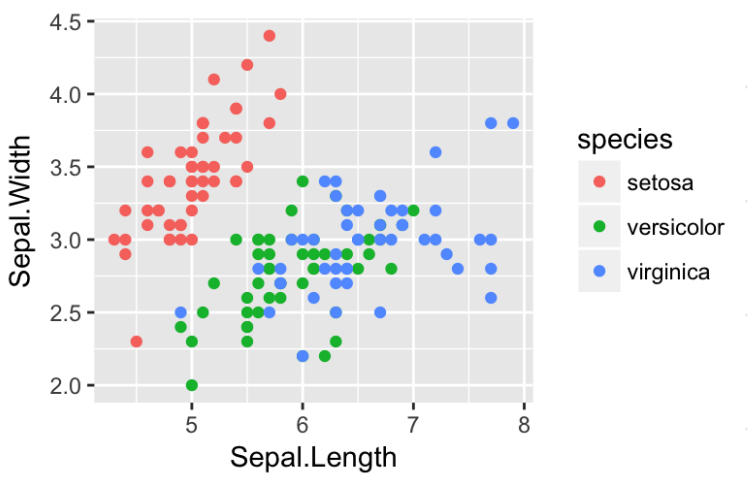
Plot



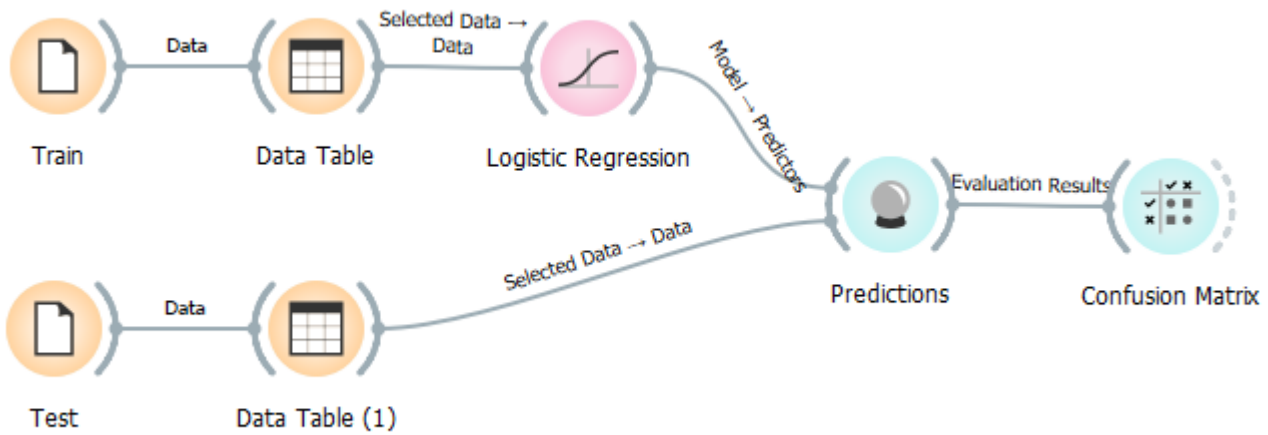

```
plt.figure(figsize=(8, 6))
plt.scatter(X[:, 0], X[:, 1], c=cluster_labels,
cmap='viridis', edgecolor='k')
plt.xlabel(iris.feature_names[0])
plt.ylabel(iris.feature_names[1])
plt.title("K-Means Clustering of Iris Dataset")
plt.show()
```







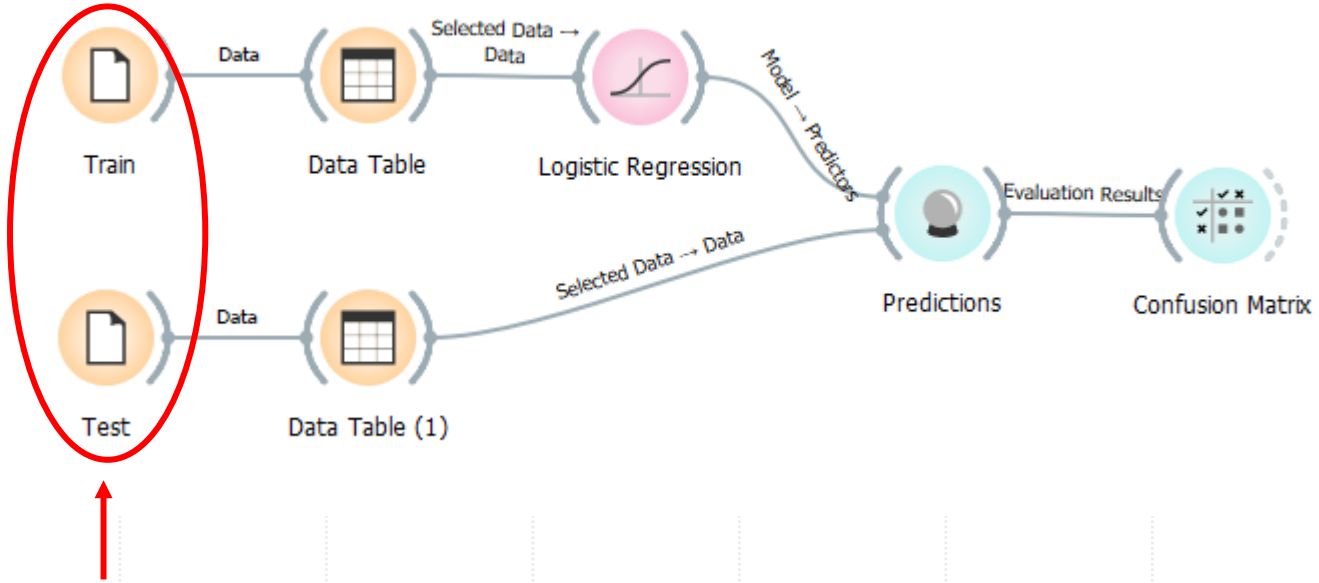
Classification



Import Libraries

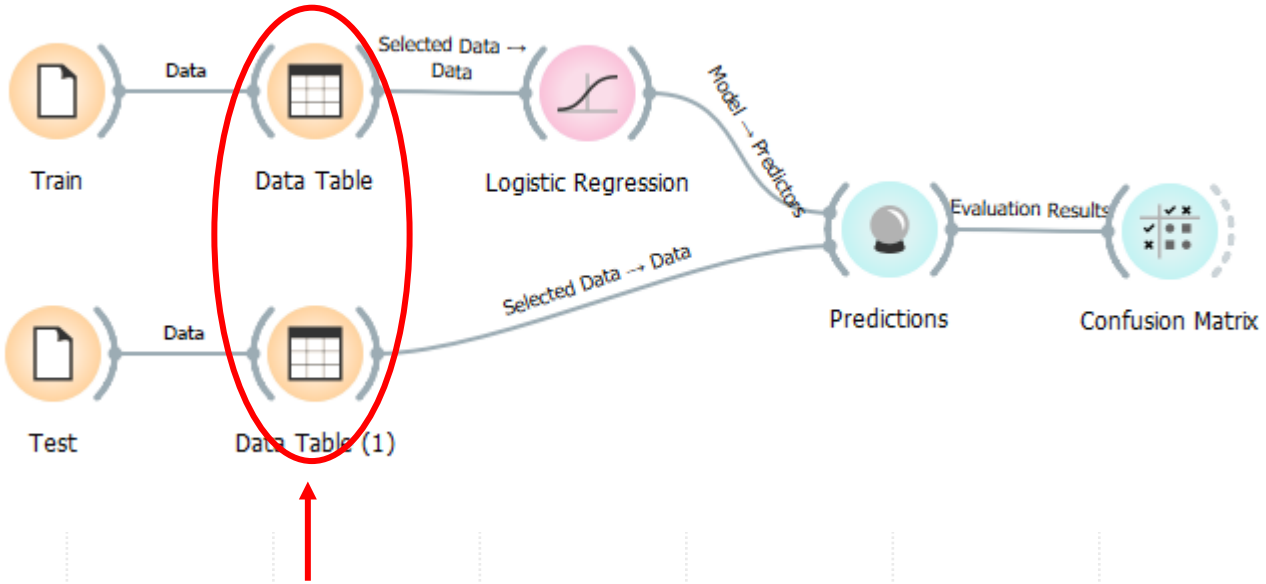
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, classification_report
```

Load Data



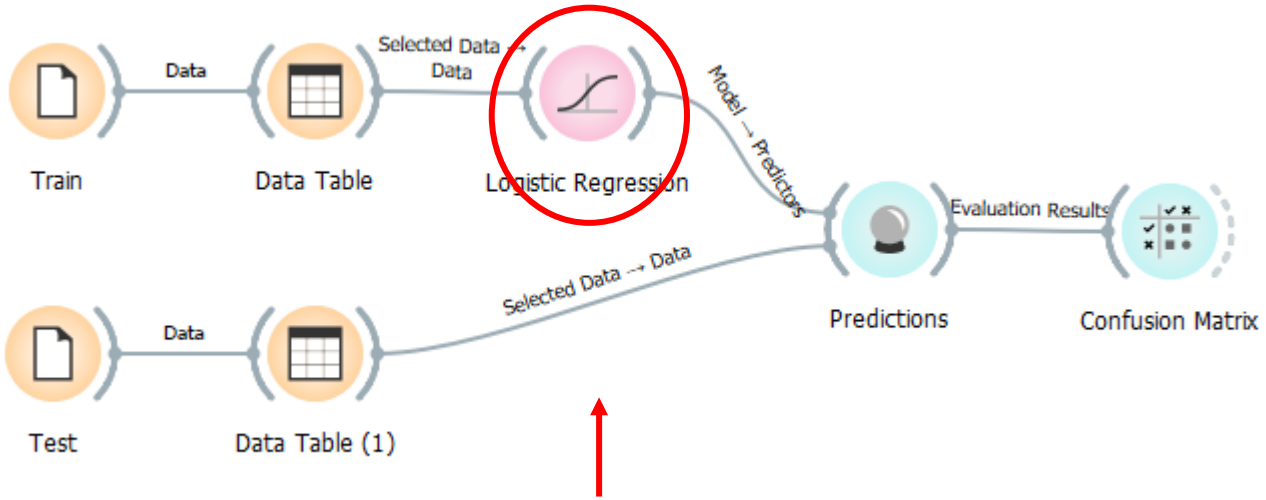

```
iris = load_iris()
X = iris.data
y = iris.target
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.4,
random_state=10)
```

Preprocess



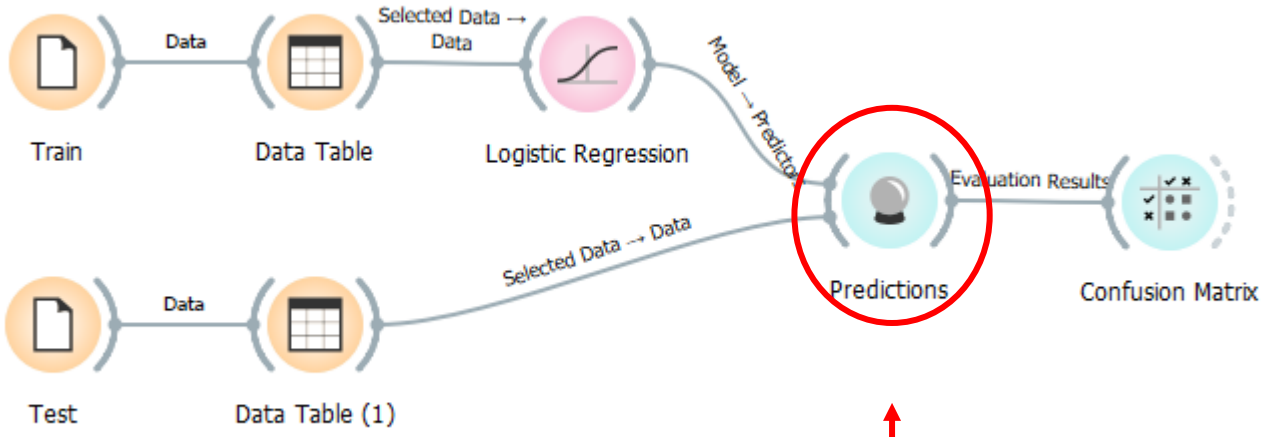

```
df = pd.DataFrame(X, columns=iris.feature_names)
df['target'] = y
df
```

Train Logistic Regression Model



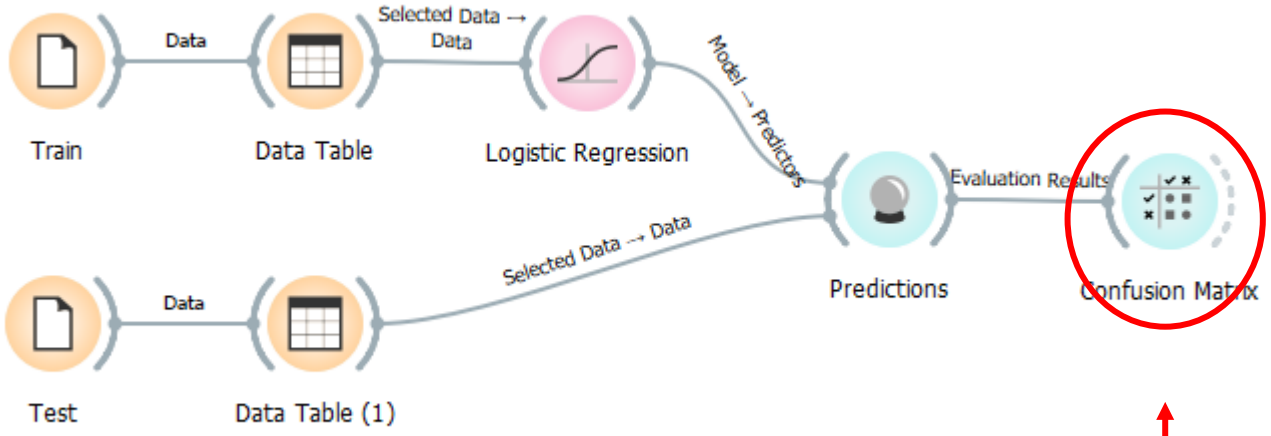

```
logreg = LogisticRegression(max_iter=200)
logreg.fit(X_train, y_train)
```

Prediction




```
y_pred = logreg.predict(X_test)
```

Evaluation




```
print("Accuracy:", accuracy_score(y_test, y_pred))
print("\nClassification Report:\n", classification_report(y_test,
y_pred, target_names=iris.target_names))
```

Now your turn!