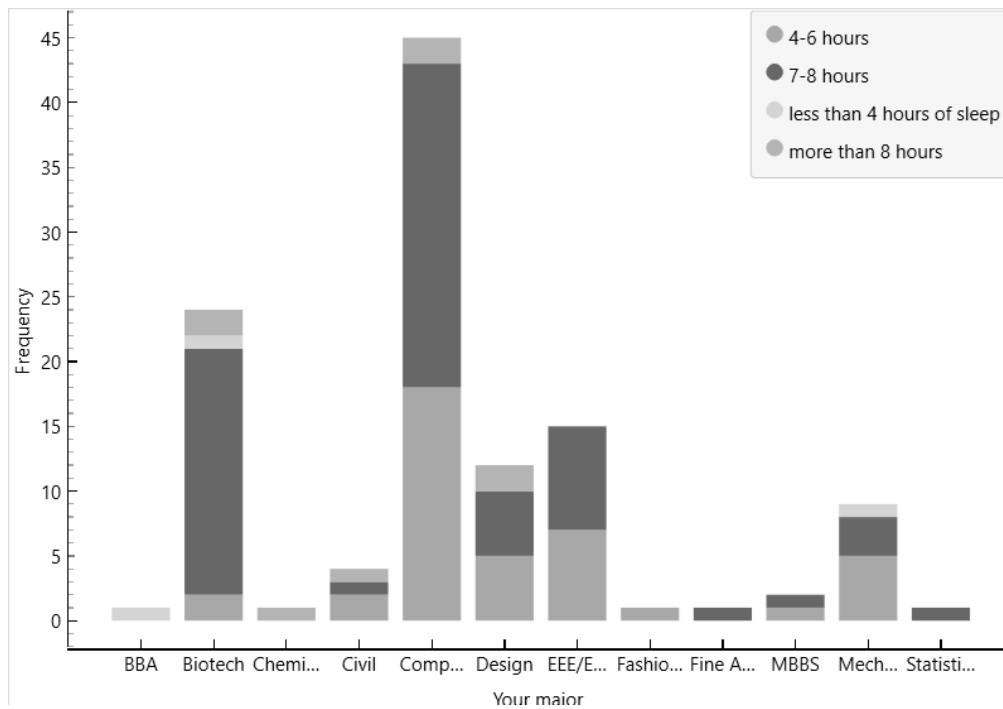
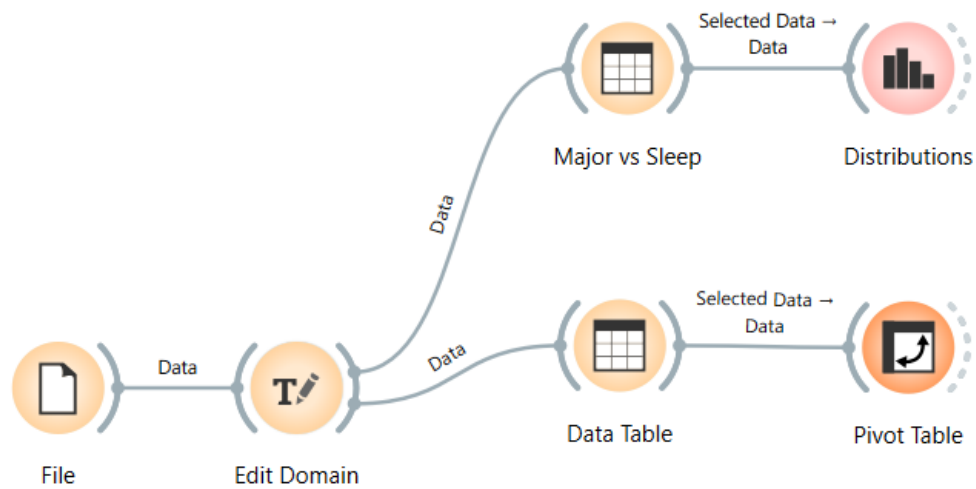


## OUTPUT:



Pivot Table - Orange

Rows

Age

Columns

Gender

Values

Your major

Aggregations

☒ Count

☐ Count defined

☒ Apply Automatically

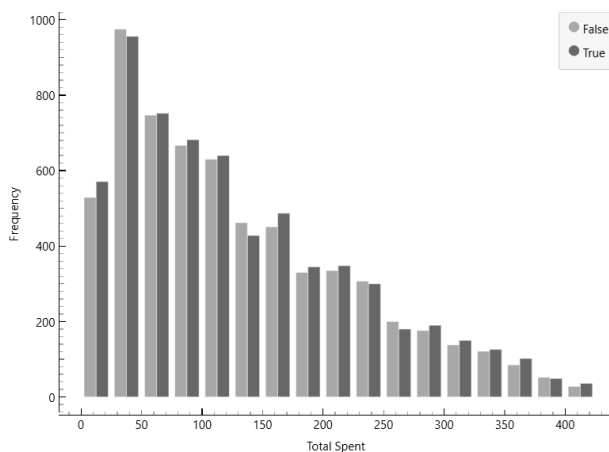
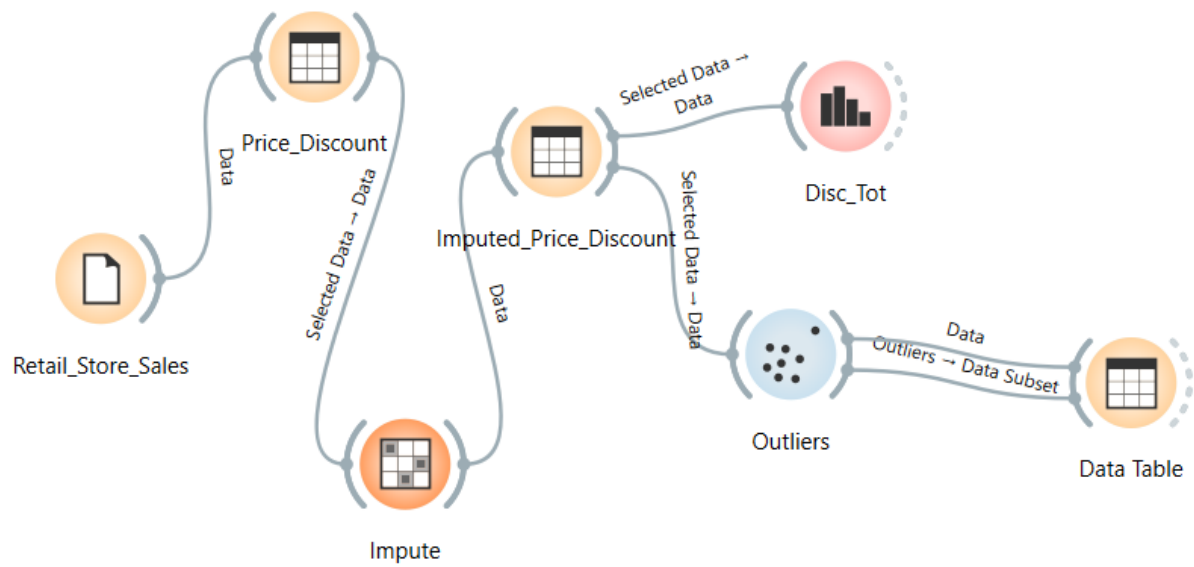
		Gender			
	Count	Female	Male	Non-binary	Total
Age	18-20	48.0	46.0	2.0	96.0
	21-23	14.0	5.0	0.0	19.0
	24-25	0.0	1.0	0.0	1.0
	Total	62.0	52.0	2.0	116.0

116 | 3 | 19

## PROCEDURE:

1. **Load the Dataset:** Begin the workflow by adding a File widget and loading the “sleep pattern” dataset.
2. **Pre-process the Data:** Connect the File widget to an Edit Domain widget to perform data type conversions, specifically changing necessary attributes to categorical data or vice versa.
3. **Create Data Branch:** Create a separate link from the Edit Domain widget's output to a Data Table widget.
4. **Explore Distributions:** From the Data Table, connect to a Distributions widget. Select the major and hours of sleep attributes to visualize their distributions.
5. **Create a Data Cube:** Create a new link from the Edit Domain widget's output to a Pivot Table widget.
6. **Configure Pivot Table:** In the Pivot Table widget, set age as the rows, gender as the columns, and major as the values to create a data cube summarizing the population demographics.

## OUTPUT:



**Outliers - Orange**

Method: Isolation Forest

Parameters:

Contamination: 33 %

☒ Replicable training

☒ Apply Automatically

12.6k | 8439 | 4136 | 12.6k

**Data Table - Orange**

Info: 12575 instances (no missing data), 2 features, No target variable, 1 meta attribute.

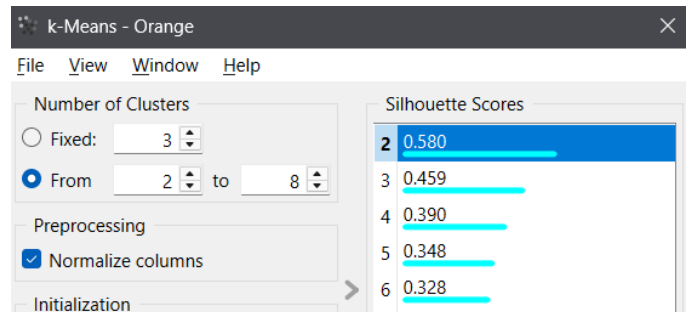
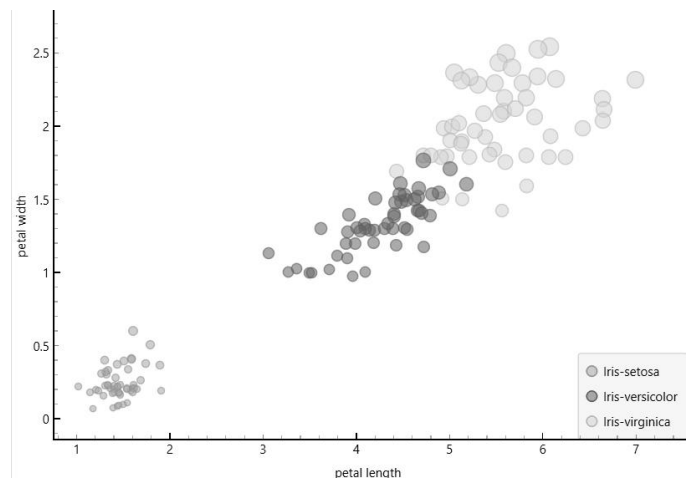
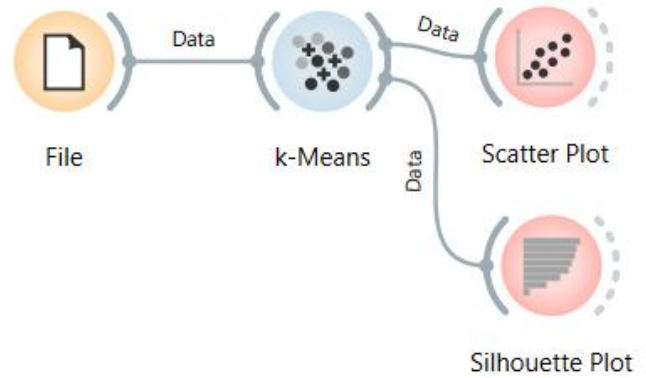
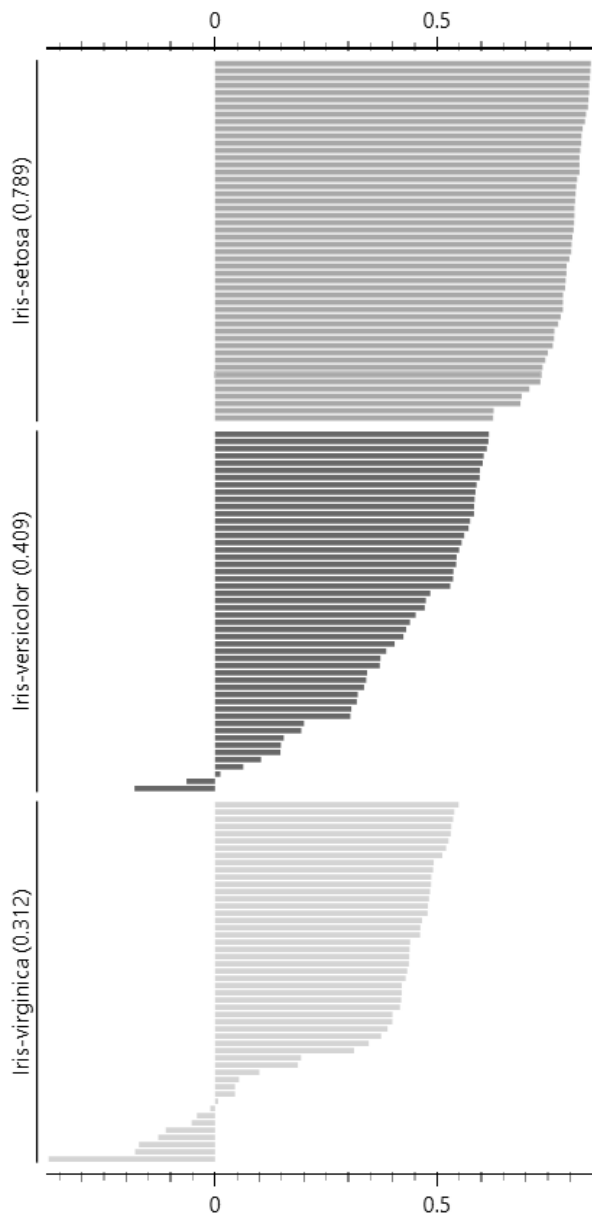
Variables: ☒ Show variable labels (if present),  ☒ Send Automatically

	Outlier	Total Spent	Discount Applied
• 1	Yes	185.0	True
• 2	Yes	261.0	True
3	No	43.0	False
• 4	Yes	247.5	False
5	No	87.5	False
• 6	Yes	200.0	False
7	No	40.0	True

## PROCEDURE:

1. **Load the Dataset:** Begin the workflow by loading the “Retail\_Store\_Sales” dataset in a File widget.
2. **Select Columns:** Connect the File widget to a Data Table widget to select the price and discount columns for the experiment.
3. **Impute Missing Values:** Connect the data to an Impute widget. Fill any missing data points by using the random values method.
4. **Analyze Distributions:** Connect the imputed data to a Distributions widget. Analyze the distribution of the total spent and discount applied attributes.
5. **Detect Outliers:** Connect the imputed data to an Outliers widget. Use the Isolation Forest method with a 33% contamination rate to detect outliers.
6. **Inspect Outliers:** Connect the output of the Outliers widget to a final Data Table widget to view and inspect the identified outlier data.

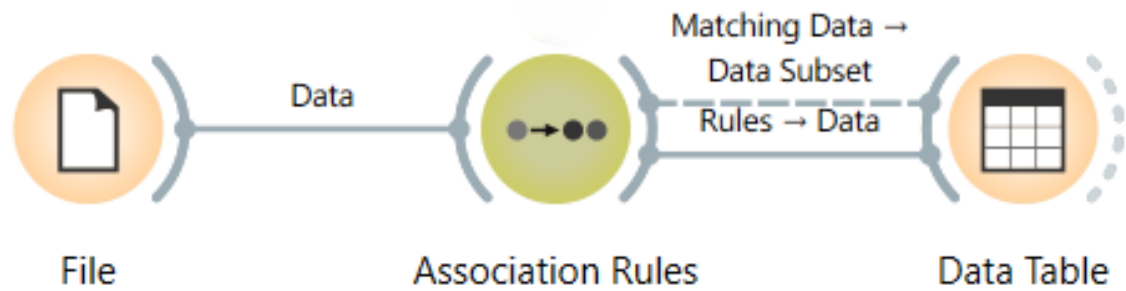
## OUTPUT:



## PROCEDURE:

1. **Load the Dataset:** Begin by loading the “iris” dataset into a File widget.
2. **Perform Clustering:** Connect the dataset to a k-Means widget to perform clustering on the petal length, petal width, sepal length, and sepal width attributes.
3. **Set Cluster Range:** Set the number of clusters (k) to a range of 2 to 8 to find the optimal number of groups within the data.
4. **Visualize Clusters:** Connect the output to a Scatter Plot widget. Plot the clusters by setting petal length on the X-axis and petal width on the Y-axis.
5. **Evaluate Cluster Quality:** To evaluate the quality of the clusters, connect the output to a Silhouette Plot widget.
6. **Analyze Results:** Inspect the results from both the Scatter Plot and the Silhouette Plot to analyze the identified clusters and determine the most effective number of clusters for the dataset.

## OUTPUT:



**Data Table - Orange**

File Edit View Window Help

Info  
16 instances (no missing data)  
6 features  
No target variable.  
2 meta attributes

Variables  
☒ Show variable labels (if present)  
☐ Visualize numeric values  
☒ Color by instance classes

Selection  
☐ Select full rows

Restore Original Order  
☒ Send Automatically

	Antecedent	Consequent	Support	Confidence	Coverage	Strength	Lift	Leverage
1	antioxydant juic...	spinach=mayo...	0	1	0	1	7500	0
2	spinach=mayo...	antioxydant juic...	0	1	0	1	7500	0
3	antioxydant juic...	frozen smoothi...	0	1	0	2	3750	0
4	frozen smoothi...	antioxydant juic...	0	1	0	2	3750	0
5	spinach=mayo...	frozen smoothi...	0	1	0	2	3750	0
6	spinach=mayo...	antioxydant juic...	0	1	0	2	3750	0
7	frozen smoothi...	antioxydant juic...	0	1	0	1	7500	0
8	antioxydant juic...	spinach=cereals	0	1	0	1	7500	0
9	spinach=cereals	antioxydant juic...	0	1	0	1	7500	0
10	antioxydant juic...	frozen smoothi...	0	1	0	1	7500	0
11	frozen smoothi...	antioxydant juic...	0	1	0	2	3750	0
12	frozen smoothi...	spinach=cereals	0	1	0	1	7500	0
13	spinach=cereals	frozen smoothi...	0	1	0	1	7500	0
14	spinach=cereals	antioxydant juic...	0	1	0	2	3750	0
15	frozen smoothi...	antioxydant juic...	0	1	0	2	3750	0
16	antioxydant juic...	frozen smoothi...	0	1	0	2	3750	0

16 | 16

**Association Rules - Orange**

File View Window Help

Info  
Rules: 16 (shown 16)

Find association rules  
Min. supp.: 0.01 %  
Min. conf.: 60 %  
Max. rules: 10k  
☐ Induce only classification rules  
☒ Restrict search by below filters

Find Rules

Filter by Antecedent  
Contains:  
Items, min: 1 max: 999

Filter by Consequent  
Contains:  
Items, min: 1 max: 999

☒ Send selection

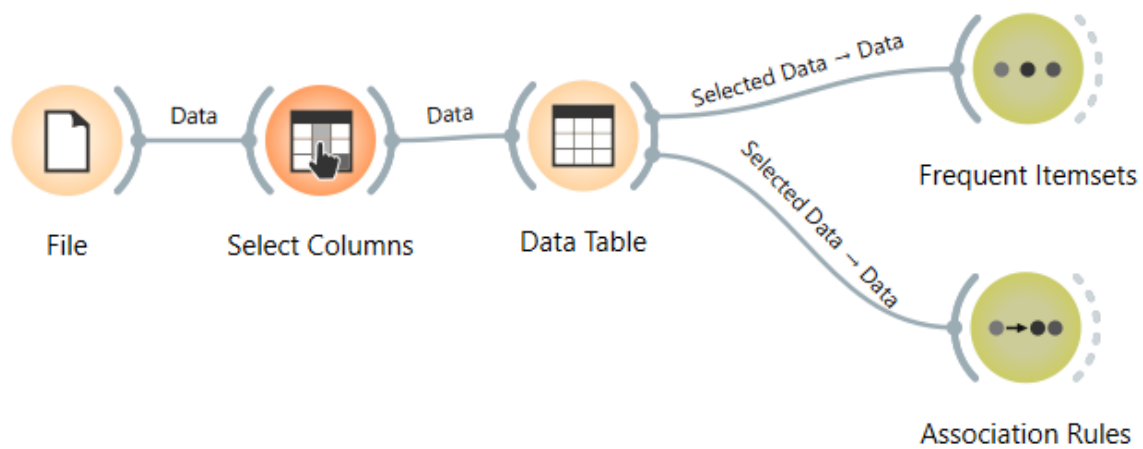
Supp	Conf	Covr	Strg	Lift	Levr	Antecedent	Consequent
0....	1....	0....	1....	7500....	0....	antioxydant juice=frozen smoothie, ...	spinach=mayonnaise
0....	1....	0....	2....	3750....	0....	antioxydant juice=frozen smoothie, ...	frozen smoothie=protein bar
0....	1....	0....	2....	3750....	0....	frozen smoothie=protein bar, ...	antioxydant juice=frozen smoothie
0....	1....	0....	1....	7500....	0....	antioxydant juice=frozen smoothie, ...	spinach=cereals
0....	1....	0....	1....	7500....	0....	antioxydant juice=frozen smoothie, ...	frozen smoothie=spinach
0....	1....	0....	2....	3750....	0....	frozen smoothie=spinach, ...	antioxydant juice=frozen smoothie
0....	1....	0....	1....	7500....	0....	spinach=mayonnaise	antioxydant juice=frozen smoothie, frozen smoothie=
0....	1....	0....	2....	3750....	0....	spinach=mayonnaise	frozen smoothie=protein bar
0....	1....	0....	2....	3750....	0....	spinach=mayonnaise	antioxydant juice=frozen smoothie
0....	1....	0....	1....	7500....	0....	frozen smoothie=spinach	antioxydant juice=frozen smoothie, spinach=cereals
0....	1....	0....	1....	7500....	0....	spinach=cereals	antioxydant juice=frozen smoothie, frozen smoothie=
0....	1....	0....	1....	7500....	0....	frozen smoothie=spinach	spinach=cereals
0....	1....	0....	1....	7500....	0....	spinach=cereals	frozen smoothie=spinach
0....	1....	0....	2....	3750....	0....	spinach=cereals	antioxydant juice=frozen smoothie
0....	1....	0....	2....	3750....	0....	frozen smoothie=spinach	antioxydant juice=frozen smoothie
0....	1....	0....	2....	3750....	0....	antioxydant juice=french fries	frozen smoothie=protein bar

## PROCEDURE:

1. **Load the Dataset:** Begin the workflow by loading the “store\_data” CSV file into a File widget.
2. **Establish Analysis Goal:** The objective is to identify strong relationships and co-occurrence patterns among products in a transaction dataset.
3. **Generate Association Rules:** Connect the data to an Association Rules widget to find the relationships between products in the billing data.
4. **Set Support Threshold:** In the widget's settings, set the Support to 0.01%. This controls the minimum frequency of the product sets.
5. **Set Confidence Threshold:** Set the Confidence to 60%. This defines the likelihood that a rule is true.
6. **Analyze the Rules:** Connect the output of the Association Rules widget to a Data Table to view and analyze the generated rules and their confidence scores.



## OUTPUT:



**Frequent Itemsets - Orange**

View Window Help

Info  
 Number of itemsets: 11  
 Selected itemsets: 0  
 Selected examples: 0  
 Expand all Collapse all

Find itemsets  
 Minimal support: 35%  
 Max. number of itemsets: 10000  
☐ Find Itemsets

Itemsets	Support	%
✓ item1=bread	5	83.33
✓ item5=sugar	4	66.67
item6=juice	3	50
item3=jam	3	50
item6=juice	3	50
✓ item4=milk	3	50
item5=sugar	3	50
✓ item5=sugar	5	83.33
item6=juice	3	50
item3=jam	3	50
item6=juice	3	50

**Association Rules - Orange**

File View Window Help

Info  
 Rules: 44 (shown 44)

Find association rules  
 Min. supp.: 0.01 %  
 Min. conf.: 60 %  
 Max. rules: 10k  
☐ Induce only classification rules  
☒ Restrict search by below filters  
 Find Rules

Filter by Antecedent  
 Contains:  
 Items, min: 1 max: 999

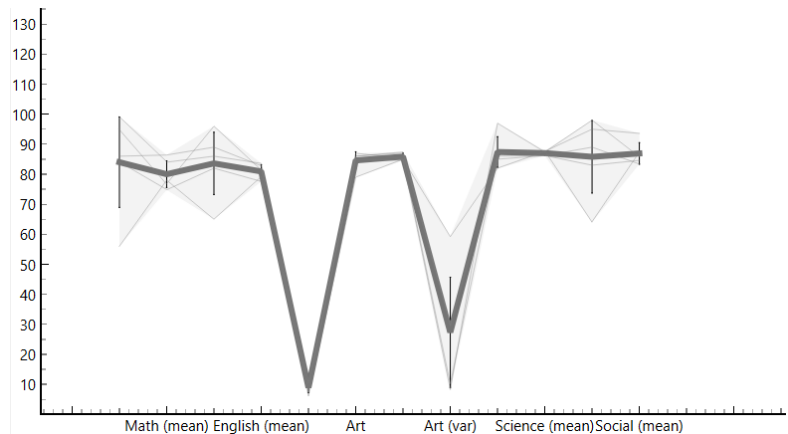
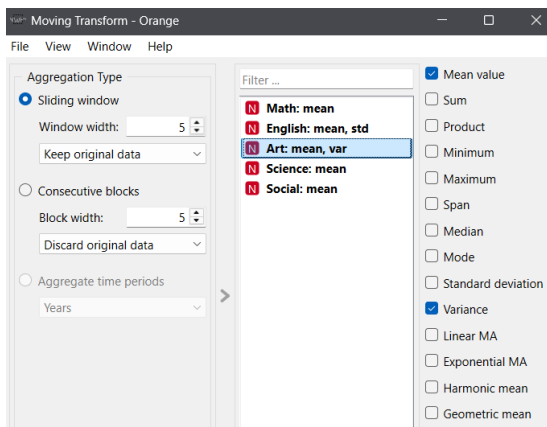
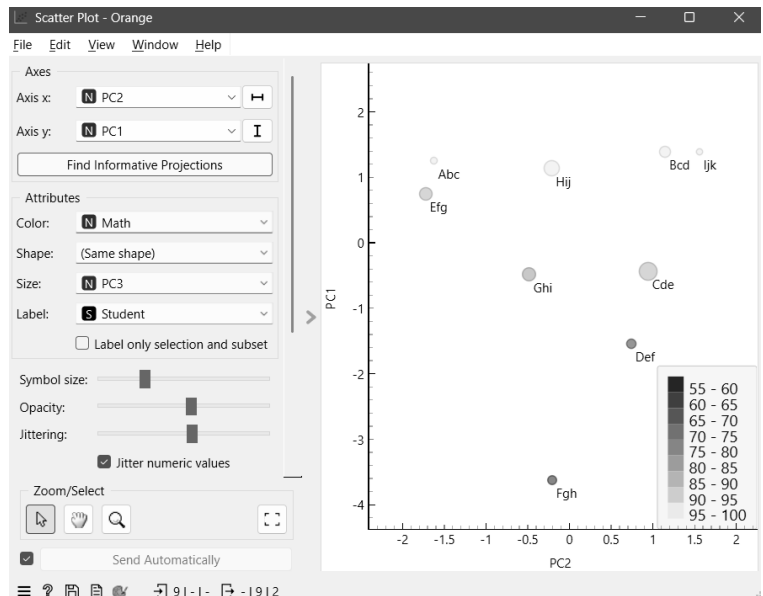
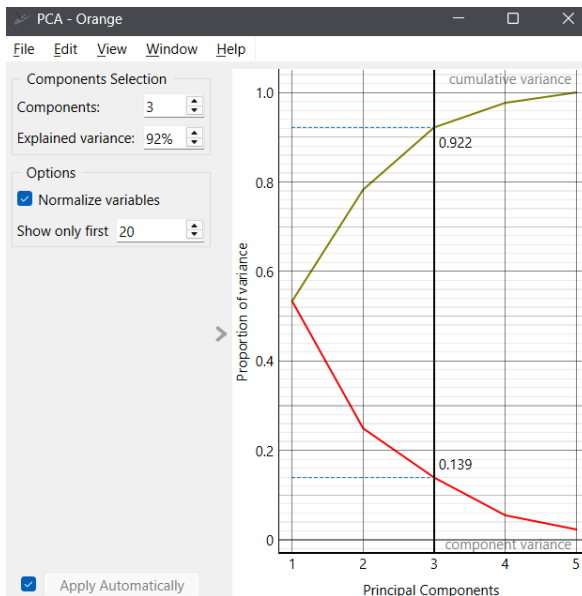
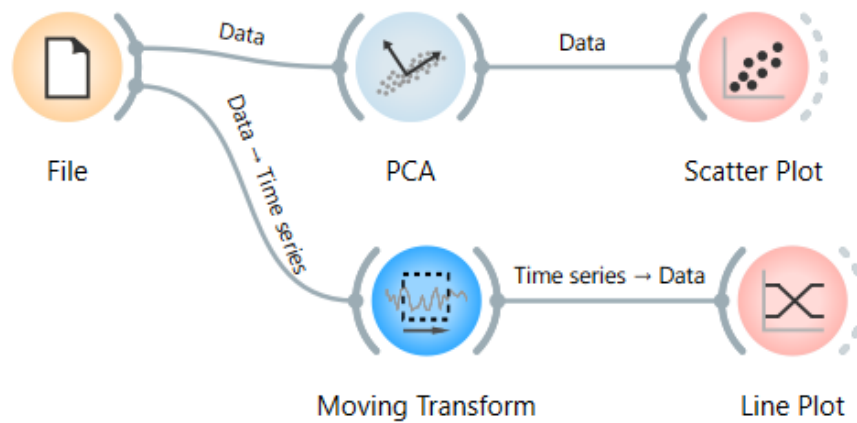
Filter by Consequent  
 Contains:  
 Items, min: 1 max: 999

Supp	Conf	Covr	Strg	Lift	Levr	Antecedent	Consequent
0.167	1.000	0.167	4.000	1.500	0.056	item3=jam, item6=juice	→ item1=bread, item5=sugar
0.167	1.000	0.167	5.000	1.200	0.028	item1=bread, item3=jam, item6=juice	→ item5=sugar
0.167	1.000	0.167	5.000	1.200	0.028	item5=sugar, item3=jam, item6=juice	→ item1=bread
0.167	1.000	0.167	5.000	1.200	0.028	item3=jam, item6=juice	→ item5=sugar
0.167	1.000	0.167	5.000	1.200	0.028	item3=jam, item6=juice	→ item1=bread
0.167	1.000	0.167	4.000	1.500	0.056	item4=milk, item6=juice	→ item1=bread, item5=sugar
0.167	1.000	0.167	5.000	1.200	0.028	item1=bread, item4=milk, item6=juice	→ item5=sugar
0.167	1.000	0.167	5.000	1.200	0.028	item4=milk, item5=sugar, item6=juice	→ item1=bread
0.167	1.000	0.167	5.000	1.200	0.028	item4=milk, item6=juice	→ item5=sugar
0.500	0.600	0.833	0.600	1.200	0.083	item1=bread	→ item5=sugar, item6=juice
0.500	0.600	0.833	0.600	1.200	0.083	item5=sugar	→ item1=bread, item6=juice
0.500	0.750	0.667	0.750	1.500	0.167	item1=bread, item5=sugar	→ item6=juice
0.500	1.000	0.500	1.333	1.500	0.167	item6=juice	→ item1=bread, item5=sugar
0.500	1.000	0.500	1.667	1.200	0.083	item1=bread, item6=juice	→ item5=sugar
0.500	1.000	0.500	1.667	1.200	0.083	item5=sugar, item6=juice	→ item1=bread

## **PROCEDURE:**

1. **Load the Dataset:** Begin the workflow by loading a transactional dataset into a File widget.
2. **Select Attributes:** Connect the data to a Select Columns widget to choose the attributes containing the transaction items for analysis.
3. **Prepare the Data:** Connect the selected data to a Data Table widget to inspect and prepare the attributes for rule generation.
4. **Find Frequent Itemsets:** Connect the data to a Frequent Itemsets widget. In the settings, find itemsets with a minimum support of 35%.
5. **Generate Association Rules:** Create a separate link from the Data Table to an Association Rules widget. Here, set the minimum support to 0.01% and the minimum confidence to 60% to generate the rules.
6. **Analyze Hypotheses:** Examine the Frequent Itemsets and Association Rules outputs. Use these results to generate and test hypotheses, such as the likelihood that customers who buy one product will also buy another.

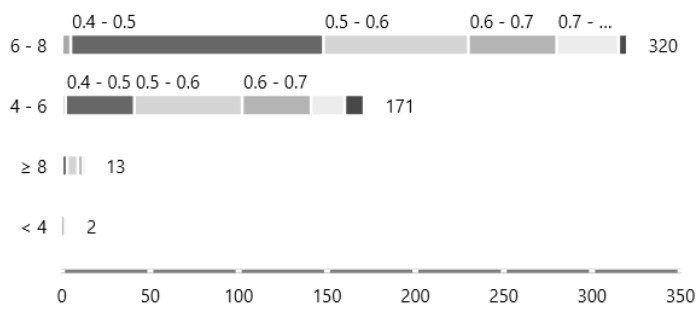
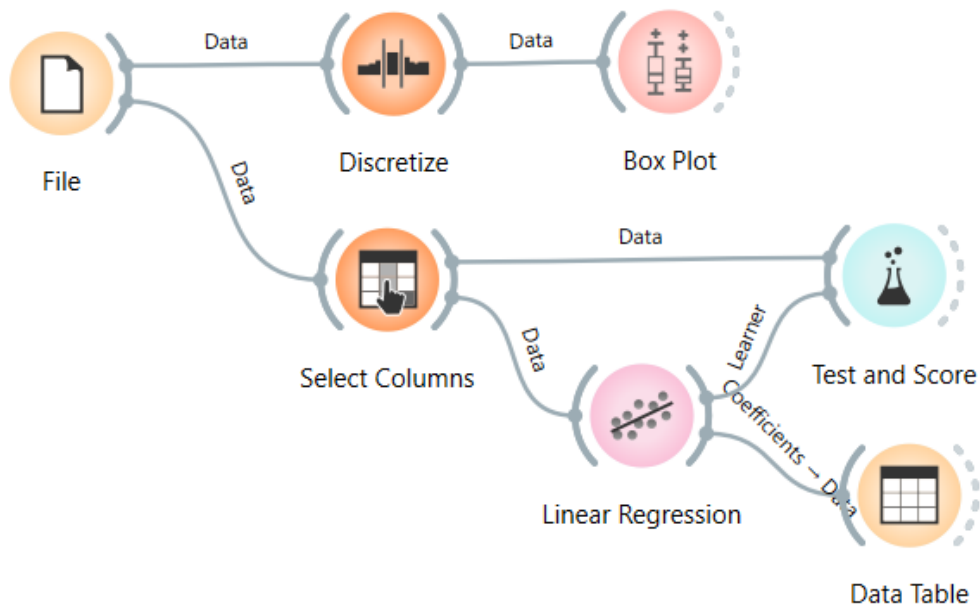
## OUTPUT:



## **PROCEDURE:**

1. **Load the Dataset:** Import the student marks dataset using the File widget to begin the analysis.
2. **Apply PCA:** Connect the dataset to a PCA widget to perform dimensionality reduction and capture key patterns in the marks.
3. **Visualize PCA Output:** Link the PCA results to a Scatter Plot widget to observe relationships and clustering among the students' performance.
4. **Transform Time Series:** Send the dataset from the File widget to a Moving Transform widget to generate transformed values over a moving window.
5. **Plot Transformed Data:** Connect the Moving Transform output to a Line Plot widget to visualize the smoothed trends in student marks.
6. **Compare Insights:** Examine both the scatter plot and line plot to understand performance variations and transformations in the dataset.

## OUTPUT:



$\chi^2$ : 38.12 (p=0.001, dof=15)

	name	coef
1	CRIM	-0.108011
2	ZN	0.0464205
3	INDUS	0.0205586
4	NOX	-17.7666
5	RM	3.80987
6	AGE	0.000692225
7	DIS	-1.47557
8	RAD	0.306049

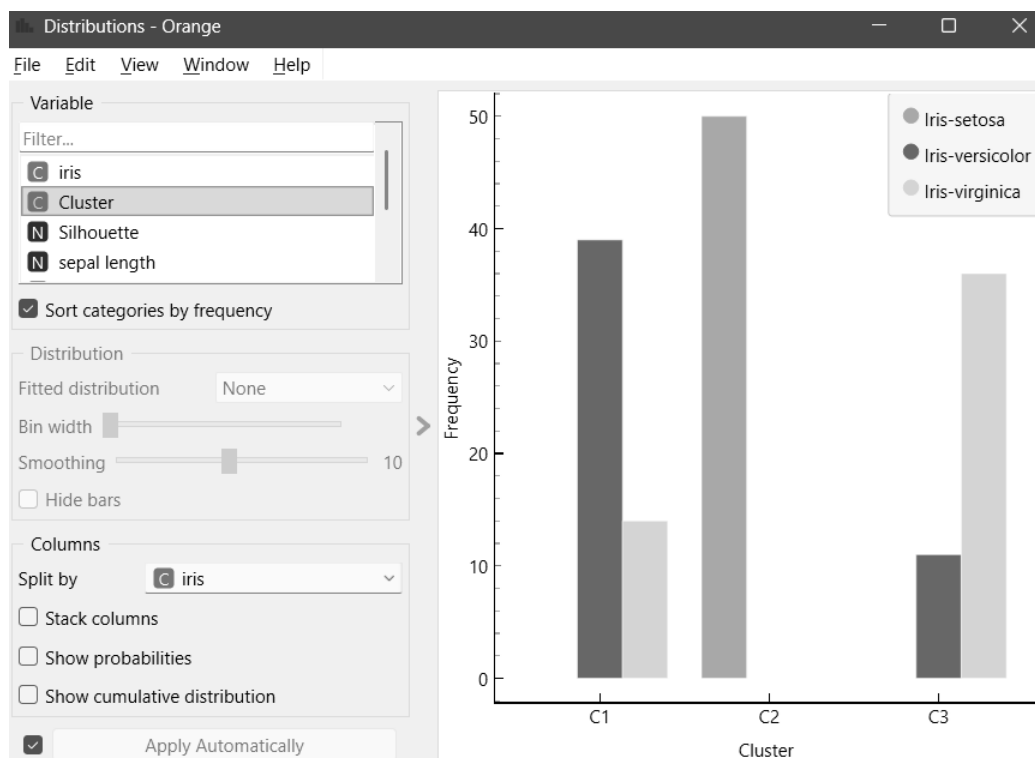
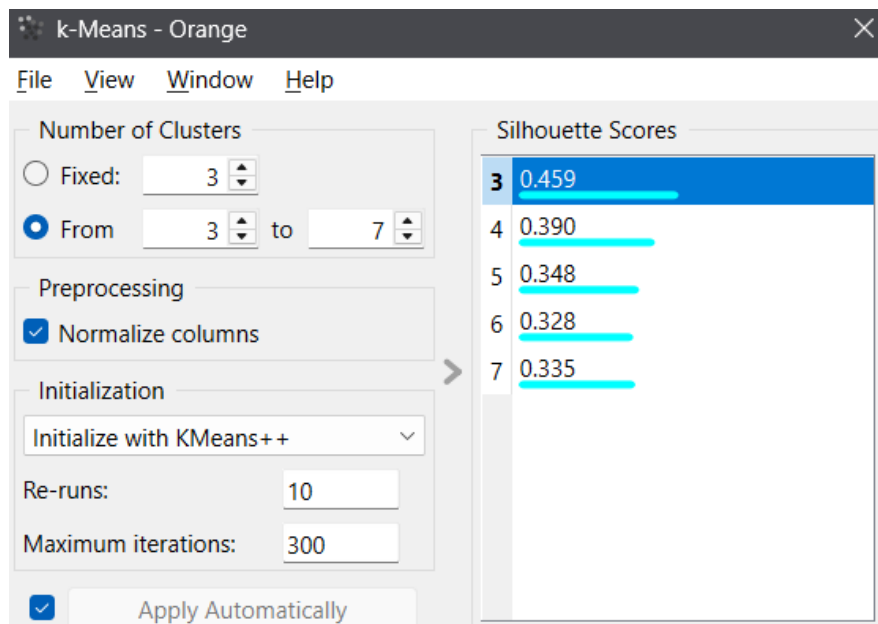
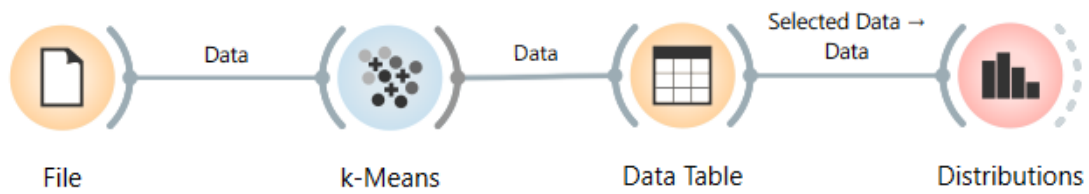
File Edit View Window Help

<input checked="" type="radio"/> Cross validation Number of folds: 10 <input type="checkbox"/> Stratified <input type="radio"/> Cross validation by feature						
Model	MSE	RMSE	MAE	MAPE	R2	
Linear Regression	23....	4.834	3.376	17.017	0.723	

## PROCEDURE:

1. **Load the Dataset:** Begin by importing the Boston Housing dataset through the File widget to make the data available for analysis.
2. **Discretize Data:** Connect the dataset to a Discretize widget and convert continuous attributes into categorical bins for easier interpretation.
3. **Visualize Distribution:** Link the Discretize output to a Distribution widget to display the frequency patterns of the binned variables.
4. **Visualize Box Plot:** Add a Box Plot widget to the workflow and connect it to study how attribute values vary across different bins.
5. **Apply Linear Regression:** Use the Linear Regression widget by specifying the target variable and predictors to generate a regression model.
6. **Evaluate Model:** Connect the regression model to the Test & Score widget to obtain performance metrics such as  $R^2$  and RMSE for evaluation.

## OUTPUT:

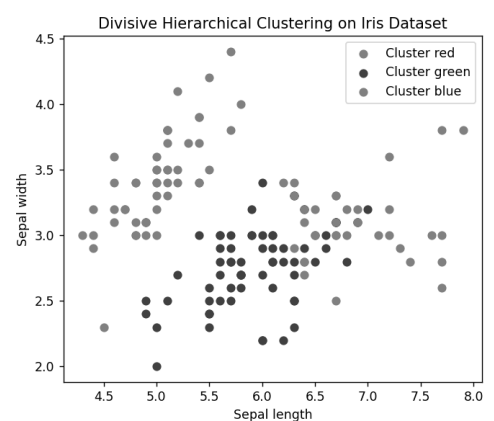
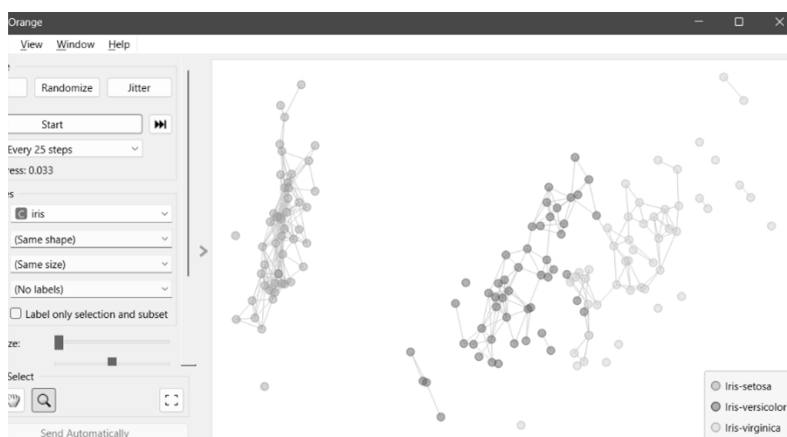
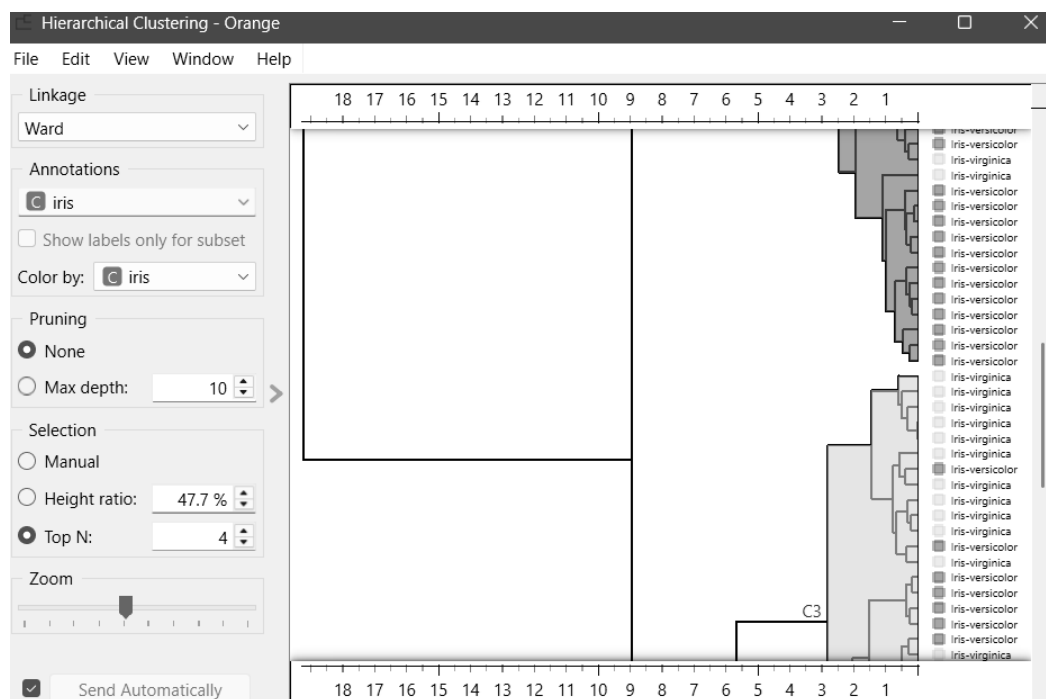
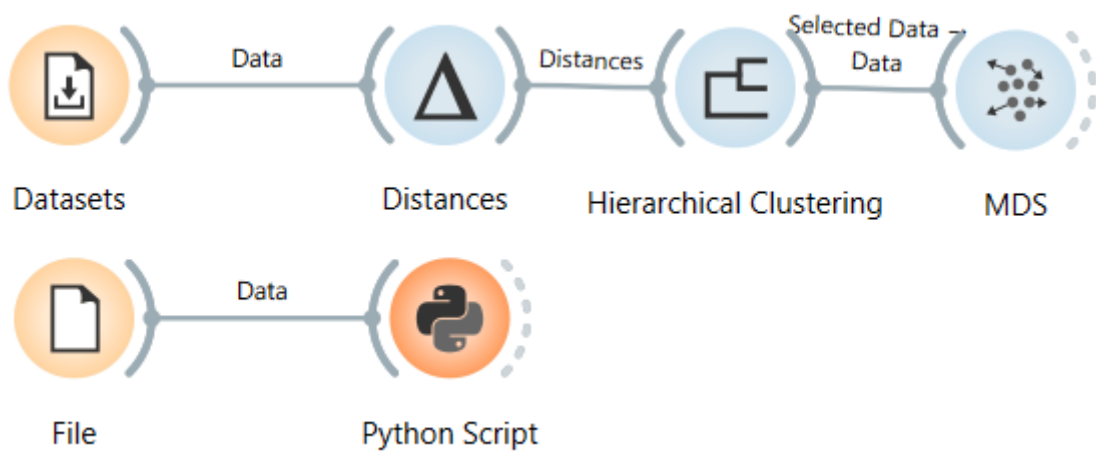


## PROCEDURE:

1. **Load Dataset:** Begin by importing the Iris dataset using the File widget to provide the data for clustering.
2. **Apply Clustering:** Connect the dataset to the k-Means widget and configure the number of clusters to range between three and seven.
3. **Generate Cluster Labels:** The k-Means widget assigns each data point to a cluster, adding a new cluster attribute to the dataset.
4. **Inspect Data:** Connect the output of the k-Means widget to a Data Table widget to examine the cluster assignments alongside the original attributes.
5. **Visualize Distributions:** Link the Data Table to the Distributions widget and configure it to plot attribute distributions grouped by the cluster attribute.
6. **Analyze Results:** Observe the histograms in the Distributions widget to compare how different clusters vary in terms of attribute values.



## OUTPUT:



## PROCEDURE:

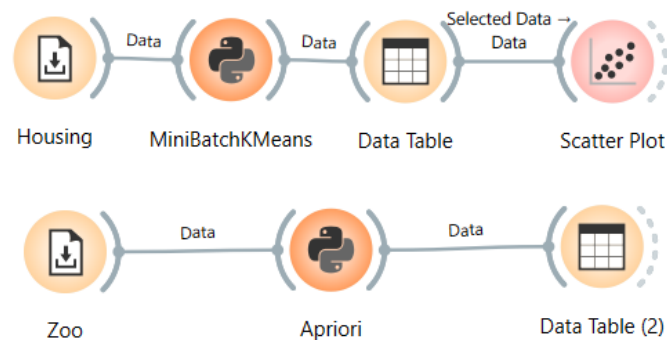
1. **Load the Dataset:** Start by loading the Iris dataset using the File widget as the common input source.
2. **Branch the Workflow:** From the File widget, create two branches: one for hierarchical clustering and one for divisive clustering.
3. **Compute Distances:** In the hierarchical branch, connect the data to the Distances widget to calculate pairwise distances.
4. **Perform Agglomerative Clustering:** Pass the distances to the Hierarchical Clustering widget to generate a dendrogram for agglomerative clustering.
5. **Apply Divisive Clustering:** In the second branch, connect the File widget to a Python Script widget and run the provided code to perform divisive clustering.

```
import numpy as np
from sklearn.cluster import AgglomerativeClustering
from Orange.data import Table
import matplotlib.pyplot as plt
data = Table("iris")
X = np.array([list(row)[-1] for row in data])
def divisive_clustering(X, n_clusters=3):
    clusters = [X]
    while len(clusters) < n_clusters:
        cluster_to_split = max(clusters, key=len)
        clusters.remove(cluster_to_split)
        clustering = AgglomerativeClustering(n_clusters=2)
        labels = clustering.fit_predict(cluster_to_split)
        sub_cluster_1 = cluster_to_split[labels == 0]
        sub_cluster_2 = cluster_to_split[labels == 1]
        clusters.append(sub_cluster_1)
        clusters.append(sub_cluster_2)
    return clusters
clusters = divisive_clustering(X, n_clusters=3)
colors = ['red', 'green', 'blue']
plt.figure(figsize=(6, 5))
```

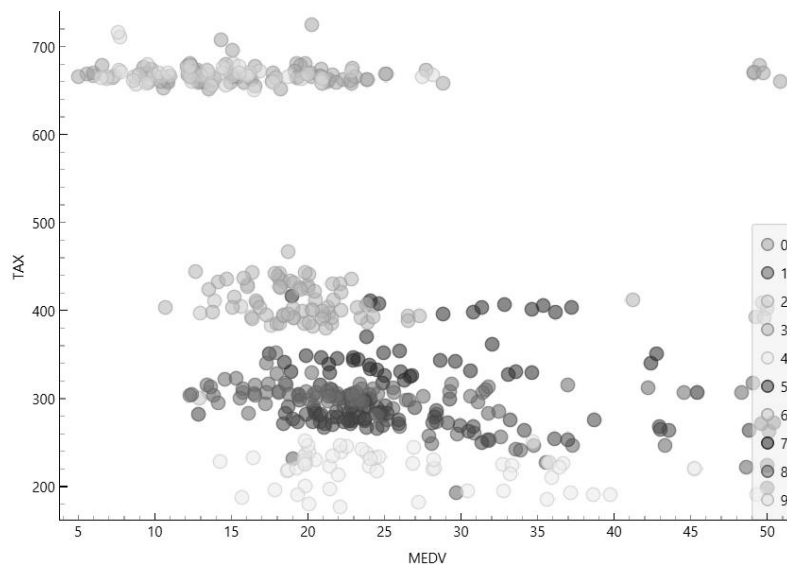
```
for cluster, color in zip(clusters, colors):  
    plt.scatter(cluster[:, 0], cluster[:, 1], c=color, label=f'Cluster {color}')  
plt.xlabel('Sepal length')  
plt.ylabel('Sepal width')  
plt.title('Divisive Hierarchical Clustering on Iris Dataset')  
plt.legend()  
plt.show()
```

6. **Visualize and Compare:** Use the MDS widget in the hierarchical branch to visualize clusters in two dimensions, then compare the dendrogram (agglomerative) and script output (divisive).

## OUTPUT:



Data Table - Orange				
File Edit View Window Help				
<b>Info</b> 506 instances (no missing data) 13 features Numeric outcome 1 meta attribute				
<b>Variables</b> <input checked="" type="checkbox"/> Show variable labels (if present)				
		MEDV	Cluster	CRIM
1		24.0	5	0.00632
2		21.6	4	0.02731
3		34.7	4	0.02729
4		33.4	4	0.03237
5		36.2	4	0.06905



	antecedents	consequents	ntecedent suppor	onsequent suppor	support	confidence
1	tail	backbone	0.75	0.82	0.74	0.986667
2	catsize	backbone	0.44	0.82	0.43	0.977273
3	milk	toothed	0.41	0.6	0.4	0.97561
4	milk	hair	0.41	0.43	0.39	0.95122
5	hair	milk	0.43	0.41	0.39	0.906977
6	hair	backbone	0.43	0.82	0.39	0.906977
7	backbone	tail	0.82	0.75	0.74	0.902439
8	catsize	breathes	0.44	0.79	0.39	0.886364
9	hair	toothed	0.43	0.6	0.38	0.883721

## PROCEDURE:

1. **Load the Dataset:** Drag the Dataset widget onto the Orange canvas and choose the Housing dataset. This dataset is suitable for clustering experiments because it contains multiple numeric attributes.
2. **Connect for Clustering:** Link the Housing dataset to a Python Script widget. Inside the script, import the MiniBatchKMeans library from scikit-learn to carry out scalable clustering.

```
import numpy as np
from sklearn.cluster import MiniBatchKMeans
import Orange
data = in_data
X = np.asarray(data.X, dtype=float)
k = 10
batch_size = 100
n_init = 10
mbk = MiniBatchKMeans(n_clusters=k, batch_size=batch_size, n_init=n_init)
labels = mbk.fit_predict(X).astype(int)
cluster_var = Orange.data.DiscreteVariable(
    "Cluster", values=[str(i) for i in range(k)]
)
new_domain = Orange.data.Domain(
    data.domain.attributes,
    data.domain.class_vars,
    metas=list(data.domain.metas) + [cluster_var]
)
if data.metas.size == 0:
    new_metas = labels.reshape(-1, 1)
else:
    new_metas = np.column_stack([data.metas, labels.reshape(-1, 1)])
out_data = Orange.data.Table(new_domain, data.X, data.Y, new_metas)
```

3. **Generate Clusters:** Run MiniBatch K-Means on the Housing dataset by specifying the number of clusters and batch size. Ensure the script appends a new “Cluster” column to the dataset.
4. **View Cluster Labels:** Connect the Python Script output to a Data Table widget to confirm that the cluster column has been added, showing the cluster assignment of each instance.
5. **Visualize Clusters:** Send the clustered data to a Scatter Plot widget. Set MEDV (median value of homes) as the x-axis, TAX (property tax rate) as the y-axis, and color points by cluster. Observe the separation of data points across clusters.
6. **Load the Zoo Dataset:** Drag another Dataset widget and choose the Zoo dataset. This dataset is appropriate for association rule mining because it contains categorical “yes/no” attributes describing animals.
7. **Prepare Data for Apriori:** Connect the Zoo dataset to a Python Script widget. Inside the script, convert the “yes/no” attributes into Boolean values so they are suitable for frequent itemset mining.

```
import numpy as np
import pandas as pd
import Orange

from Orange.data.pandas_compat import table_to_frame, table_from_frame
from mlxtend.frequent_patterns import apriori, association_rules

data = in_data

df = table_to_frame(data, include metas=True)
df_bool = df.applymap(lambda x: True if str(x).lower() == "yes" else False)

frequent_itemsets = apriori(
    df_bool,
    min_support=0.3,
    use_colnames=True,
    max_len=2
)

rules = association_rules(
    frequent_itemsets,
    metric="confidence",
    min_threshold=0.6
).sort_values(["confidence", "lift"], ascending=False)
```

```
rules["antecedents"] = rules["antecedents"].apply(lambda x: ", ".join(list(x)))
rules["consequents"] = rules["consequents"].apply(lambda x: ", ".join(list(x)))
rules = rules.replace([np.inf, -np.inf], np.nan).dropna()
out_data = table_from_frame(rules.reset_index(drop=True))
```

8. **Run Apriori Algorithm:** Use the Apriori function from the mlxtend library to generate frequent itemsets. Specify minimum support and confidence thresholds to control the number of rules generated.
9. **Generate Association Rules:** From the frequent itemsets, produce association rules with support, confidence, and lift values. Convert antecedents and consequents into readable strings for easy interpretation.
10. **View and Analyze Rules:** Output the rules to a Data Table widget. Examine patterns such as {tail} → {backbone}, which confirm biologically meaningful associations. Compare the results of clustering on Housing and rule mining on Zoo to highlight the scalability of both approaches.