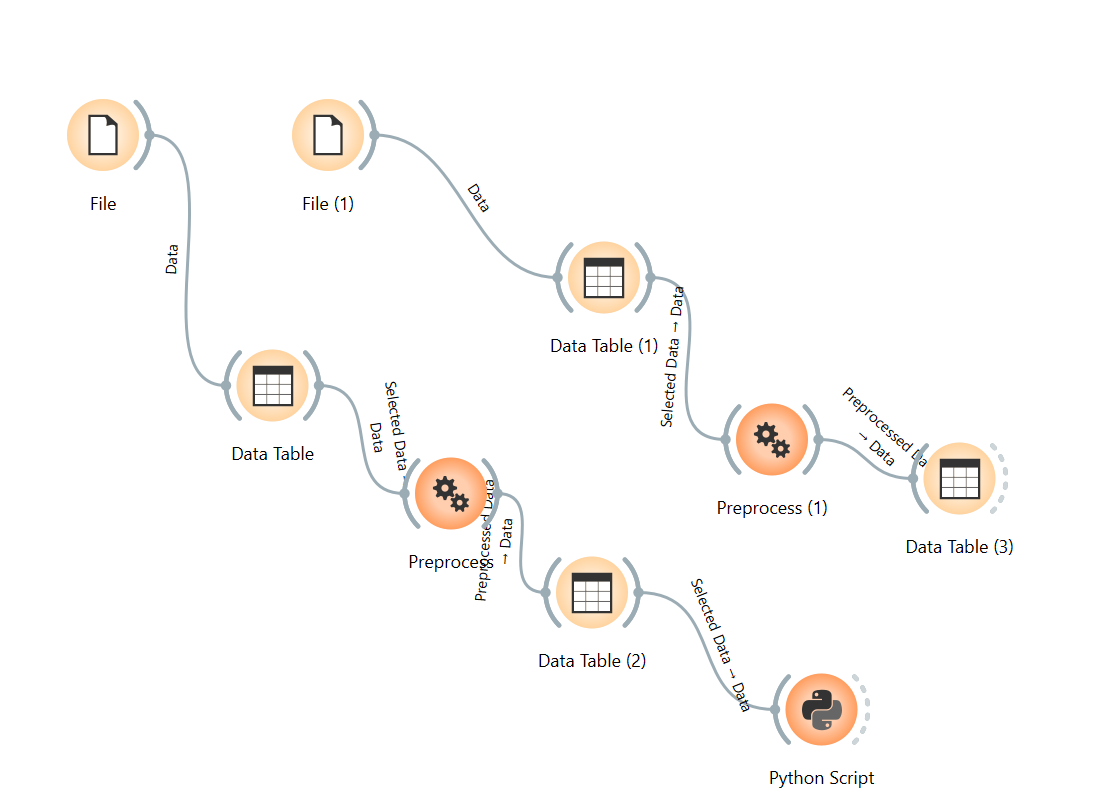
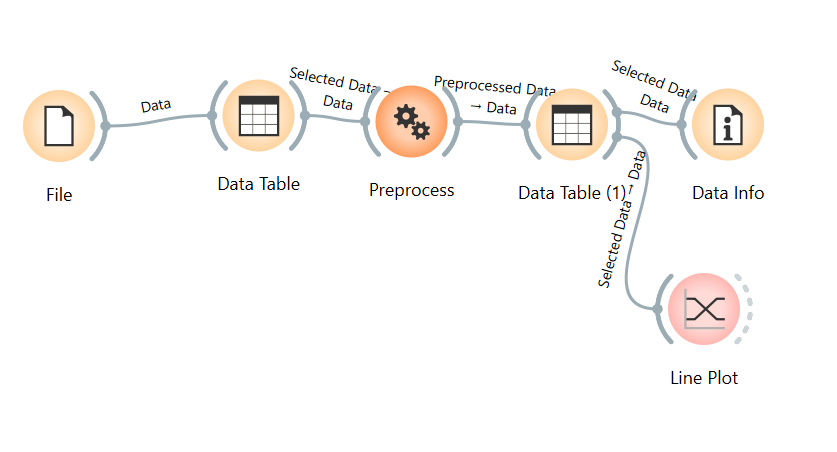
**EXP 1 DATA REPROCESSING AND DATA CUBE**

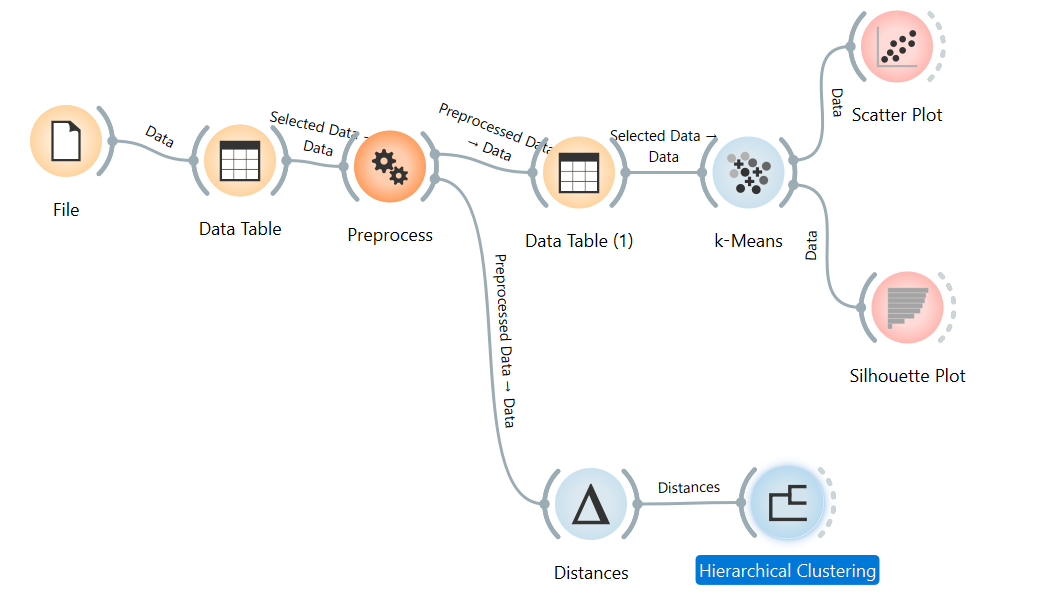


CODE AND CLICK RUN IN PYTHON SCRIPT TO SEE 3D BAR PLOT:

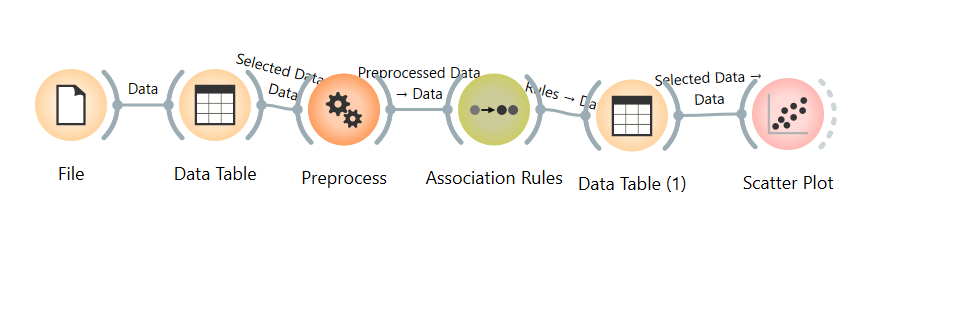
**EXP 2 DATA CLEANING**



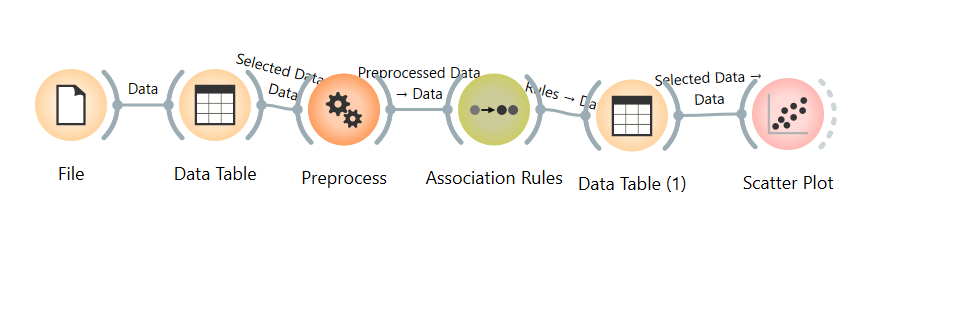
**EXP 3**



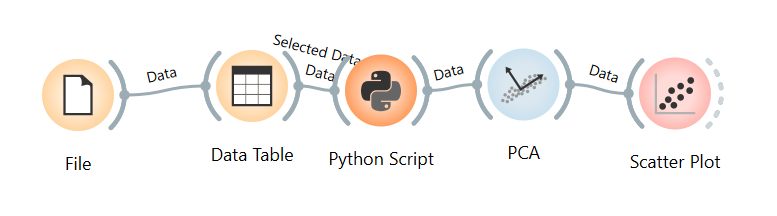
**EXP4:**



**Exp 5**



**Exp 6:**



import numpy as np

from Orange.data import Table, Domain, ContinuousVariable

def haar(x):

r=[]

for v in x:

c=[]

while len(v)>1:

a,d=(v[::2]+v[1::2])/2,(v[::2]-v[1::2])/2

c+=d.tolist();v=a

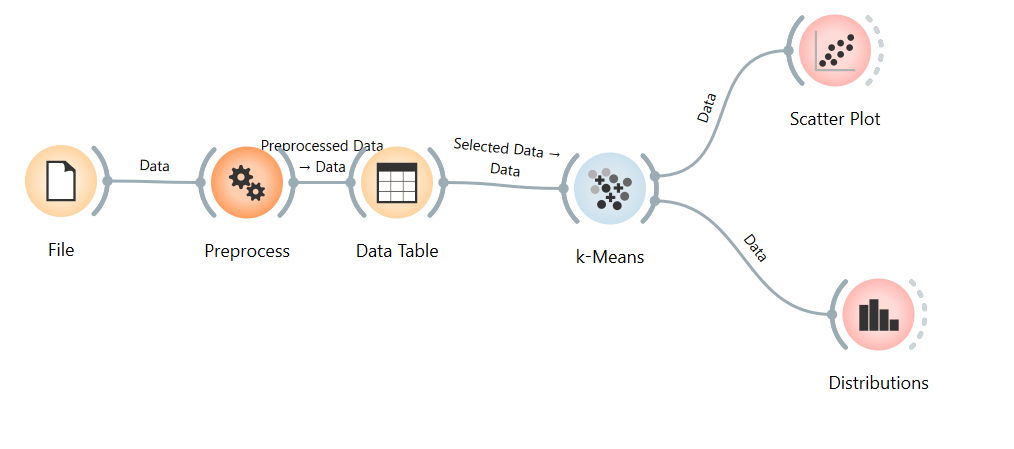
r.append(c+v.tolist())

return np.array(r)

out\_data=Table(Domain([ContinuousVariable(f"H{i}") for i in range(len(in\_data.domain))]), haar(in\_data.X))

**EXP 7:**

**EXP 8:**



**EXP 9:**

**CODE 1:**

import numpy as np

import matplotlib.pyplot as plt

from sklearn.cluster import AgglomerativeClustering

X = np.array(in\_data.X)

# Perform Agglomerative Clustering

model = AgglomerativeClustering(n\_clusters=3)

labels = model.fit\_predict(X)

# Scatter plot of clusters

plt.figure()

plt.scatter(X[:,0], X[:,1], c=labels, cmap='viridis', s=50)

plt.title("Agglomerative Clustering (Bottom-Up)")

plt.xlabel("Feature 1")

plt.ylabel("Feature 2")

plt.show()

**CODE 2:**

import numpy as np

import matplotlib.pyplot as plt

from sklearn.cluster import KMeans

X = np.array(in\_data.X)

# Simulate Divisive Clustering (recursive top-down splitting)

labels = np.zeros(len(X))

k = 2

for i in range(k):

km = KMeans(n\_clusters=2, random\_state=i).fit(X[labels==i])

sub\_labels = km.labels\_ + 2\*i

labels[labels==i] = sub\_labels

# Scatter plot

plt.figure()

plt.scatter(X[:,0], X[:,1], c=labels, cmap='plasma', s=50)

plt.title("Divisive Clustering (Top-Down)")

plt.xlabel("Feature 1")

plt.ylabel("Feature 2")

plt.show()

**EXP 10:**