

# DATAMINING PROJECT [2024]



CLASSIFICATION

CLUSTERING

ASSOCIATE RULE

# CLASSIFICATION ALGORITHMS



DỰ ĐOÁN NGÀY MÀU  
CÓ MƯA HAY KHÔNG  
TẠI KHU VỰC ALBURY - ÚC





# ALBURY OVERVIEW

Nằm trên bờ phía bắc của sông Murray, cách Sydney khoảng 554 km về phía nam và cách Melbourne khoảng 326 km về phía đông bắc.

Thành phố này có một địa hình kết hợp giữa đồng bằng và đồi núi, tạo điều kiện thuận lợi cho nhiều hoạt động kinh tế.

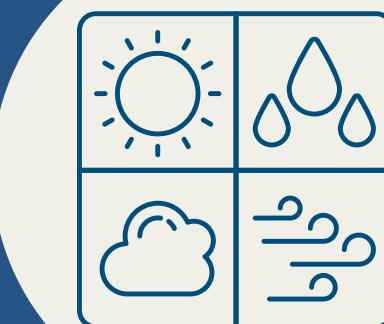
Nông nghiệp và chăn nuôi



Hoạt động kinh doanh và du lịch



Ứng phó với biến đổi khí hậu



# WHY?



DATASET: Albury Climate - 2017  
Size: 14 trường và 3040 bản ghi

# ALBURY DATASET

TÊN TRƯỜNG	Ý NGHĨA
MinTemp , MaxTemp	Nhiệt độ tối thiểu và tối đa trong ngày
Rainfall	Lượng mưa trong ngày (đơn vị mm)
Humidity9am,Humidity3pm	Độ ẩm vào lúc 9 giờ sáng và 3 giờ chiều
Pressure9am,Pressure3pm	Áp suất khí quyển vào lúc 9 giờ sáng và 3 giờ chiều
Cloud9am,Cloud3pm	Mức độ mây vào lúc 9 giờ sáng và 3 giờ chiều
Temp9am, Temp3pm	Nhiệt độ vào lúc 9 giờ sáng và 3 giờ chiều
RainToday, RainTomorrow	Thông tin mưa trong ngày/ngày tiếp theo (Yes/No)

# OUR STEPS

Date	MinTemp	MaxTemp	Rainfall	Humidity9a	Humidity3p	Pressure9a	Pressure3p	Cloud9am	Cloud3pm	Temp9am	Temp3pm	RainToday	RainTomorrow
01/12/2008	13.4	22.9	0.6	71	22	1007.7	1007.1	8	NA	16.9	21.8	No	No
02/12/2008	7.4	25.1	0	44	25	1010.6	1007.8	NA	NA	17.2	24.3	No	No
03/12/2008	12.9	25.7	0	38	30	1007.6	1008.7	NA	2	21	23.2	No	No
04/12/2008	9.2	28	0	45	16	1017.6	1012.8	NA	NA	18.1	26.5	No	No
05/12/2008	17.5	32.3	1	82	33	1010.8	1006	7	8	17.8	29.7	No	No
06/12/2008	14.6	29.7	0.2	55	23	1009.2	1005.4	NA	NA	20.6	28.9	No	No
07/12/2008	14.3	25	0	49	19	1009.6	1008.2	1	NA	18.1	24.6	No	No
08/12/2008	7.7	26.7	0	48	19	1013.4	1010.1	NA	NA	16.3	25.5	No	No
09/12/2008	9.7	31.9	0	42	9	1008.9	1003.6	NA	NA	18.3	30.2	No	Yes
10/12/2008	13.1	30.1	1.4	58	27	1007	1005.7	NA	NA	20.1	28.2	Yes	No
11/12/2008	13.4	30.4	0	48	22	1011.8	1008.7	NA	NA	20.4	28.8	No	Yes
12/12/2008	15.9	21.7	2.2	89	91	1010.5	1004.2	8	8	15.9	17	Yes	Yes
13/12/2008	15.9	18.6	15.6	76	93	994.3	993	8	8	17.4	15.8	Yes	Yes
14/12/2008	12.6	21	3.6	65	43	1001.2	1001.8	NA	7	15.8	19.8	Yes	No
15/12/2008	8.4	24.6	0	57	32	1009.7	1008.7	NA	NA	15.9	23.5	No	NA
16/12/2008	9.8	27.7	NA	50	28	1013.4	1010.3	0	NA	17.3	26.2	NA	No
17/12/2008	14.1	20.9	0	69	82	1012.2	1010.4	8	1	17.2	18.1	No	Yes
18/12/2008	13.5	22.9	16.8	80	65	1005.8	1002.2	8	1	18	21.5	Yes	Yes
19/12/2008	11.2	22.5	10.6	47	32	1009.4	1009.7	NA	2	15.5	21	Yes	No
20/12/2008	9.8	25.6	0	45	26	1019.2	1017.1	NA	NA	15.8	23.2	No	No
21/12/2008	11.5	29.3	0	56	28	1019.3	1014.8	NA	NA	19.1	27.3	No	No
22/12/2008	17.1	33	0	38	28	1013.6	1008.1	NA	1	24.5	31.6	No	No
23/12/2008	20.5	31.8	0	54	24	1007.8	1005.7	NA	NA	23.8	30.8	No	No

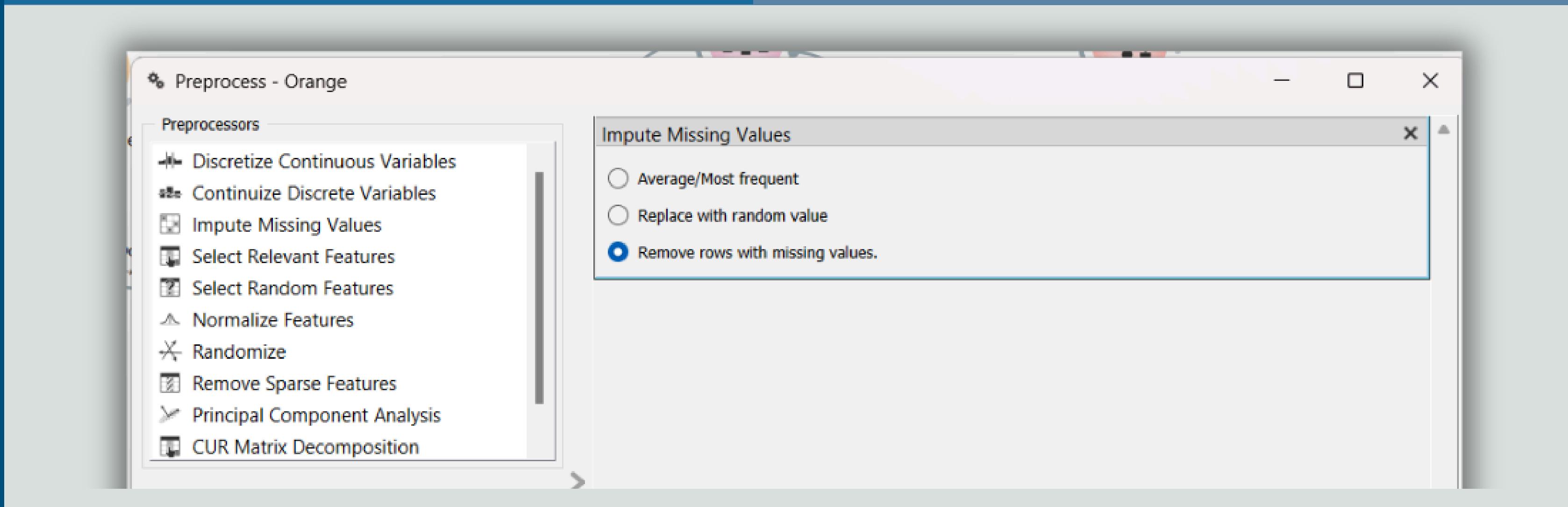


# STEP 1:

# TIỀN XỬ LÝ DỮ LIỆU

## 1. Xử lý giá trị thiếu (Missing Values)

xóa các dòng hoặc cột chứa  
giá trị thiếu

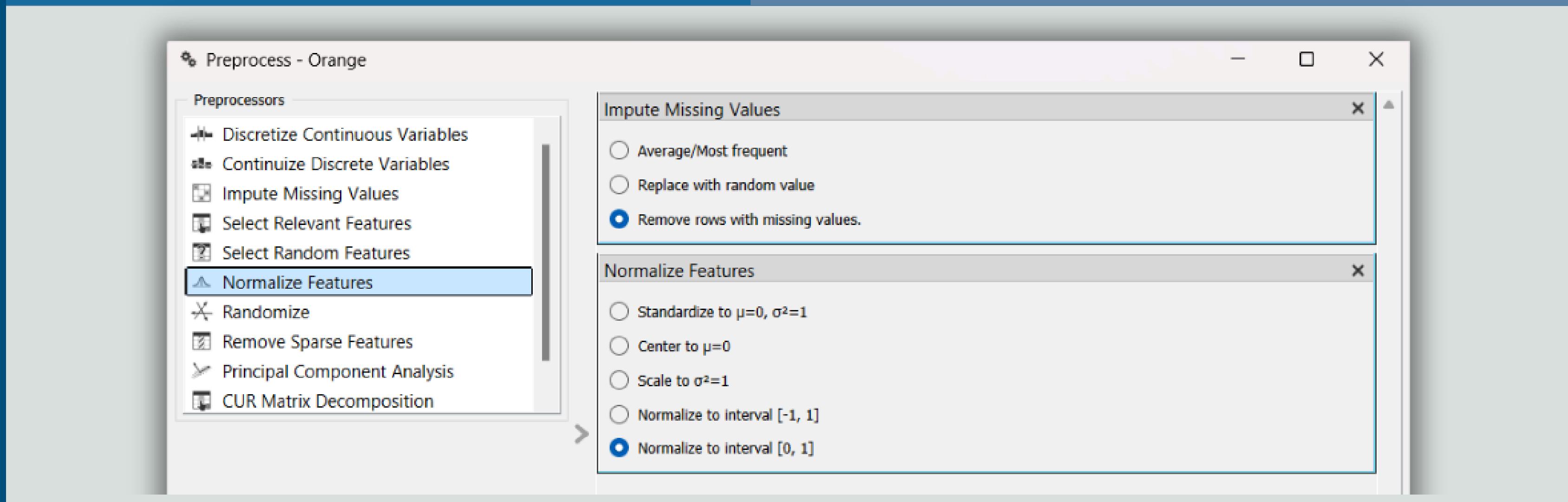


# STEP 1:

# TIỀN XỬ LÝ DỮ LIỆU

## 2. Chuẩn hoá dữ liệu:

nhiệt độ có thể có giá trị rất lớn  
độ ẩm có giá trị nhỏ hơn

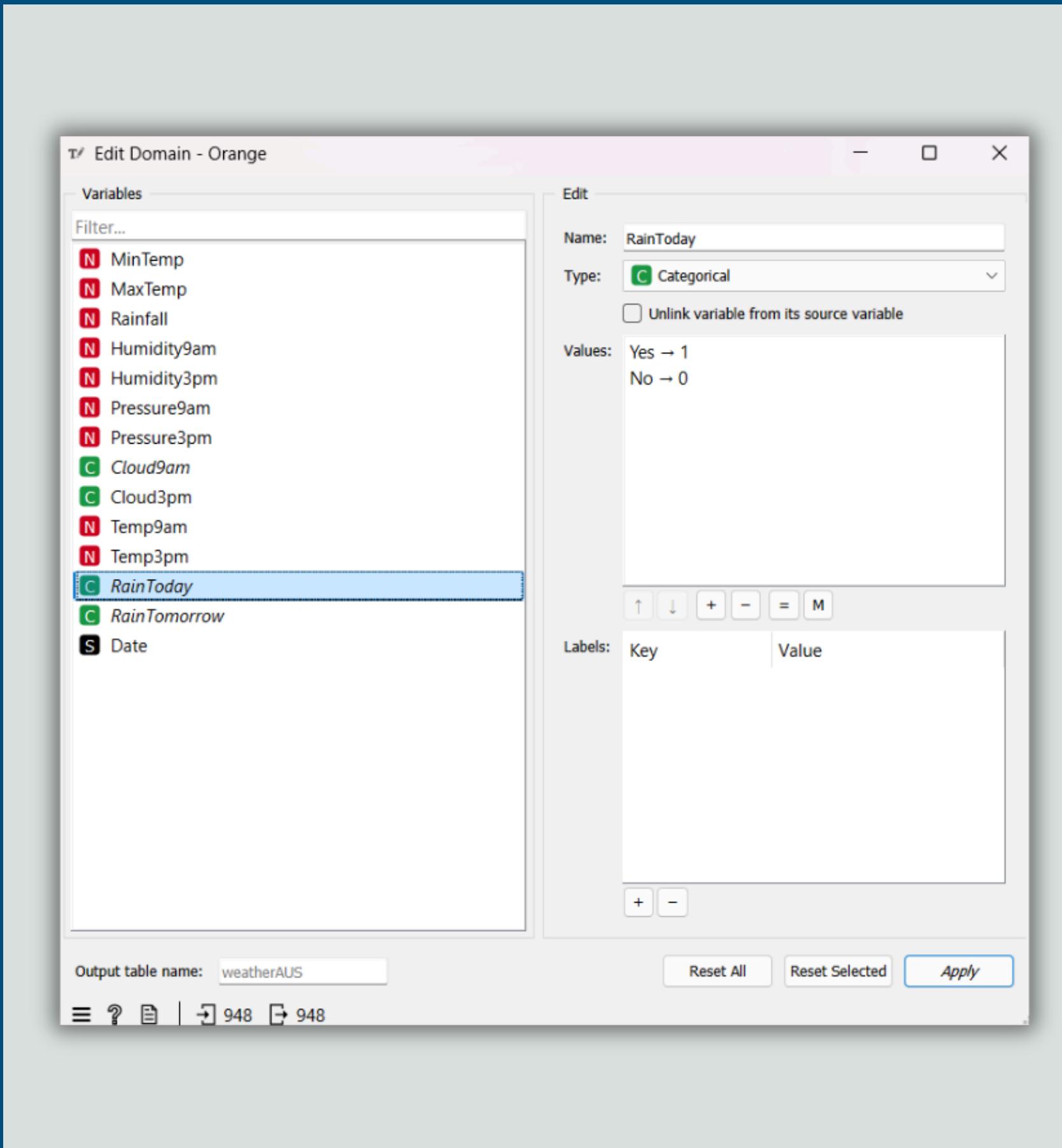


# STEP 1:

# TIỀN XỬ LÝ DỮ LIỆU

## 3. Chuyển đổi dữ liệu phân loại:

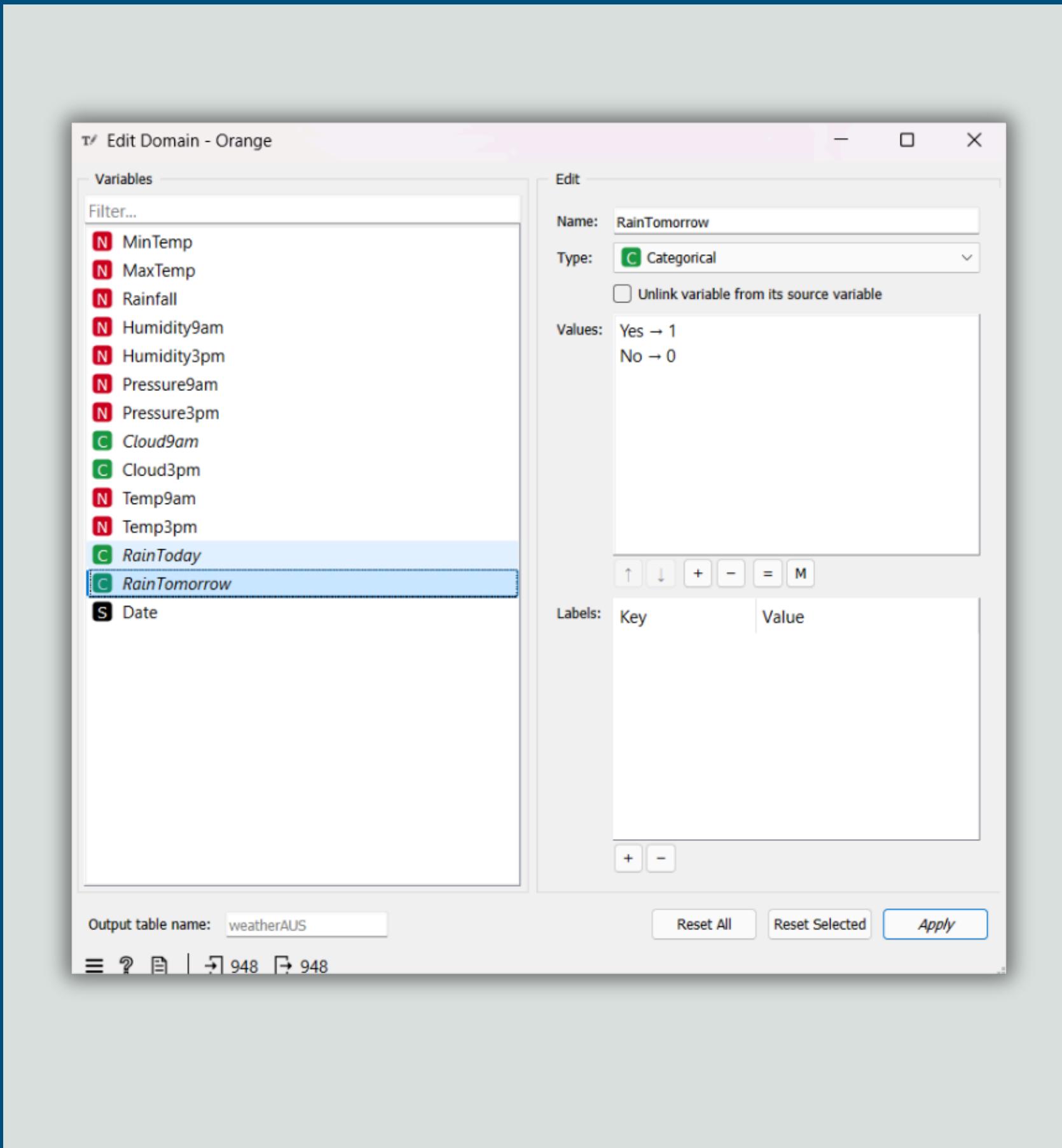
các cột , "RainToday" ,  
"RainTomorrow" với các giá trị  
"Yes", "No"



# STEP 1: TIỀN XỬ LÝ DỮ LIỆU

## 3. Chuyển đổi dữ liệu phân loại:

các cột , "RainToday" ,  
"RainTomorrow" với các giá trị  
"Yes", "No"



# BEFORE

**Data Table - Orange**

**Info**  
3040 instances  
12 features (9.6 % missing data)  
Target with 2 values (1.0 % missing data)  
1 meta attribute

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

**Selection**  
 Select full rows

Restore Original Order

3040 | 3040.1 3040

	RainTomorrow	Date	MinTemp	MaxTemp	Rainfall	Humidity9am	Humidity3pm	Pressure9am	F
1	No	01/12/2008	13.4	22.9	0.6	71	22	1007.7	
2	No	02/12/2008	7.4	25.1	0.0	44	25	1010.6	
3	No	03/12/2008	12.9	25.7	0.0	38	30	1007.6	
4	No	04/12/2008	9.2	28.0	0.0	45	16	1017.6	
5	No	05/12/2008	17.5	32.3	1.0	82	33	1010.8	
6	No	06/12/2008	14.6	29.7	0.2	55	23	1009.2	
7	No	07/12/2008	14.3	25.0	0.0	49	19	1009.6	
8	No	08/12/2008	7.7	26.7	0.0	48	19	1013.4	
9	Yes	09/12/2008	9.7	31.9	0.0	42	9	1008.9	
10	No	10/12/2008	13.1	30.1	1.4	58	27	1007.0	
11	Yes	11/12/2008	13.4	30.4	0.0	48	22	1011.8	
12	Yes	12/12/2008	15.9	21.7	2.2	89	91	1010.5	
13	Yes	13/12/2008	15.9	18.6	15.6	76	93	994.3	
14	No	14/12/2008	12.6	21.0	3.6	65	43	1001.2	
15	?	15/12/2008	8.4	24.6	0.0	57	32	1009.7	
16	No	16/12/2008	9.8	27.7	?	50	28	1013.4	
17	Yes	17/12/2008	14.1	20.9	0.0	69	82	1012.2	
18	Yes	18/12/2008	13.5	22.9	16.8	80	65	1005.8	
19	No	19/12/2008	11.2	22.5	10.6	47	32	1009.4	
20	No	20/12/2008	9.8	25.6	0.0	45	26	1019.2	
21	No	21/12/2008	11.5	29.3	0.0	56	28	1019.3	
22	No	22/12/2008	17.1	33.0	0.0	38	28	1013.6	
23	No	23/12/2008	20.5	31.8	0.0	54	24	1007.8	
24	No	24/12/2008	15.3	30.9	0.0	55	23	1011.0	
25	No	25/12/2008	12.6	32.4	0.0	49	17	1012.9	
26	No	26/12/2008	16.2	33.9	0.0	45	19	1010.9	
27	No	27/12/2008	16.9	33.0	0.0	41	28	1006.8	
28	No	28/12/2008	20.1	32.7	0.0	56	15	1005.2	
29	Yes	29/12/2008	19.7	27.2	0.0	49	22	1004.8	
30	No	30/12/2008	12.5	24.2	1.2	78	70	1005.6	
31	No	31/12/2008	12.0	24.4	0.8	48	28	1006.1	
32	No	01/01/2009	11.3	26.5	0.0	46	26	1004.5	
33	No	02/01/2009	9.6	23.9	0.0	44	22	1014.4	
34	No	03/01/2009	10.5	28.8	0.0	43	22	1018.7	
35	No	04/01/2009	12.3	34.6	0.0	41	12	1015.1	
36	No	05/01/2009	12.9	35.8	0.0	41	9	1012.6	

# AFTER

**Data Table (1) - Orange**

**Info**

948 instances  
12 features  
Target with 2 values (0.2 % missing data)  
1 meta attribute

**Variables**

Show variable labels (if present)

Visualize numeric values

Color by instance classes

**Selection**

Select full rows

iTomorrow	Date	MinTemp	MaxTemp	Rainfall	Humidity9am	Humidity3pm	Pressure9am	Pressure3
1	05/12/2008	0.67368	0.74780	0.009597	0.7662	0.2386	0.45455	0.
2	12/12/2008	0.61754	0.43695	0.021113	0.8571	0.8977	0.44805	0.
3	13/12/2008	0.61754	0.34604	0.149712	0.6883	0.9205	0.09740	0.
4	17/12/2008	0.55439	0.41349	0.0000	0.5974	0.7955	0.48485	0.
5	18/12/2008	0.53333	0.47214	0.161228	0.7403	0.6023	0.34632	0.
6	30/12/2008	0.49825	0.51026	0.011516	0.7143	0.6591	0.34199	0.
7	22/01/2009	0.91579	0.79765	0.005758	0.4805	0.3068	0.33550	0.
8	09/02/2009	0.70526	0.71554	0.003839	0.5584	0.2159	0.42208	0.
9	12/02/2009	0.62456	0.43402	0.0000	0.4545	0.6477	0.58874	0.
10	24/02/2009	0.51228	0.66862	0.0000	0.4026	0.0227	0.52381	0.
11	25/02/2009	0.37193	0.73607	0.0000	0.3377	0.0455	0.58009	0.
12	22/03/2009	0.57193	0.68035	0.0000	0.4286	0.2045	0.54113	0.
13	25/03/2009	0.600	0.45455	0.003839	0.5974	0.75	0.59740	0.
14	03/04/2009	0.70526	0.62463	0.082534	0.8961	0.4205	0.62771	0.
15	10/04/2009	0.46667	0.58358	0.0000	0.4675	0.3409	0.68615	0.
16	11/04/2009	0.51579	0.53079	0.080614	0.6753	0.3977	0.74892	0.
17	17/04/2009	0.29123	0.43402	0.0000	0.4675	0.3068	0.62121	0.
18	24/04/2009	0.43158	0.27566	0.0000	0.6623	0.8409	0.41342	0.
19	25/04/2009	0.51228	0.26393	0.191939	0.7792	0.8750	0.32468	0.
20	26/04/2009	0.36140	0.17889	0.201536	0.7013	0.5682	0.30736	0.
21	27/04/2009	0.21754	0.13783	0.030710	0.7662	0.7614	0.51948	0.
22	28/04/2009	0.32632	0.22581	0.046065	0.7792	0.4091	0.61039	0.
23	02/05/2009	0.31228	0.36364	0.0000	0.7532	0.4205	0.80303	0.
24	14/05/2009	0.30175	0.27859	0.0000	0.7013	0.5341	0.55195	0.
25	15/05/2009	0.42105	0.28739	0.0000	0.7662	0.5795	0.54545	0.
26	16/05/2009	0.49474	0.28152	0.017274	0.7662	0.4773	0.47619	0.
27	20/05/2009	0.23860	0.34604	0.0000	0.6753	0.5114	0.71429	0.
28	25/05/2009	0.38596	0.39589	0.0000	0.7013	0.4318	0.75325	0.
29	26/05/2009	0.46667	0.33138	0.040307	0.9610	0.7045	0.68398	0.
30	01/06/2009	0.34035	0.21994	0.011516	0.8442	0.7159	0.91126	0.
31	02/06/2009	0.35439	0.19355	0.013436	0.7143	0.7159	0.85498	0.
32	03/06/2009	0.43158	0.21994	0.046065	0.8052	0.9659	0.72944	0.
33	04/06/2009	0.37193	0.31085	0.076775	0.9870	0.5227	0.67532	0.
34	06/06/2009	0.11930	0.10850	0.001919	0.9870	0.8977	0.54978	0.
35	07/06/2009	0.22456	0.14076	0.138196	0.9091	0.7727	0.38312	0.
36	08/06/2009	0.37544	0.15249	0.044146	0.8442	0.7614	0.42857	0.

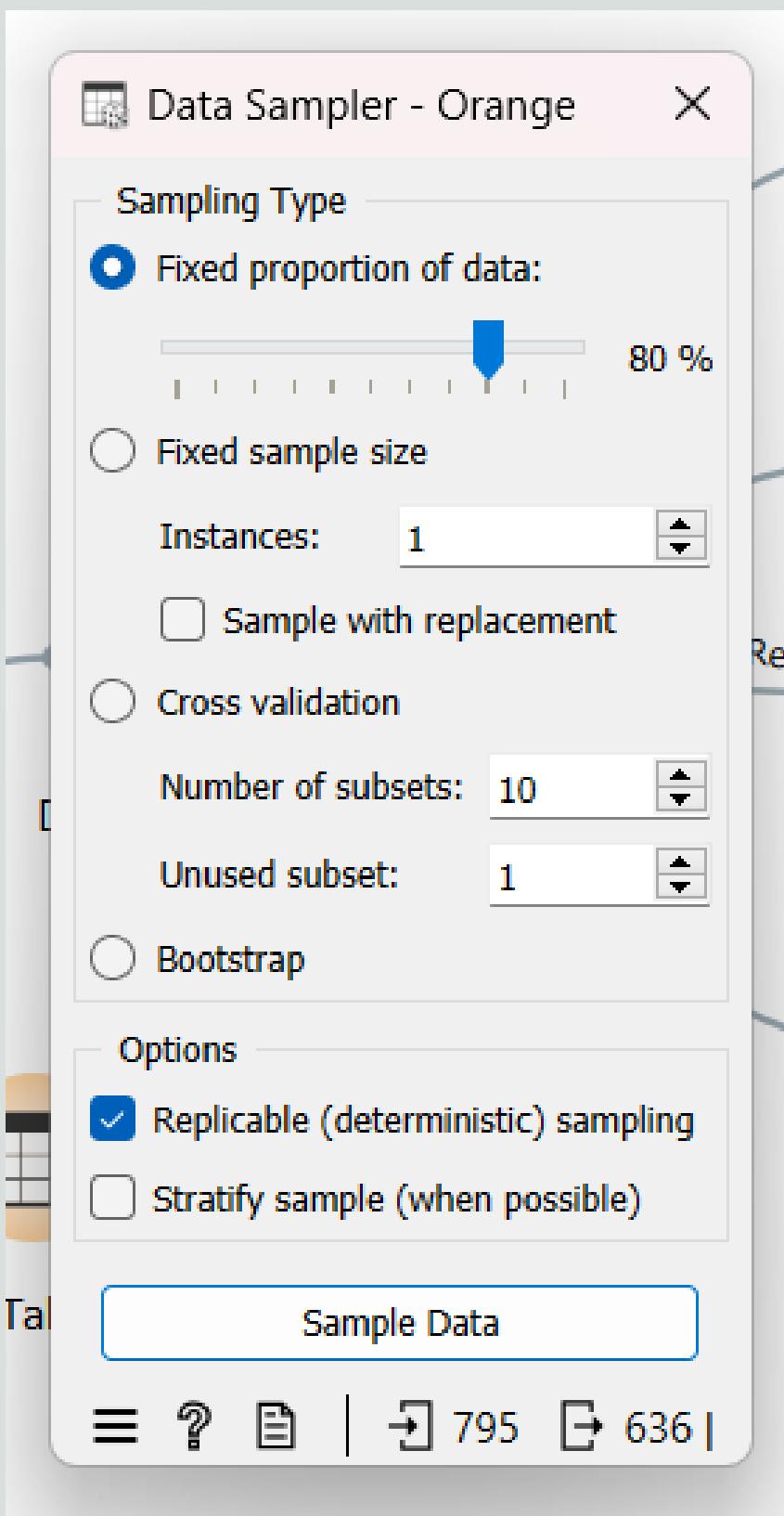
Restore Original Order

☰ ? ⌂ | 948 ⌂ 948 | 948

## STEP 2:

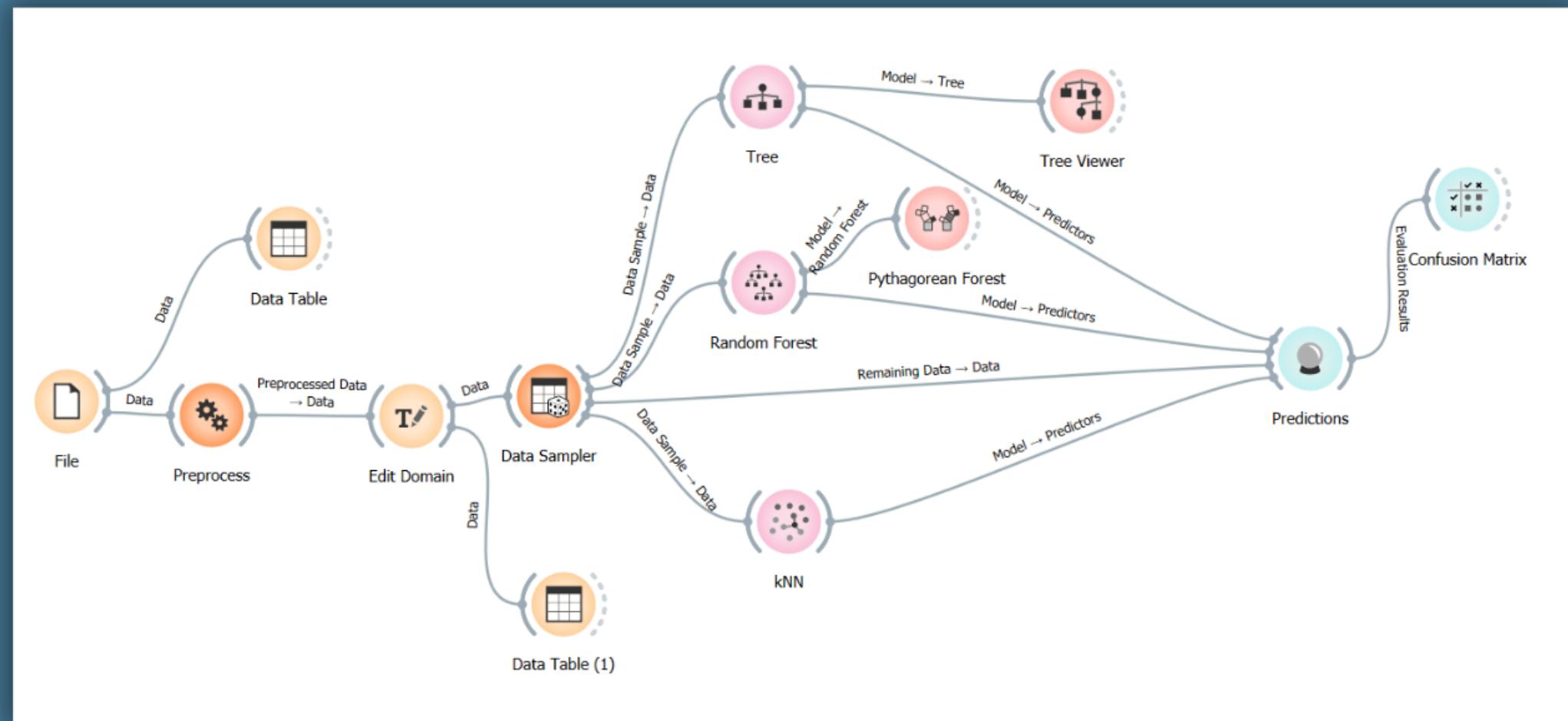
# PHÂN CHIA DỮ LIỆU

Tập training : 80%  
Tập test : 20%



1. ID3
2. Random Forest
3. KNN

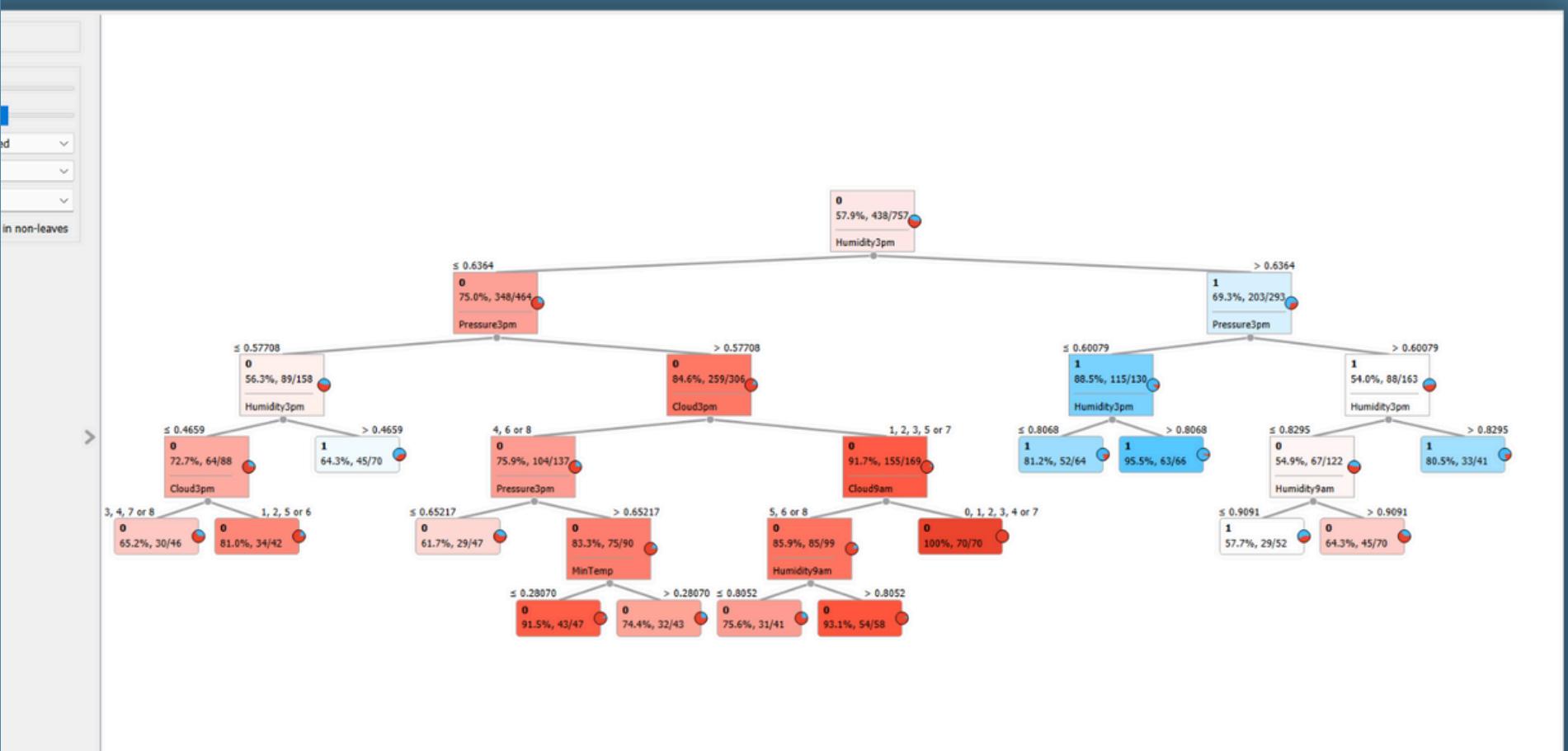
## STEP 3: ÁP DỤNG THUẬT TOÁN



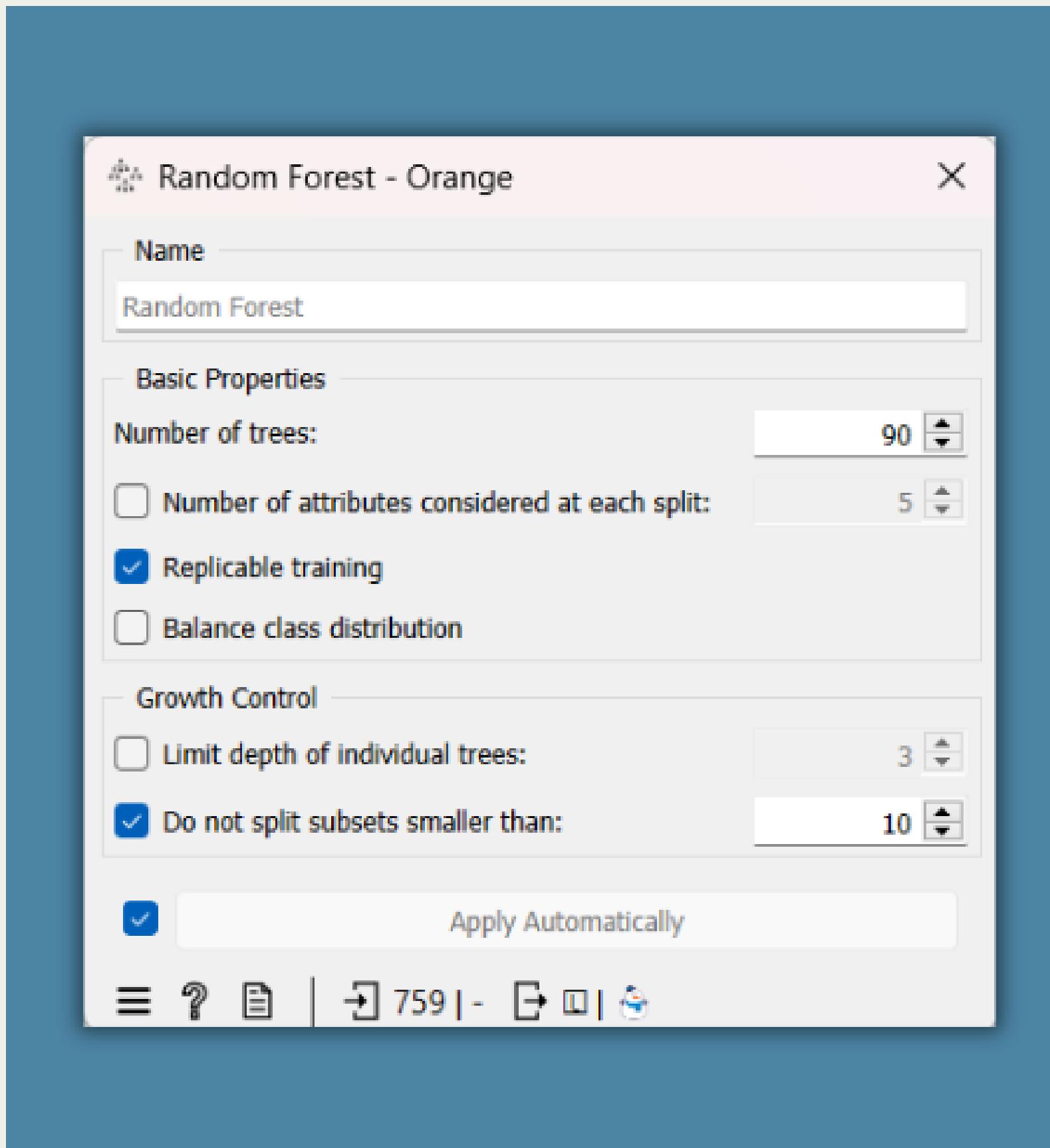
# STEP 3: ÁP DỤNG THUẬT TOÁN

## 1. ID3

- Chọn thuộc tính tốt nhất
- Chia nhỏ dữ liệu
- Lặp lại
- Hoàn thành cây quyết định



# STEP 3: ÁP DỤNG THUẬT TOÁN



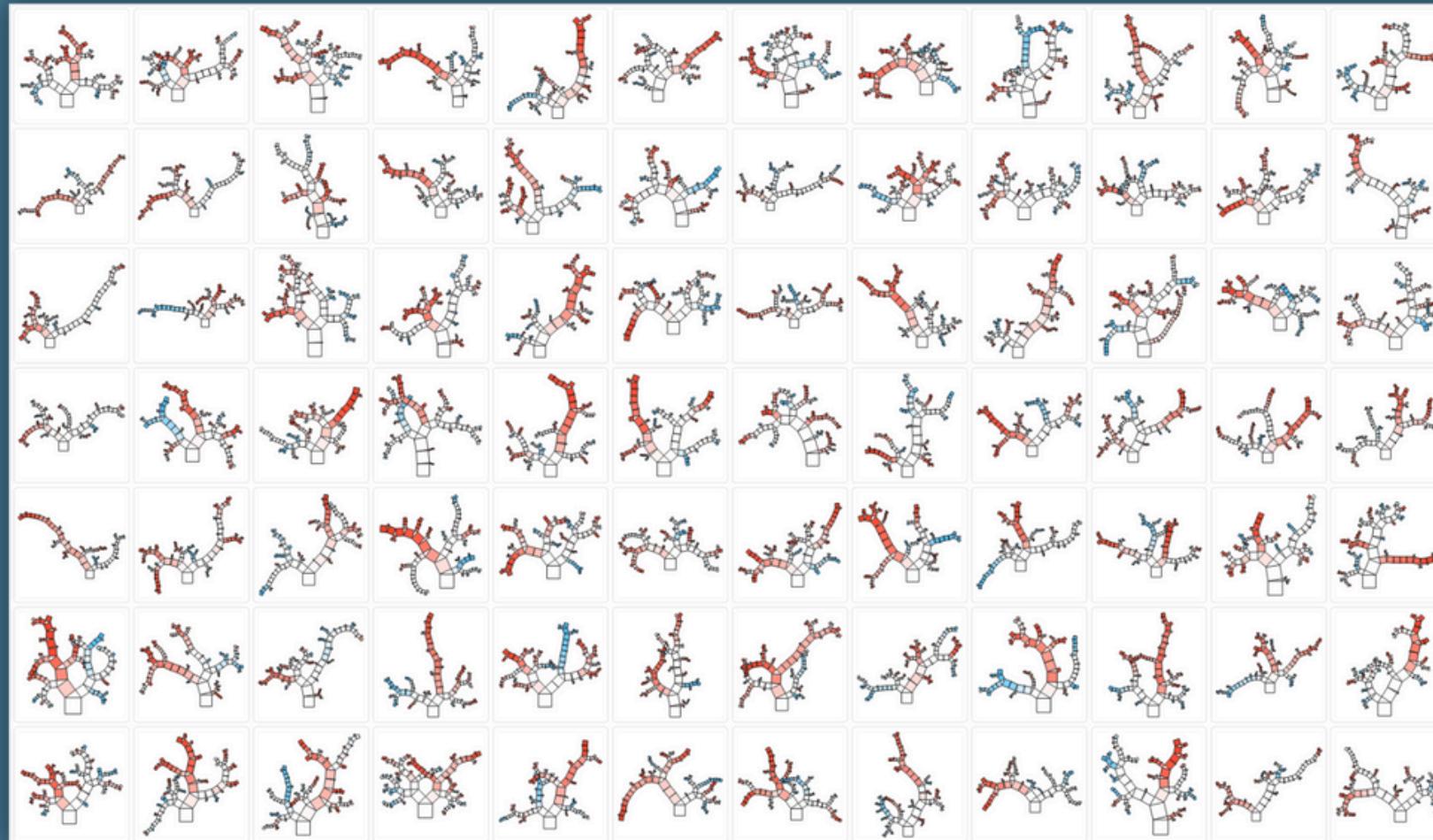
## 2. Random Forest

- Tạo ra nhiều cây quyết định (decision trees) độc lập
- Kết hợp kết quả của các cây này để đưa ra dự đoán cuối cùng

## 2. Random Forest

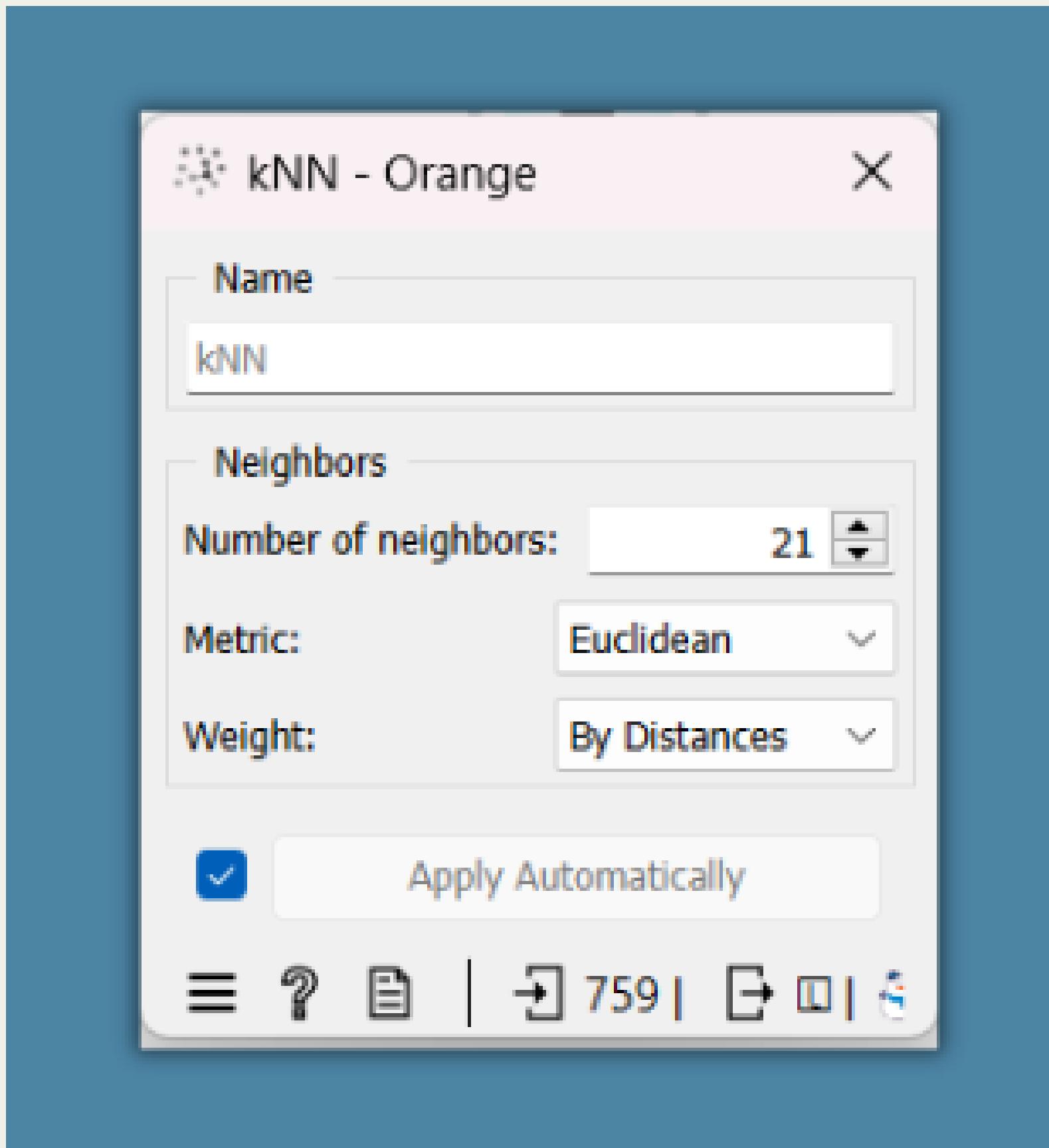
- Số lượng cây trong rừng là : 90

# STEP 3: ÁP DỤNG THUẬT TOÁN



## STEP 3:

# ÁP DỤNG THUẬT TOÁN



### 3. KNN

- Tính toán khoảng cách
- Xác định K điểm gần nhất
- Tiến hành dự đoán

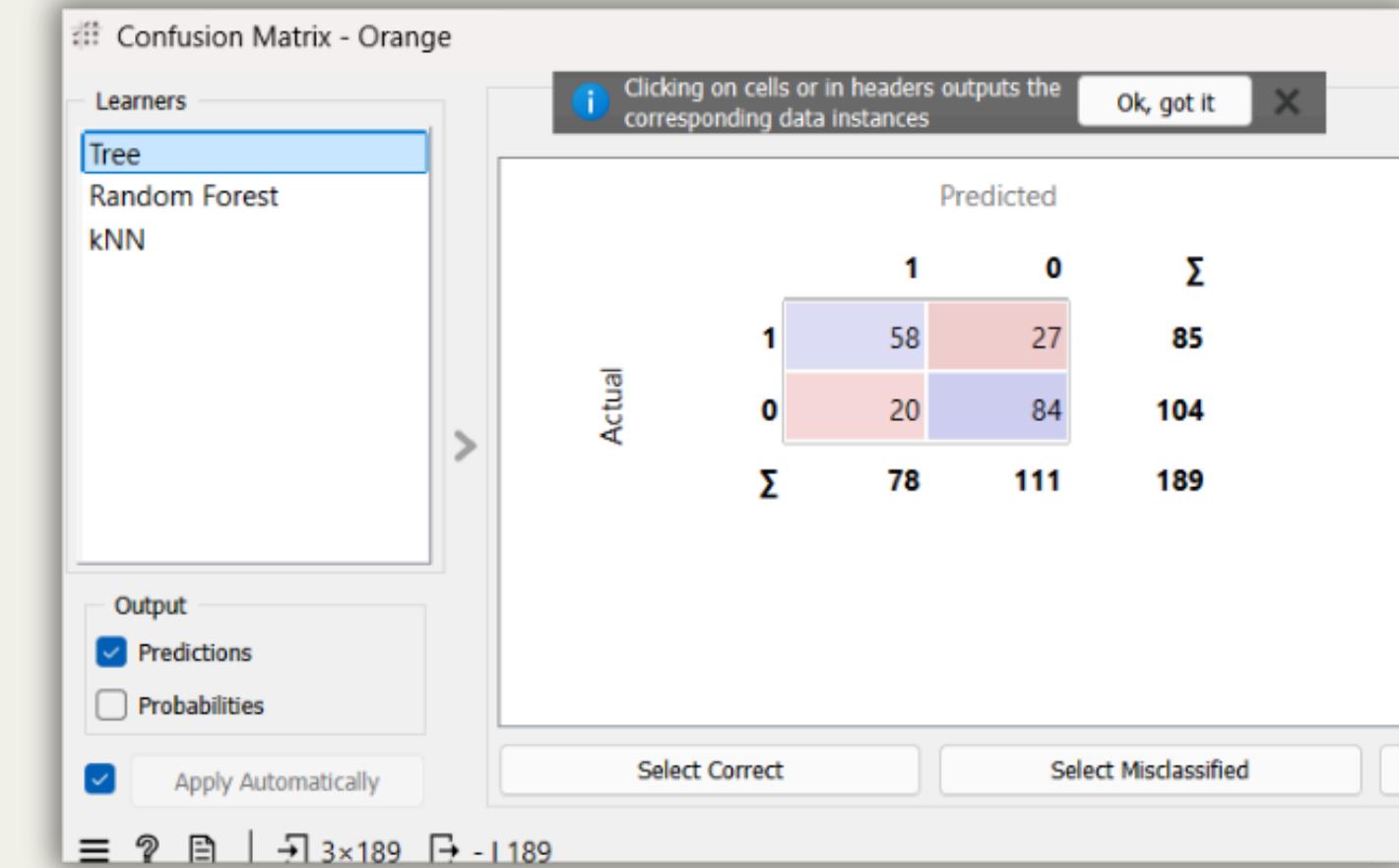


# MODEL EVALUATION

Model	AUC	CA	F1	Prec	Recall	MCC
Tree	0.775	0.751	0.750	0.751	0.751	0.495
Random Forest	0.803	0.757	0.756	0.756	0.757	0.506
kNN	0.770	0.725	0.722	0.725	0.725	0.441

# ID3

Model	AUC	CA	F1	Prec	Recall
Tree	0.775	0.751	0.750	0.751	0.751



## [ STRENGTH ]

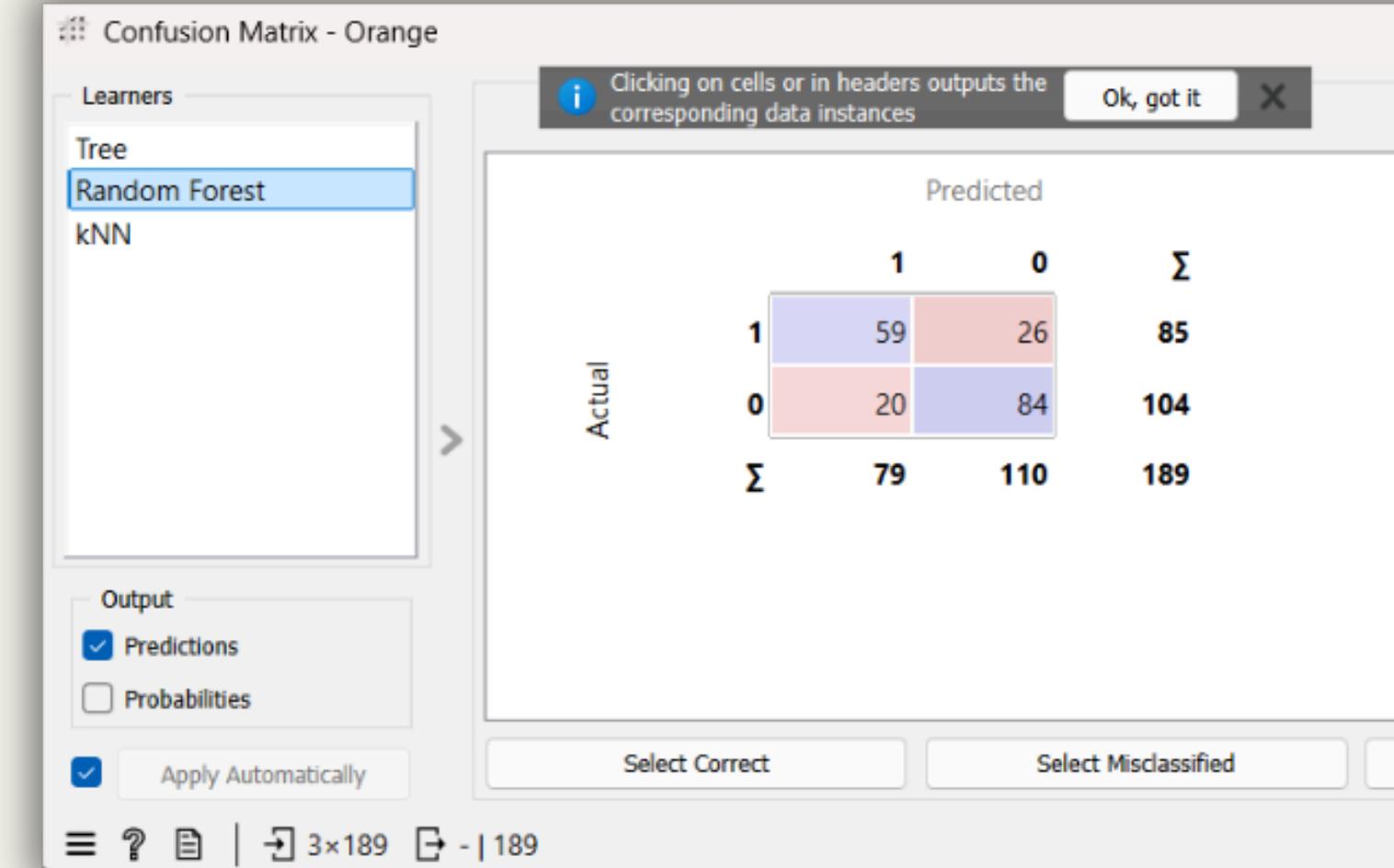
- Độ chính xác và khả năng phân biệt 2 lớp khá ổn

## [ WEAKNESS ]

- Dự đoán nhầm không mưa.

# RANDOM FOREST

Model	AUC	CA	F1	Prec	Recall
Random Forest	0.803	0.757	0.756	0.756	0.757



## [ STRENGTH ]

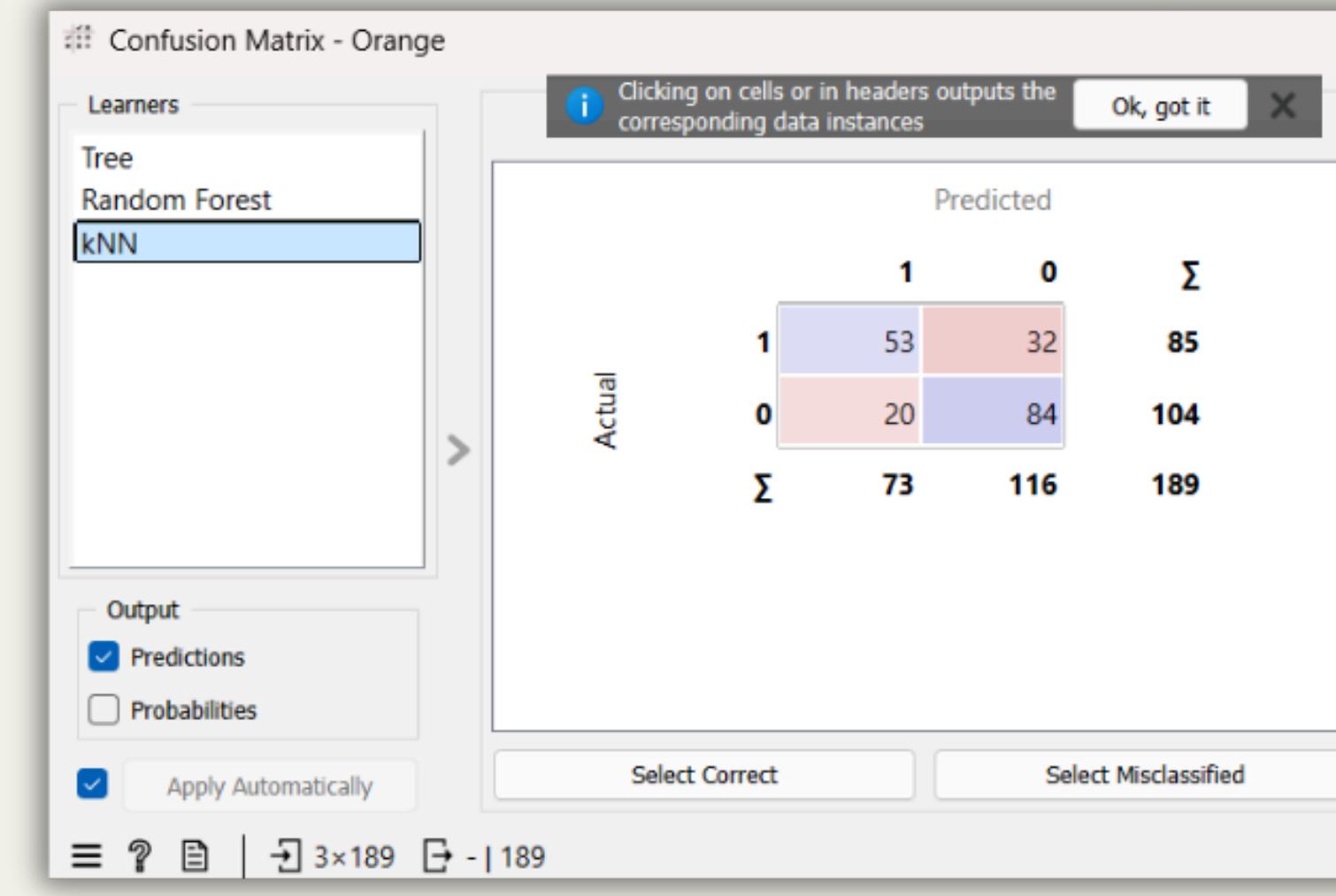
- Chính xác hơn ID3 ( $0.757 > 0.751$ )
- Khả năng phân biệt hai lớp tốt.

## [ WEAKNESS ]

- Chênh lệch không quá lớn so với ID3

KNN

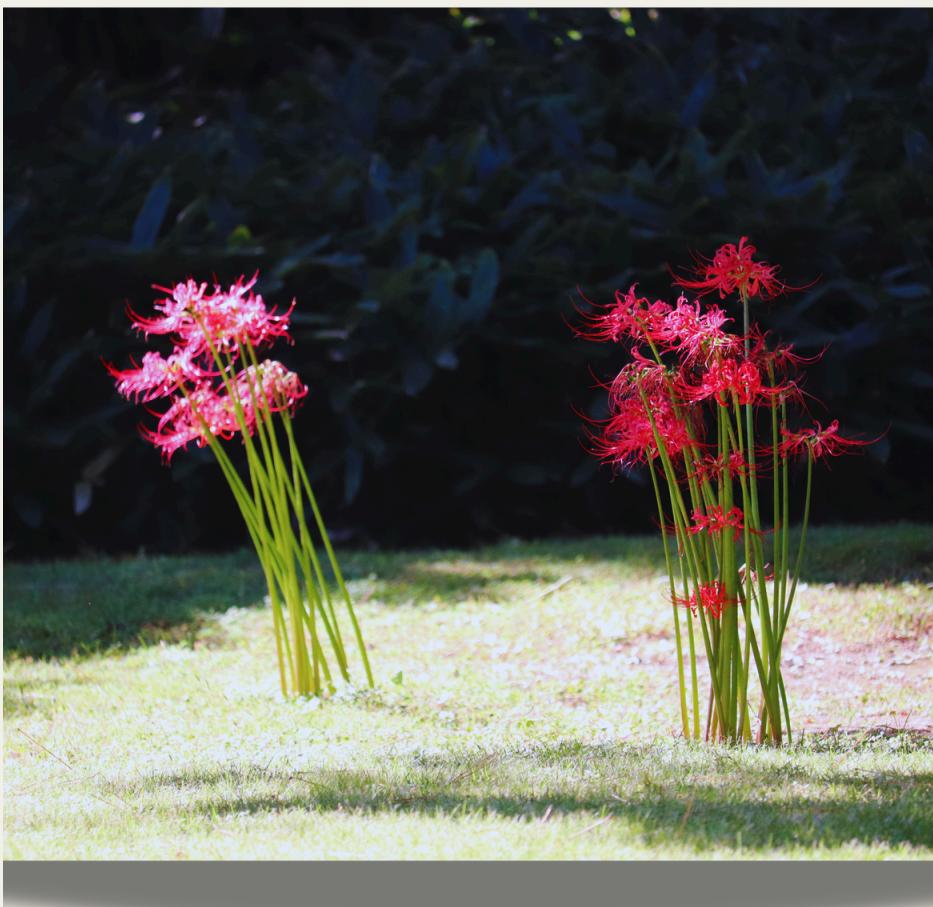
Model	AUC	CA	F1	Prec	Recall
KNN	0.770	0.725	0.722	0.725	0.725



## [ OVERVIEW ]

- Độ chính xác thấp hơn so với ID3 và Random Forest.
- Khả năng phân biệt hai lớp kém hơn ID3 và Random Forest.
- Hiệu suất tổng quan kém hơn so với ID3 và Random Forest.

# CLUSTERING



[2 ALGORITHMS]

K-MEANS

HIERARCHICAL

# BANKLOANS DATASET

DATASET: bankloans - 2022  
Size: 4 trường và 105 bản ghi

TÊN TRƯỜNG	Ý NGHĨA
Tuoi	Tuổi của chủ thẻ
Kinh Nghiem Lam Viec	Kinh nghiệm làm việc (năm)
Thu Nhap	Thu nhập chủ thẻ
No Nan	Nợ nần trong năm

# OUR STEPS

Tuoi	Kinh Nghiem Lam Viec	Thu Nhap	No Nan
57	27	176	25
31	14	41	20
57	26	109	29
54	21	120	2.9
45	11	28	17.3
65	28	130	16
41	14	67	30.6
27	0	49	26
40	16	19	24.4
36	14	25	19.7
27	3	16	1.7
25	2	23	5.2
52	29	64	10
37	14	31	24.1
46	25.5	100	9.1

**Step 1**

tìm số cụm  
tối ưu

**Step 2**

tiền xử lý  
dữ liệu

**Step 3**

thuật toán



# STEP 1: TÌM K TỐI UU

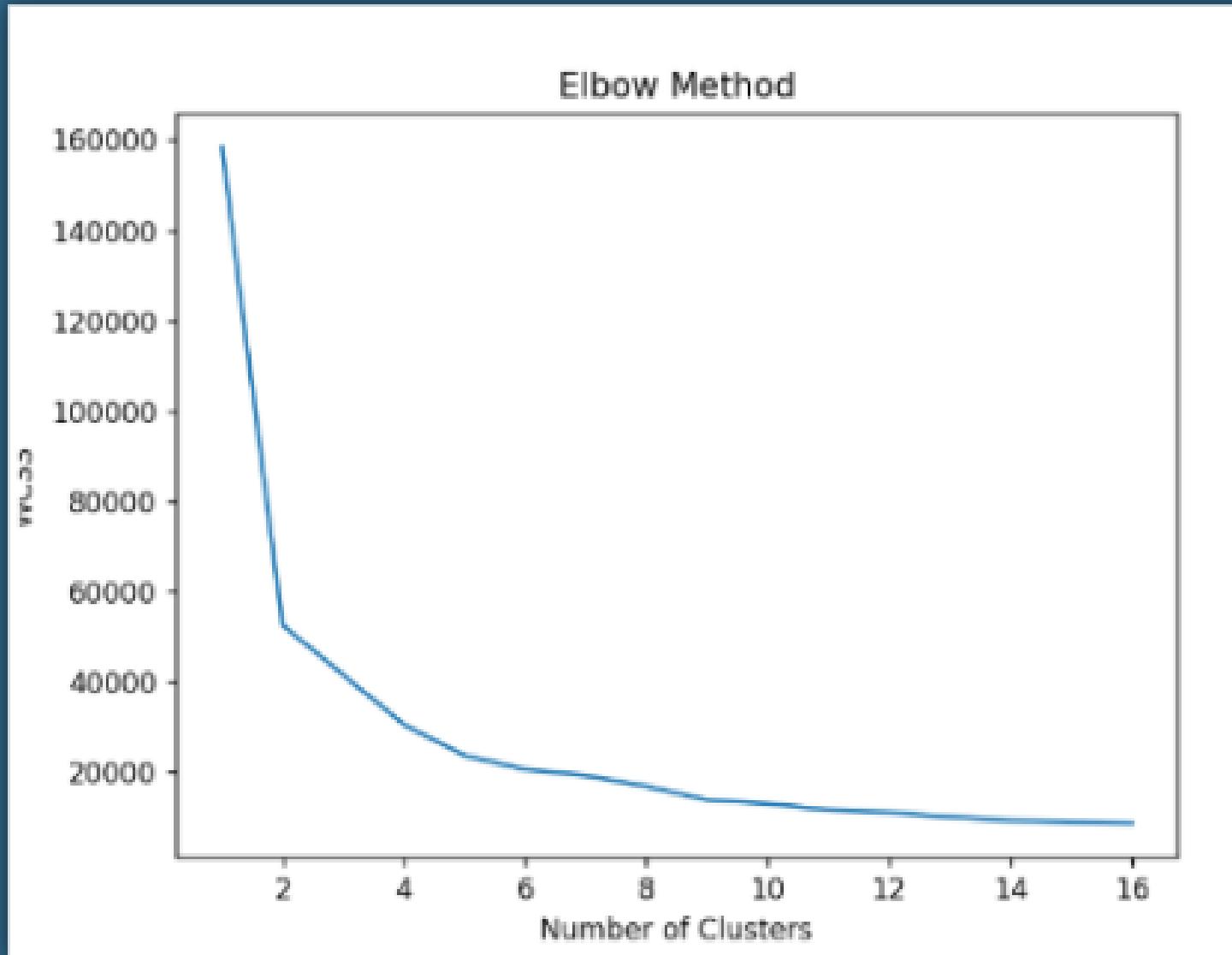
```
1 import matplotlib.pyplot as plt
2 import pandas as pd
3 from sklearn.cluster import KMeans
4
5 #Đọc file CSV
6 df = pd.read_csv('bankloans.csv')
7 print(df)
8
9 #Chọn những trường mà mình dùng
10 selected_features = ['Tuoi', 'Kinh Nghiem Lam Viec', 'Thu Nhap', 'No Nan']
11 X = df[selected_features]
12
13 #Tìm cụm tối ưu bằng Elbow
14 wcss = []
15 for k in range(1, 17):
16     Kmeans = KMeans(n_clusters=k, random_state=42)
17     Kmeans.fit(X)
18     wcss.append(Kmeans.inertia_)
19
20 #Vẽ đồ thị
21
22 plt.plot(range(1, 17), wcss)
23 plt.title('Elbow Method')
24 plt.xlabel('Number of Clusters')
25 plt.ylabel('WCSS')
26 plt.show()
27
```

## Elbow Method:

- Trực quan hóa và xác định giá trị  $k$  tốt nhất.
- Phân tích tổng độ biến thiên trong cụm (WCSS) ở các giá trị  $k$  khác nhau.

# STEP 1:

## TÌM K TỐI ƯU



### Elbow Method:

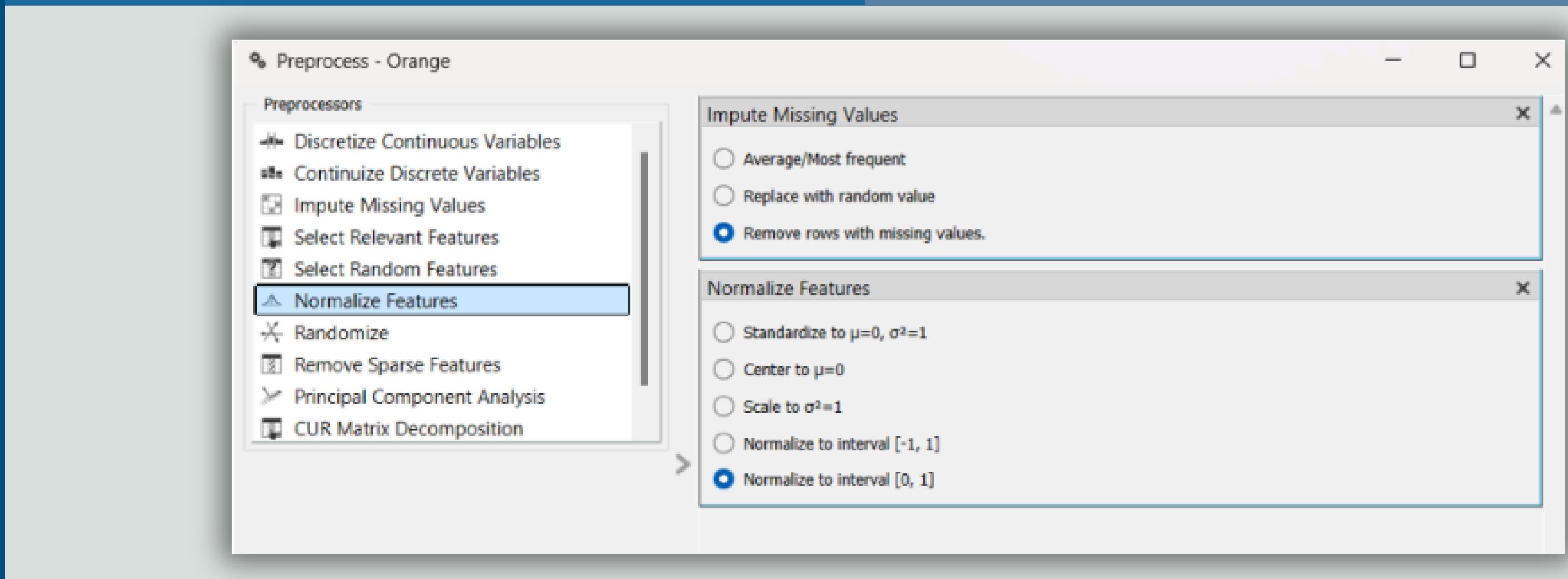
- Trực quan hóa và xác định giá trị k tốt nhất.
- Phân tích tổng độ biến thiên trong cụm (WCSS) ở các giá trị k khác nhau.

# STEP 2:

# TIỀN XỬ LÝ DỮ LIỆU

## Chuẩn hoá dữ liệu:

thu nhập, nợ nần có thể có giá trị rất  
lớn so với tuổi và năm kinh nghiệm



# BEFORE

Data Table - Orange

Info  
105 instances (no missing data)  
4 features  
No target variable.  
No meta attributes.

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

Selection  
 Select full rows

>

	Tuoi	Sinh Nghiem Lam Viec	Thu Nhap	No Nan
1	57	27	176	25
2	31	14	41	20
3	57	26	109	29
4	54	21	120	2.9
5	45	11	28	17.3
6	65	28	130	16
7	41	14	67	30.6
8	27	0	49	26
9	40	16	19	24.4
10	36	14	25	19.7
11	27	3	16	1.7
12	25	2	23	5.2
13	52	29	64	10
14	37	14	31	24.1
15	46	25.5	100	9.1
16	66	31	111	28
17	36	12	41	23
18	21	0	21	6
19	60	33	89	30
20	54	34	91	16
21	52	23	92	3
22	49	26	89	10

Restore Original Order

Send Automatically

☰ ? ⌂ | ↵ 105 ⌂ 105 | 105

# AFTER

Data Table (1) - Orange

	Tuoi	Ginh Nghiem Lam Viec	Thu Nhap	No Nan
1	0.7959	0.67500	1.000	0.8133333
2	0.2653	0.35000	0.16667	0.6466667
3	0.7959	0.65000	0.58642	0.9466667
4	0.7347	0.52500	0.65432	0.0766667
5	0.5510	0.27500	0.08642	0.5566667
6	0.9592	0.70000	0.71605	0.5133333
7	0.4694	0.35000	0.32716	1.00000
8	0.1837	0.00000	0.21605	0.8466667
9	0.4490	0.40000	0.03086	0.7933333
10	0.3673	0.35000	0.06790	0.6366667
11	0.1837	0.07500	0.01235	0.0366667
12	0.1429	0.05000	0.05556	0.1533333
13	0.6939	0.72500	0.30864	0.3133333
14	0.3878	0.35000	0.10494	0.7833333
15	0.5714	0.63750	0.53086	0.2833333
16	0.9796	0.77500	0.59877	0.9133333
17	0.3673	0.30000	0.16667	0.7466667
18	0.0612	0.00000	0.04321	0.18000
19	0.8571	0.82500	0.46296	0.98000
20	0.7347	0.85000	0.47531	0.5133333
21	0.6939	0.57500	0.48148	0.08000
22	0.6327	0.65000	0.46296	0.3133333

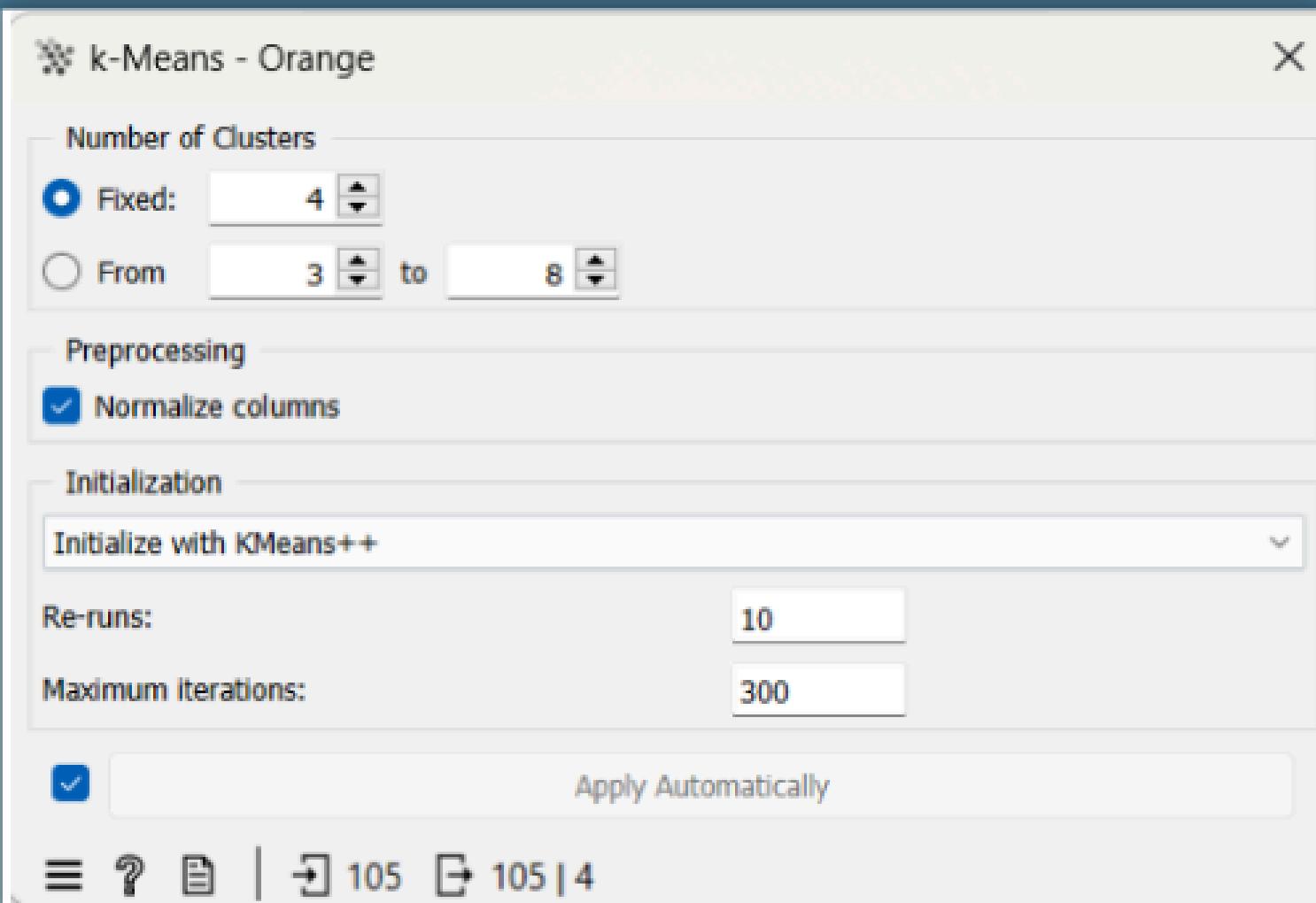
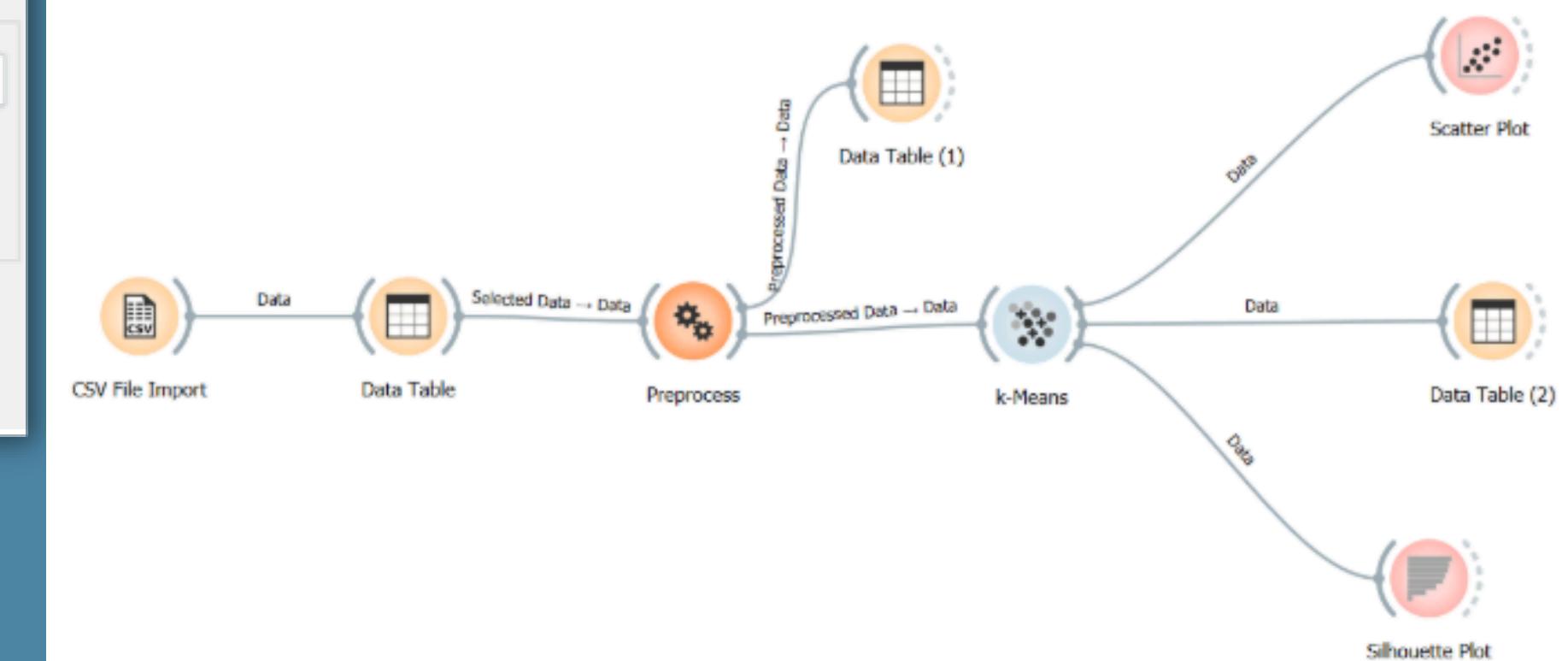
- 1. K-Means
- 2. Hierarchical

**STEP 3:**  
**ÁP DỤNG THUẬT TOÁN**

# STEP 3:

# ÁP DỤNG THUẬT TOÁN

## 1. K-Means



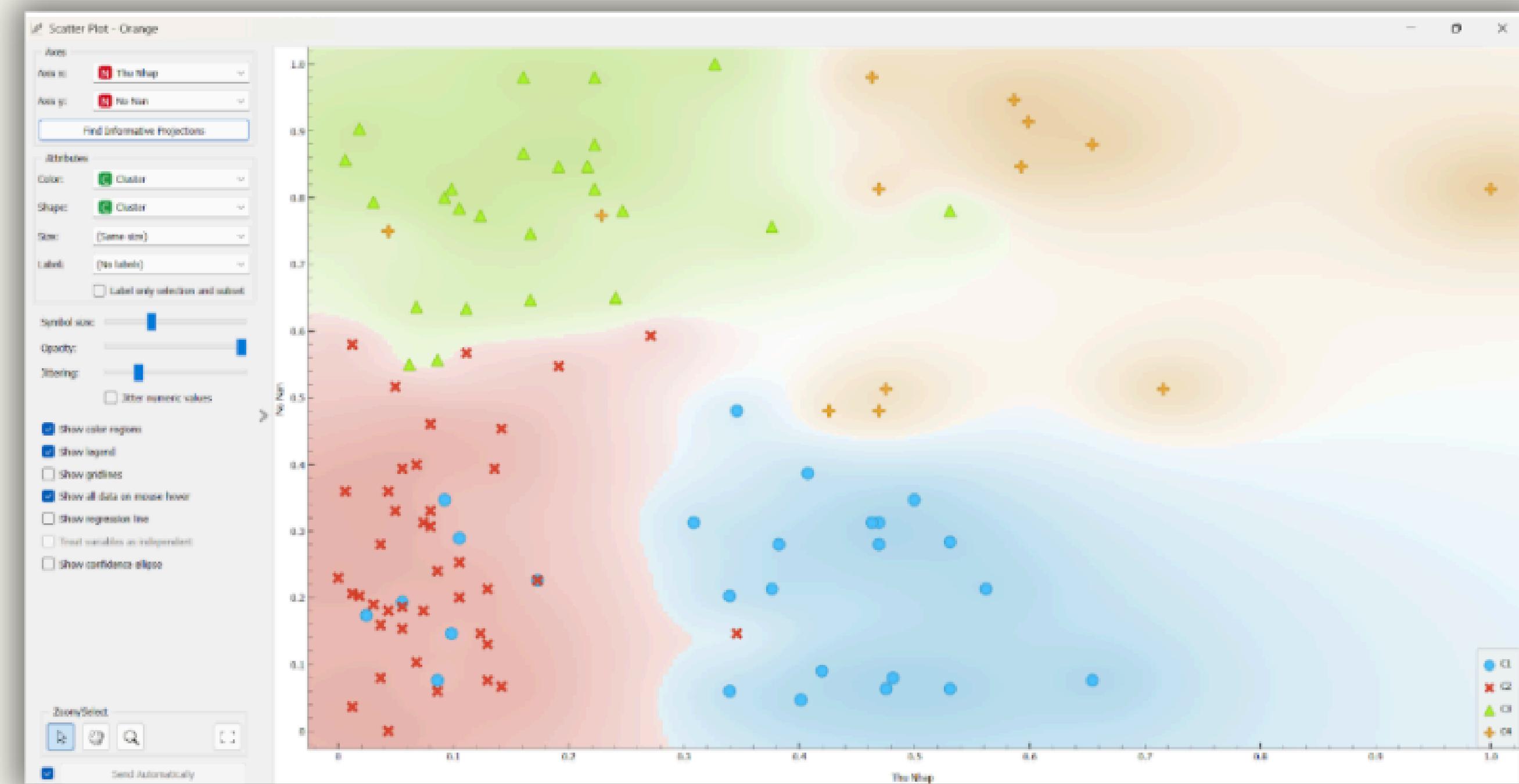
Chọn k=4

# RESULT

$k = 4$

Dựa trên:

- Thu nhập.
- Nợ nần

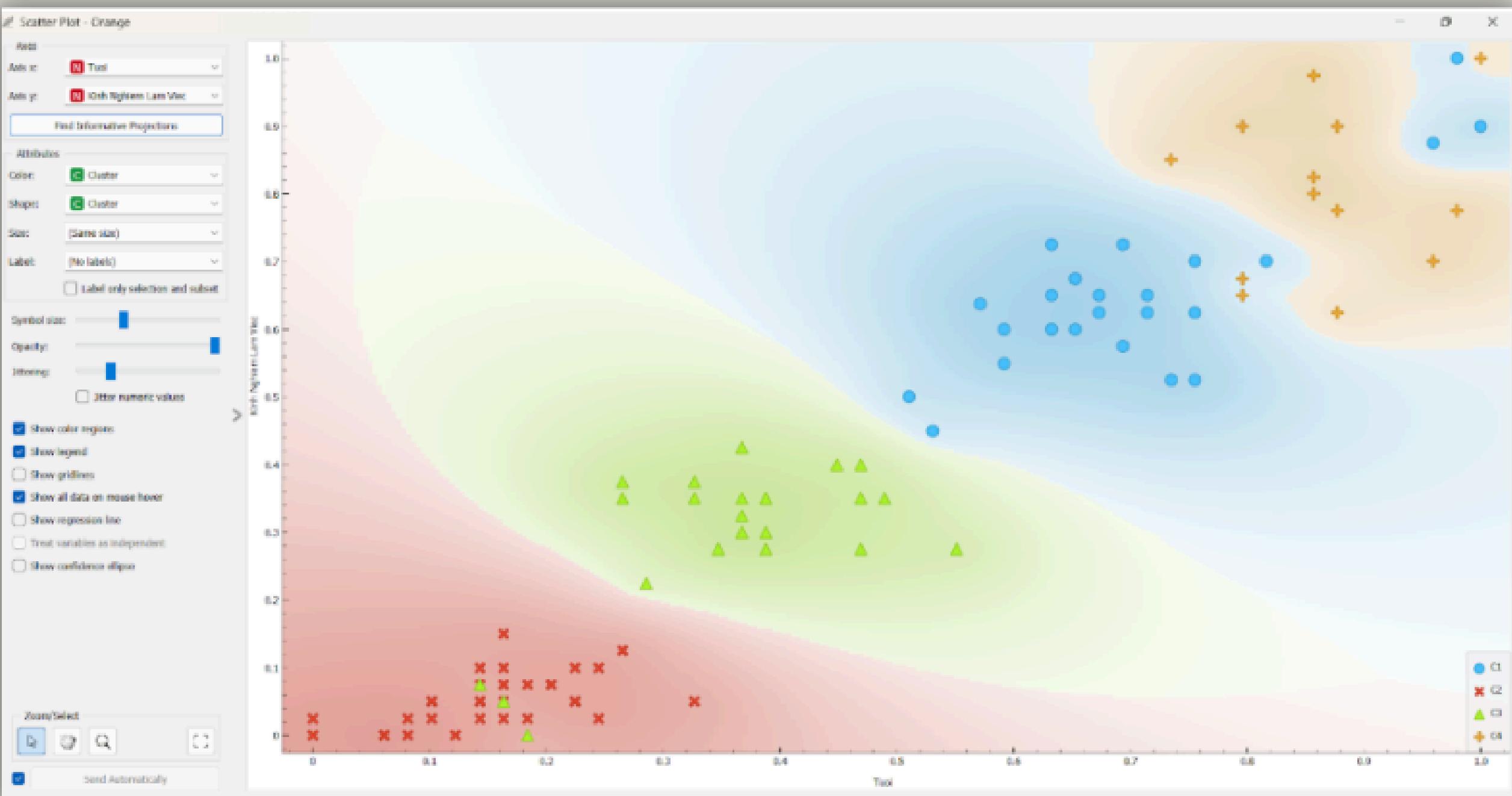


# RESULT

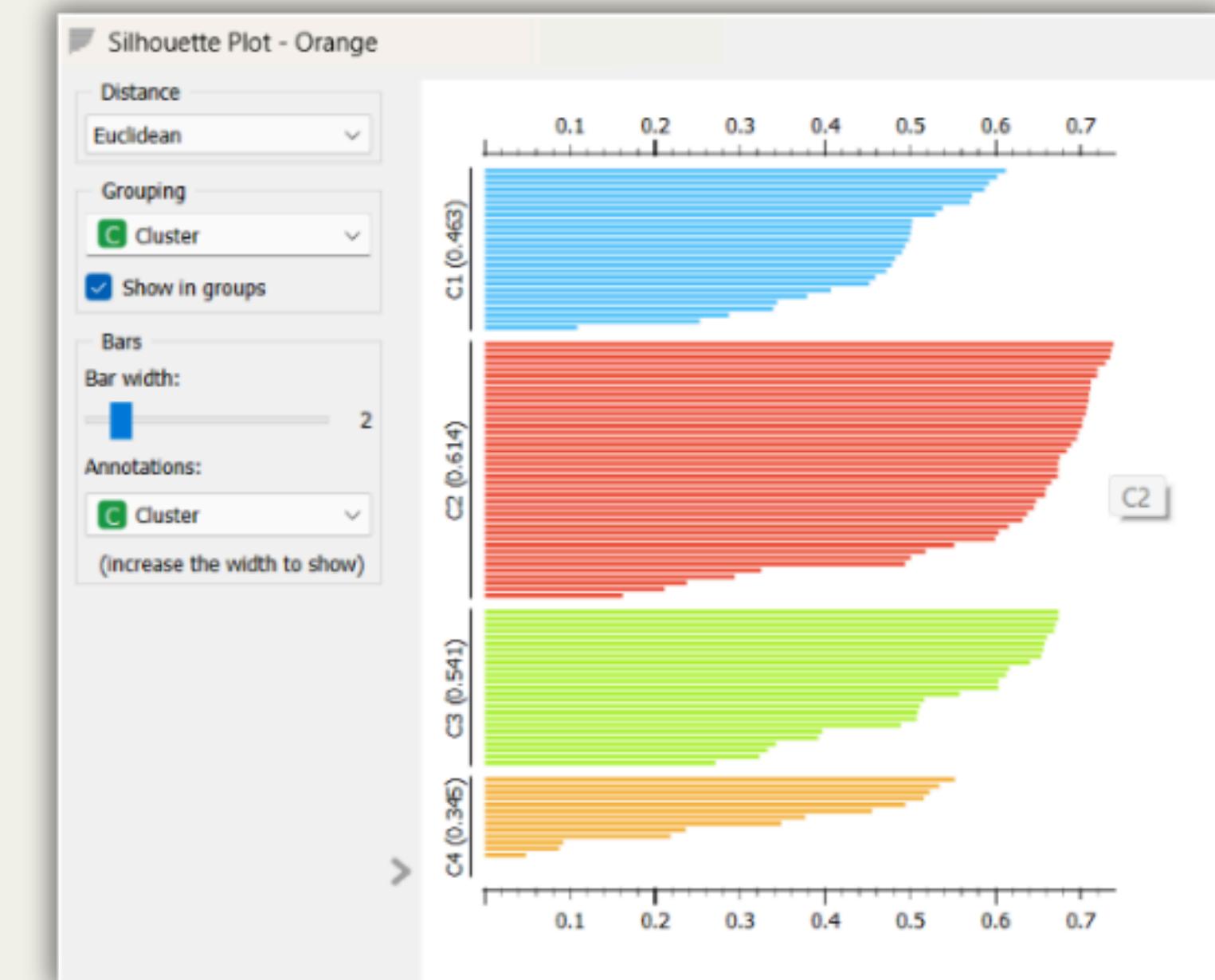
$k = 4$

Dựa trên:

- Tuổi.
- Kinh nghiệm làm việc.



# K-MEANS EVALUATION

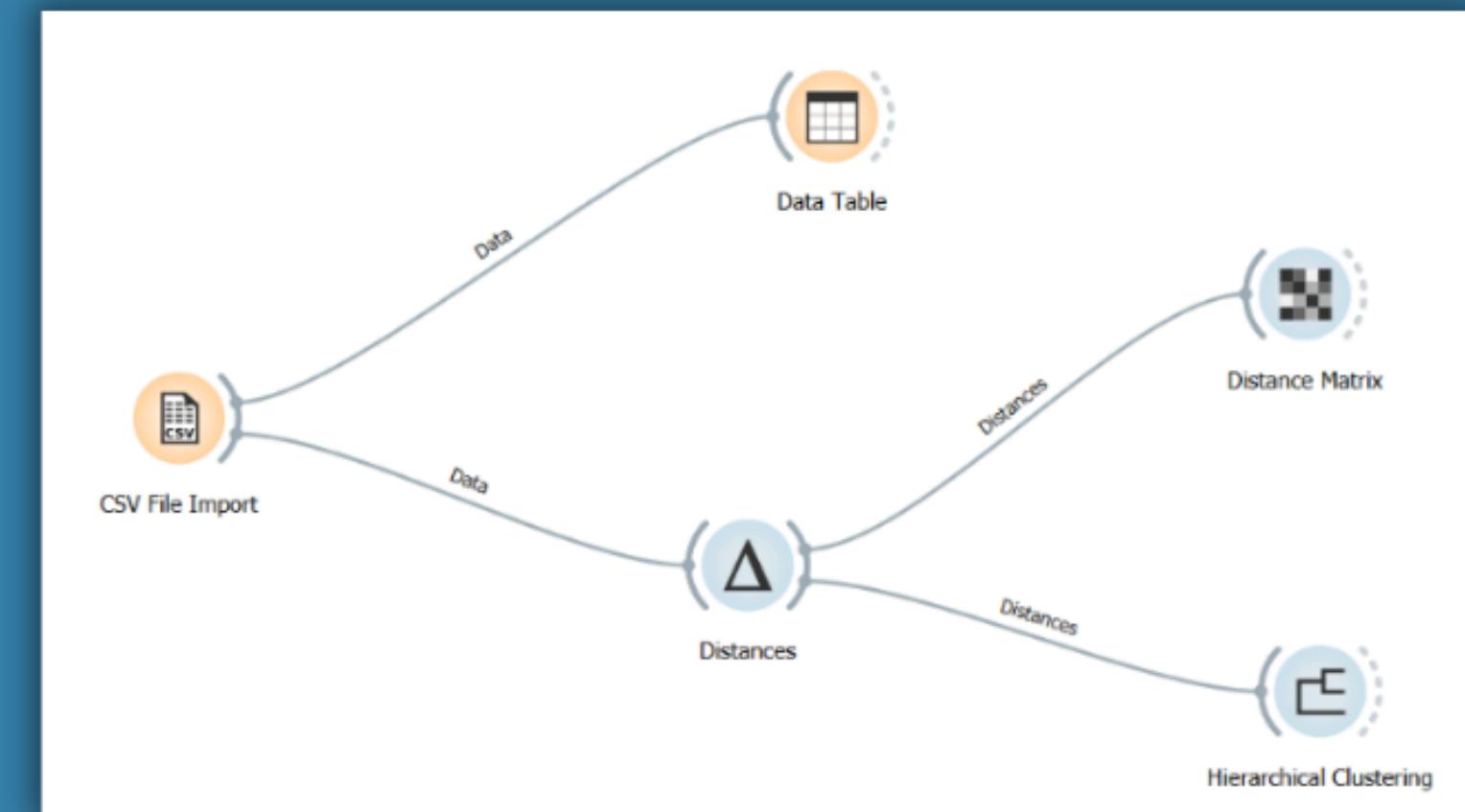


Silhouette Score

# STEP 3:

## ÁP DỤNG THUẬT TOÁN

### 2. Hierarchical



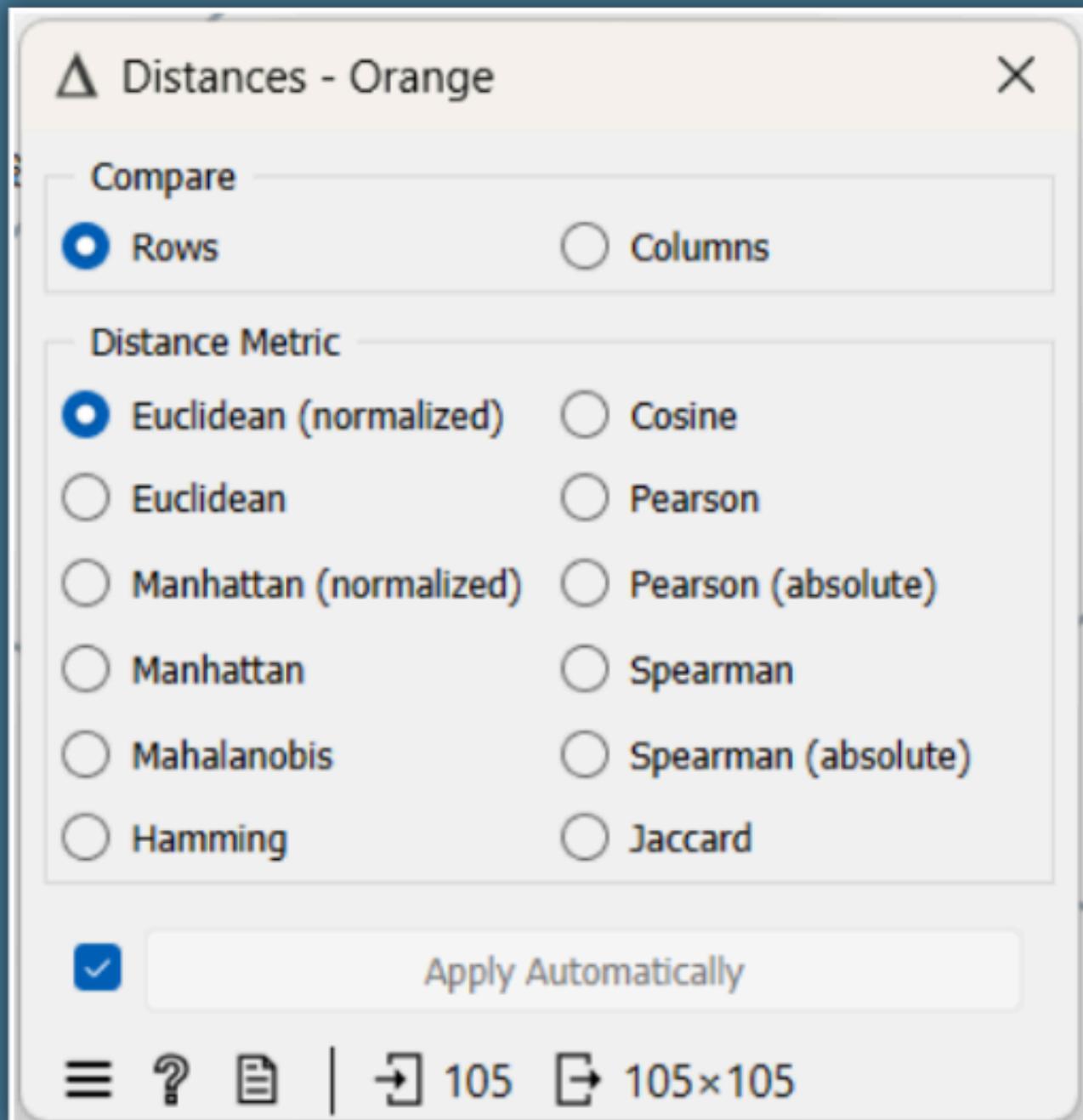
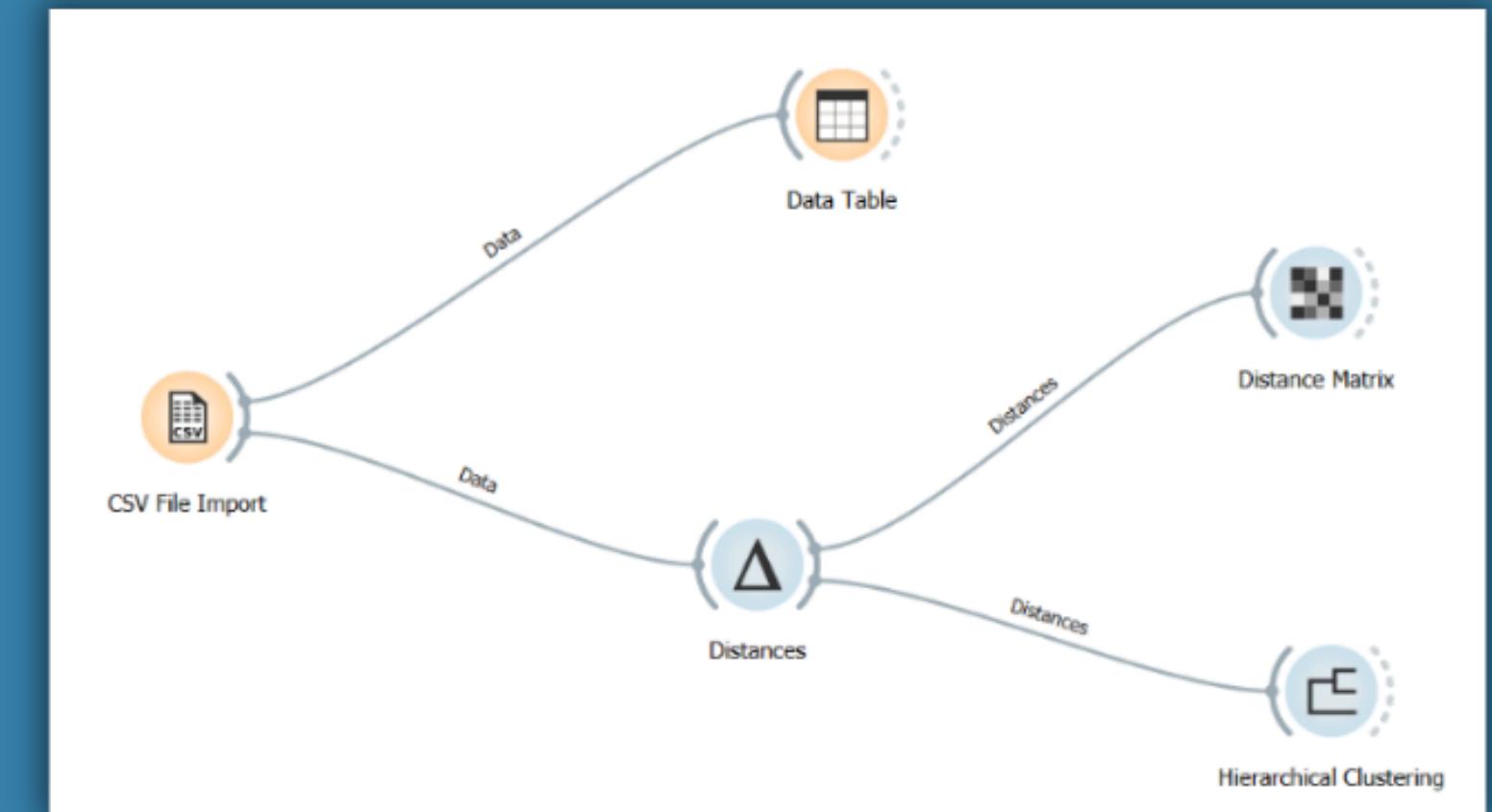
	Tuoi	Kinh Nghiem Lam Viec	Thu Nhap	No Nan
1	57		27	176
2	31		14	41
3	57		26	109
4	54		21	120
5	45		11	28
6	65		28	130
7	41		14	67
8	27		0	49
9	40		16	19
10	36		14	25
11	27		3	16
12	25		2	23
13	52		29	64
14	37		14	31
15	46		25.5	100
16	66		31	111
17	36		12	41
18	21		0	21
19	60		33	89
20	54		34	91
21	52		23	92
22	49		26	89

Dữ liệu tại Data Table

# STEP 3:

## ÁP DỤNG THUẬT TOÁN

### 2. Hierarchical



Chọn rows và Euclidean

# RESULT

## Distance Matrix

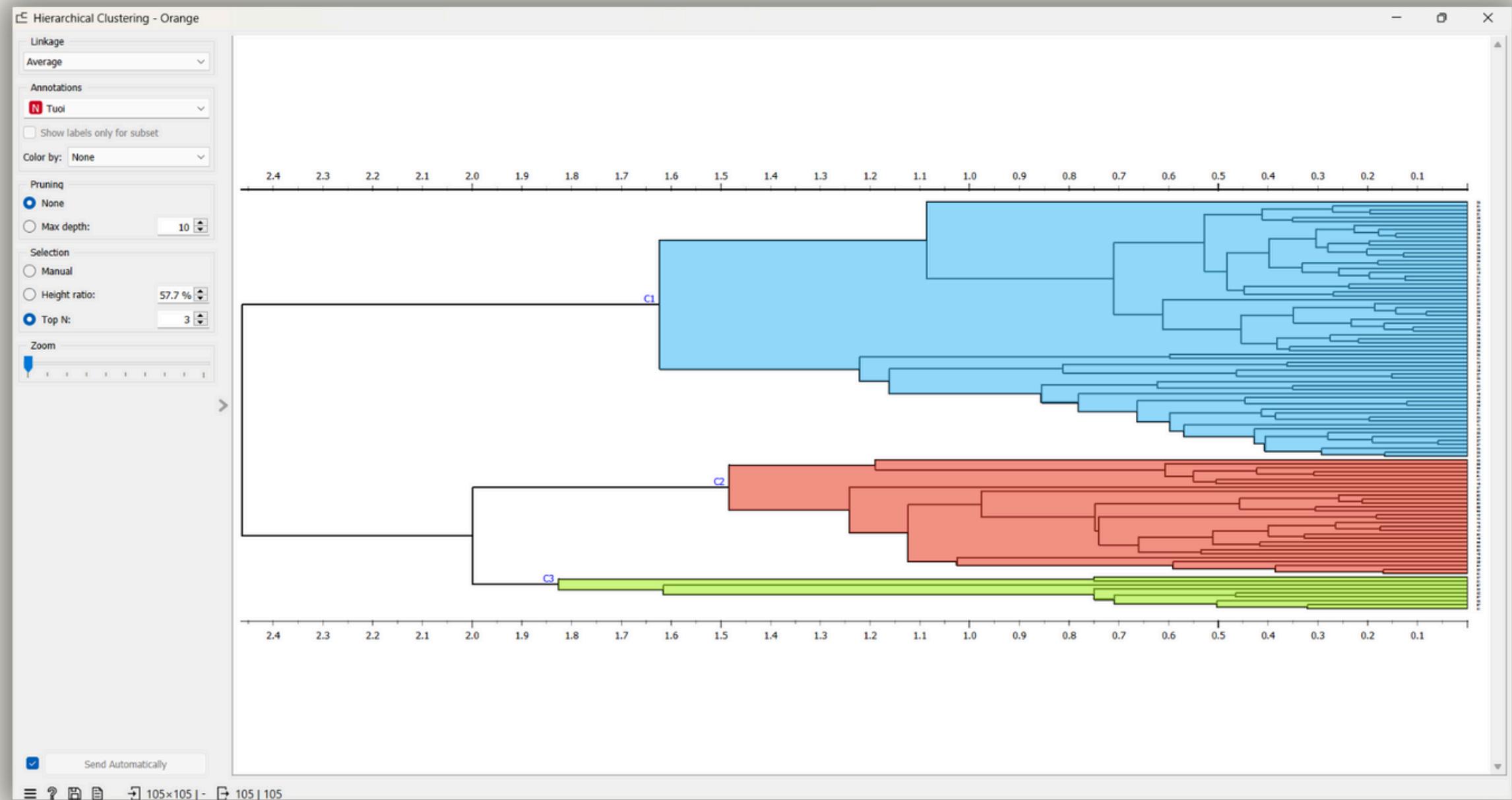
Distance Matrix - Orange

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
1		3.292	1.475	2.184	3.421	1.291	2.625	3.488	3.534	3.517	4.421	4.246	2.705	3.357	2.149	1.504	3.208	4.376	1.947	2.011	2.51
2	3.292		2.203	2.506	0.807	2.708	1.138	0.983	0.753	0.426	1.708	1.467	1.664	0.497	1.842	2.597	0.368	1.541	2.249	1.986	2.1
3	1.475	2.203		2.145	2.244	1.216	1.401	2.505	2.222	2.327	3.591	3.410	1.838	2.111	1.710	0.545	2.035	3.545	0.616	1.220	2.11
4	2.184	2.506	2.145		2.406	1.279	2.624	3.021	2.879	2.642	2.815	2.777	1.414	2.733	0.815	2.201	2.569	2.969	2.409	1.441	0.61
5	3.421	0.807	2.244	2.406		2.605	1.387	1.387	0.716	0.526	1.639	1.500	1.470	0.705	1.881	2.526	0.705	1.649	2.229	1.957	1.91
6	1.291	2.708	1.216	1.279	2.605		2.309	3.157	2.863	2.820	3.616	3.515	1.634	2.754	1.288	1.066	2.637	3.696	1.483	1.062	1.50
7	2.625	1.138	1.401	2.624	1.387	2.309		1.202	1.152	1.284	2.746	2.495	1.958	0.955	2.005	1.868	0.874	2.567	1.535	1.852	2.4
8	3.488	0.983	2.505	3.021	1.387	3.157	1.202		1.312	1.179	2.092	1.776	2.488	1.043	2.483	2.981	0.883	1.749	2.696	2.685	2.7
9	3.534	0.753	2.222	2.879	0.716	2.863	1.152	1.312		0.463	2.088	1.908	1.794	0.321	2.223	2.540	0.575	1.996	2.109	2.106	2.4
10	3.517	0.426	2.327	2.642	0.526	2.820	1.284	1.179	0.463		1.662	1.471	1.649	0.381	2.005	2.665	0.450	1.568	2.292	2.065	2.19
11	4.421	1.708	3.591	2.815	1.639	3.616	2.746	2.092	2.088	1.662		0.340	2.318	2.008	2.495	3.892	1.929	0.503	3.669	3.001	2.36
12	4.246	1.467	3.410	2.777	1.500	3.515	2.495	1.776	1.908	1.471	0.340		2.286	1.791	2.410	3.747	1.690	0.245	3.514	2.910	2.38
13	2.705	1.664	1.838	1.414	1.470	1.634	1.958	2.488	1.794	1.649	2.318	2.286		1.768	0.857	1.907	1.722	2.495	1.763	0.815	0.89
14	3.357	0.497	2.111	2.733	0.705	2.754	0.955	1.043	0.321	0.381	2.008	1.791	1.768		2.076	2.479	0.264	1.868	2.081	2.041	2.38
15	2.149	1.842	1.710	0.815	1.881	1.288	2.005	2.483	2.223	2.005	2.495	2.410	0.857	2.076		1.870	1.931	2.591	1.894	0.868	0.61
16	1.504	2.597	0.545	2.201	2.526	1.066	1.868	2.981	2.540	2.665	3.892	3.747	1.907	2.479	1.870		2.432	3.904	0.594	1.231	2.21

# RESULT

## Hierachical Clustering

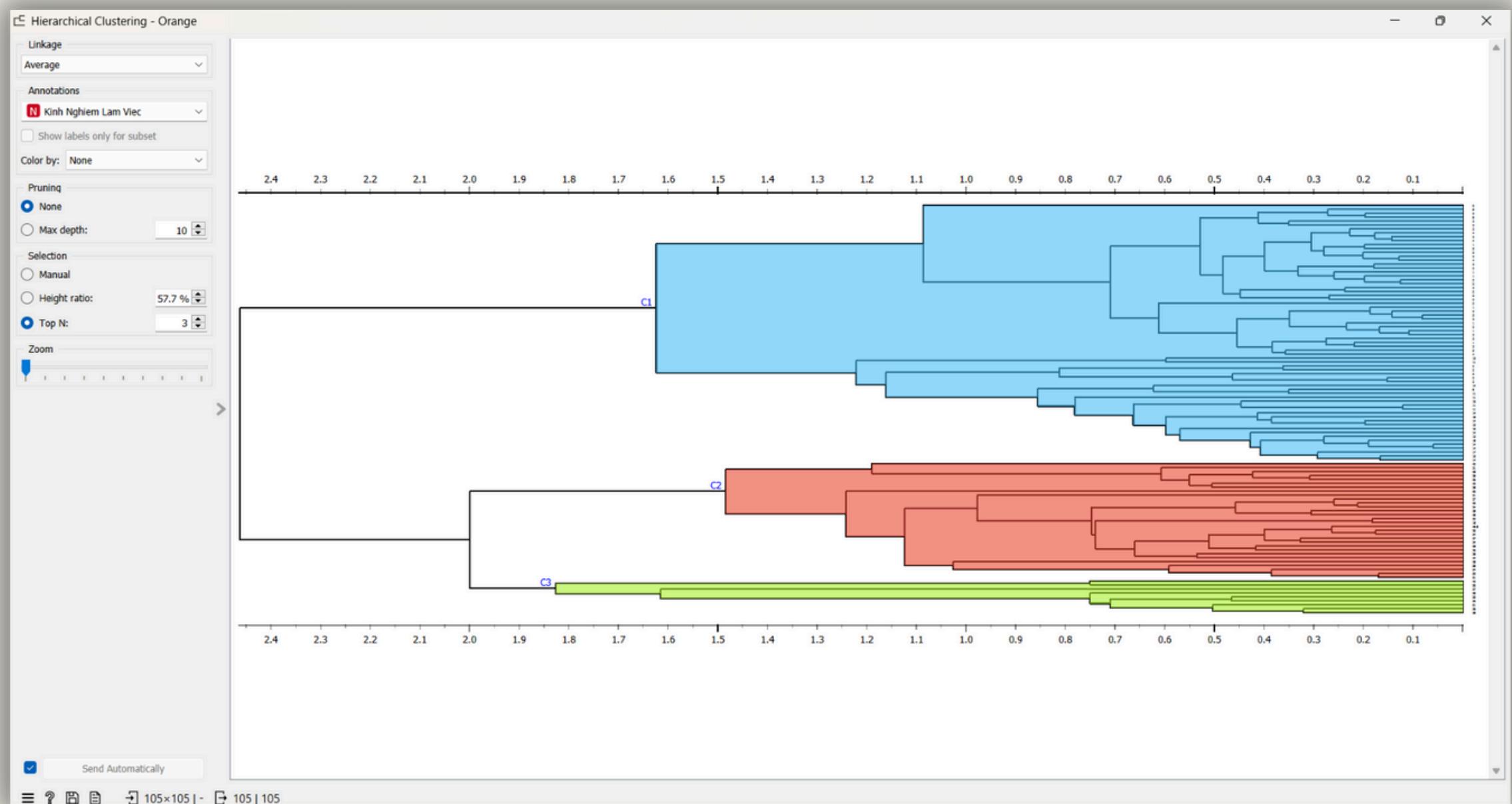
Theo tuổi



# RESULT

## Hierachical Clustering

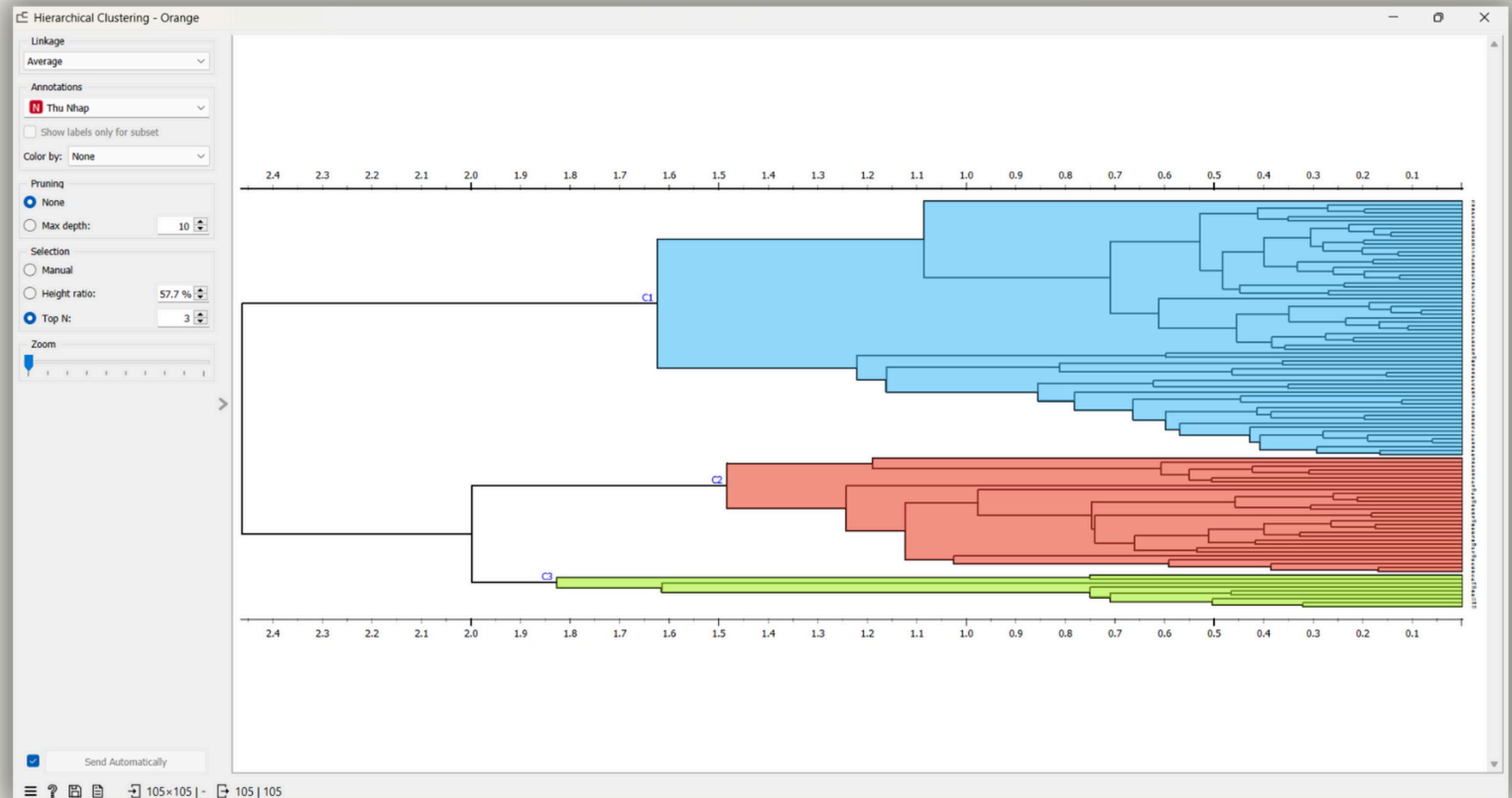
Theo kinh nghiệm làm việc



# RESULT

## Hierachical Clustering

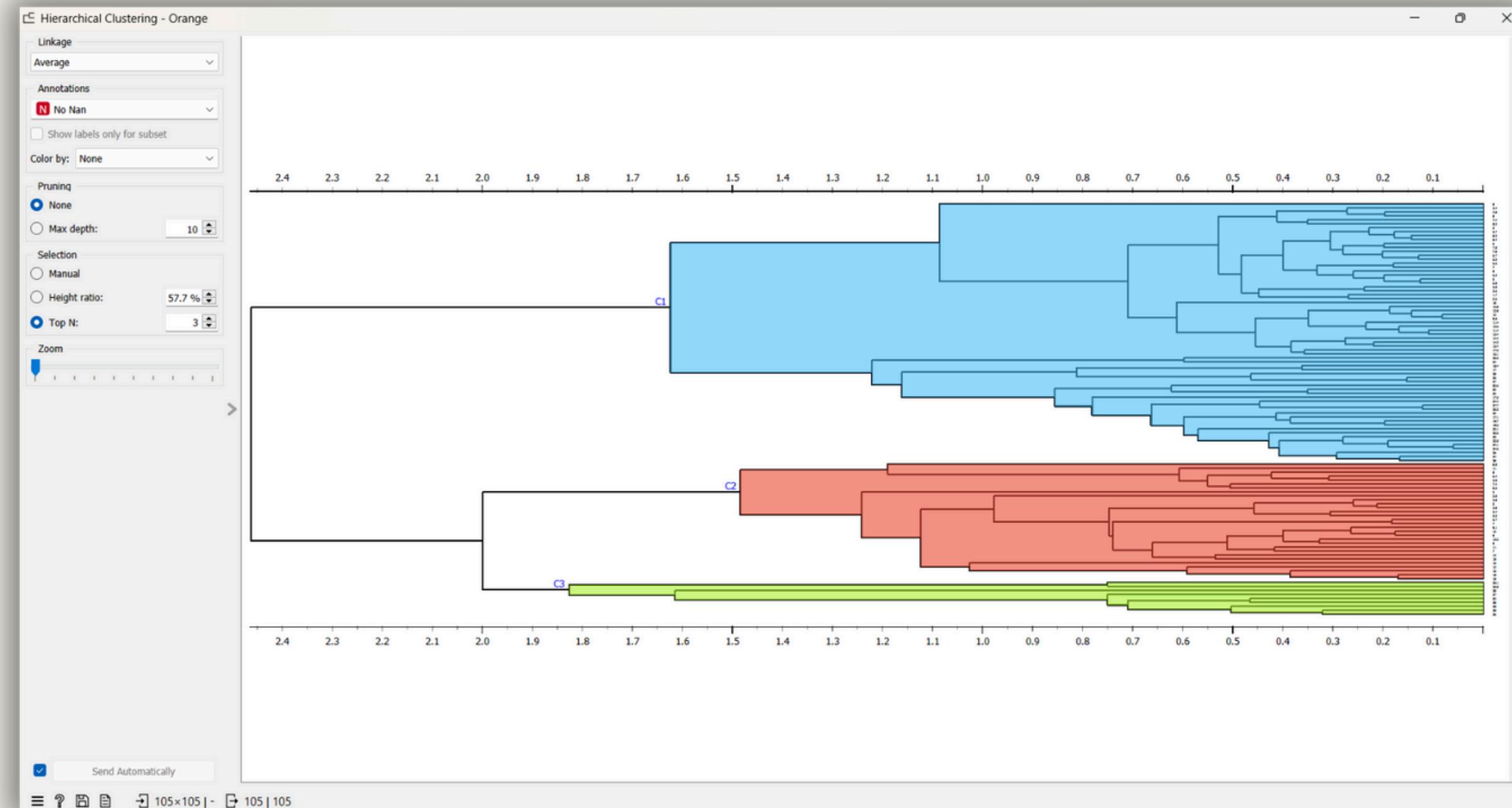
Theo thu nhập



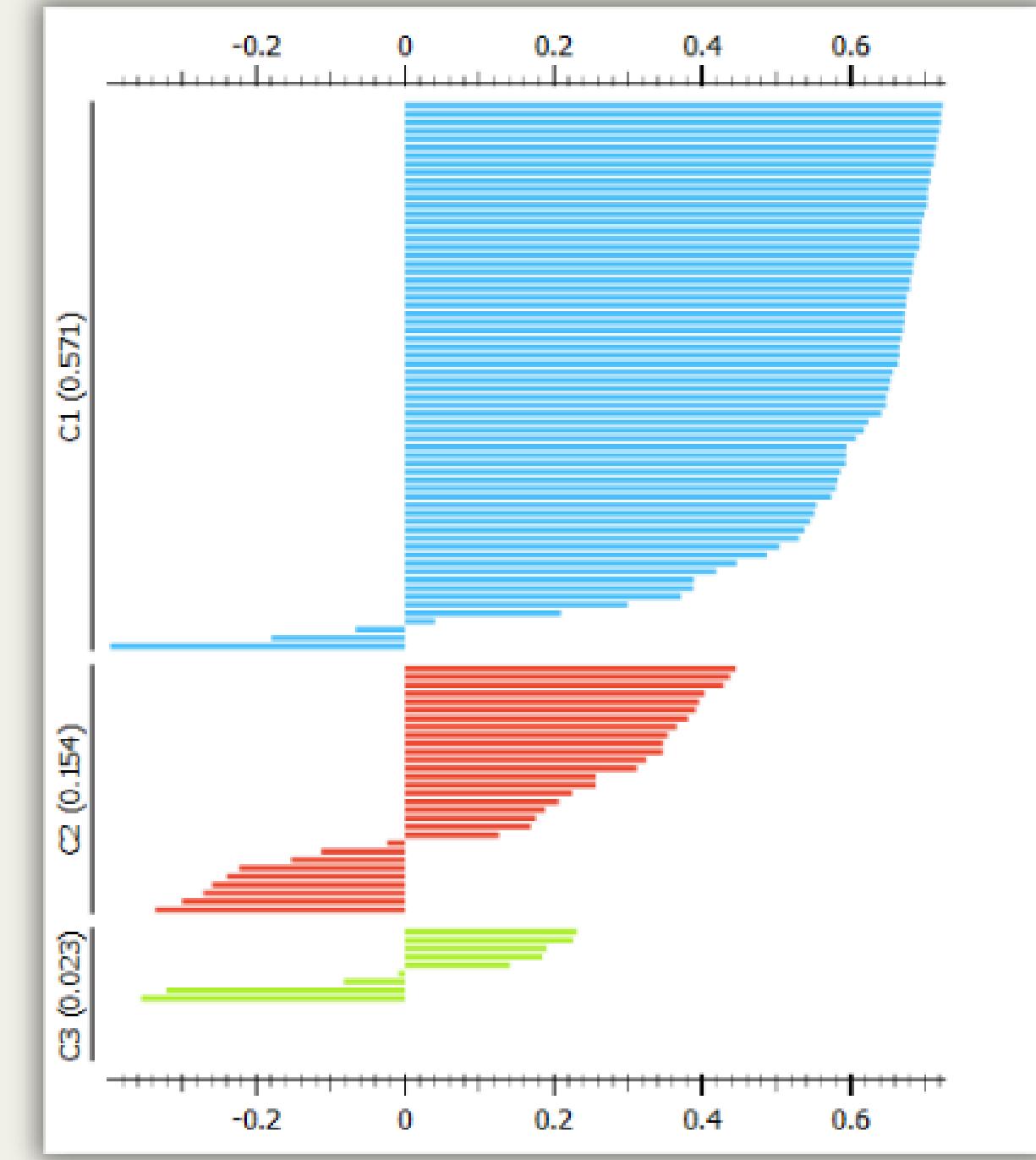
# RESULT

## Hierachical Clustering

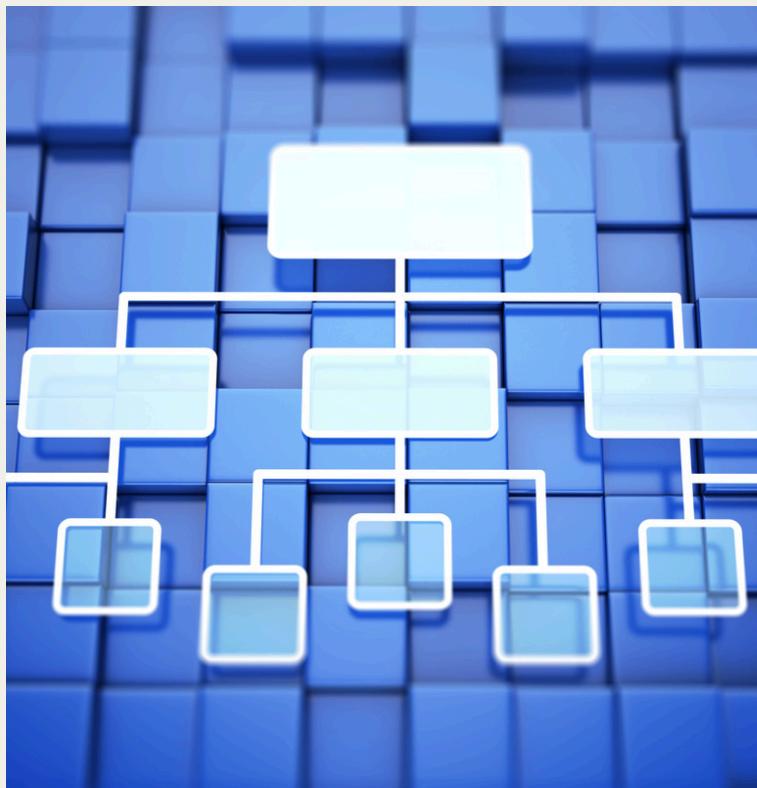
Theo nơ nần



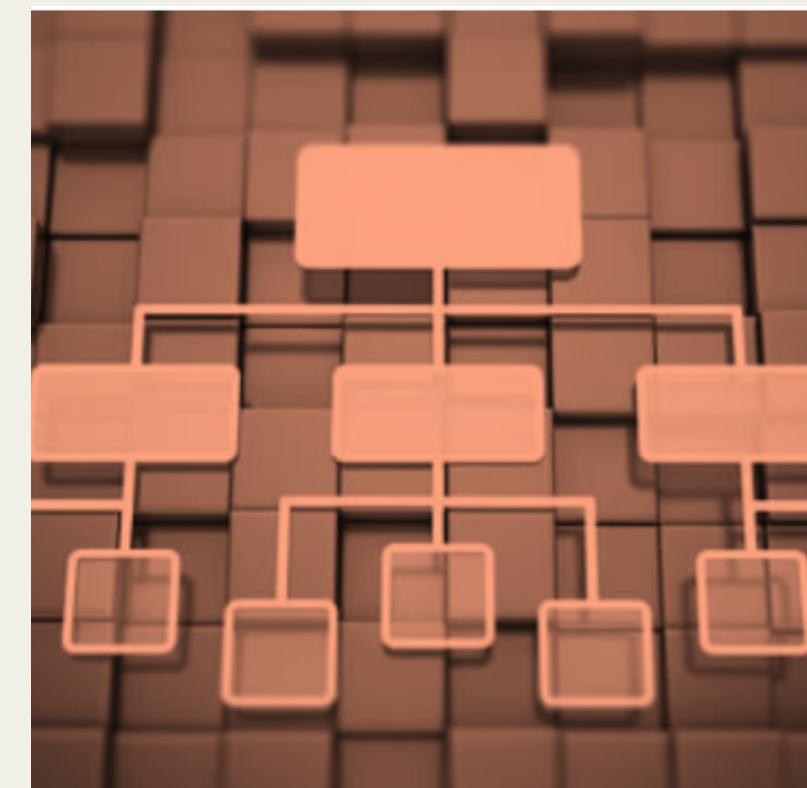
# HIERARCHICAL EVALUATION



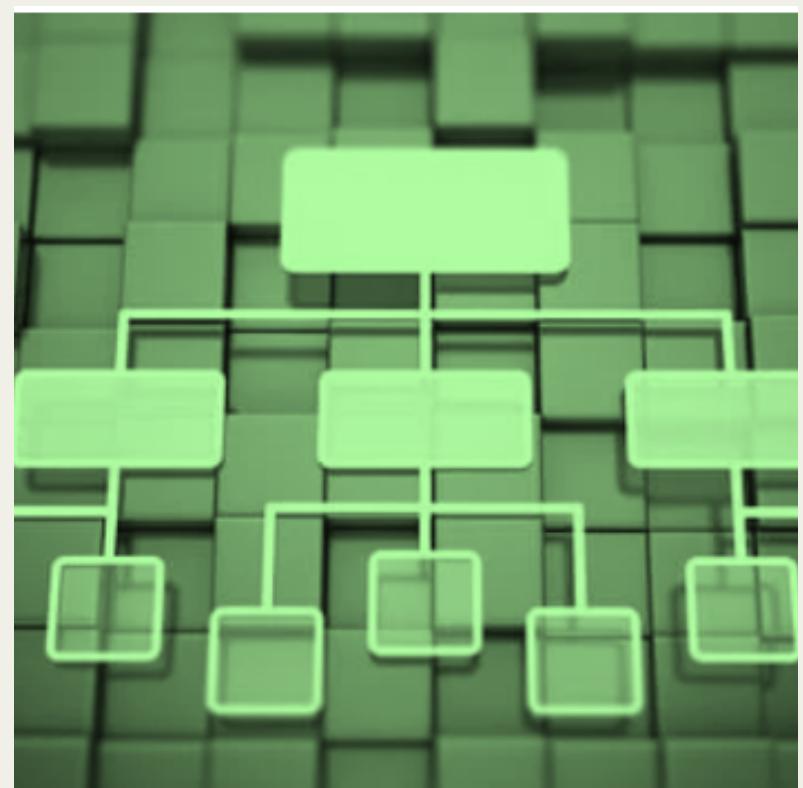
Silhouette Score



[ C1 ]



[ C2 ]

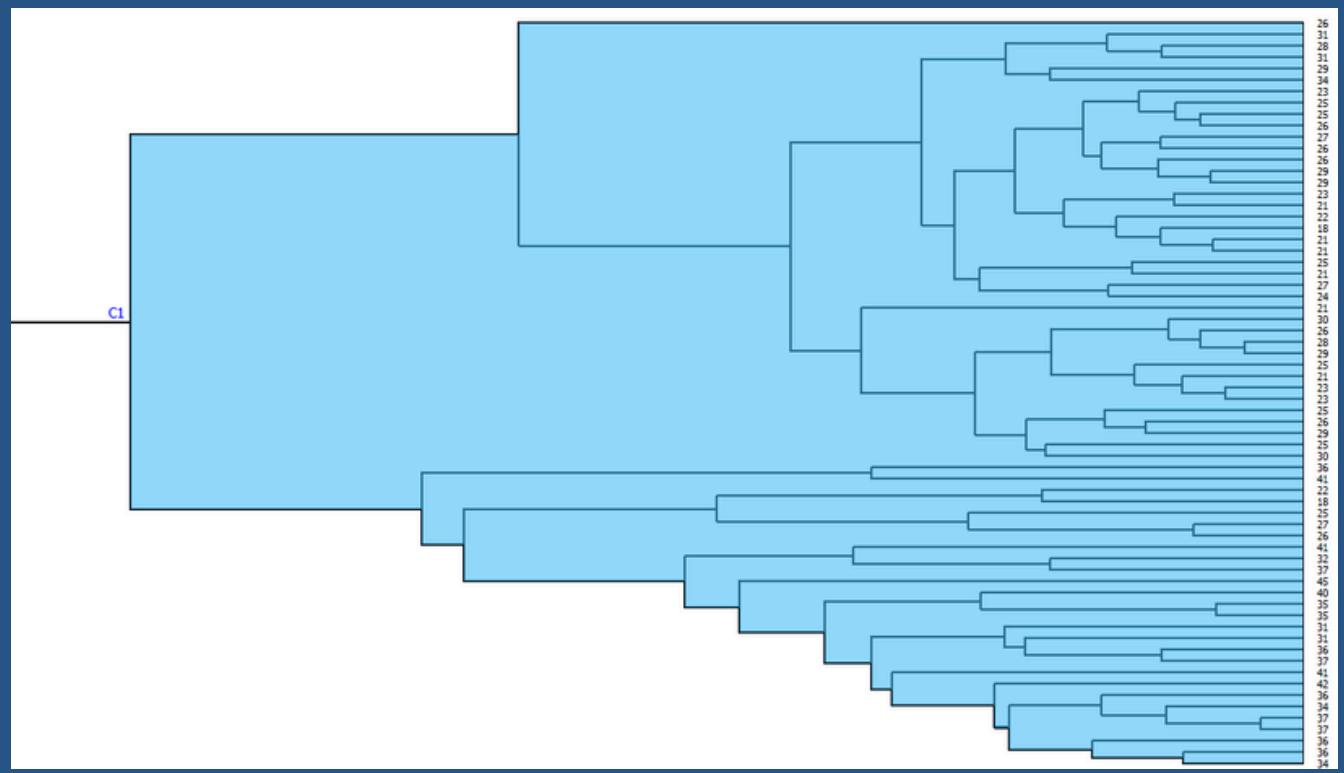


[ C3 ]

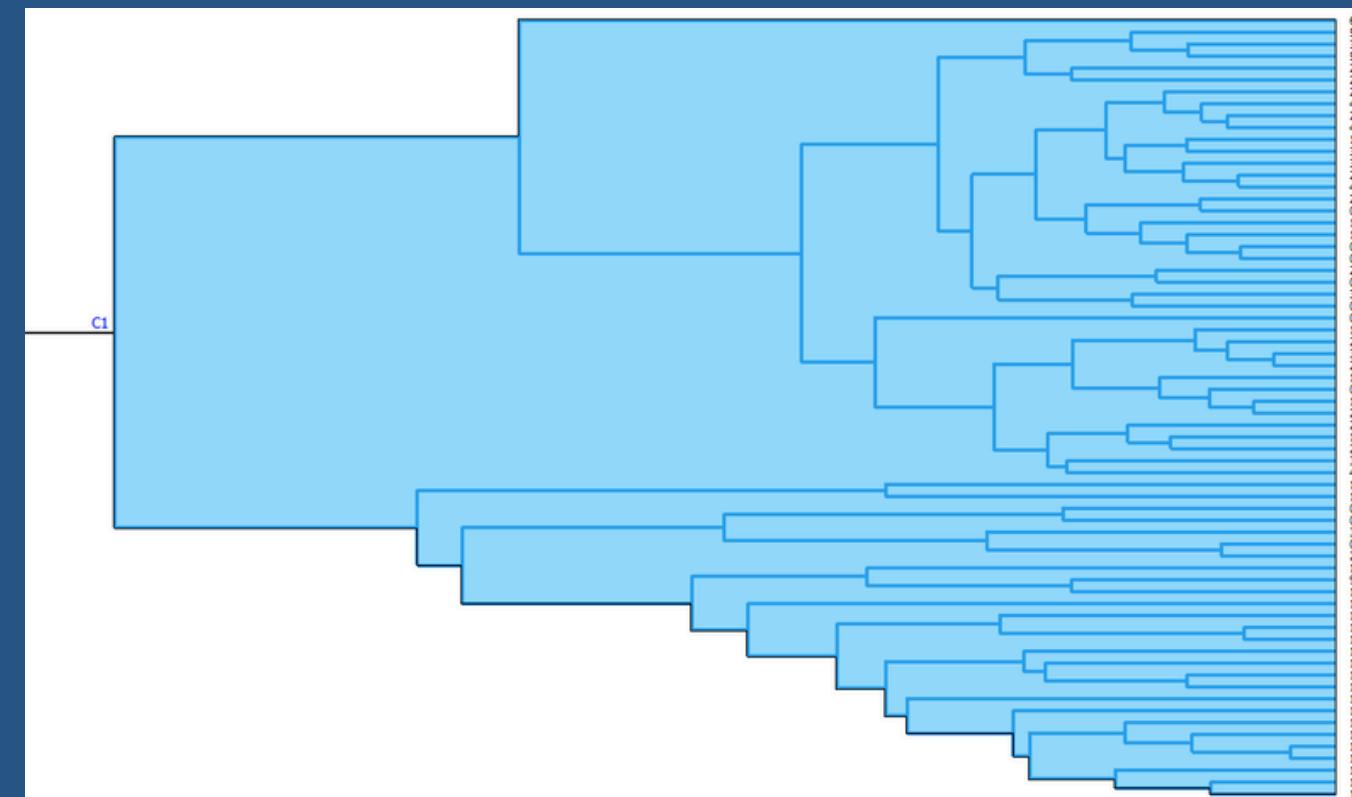
# ANALYZE

C1

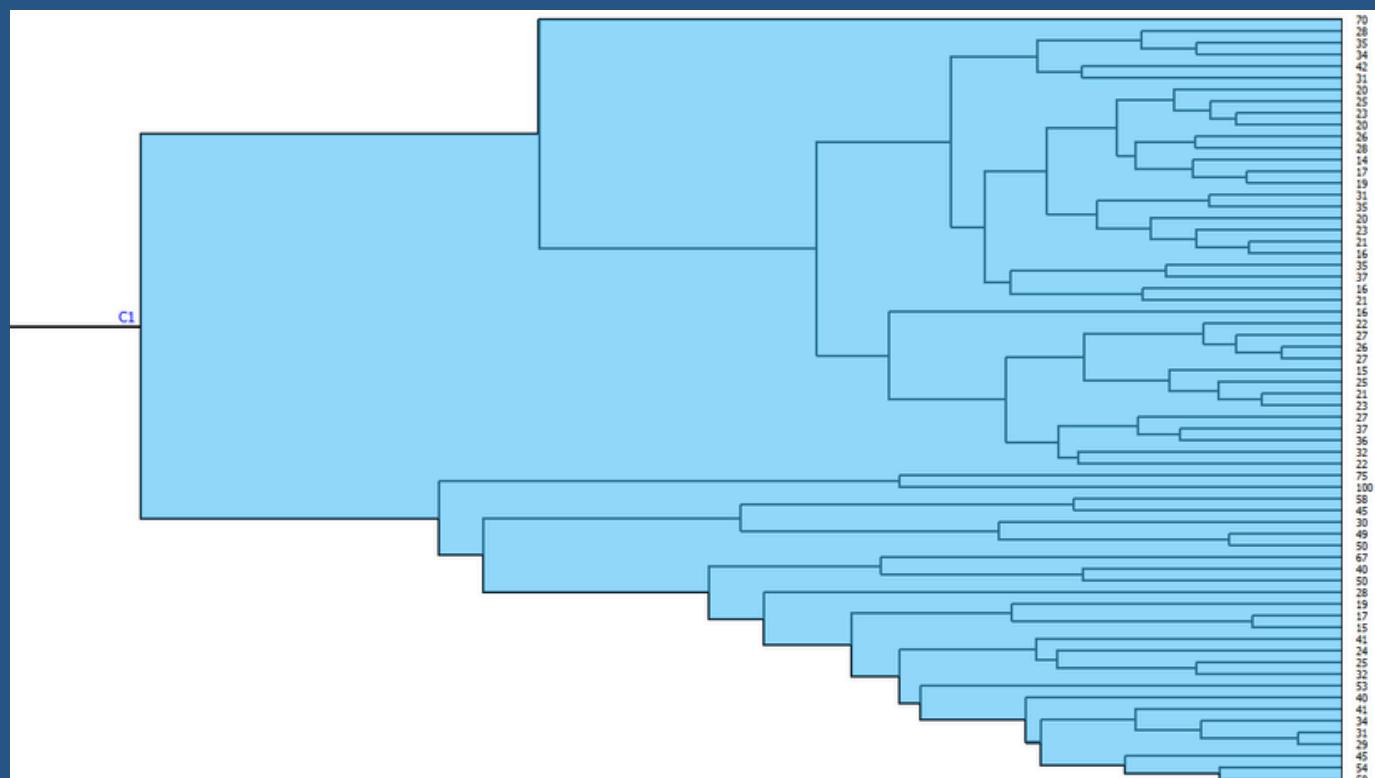
## Theo tuổi (20-31)



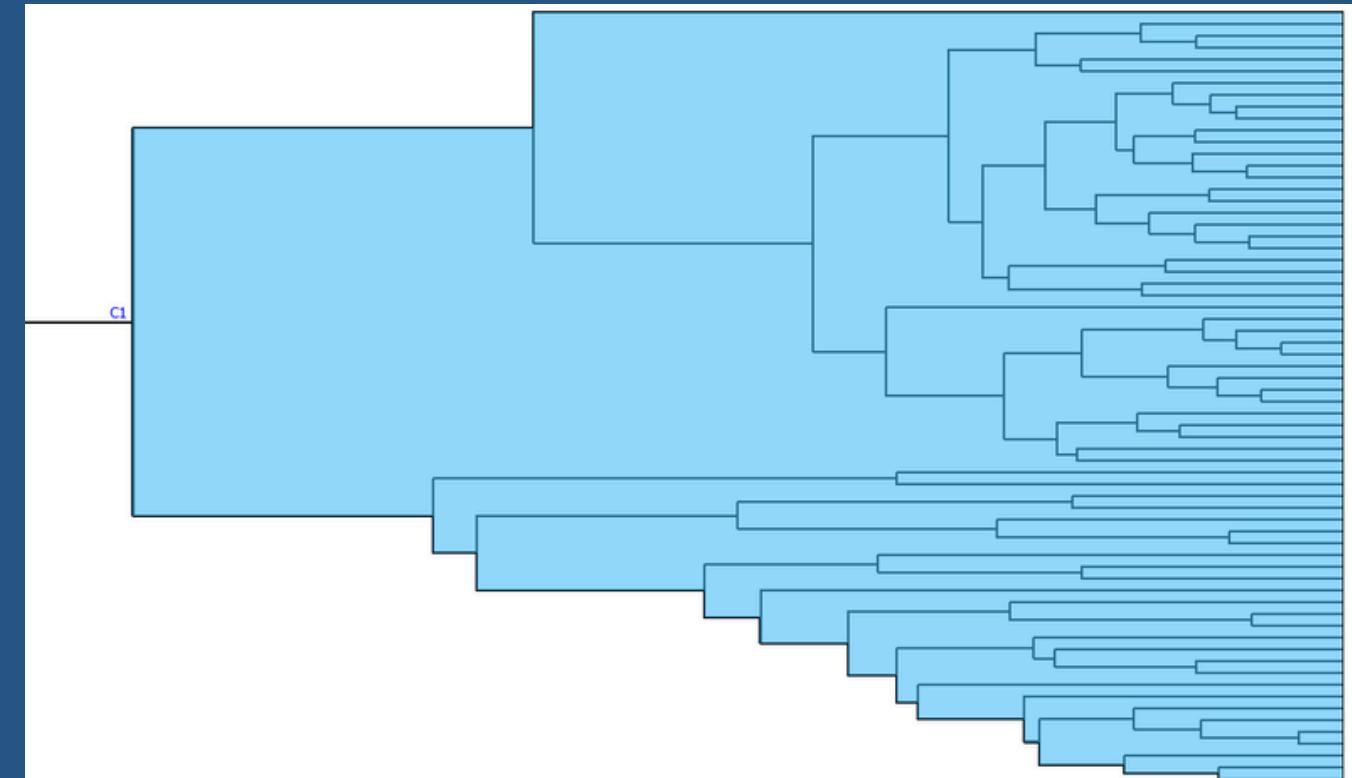
## Theo kinh nghiệm(khoảng dưới 5 năm)



## Theo thu nhập(dưới 30tr)

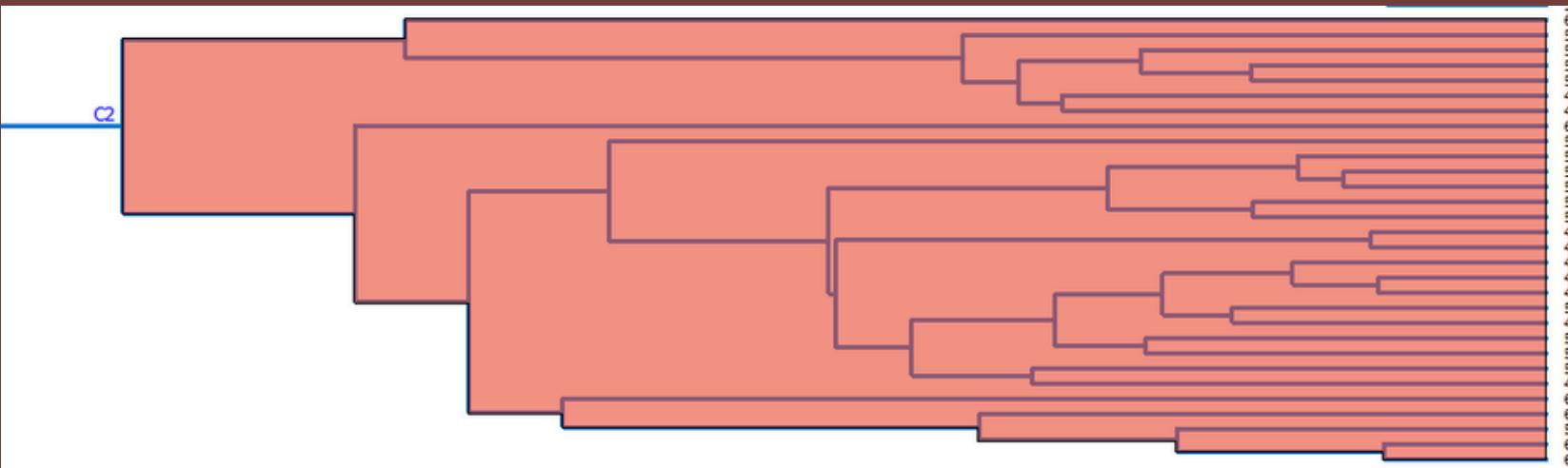


## Theo nợ(dưới 20tr)

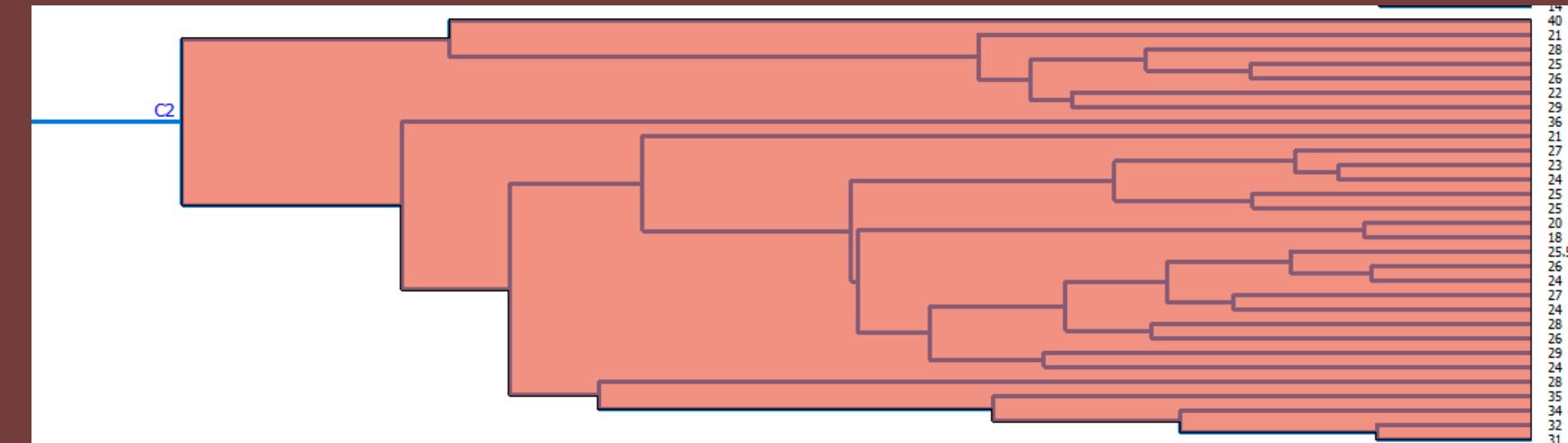


C2

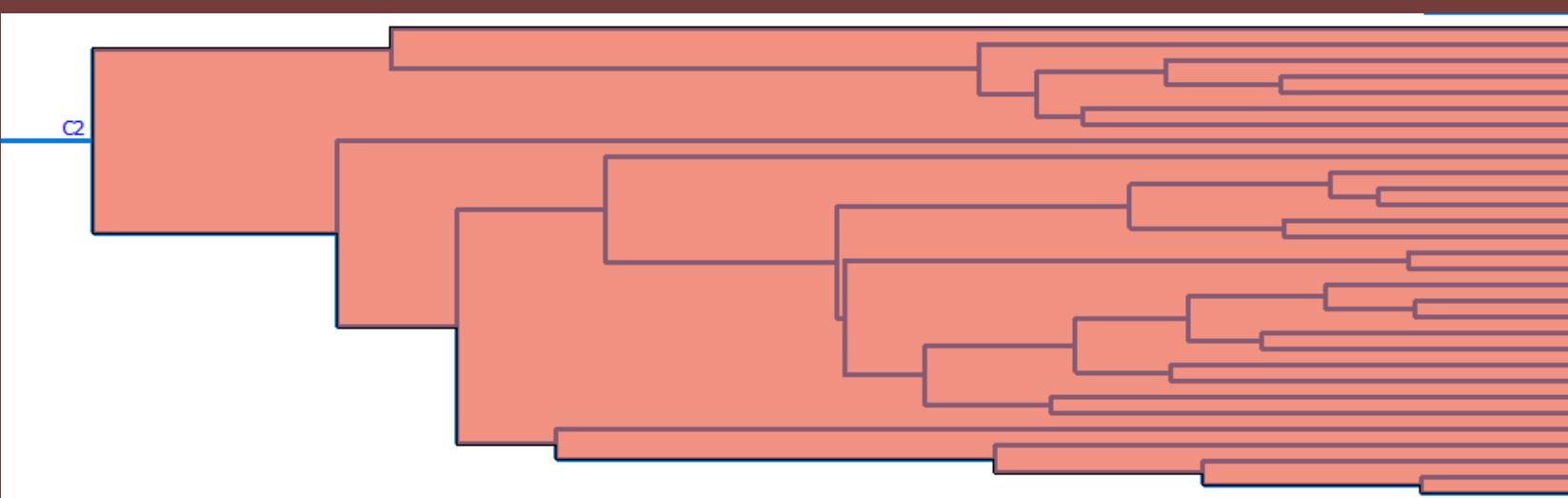
Theo tuổi(dưới 55t)



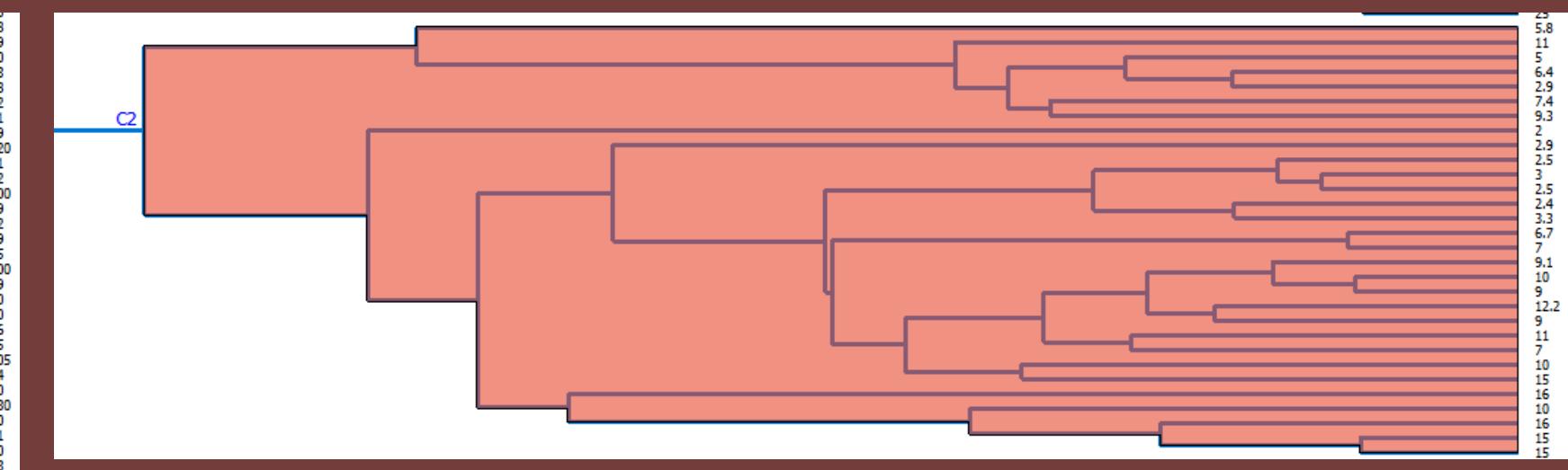
Theo kinh nghiệm (dưới 30 năm)



Theo thu nhập(dưới 100tr)



Theo nợ(dưới 15tr)



# C3



Theo tuổi  
(trên 60t)



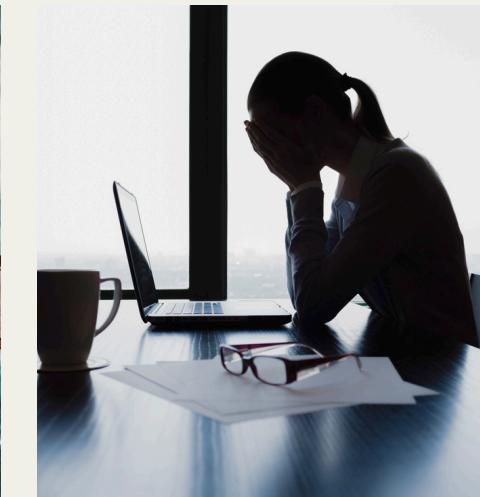
Theo kinh nghiệm  
(trên 33 năm)



Theo thu nhập  
(trên 100tr)



Theo nợ  
(25-30tr)



# SUICIDE

“vấn đề sức khỏe cộng  
đồng nghiêm trọng”

“khoảng 1,5% tổng số ca  
tử vong toàn cầu”



“thanh thiếu niên và  
người trẻ tuổi”

“khoảng 800.000 sinh  
mạng”

# ASSOCIATION RULES

Bộ dữ liệu

Human Suicide Risks

Mục tiêu

Xác định các yếu tố rủi ro  
Nhận diện các mô hình  
Phát triển giải pháp

# SUICIDE RISKS DATASET

ASSOCIATION RULES

2024

TÊN TRƯỜNG	Ý NGHĨA
ID	Mã định danh cá nhân
Gender	Giới tính (Male, Female)
Mental_Health_Issue	Vấn đề sức khỏe tâm thần/tâm lý (Yes/No)
Substance_Abuse	Sử dụng chất kích thích không (Yes/No)
Abuse_History	Từng bị lạm dụng (tinh thần, thể chất, tình dục) (Y/N)
Social_Isolation	Bị cô lập xã hội (Yes/No)
Financial_Problem	Gặp vấn đề tài chính (Yes/No)
Previous_Attempts	Từng cố tự sát chưa (Yes/No)

# OUR STEPS



# STEP 1: LOAD DATA

```
from google.colab import files
from google.colab import drive

drive.mount('/content/drive')

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import networkx as nx
from itertools import combinations

df = pd.read_csv('/content/drive/MyDrive/ML_Data/human_suicide_risks.csv', sep=',')
print(df.shape)
df.head()
```

# STEP 1: LOAD DATA

```
Mounted at /content/drive  
(500, 8)
```

	ID	Gender	Mental_Health_Issue	Substance_Abuse	Previous_Attempts	Abuse_History	Social_Isolation	Financial_Problem	
0	1	Male	No	No	Yes	No	Yes	Yes	
1	2	Male	Yes	Yes	No	No	No	No	
2	3	Female	Yes	Yes	No	Yes	Yes	Yes	
3	4	Male	Yes	Yes	Yes	No	No	No	
4	5	Female	Yes	Yes	No	Yes	Yes	Yes	

# STEP 2: CHUẨN BỊ DỮ LIỆU

```
def prepare_transactions(data, columns):
    transactions = []
    for _, row in data.iterrows():
        transaction = [col for col in columns if row[col] == 'Yes']
        transactions.append(transaction)
    return transactions

selected_columns = ['Mental_Health_Issue', 'Substance_Abuse', 'Previous_Attempts',
                    'Abuse History', 'Social Isolation', 'Financial Problem']
transactions = prepare_transactions(df, selected_columns)
```

# STEP 3:

# FREQUENT ITEMSET

# APRIORI ALGORITHMS

```
def apriori_analysis(transactions, min_support):
    C1 = {}
    for transaction in transactions:
        for item in transaction:
            C1[item] = C1.get(item, 0) + 1

    L1 = {key: value for key, value in C1.items() if value / len(transactions) >= min_support}

    L = [L1]
    k = 2
    while L[k - 2]:
        Ck = {}
        for transaction in transactions:
            for combo in combinations(transaction, k):
                Ck[combo] = Ck.get(combo, 0) + 1

        Lk = {key: value for key, value in Ck.items() if value / len(transactions) >= min_support}
        L.append(Lk)
        k += 1

    frequent_itemsets = [item for sublist in L for item in sublist.keys()]
    return frequent_itemsets
```

# STEP 3:

# FREQUENT

# ITEMSET

# FP GROWTH

# ALGORITHMS

## Phân 1: Lớp FPNode

```
class FPNode:  
    def __init__(self, item, count, parent):  
        self.item = item  
        self.count = count  
        self.parent = parent  
        self.children = {}  
        self.link = None
```

# STEP 3: FREQUENT ITEMSET

## FP GROWTH ALGORITHMS

## Phần 2: Lớp FPTree

Hàm khởi tạo

```
class FPTree:  
    def __init__(self, transactions, min_support):  
        self.min_support = min_support  
        self.header_table = {}  
        self.root = FPNode(None, 1, None)  
        self._build_tree(transactions)
```

# FP GROWTH ALGORITHMS

## STEP 3: FREQUENT ITEMSET

```
def _build_tree(self, transactions):
    # Count item frequencies
    item_counts = {}
    for transaction in transactions:
        for item in transaction:
            item_counts[item] = item_counts.get(item, 0) + 1

    # Filter items by min_support
    self.header_table = {item: [count, None] for item, count in item_counts.items() if count >= self.min_support}
    if not self.header_table:
        return

    # Sort items in each transaction by frequency
    for transaction in transactions:
        sorted_items = [item for item in sorted(transaction, key=lambda x: (-item_counts[x], x)) if item in self.header_table]
        self._insert_tree(sorted_items, self.root)
```

Phần 2: Lớp FPTree  
Hàm xây dựng FP-Tree

## Phần 2: Lớp FPTree

### Insert vào FP- Tree

```
def _insert_tree(self, items, node):
    if not items:
        return

    first_item = items[0]
    if first_item in node.children:
        node.children[first_item].count += 1
    else:
        new_node = FPNode(first_item, 1, node)
        node.children[first_item] = new_node

        if self.header_table[first_item][1] is None:
            self.header_table[first_item][1] = new_node
        else:
            current = self.header_table[first_item][1]
            while current.link is not None:
                current = current.link
            current.link = new_node

    self._insert_tree(items[1:], node.children[first_item])
```

**STEP 3:**  
**FREQUENT ITEMSET**  
**FP GROWTH ALGORITHMS**

## Phần 2: Lớp FPTree

### Khai thác mẫu thường xuyên

```
def mine_patterns(self):
    patterns = {}
    for item, (count, node) in self.header_table.items():
        conditional_patterns = []
        while node is not None:
            path = []
            parent = node.parent
            while parent is not None and parent.item is not None:
                path.append(parent.item)
                parent = parent.parent
            path.reverse()
            for _ in range(node.count):
                conditional_patterns.append(path)
            node = node.link

        # Recursively mine conditional tree
        conditional_tree = FPTree(conditional_patterns, self.min_support)
        conditional_patterns = conditional_tree.mine_patterns()
        for pattern, freq in conditional_patterns.items():
            patterns[tuple([item] + list(pattern))] = freq

    patterns[(item,)] = count
return patterns
```

**STEP 3:**  
**FREQUENT ITEMSET**  
**FP GROWTH ALGORITHMS**

# STEP 4: RULES

```
def generate_association_rules(frequent_itemsets, transactions, min_confidence):
    rules = []
    for itemset in frequent_itemsets:
        if len(itemset) > 1:
            for i in range(1, len(itemset)):
                for antecedent in combinations(itemset, i):
                    consequent = tuple(set(itemset) - set(antecedent))
                    antecedent_support = sum(1 for transaction in transactions if set(antecedent).issubset(transaction))
                    both_support = sum(1 for transaction in transactions if set(itemset).issubset(transaction))
                    if antecedent_support > 0:
                        confidence = both_support / antecedent_support
                        if confidence >= min_confidence:
                            rules.append((antecedent, consequent, confidence))
    return rules
```

# APRIORI EXECUTION

```
min_support = 0.25
min_confidence = 0.5
frequent_itemsets = apriori_analysis(transactions, min_support)
rules = generate_association_rules(frequent_itemsets, transactions, min_confidence)

print("Frequent Itemsets:", frequent_itemsets)
print("Association Rules:")
for antecedent, consequent, confidence in rules:
    print(f'{antecedent} => {consequent} (Confidence: {confidence:.2f})')
```

# FP-GROWTH EXECUTION

```
min_support = 0.25 * len(transactions)
min_confidence = 0.5
fp_tree = FPTree(transactions, min_support)
patterns = fp_tree.mine_patterns()
rules_fp = generate_association_rules_fp(patterns, min_confidence, transactions)

filtered_patterns = {itemset: support for itemset, support in patterns.items() if support / len(transactions) >= 0.2}
filtered_rules = [rule for rule in rules_fp if rule[3] >= min_confidence]

print("Frequent Itemsets (min_support >= {:.2f}):".format(min_support))
print(filtered_patterns)
print("\nAssociation Rules (min_confidence >= {:.2f}):".format(min_confidence))
for rule in filtered_rules:
    antecedent, consequent, support, confidence = rule
    print(f"{antecedent} => {consequent} (Support: {support:.2f}, Confidence: {confidence:.2f})")
```

# RESULT

Apriori

minsup = 0.25

minconf = 0.5

Association Rules:

```
('Previous_Attempts',) => ('Social Isolation',) (Confidence: 0.54)
('Social Isolation',) => ('Previous_Attempts',) (Confidence: 0.56)
('Previous_Attempts',) => ('Financial Problem',) (Confidence: 0.51)
('Financial Problem',) => ('Previous_Attempts',) (Confidence: 0.50)
('Social Isolation',) => ('Financial Problem',) (Confidence: 0.52)
('Mental_Health_Issue',) => ('Substance_Abuse',) (Confidence: 0.54)
('Abuse History',) => ('Substance_Abuse',) (Confidence: 0.50)
('Social Isolation',) => ('Substance_Abuse',) (Confidence: 0.51)
('Substance_Abuse',) => ('Financial Problem',) (Confidence: 0.54)
('Financial Problem',) => ('Substance_Abuse',) (Confidence: 0.53)
('Abuse History',) => ('Social Isolation',) (Confidence: 0.51)
('Social Isolation',) => ('Abuse History',) (Confidence: 0.52)
('Abuse History',) => ('Financial Problem',) (Confidence: 0.55)
('Financial Problem',) => ('Abuse History',) (Confidence: 0.52)
('Substance_Abuse',) => ('Previous_Attempts',) (Confidence: 0.52)
('Previous_Attempts',) => ('Substance_Abuse',) (Confidence: 0.52)
('Previous_Attempts',) => ('Abuse History',) (Confidence: 0.51)
('Abuse History',) => ('Previous_Attempts',) (Confidence: 0.52)
```

# RESULT

## FP - Growth

minsup = 0.25

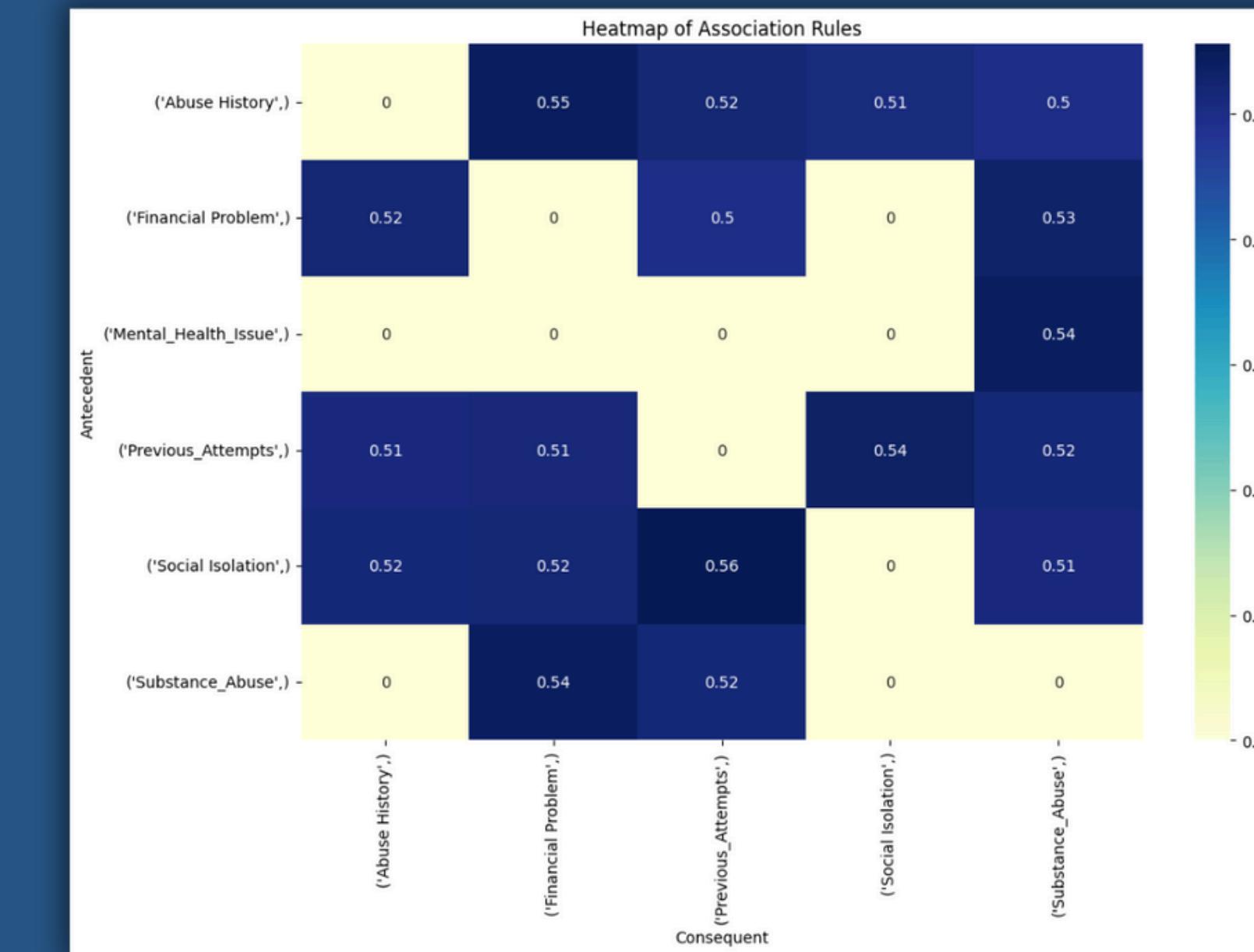
minconf = 0.5

```
Association Rules (min_confidence >= 0.50):
('Previous_Attempts',) => ('Financial Problem',) (Support: 0.27, Confidence: 0.51)
('Financial Problem',) => ('Previous_Attempts',) (Support: 0.27, Confidence: 0.50)
('Previous_Attempts',) => ('Substance_Abuse',) (Support: 0.27, Confidence: 0.52)
('Substance_Abuse',) => ('Previous_Attempts',) (Support: 0.27, Confidence: 0.52)
('Social Isolation',) => ('Financial Problem',) (Support: 0.26, Confidence: 0.52)
('Social Isolation',) => ('Previous_Attempts',) (Support: 0.28, Confidence: 0.56)
('Previous_Attempts',) => ('Social Isolation',) (Support: 0.28, Confidence: 0.54)
('Social Isolation',) => ('Substance_Abuse',) (Support: 0.26, Confidence: 0.51)
('Social Isolation',) => ('Abuse History',) (Support: 0.26, Confidence: 0.52)
('Abuse History',) => ('Social Isolation',) (Support: 0.26, Confidence: 0.51)
('Mental_Health_Issue',) => ('Substance_Abuse',) (Support: 0.25, Confidence: 0.54)
('Substance_Abuse',) => ('Financial Problem',) (Support: 0.28, Confidence: 0.54)
('Financial Problem',) => ('Substance_Abuse',) (Support: 0.28, Confidence: 0.53)
('Abuse History',) => ('Financial Problem',) (Support: 0.28, Confidence: 0.55)
('Financial Problem',) => ('Abuse History',) (Support: 0.28, Confidence: 0.52)
('Abuse History',) => ('Substance_Abuse',) (Support: 0.25, Confidence: 0.50)
('Abuse History',) => ('Previous_Attempts',) (Support: 0.26, Confidence: 0.52)
('Previous_Attempts',) => ('Abuse History',) (Support: 0.26, Confidence: 0.51)
```

# VISUALIZATION

## Heatmap

ASSOCIATION RULES



2024

Màu sắc trên heatmap phản ánh độ tin cậy (confidence) của từng luật kết hợp.

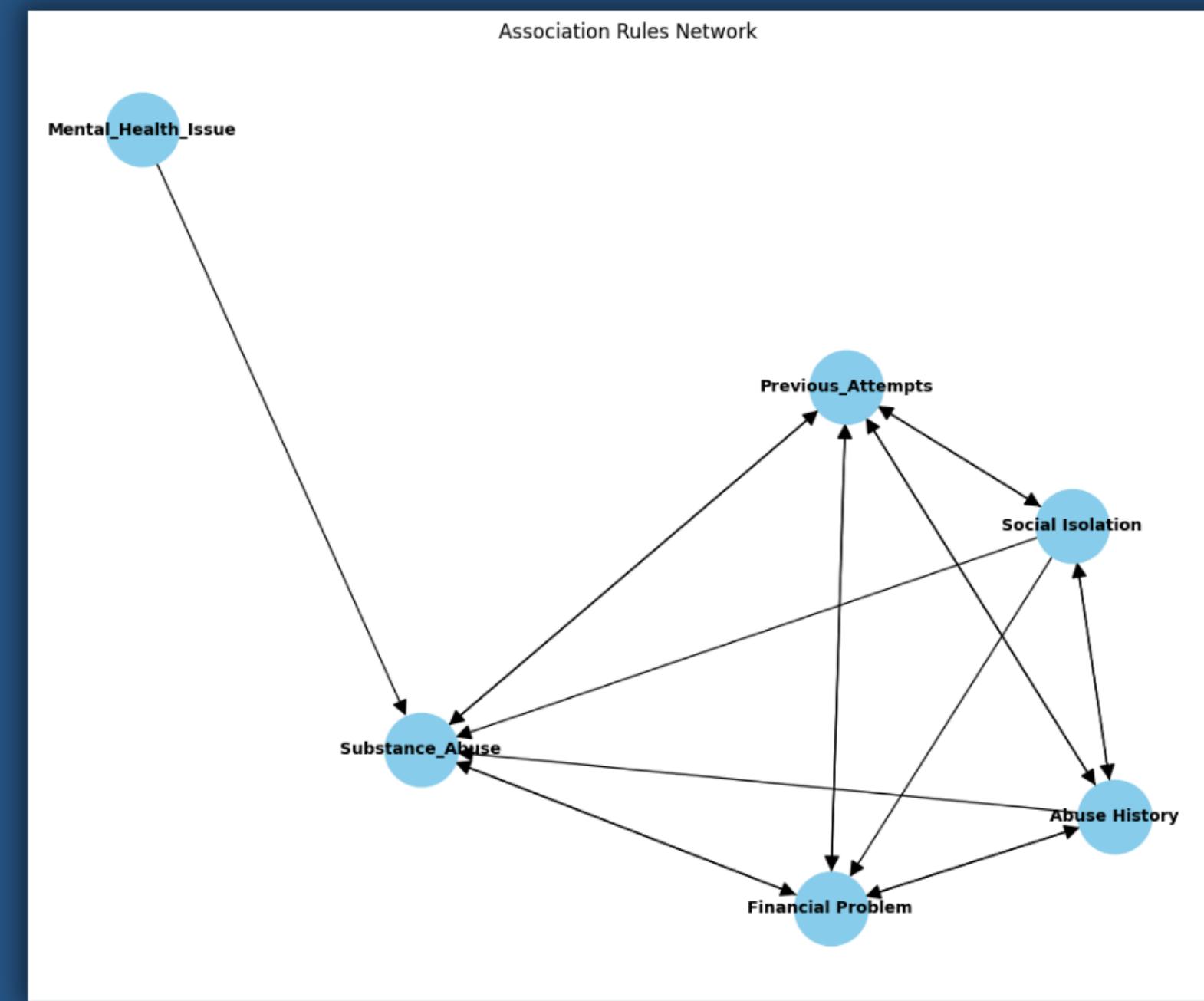
**Màu xanh đậm:** Giá trị confidence cao.

**Màu vàng nhạt:** Giá trị confidence thấp.

# VISUALIZATION

## Association Rules Network

ASSOCIATION RULES



2024

Mỗi nút (node) đại diện cho một yếu tố nguy cơ

Các mũi tên (edges) biểu thị mối quan hệ giữa hai yếu tố



## [ CHƯƠNG TRÌNH PHÒNG NGỪA TỰ TỬ ]

- PATH
- LIVE LIFE

## [ TƯ VẤN & HỖ TRỢ TÂM LÝ ]

Cung cấp dịch vụ tư vấn tâm lý miễn phí hoặc dễ tiếp cận

## [ GIA ĐÌNH ]

Trang bị cho cha mẹ kỹ năng phát hiện và can thiệp sớm

# DEAL WITH SUICIDE

# THANK YOU

