

# DATA MINING PROJECT



Palestine Technical University – Kadoorie  
College of Engineering and Technology  
Department of Computer Systems Engineering

Project title:

## **CUSTOMER SEGMENTATION CLASSIFICATION**

By:

Ruaa Qashoo – 201911196

Instructor:

Dr. Anas Melhem

1 January 2023

## INTRODUCTION

### ❖ PROBLEM DEFINITION:

An automobile company has plans to enter new markets with their existing products (P1, P2, P3, P4, and P5). After intensive market research, they've deduced that the behavior of the new market is similar to their existing market.

In their existing market, the sales team has classified all customers into 4 segments (A, B, C, D). Then, they performed segmented outreach and communication for a different segment of customers. This strategy has worked exceptionally well for them. They plan to use the same strategy for the new markets and have identified 2627 new potential customers.

The goal of this project is to study and predict the right group of new customers for an automotive company, so the company can adopt the specific proven marketing strategy to each of them and be more successful in the business.

I decided to use Weka to analyze the problem and then solve it.

## dataset description:

There is 8068 instances (customers) for training, and 2627 instance (new potential customers) for testing.

This dataset has 11 Variables (10 attributes & 1 class/target), divided as follows:

Link of The Dataset: [Customer Segmentation Classification | Kaggle](#)

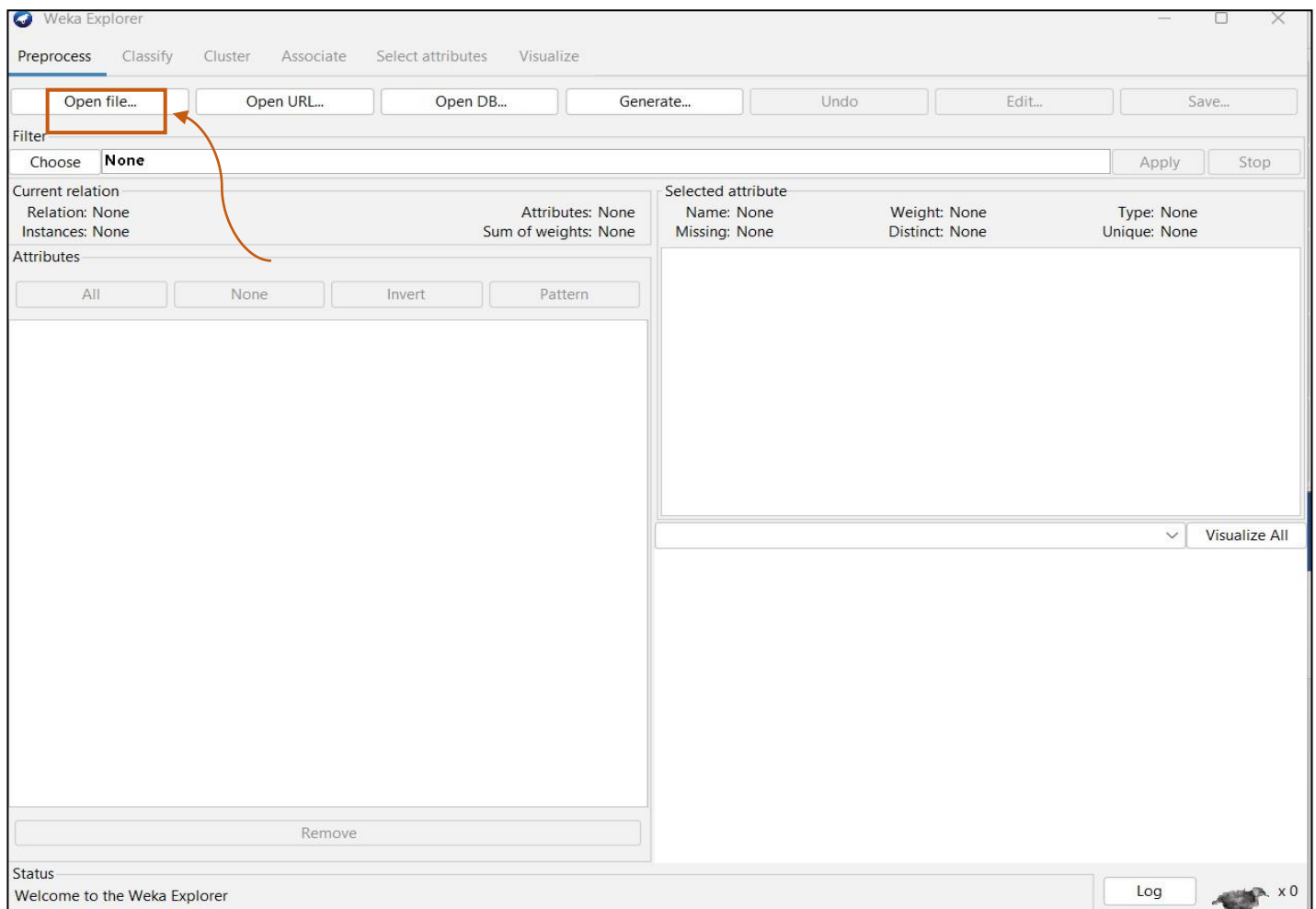
<u>VARIABLES</u>	<u>TYPE</u>	<u>DEFINITION</u>
ID	numeric	Unique ID
GENDER	binary	Gender of the customer
EVER_MARRID	binary	Marital status of the customer
AGE	numeric	Age of the customer
GRADUATED	binary	Is the customer a graduate?
PROFESSION	nominal	Profession of the customer
WORK_EXPERIANCE	numeric	Work Experience in years
SPENDING	Ordinal	Spending score of the customer
FAMILY SIZE	numeric	Number of family members for the customer (including the customer)
VAR_1	nominal	Anonymized Category for the customer
SEGMANT	nominal	(target) Customer Segment of the customer

## Problems with this data & how I solve it

<< Clean/Prepare the data >>

### # Missing Values:

after uploading the data set:



click open file → chose our training dataset → this information about each attribute will appear:

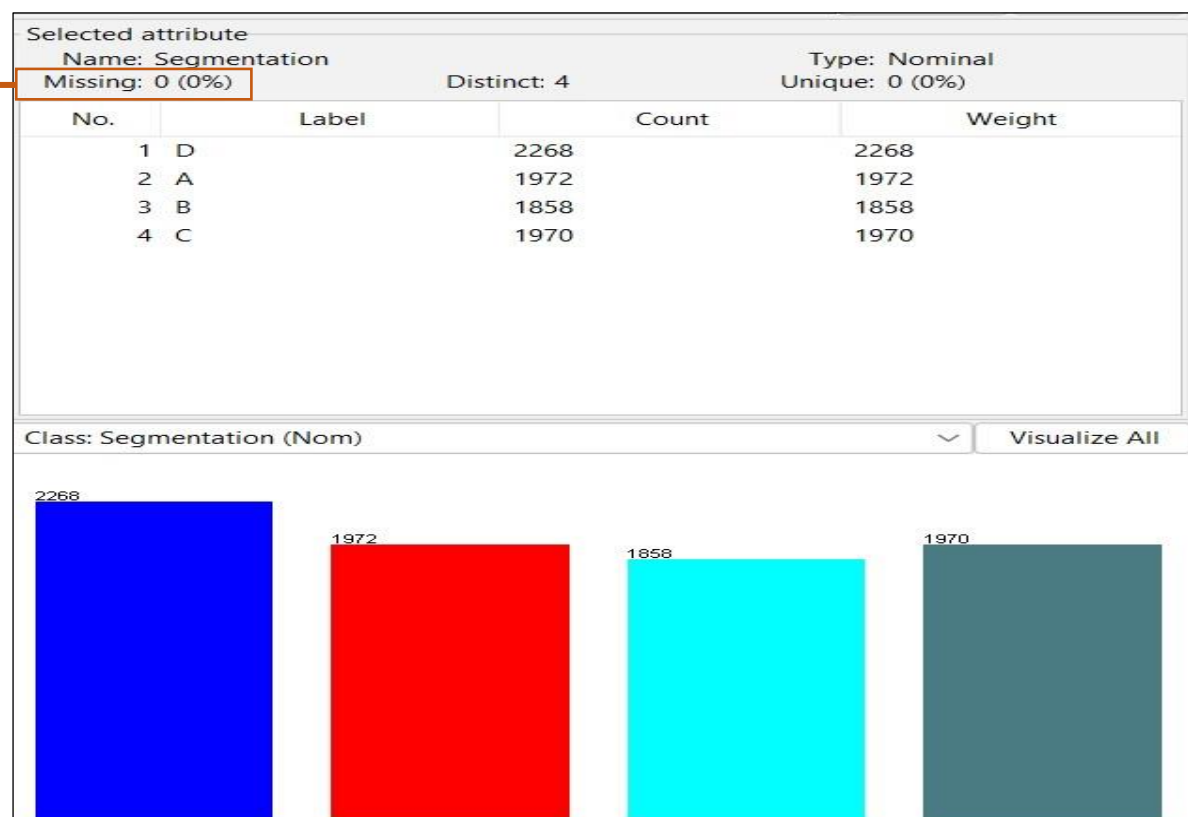
+ NOTE: as we can see there is 4 colors in each chart

■ represent data from class A

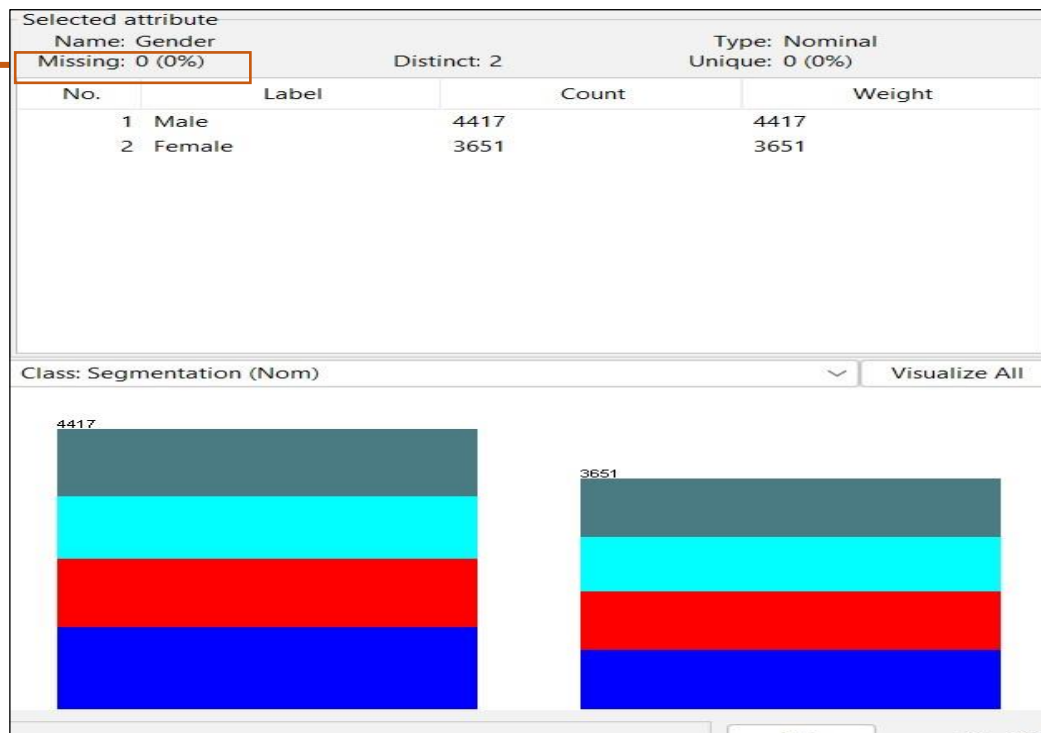
■ represent data from class B

■ represent data from class C

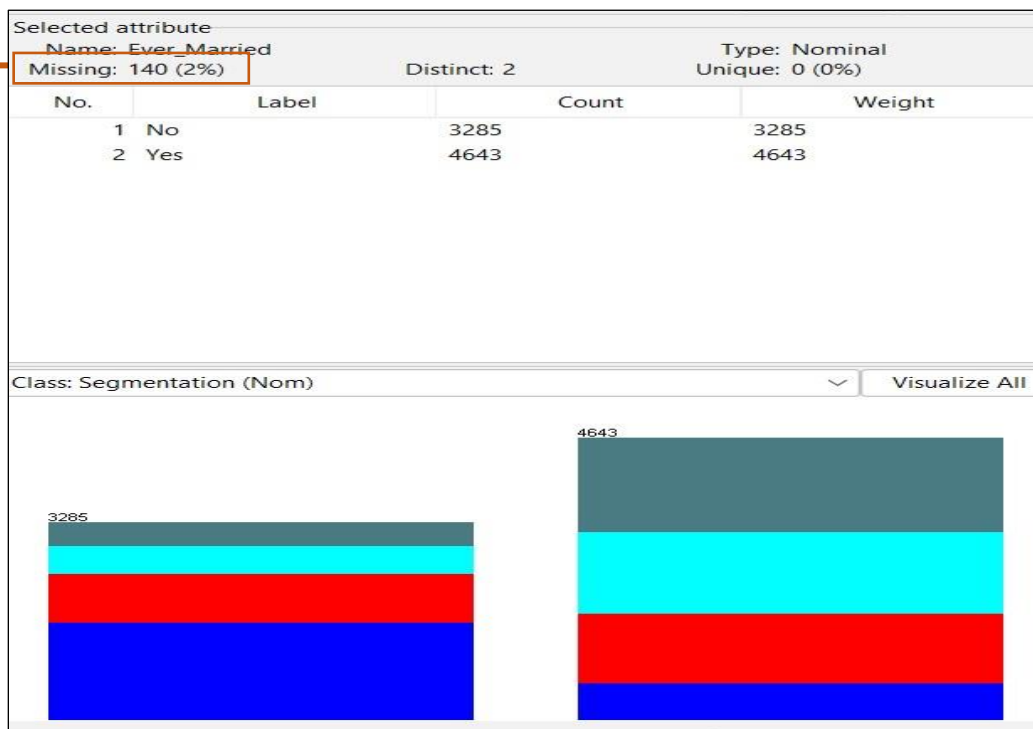
■ represent data from class D



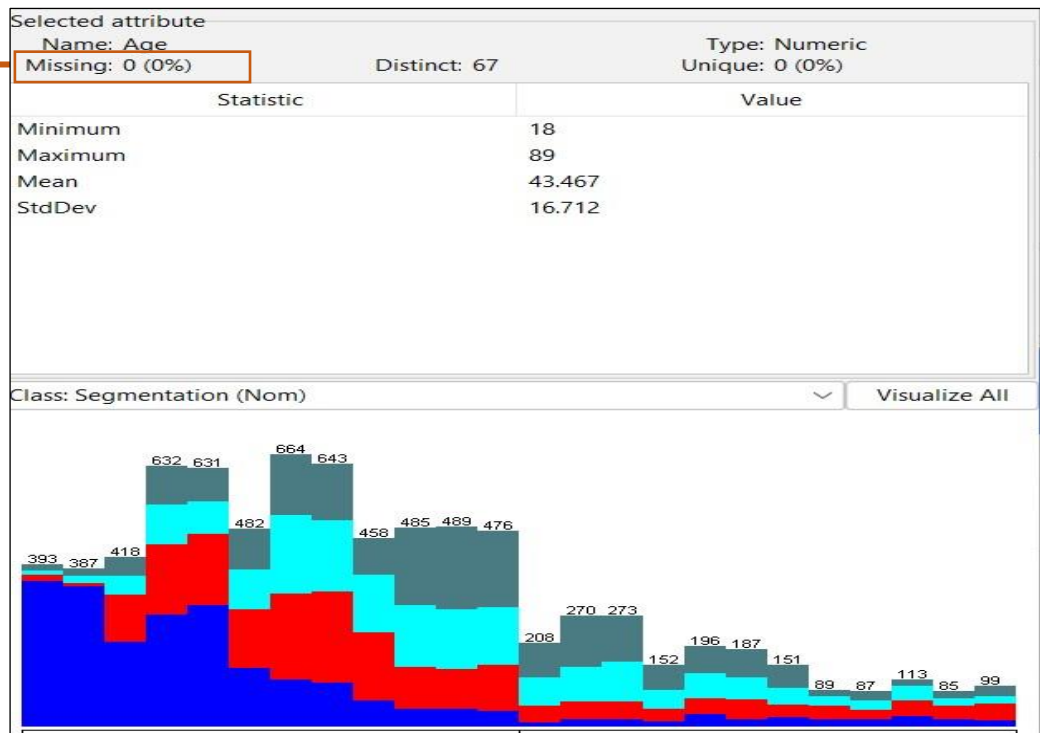
- There is no missing value in Segmentation attribute (class)



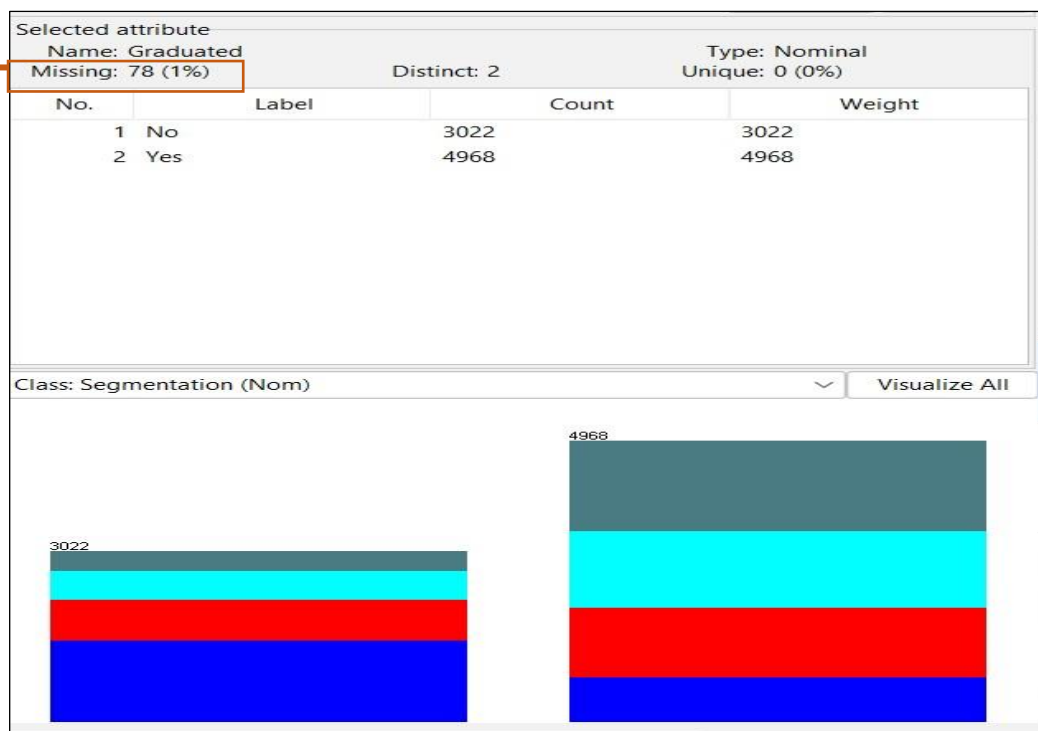
- There is no missing value in Gender attribute



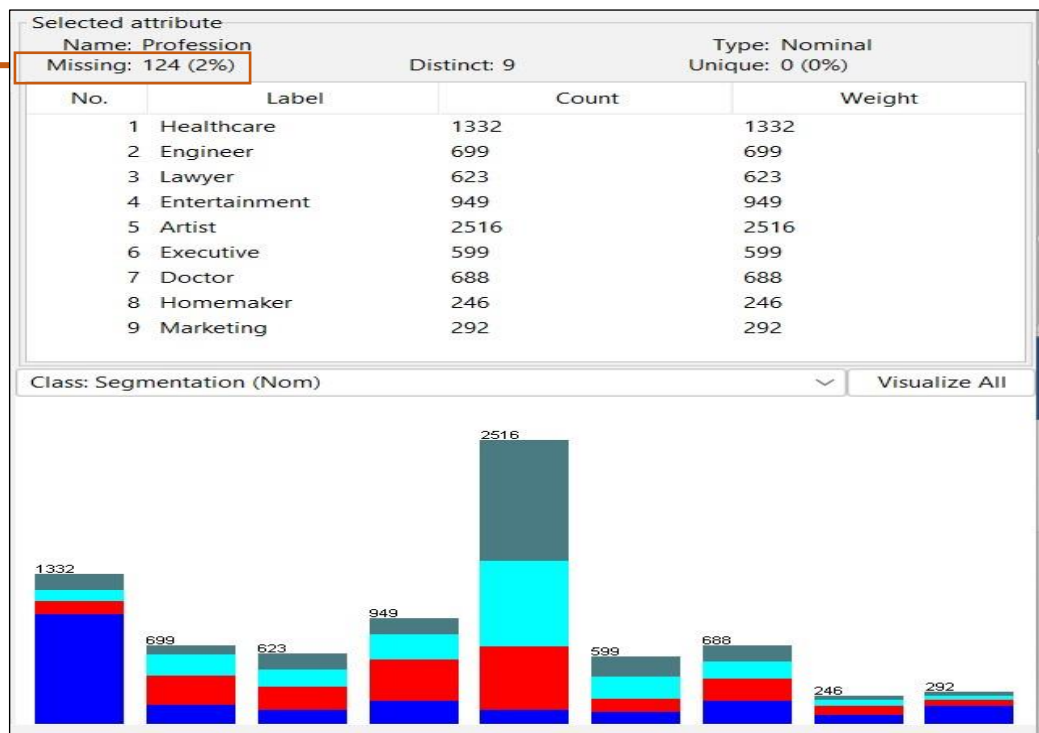
- There is 144 missing value in Ever\_Married attribute



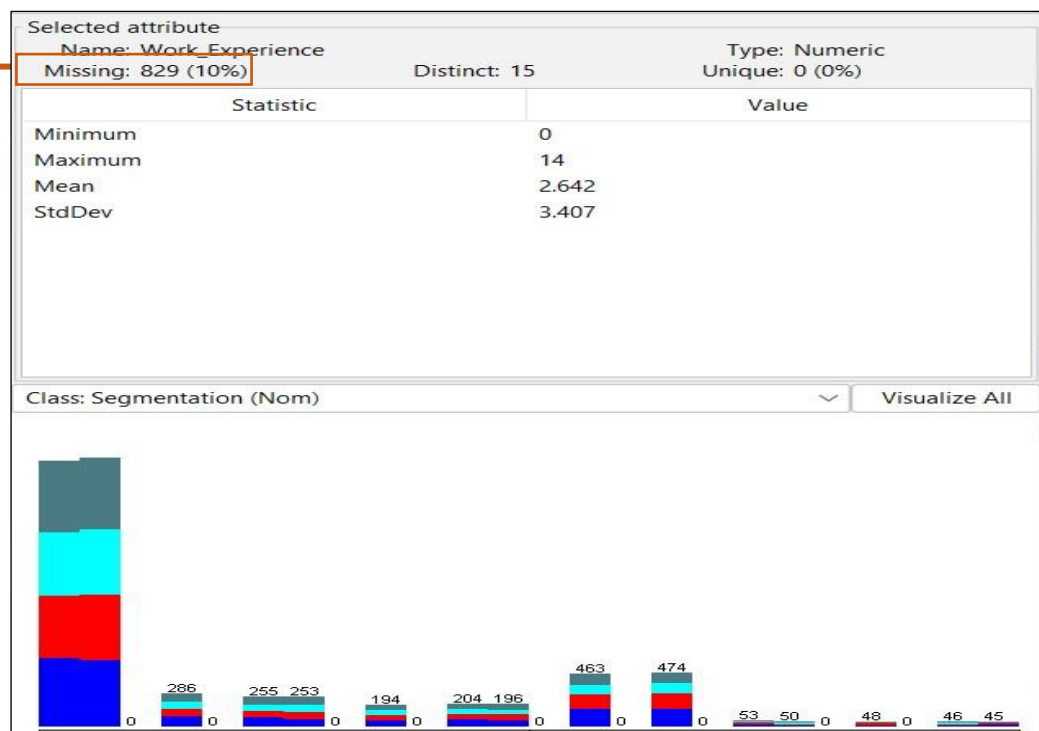
- There is no missing value in Age attribute



- There is 78 missing value in Graduated attribute

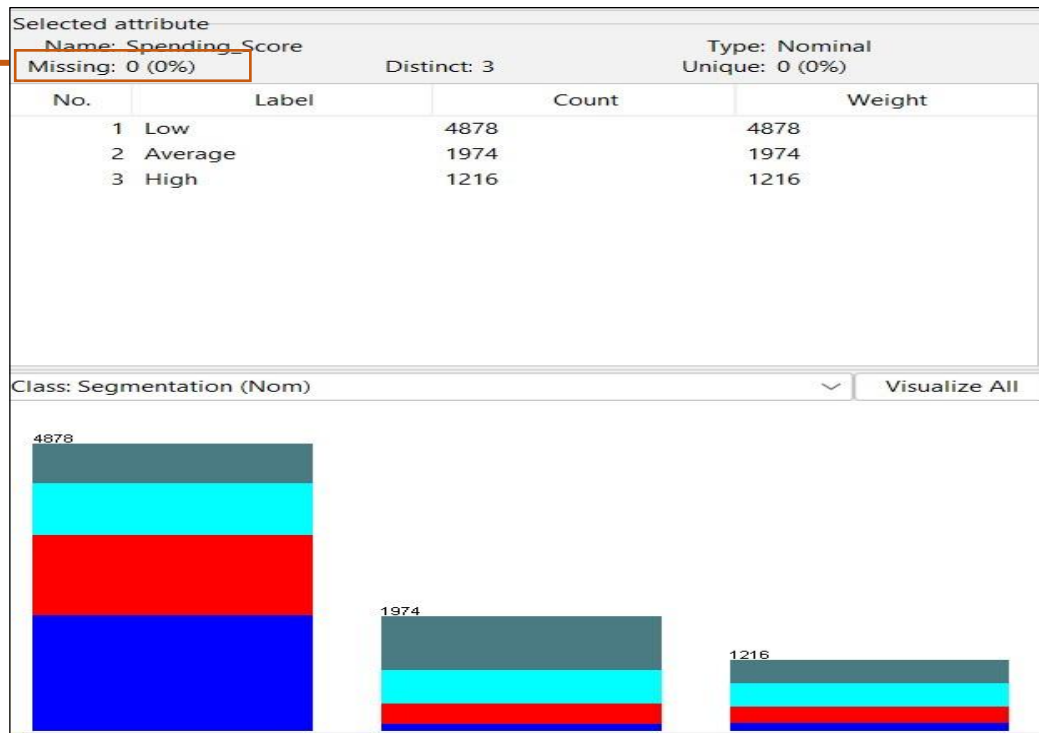


- There is 124 missing value in Profession attribute

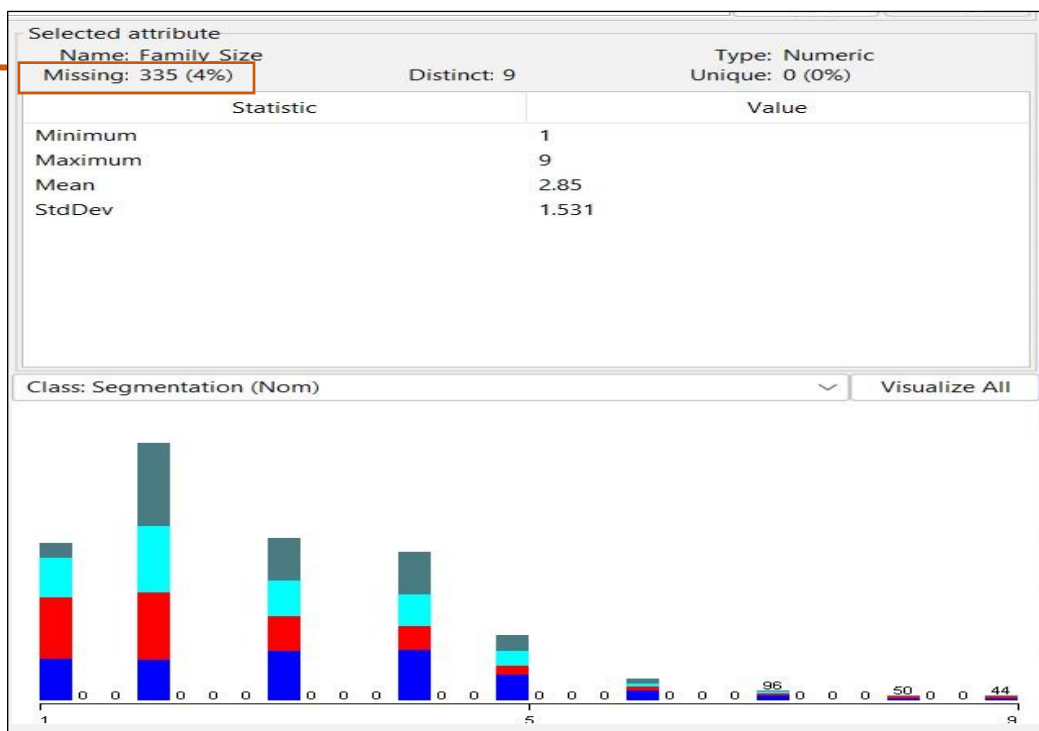


- There is 829 missing value in Work\_Experience attribute

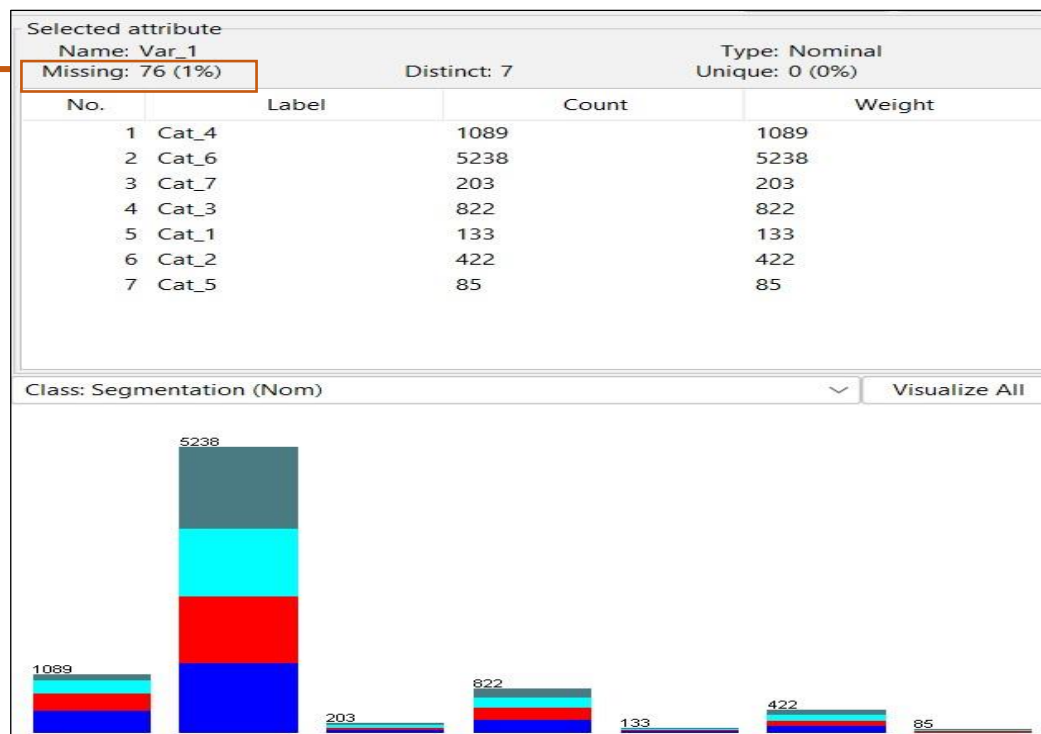




- There is no missing value in Spending attribute



- There is 335 missing value in Family\_Size attribute



- There is 76 missing value in Var\_1 attribute

## #Missing Value processing

As above, we see that data contains some missing values. there are total of 8068 entries, but some columns have less than 8068 entries, which means they have missing values:

Columns " **Family Size** "and " **Work Experience** " have huge number of null values. Both are numeric attribute, so can be replaced with their mean values. **But** because each of these features represents an integer value (1,2, 3.....) and the mean of each respectively is (2.85, 2.64) which is not an integer.

=AVERAGE(G:G)																		
	R	Q	P	O	N	M	L	K	J	I	H	G	F	E	D	C	B	A
								Segmental	Var_1	Family_Siz	Spending_	Work_Exp	Profession	Graduated	Age	Ever_Marr	Gender	ID
								D	Cat_4	4	Low	1	Healthcare	No	22	No	Male	462809
								A	Cat_4	3	Average		Engineer	Yes	38	Yes	Female	462643
								B	Cat_6	1	Low	1	Engineer	Yes	67	Yes	Female	466315
								B	Cat_6	2	High	0	Lawyer	Yes	67	Yes	Male	461735
								A	Cat_6	6	High		Entertainm	Yes	40	Yes	Female	462669
								C	Cat_6	2	Average	0	Artist	No	56	Yes	Male	461319
								C	Cat_6	3	Low	1	Healthcare	Yes	32	No	Male	460156
								D	Cat_6	3	Low	1	Healthcare	Yes	33	No	Female	464347
								D	Cat_7	3	Low	0	Engineer	Yes	61	Yes	Female	465015
								C	Cat_6	4	Average	1	Artist	Yes	55	Yes	Female	465176
								A	Cat_6	3	Low	1	Engineer	Yes	26	No	Female	464041
								D	Cat_4	4	Low	4	Healthcare	No	19	No	Male	464942
								D	Cat_3		Low	0	Executive	No	19	No	Female	461230
								A	Cat_6	1	Low		Lawyer	No	70	Yes	Male	459573
								B	Cat_3	1	Low	0	Doctor	No	58	Yes	Female	460849
								C	Cat_1	2	Low	1	Healthcare	No	41	No	Female	460563
								D	Cat_3	5	Low	9	Homemake	No	32	No	Female	466865
								B	Cat_6	6	Low	1	Healthcare	No	31	No	Male	461644
								B	Cat_6	4	Average	1	Entertainm	Yes	58	Yes	Male	466772
								C	Cat_6	1	High	0	Artist	Yes	79	Yes	Female	464291
								A	Cat_3	1	Low	12	Homemake	Yes	49	Yes	Male	466084
								D	Cat_6	4	Low	3	Healthcare	No	18	No	Female	459675
								A	Cat_3	2	Low	13	Artist	Yes	33	Yes	Male	465602
								B	Cat_6	2	Low	5	Artist	Yes	36	No	Female	459168
								B	Cat_3	3	Average	1	Executive	No	58		Female	461021
								C	Cat_6	3	Average	1	Artist	No	56	Yes	Male	465083
								A	Cat_4	8	Low	9	Healthcare	No	31	No	Male	467604
								C	Cat_6	3	Average	1	Artist	Yes	49	Yes	Male	459717

# Also replacing missing with the mean value may affect outliers.

**So**, I decided to replace the missing values in these columns with the median value of each one:

- Median of family\_size = 3
- Median of work\_Experience = 1

=MEDIAN(G:G) ←																			
S	R	Q	P	O	N	M	L	K	J	I	H	G	F	E	D	C	B	A	
								Segmental	Var_1	Family_Siz	Spending_	Work_Exp	Profession	Graduated	Age	Ever_Marr	Gender	ID	1
								D	Cat_4	4	Low	1	Healthcare	No	22	No	Male	462809	2
								A	Cat_4	3	Average		Engineer	Yes	38	Yes	Female	462643	3
median of family_Size:			3					B	Cat_6	1	Low	1	Engineer	Yes	67	Yes	Female	466315	4
median of work_Experience:			1					B	Cat_6	2	High	0	Lawyer	Yes	67	Yes	Male	461735	5
								A	Cat_6	6	High		Entertainm	Yes	40	Yes	Female	462669	6
								C	Cat_6	2	Average	0	Artist	No	56	Yes	Male	461319	7
								C	Cat_6	3	Low	1	Healthcare	Yes	32	No	Male	460156	8
								D	Cat_6	3	Low	1	Healthcare	Yes	33	No	Female	464347	9
								D	Cat_7	3	Low	0	Engineer	Yes	61	Yes	Female	465015	10
								C	Cat_6	4	Average	1	Artist	Yes	55	Yes	Female	465176	11
								A	Cat_6	3	Low	1	Engineer	Yes	26	No	Female	464041	12
								D	Cat_4	4	Low	4	Healthcare	No	19	No	Male	464942	13
								D	Cat_3		Low	0	Executive	No	19	No	Female	461230	14
								A	Cat_6	1	Low		Lawyer	No	70	Yes	Male	459573	15
								B	Cat_3	1	Low	0	Doctor	No	58	Yes	Female	460849	16
								C	Cat_1	2	Low	1	Healthcare	No	41	No	Female	460563	17
								D	Cat_3	5	Low	9	Homemake	No	32	No	Female	466865	18
								B	Cat_6	6	Low	1	Healthcare	No	31	No	Male	461644	19
								B	Cat_6	4	Average	1	Entertainm	Yes	58	Yes	Male	466772	20
								C	Cat_6	1	High	0	Artist	Yes	79	Yes	Female	464291	21
								A	Cat_3	1	Low	12	Homemake	Yes	49	Yes	Male	466084	22
								D	Cat_6	4	Low	3	Healthcare	No	18	No	Female	459675	23
								A	Cat_3	2	Low	13	Artist	Yes	33	Yes	Male	465602	24
								B	Cat_6	2	Low	5	Artist	Yes	36	No	Female	459168	25
								B	Cat_3	3	Average	1	Executive	No	58		Female	461021	26
								C	Cat_6	3	Average	1	Artist	No	56	Yes	Male	465083	27
								A	Cat_4	8	Low	9	Healthcare	No	31	No	Male	467604	28
								C	Cat_6	3	Average	1	Artist	Yes	49	Yes	Male	459717	29

## Replacing them in weka:

open the training dataset → click Edit → write click on attribute which needed to edit → then choose set missing value to and enter the median value (that we calculated above):

The screenshot shows the Weka Explorer window. The 'Edit...' button in the top toolbar is highlighted with an orange box. Below the toolbar, the 'Filter' section shows 'Choose' set to 'None'. The 'Current relation' section shows 'Relation: Train' and 'Instances: 8068'. The 'Attributes' section shows a list of attributes with checkboxes. The 'Selected attribute' section shows 'Name: ID' and 'Type: Numeric'. The 'Statistic' table shows the following values:

Statistic	Value
Minimum	458982
Maximum	467974
Mean	463479.215
StdDev	2595.381

Viewer

Relation: Train - Copy.arff

No.	1: ID	2: Gender	3: Ever_Married	4: Age	5: Graduated	6: Profession	7: Work_Experience	8: Spending_Score	9: Family_Size	10: Var_1	11: Segmentation
	Numeric	Nominal	Nominal	Numeric	Nominal	Nominal	Numeric	Nominal	Numeric	Nominal	Nominal
1	46280...	Male	No	22.0	No	Healthcare	1.0	Low	4.0	Cat_4	D
2	46264...	Female	Yes	38.0	Yes	Engineer	Average		3.0	Cat_4	A
3	46631...	Female	Yes	67.0	Yes	Engineer	1.0	Low	1.0	Cat_6	B
4	46173...	Male	Yes	67.0	Yes	Lawyer	0.0	High	2.0	Cat_6	B
5	46266...	Female	Yes	40.0	Yes	Entertainme...	High		6.0	Cat_6	A
6	46131...	Male	Yes	56.0	No	Artist	0.0	Average	2.0	Cat_6	C
7	46015...	Male	No	32.0	Yes	Healthcare	1.0	Low	3.0	Cat_6	C
8	46434...	Female	No	33.0	Yes	Healthcare	1.0	Low	3.0	Cat_6	D
9	46501...	Female	Yes	61.0	Yes	Engineer	0.0	Low	3.0	Cat_7	D
10	46517...	Female	Yes	55.0	Yes	Artist	1.0	Average	4.0	Cat_6	C
11	46404...	Female	No	26.0	Yes	Engineer	1.0	Low	3.0	Cat_6	A
12	46494...	Male	No	19.0	No	Healthcare	4.0	Low	4.0	Cat_4	D
13	46123...	Female	No	19.0	No	Executive	0.0	Low	Cat_3	D	
14	45957...	Male	Yes	70.0	No	Lawyer	Low		1.0	Cat_6	A
15	46084...	Female	Yes	58.0	No	Doctor	0.0	Low	1.0	Cat_3	B
16	46056...	Female	No	41.0	No	Healthcare	1.0	Low	2.0	Cat_1	C
17	46686...	Female	No	32.0	No	Homemaker	9.0	Low	5.0	Cat_3	D
18	46164...	Male	No	31.0	No	Healthcare	1.0	Low	6.0	Cat_6	B
19	46677...	Male	Yes	58.0	Yes	Entertainme...	1.0	Average	4.0	Cat_6	B
20	46429...	Female	Yes	79.0	Yes	Artist	0.0	High	1.0	Cat_6	C
21	46608...	Male	Yes	49.0	Yes	Homemaker	12.0	Low	1.0	Cat_3	A
22	45967...	Female	No	18.0	No	Healthcare	3.0	Low	4.0	Cat_6	D
23	46560...	Male	Yes	33.0	Yes	Artist	13.0	Low	2.0	Cat_3	A
24	45916...	Female	No	36.0	Yes	Artist	5.0	Low	2.0	Cat_6	B

Add instance

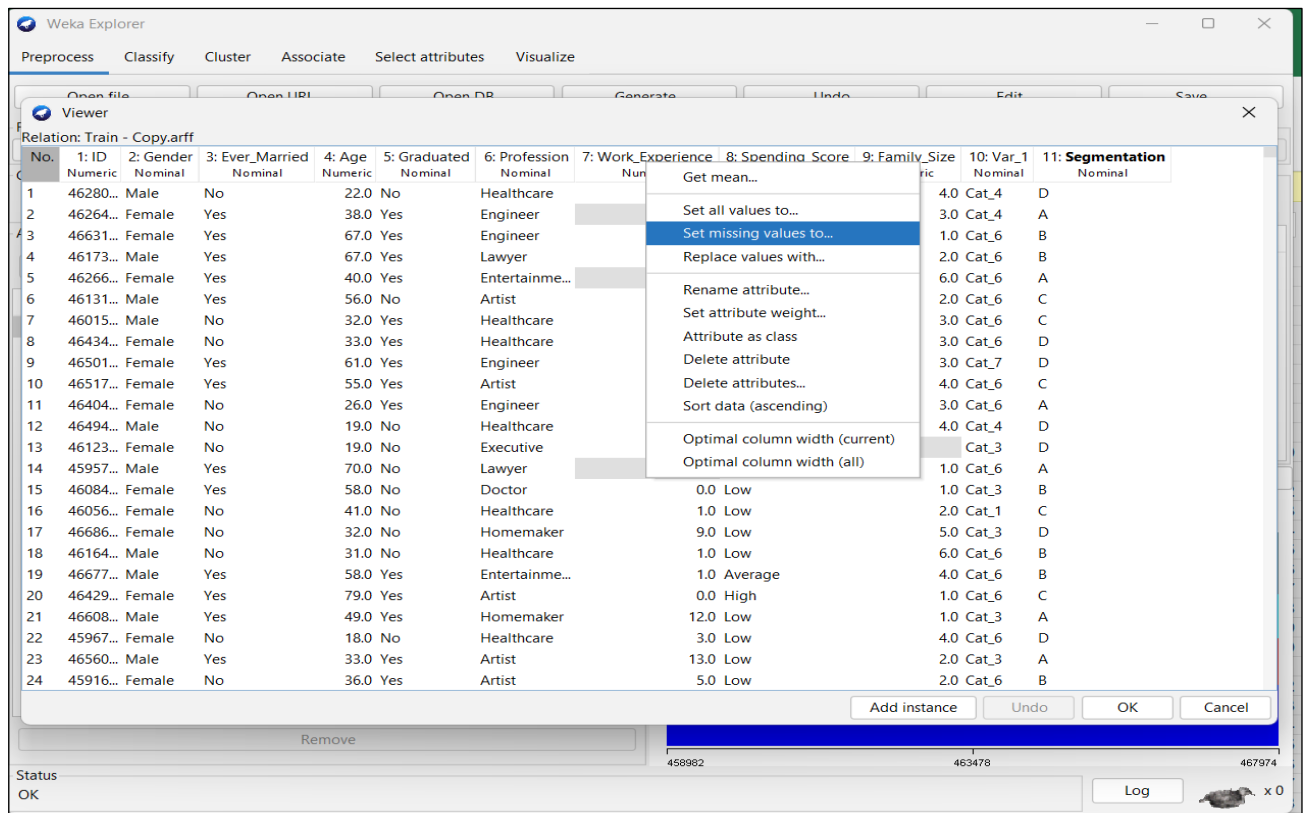
Undo

OK

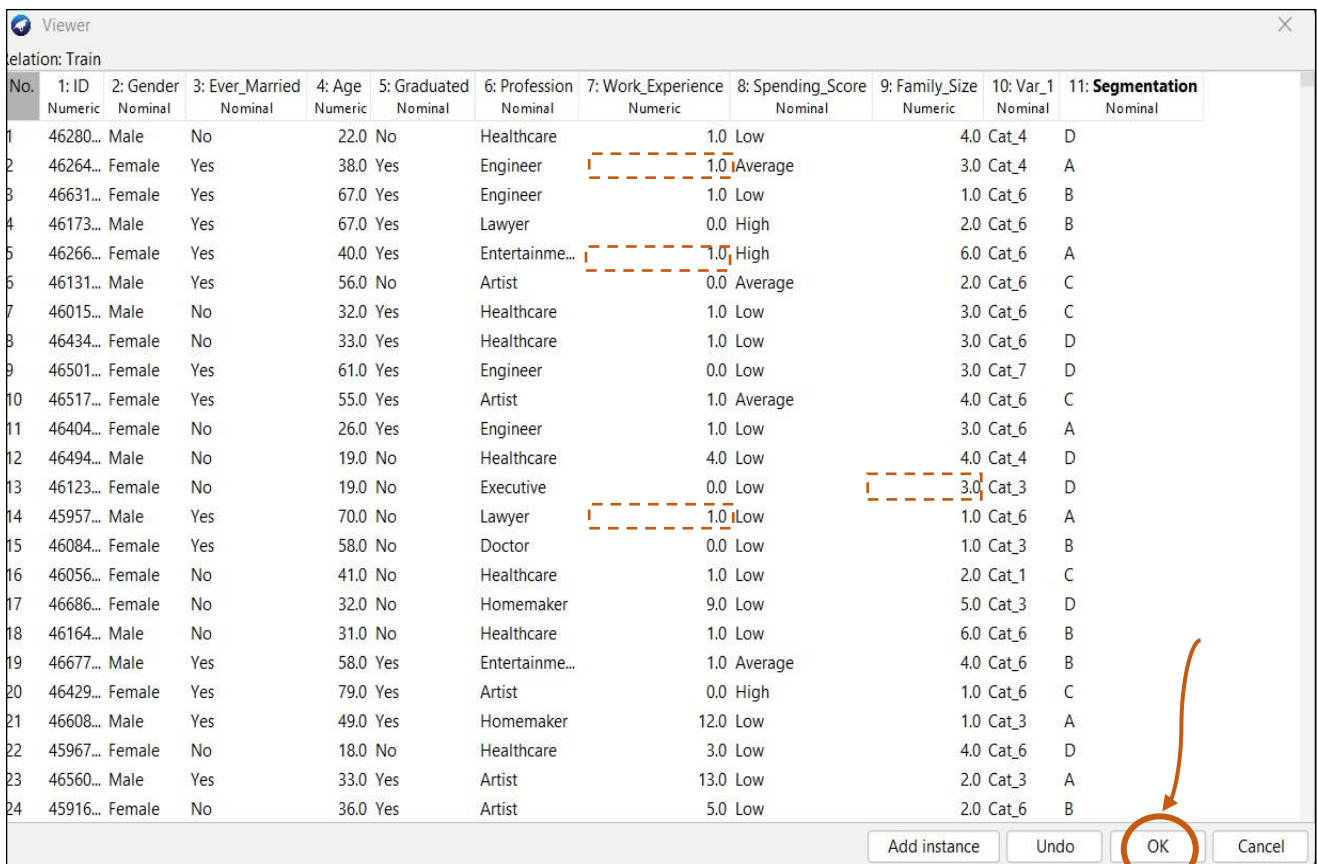
Cancel

We can see some of  
messing value





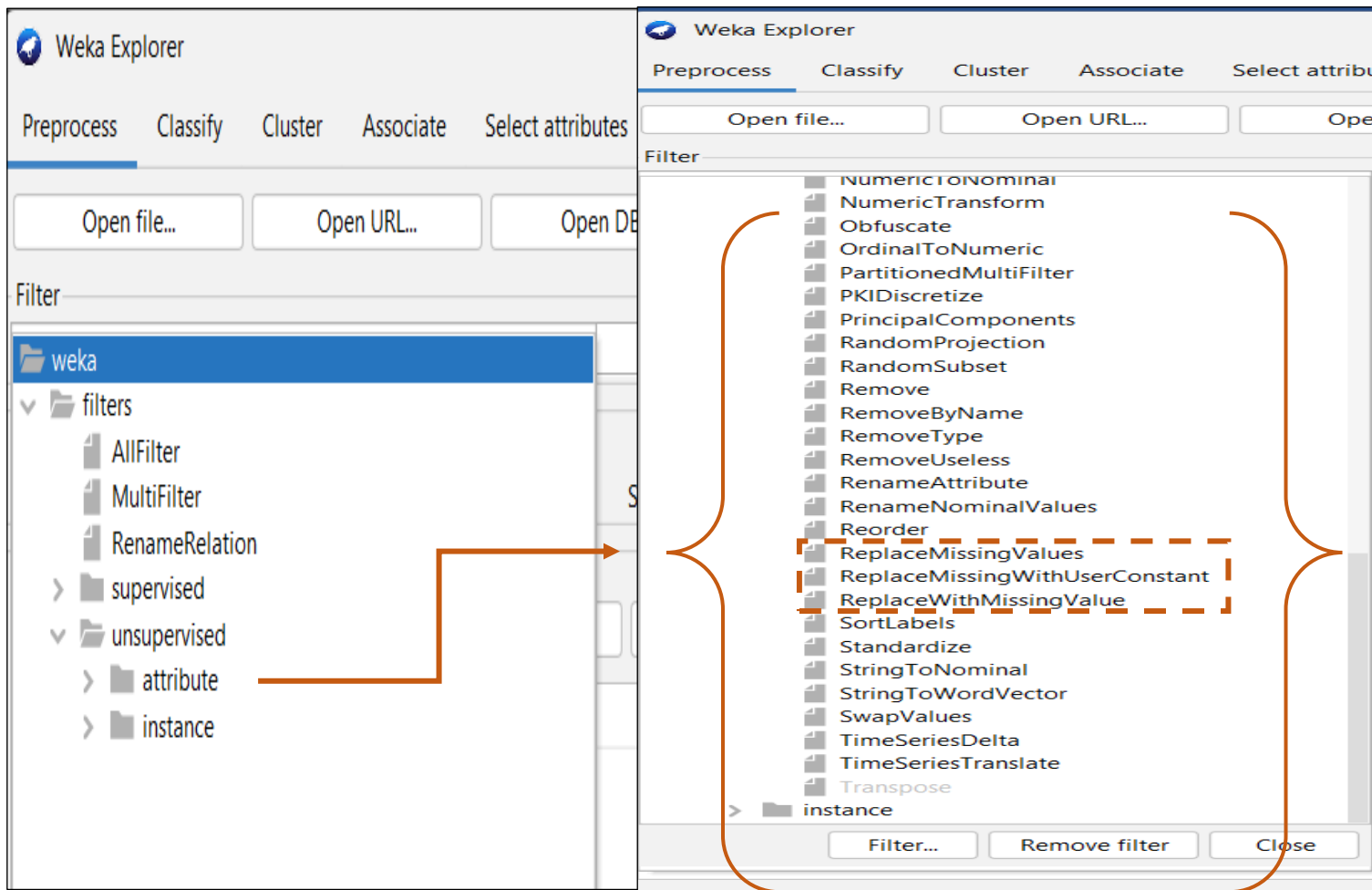
After replacing them we click "OK" to save the changes we made



Columns " Ever\_Married ", " Graduated ", " Profession ", " Var\_1 " have null values.

All of them either nominal or binary, so can be replaced with their **Mode** values.

**Note:** in weka there is 3 replacing filter I chose " ReplaceMissingValues " because this filter allow replacing the missing with Mode value.



Filter

Choose **ReplaceMissingValues** Apply Stop

Selected attribute  
Name: Gender  
Missing: 0 (0%)

Selected attribute  
Name: Ever\_Married  
Missing: 0 (0%)

Selected attribute  
Name: Age  
Missing: 0 (0%)

Selected attribute  
Name: Graduated  
Missing: 0 (0%)

Selected attribute  
Name: Profession  
Missing: 0 (0%)

Selected attribute  
Name: Work\_Experience  
Missing: 0 (0%)

Selected attribute  
Name: Spending\_Score  
Missing: 0 (0%)

Selected attribute  
Name: Family\_Size  
Missing: 0 (0%)

Selected attribute  
Name: Var\_1  
Missing: 0 (0%)

Now as we can see we replace all missing values  
(**No missing values now**)



## Outlier/ extreme values:

Now we need to find if there are any outlier/ extreme values in our data set:

A common rule for identifying suspected outliers is to single out values falling at least  $1.5 \times \text{IQR}$  above the third quartile or below the first quartile.

### # Finding them in weka:

**\*Note:** First you must process the missing values as above

Filter  $\longrightarrow$  unsupervised  $\longrightarrow$  attribute  $\longrightarrow$  InterquartileRange (IQR)

The screenshot shows the Weka Explorer interface. In the 'Filter' panel, the 'unsupervised' filter category is selected, and the 'InterquartileRange' filter is highlighted. An orange arrow points from the 'InterquartileRange' filter in the list to the 'Information' dialog box. The 'Information' dialog box provides details about the filter:

**NAME**  
weka.filters.unsupervised.attribute.InterquartileRange

**SYNOPSIS**  
A filter for detecting outliers and extreme values based on interquartile ranges. The filter skips the class attribute.

**Outliers:**  
 $Q3 + OF \cdot IQR < x \leq Q3 + EVF \cdot IQR$   
or  
 $Q1 - EVF \cdot IQR \leq x < Q1 - OF \cdot IQR$

**Extreme values:**  
 $x > Q3 + EVF \cdot IQR$   
or  
 $x < Q1 - EVF \cdot IQR$

**Key:**  
Q1 = 25% quartile

The background shows the 'Selected attribute' table for 'Age' with statistics: Minimum (18), Maximum (89), Mean (43.467), and StdDev (16.712).

This close-up shows the 'Filter' panel at the bottom of the Weka Explorer window. The 'Choose' button is active, and the command 'InterquartileRange -R first-last -O 3.0 -E 6.0' is entered in the text field. The 'Apply' button is highlighted with an orange rectangle.

# After we apply this filter 2 new column added to dataset (Outlier , ExtremeValue):

Filter

Choose **InterquartileRange** -R first-last -O 3.0 -E 6.0

Current relation  
Relation: Train-weka.filters.unsupervised.attribute.ReplaceMissin...  
Instances: 8068  
Attributes: 13  
Sum of weights: 8068

Attributes

All None Invert Pattern

No.	Name
1	<input type="checkbox"/> ID
2	<input type="checkbox"/> Gender
3	<input type="checkbox"/> Ever_Married
4	<input type="checkbox"/> Age
5	<input type="checkbox"/> Graduated
6	<input type="checkbox"/> Profession
7	<input type="checkbox"/> Work_Experience
8	<input type="checkbox"/> Spending_Score
9	<input type="checkbox"/> Family_Size
10	<input type="checkbox"/> Var_1
11	<input type="checkbox"/> Segmentation
12	<input checked="" type="checkbox"/> Outlier
13	<input checked="" type="checkbox"/> ExtremeValue

Selected attribute  
Name: Outlier  
Missing: 0 (0%)  
Distinct: 1  
Type: Nominal  
Unique: 0 (0%)

No.	Label	Count	Weight
1	no	8068	8068
2	yes	0	0

Selected attribute  
Name: ExtremeValue  
Missing: 0 (0%)  
Distinct: 1  
Type: Nominal  
Unique: 0 (0%)

No.	Label	Count	Weight
1	no	8068	8068
2	yes	0	0

In our dataset there is no Outlier/Extreme

## Note:

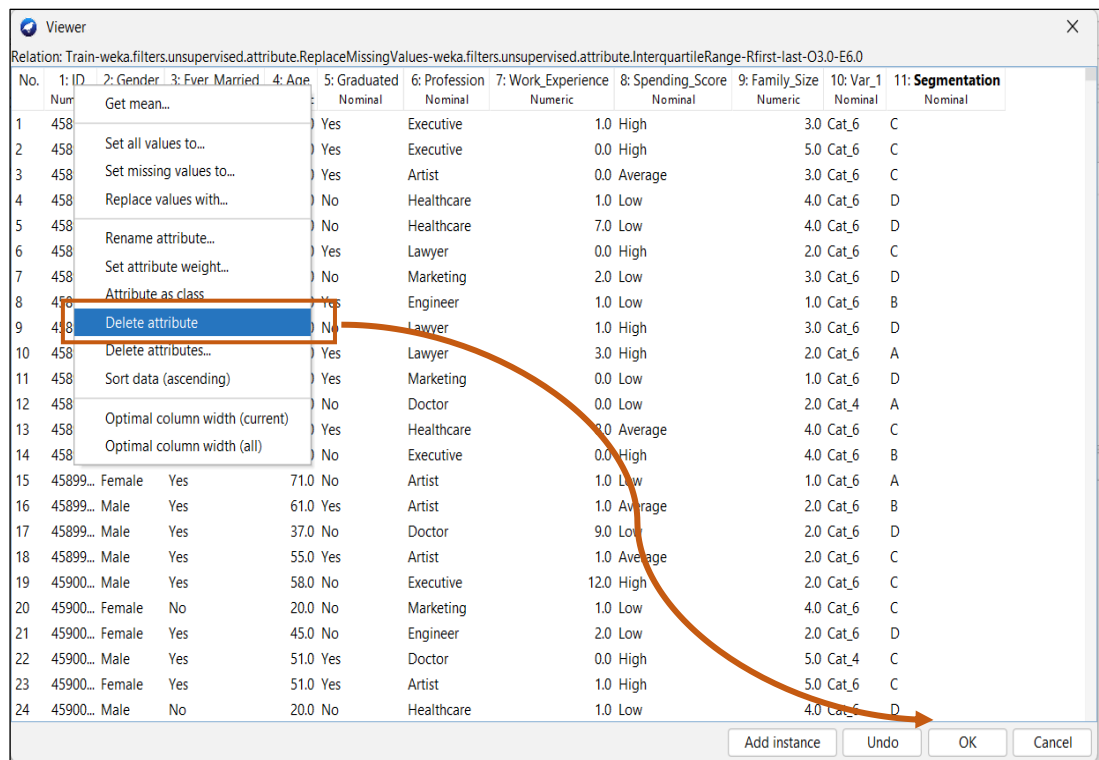
If there were outliers, we could solve them by using this filter which remove all instances that contain the outliers:

Filter → unsupervised → instance → **RemoveWithValue.**

After completion of the process, the variables (Extreme Values, outliers) deleted.

## Note:

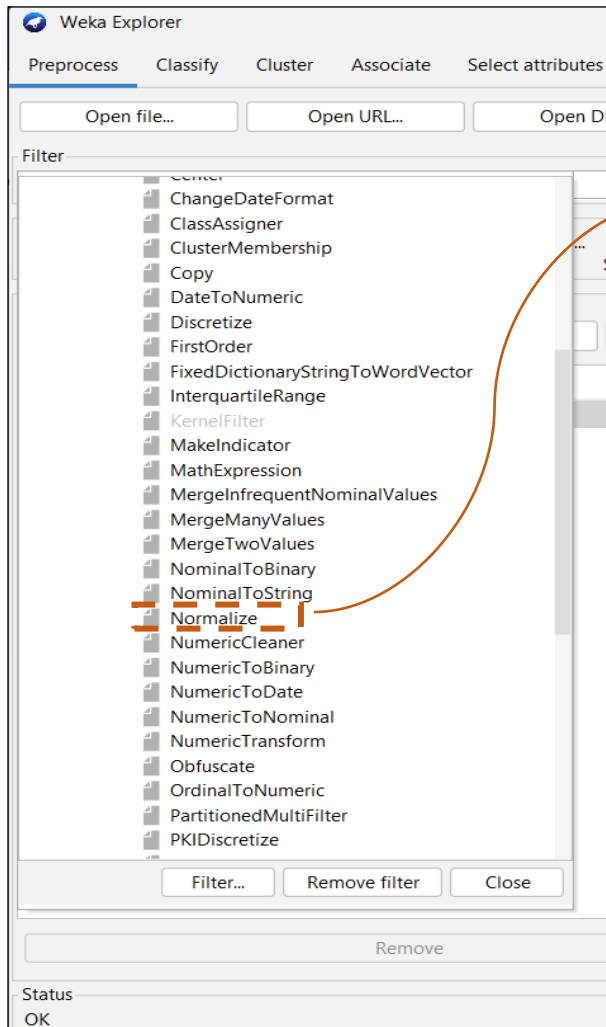
we can Remove column ID as it's not important to the model



## Note:

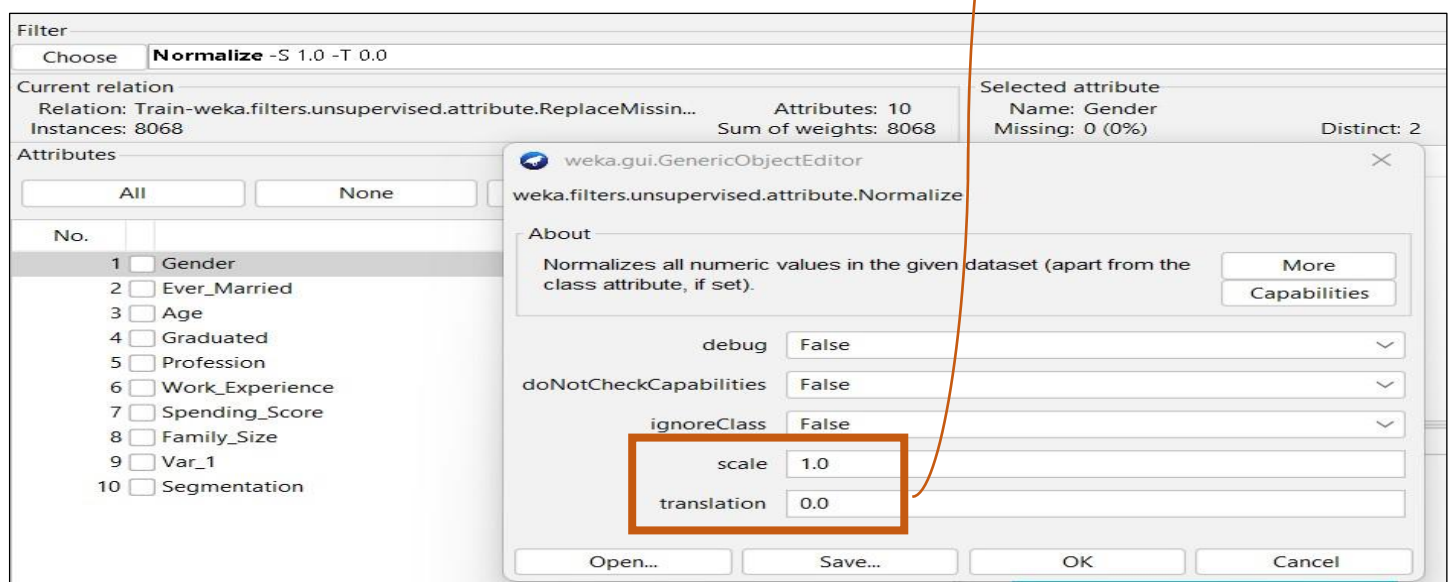
Because we need to make clustering for data, All Numeric attributes should be transformed to a similar scale to be effective:

Filter → unsupervised → instance → Normalize.



Normalizes all numeric values in the given dataset.

The resulting values are in  $[0,1]$  for the data used to compute the normalization.

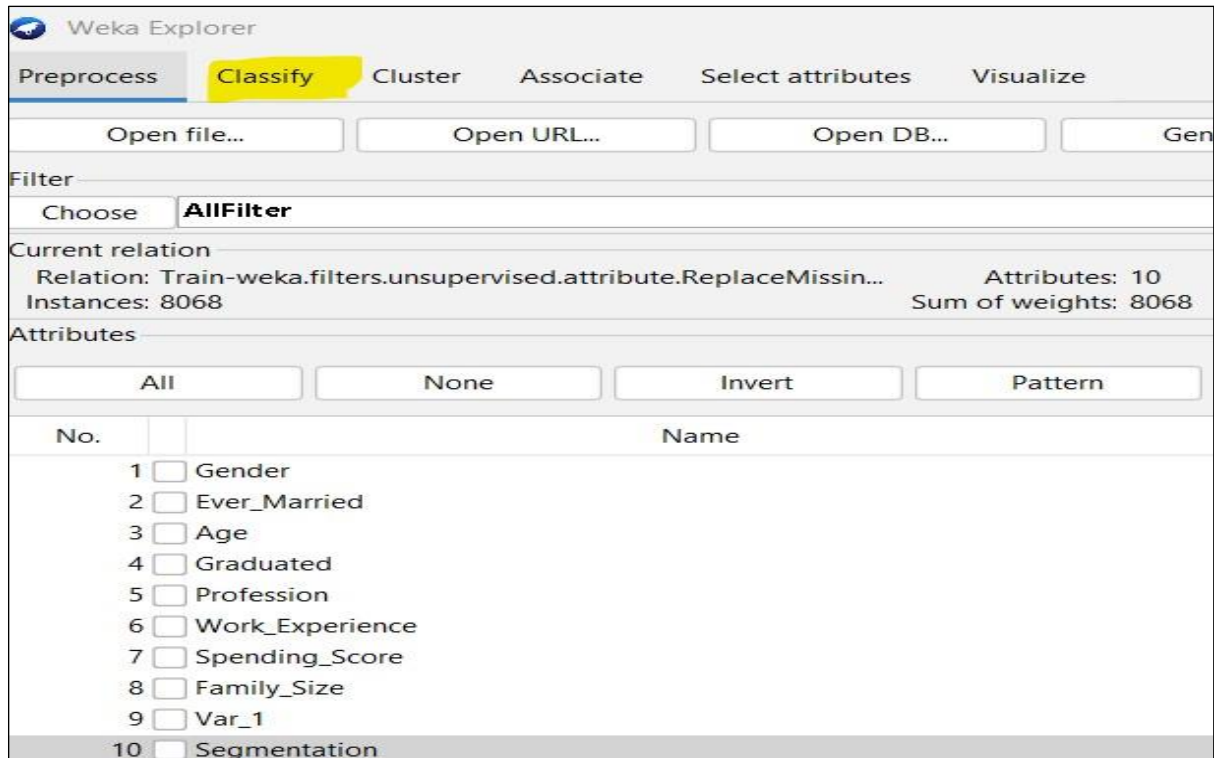


We finish data preparation (fill in missing values and get rid of the extreme values and outliers (if there is) ).

**Now our dataset ready for applying  
data mining model**

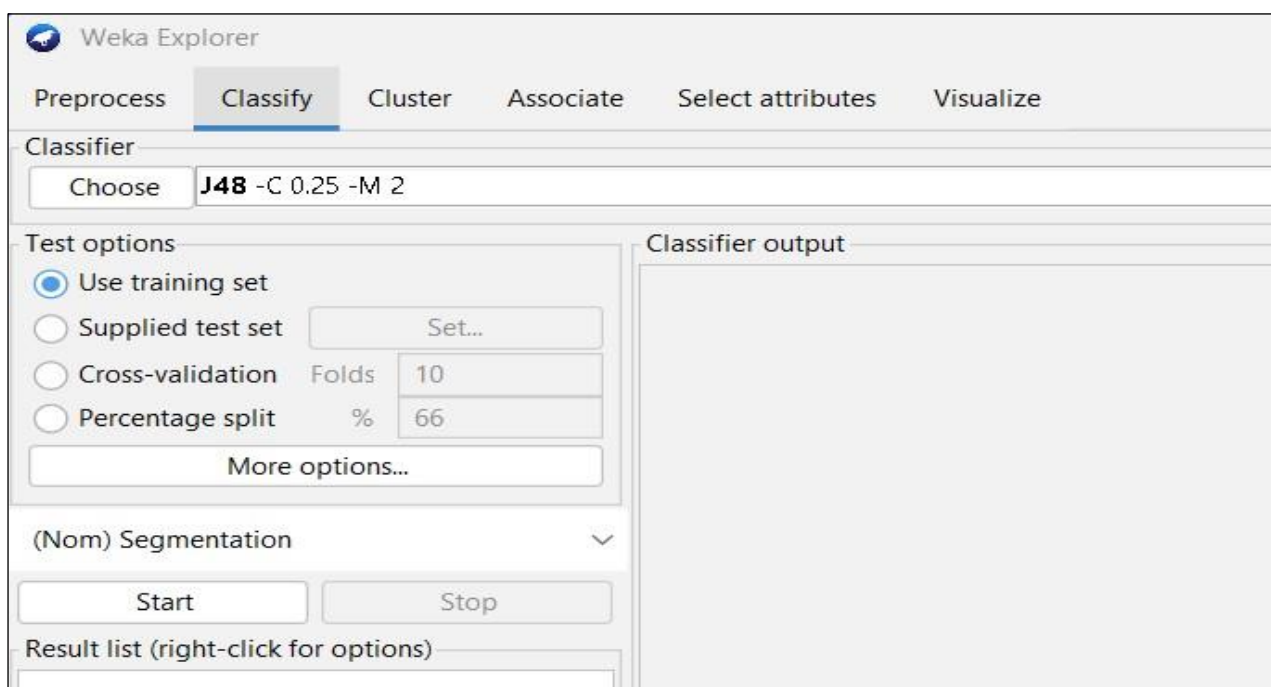
# Data Mining Modeling

## # Classification:



## # Classification using Decision Tree:

Classify → choose → tree → j48



Weka Explorer

Preprocess   **Classify**   Cluster   Associate   Select attributes   Visualize

Classifier

Choose **J48 -C 0.25 -M 2**

Test options

☒ Use training set

☐ Supplied test set   Set...

☐ Cross-validation   Folds   100

☐ Percentage split   %   66

More options...

(Nom) Segmentation   v

Start   Stop

Result list (right-click for options)

21:03:20 - trees.J48

Classifier output

=== Evaluation on training set ===

Time taken to test model on training data: 0.01 seconds

=== Summary ===

Correctly Classified Instances	5537	68.6292 %
Incorrectly Classified Instances	2531	31.3708 %
Kappa statistic	0.5799	
Mean absolute error	0.2193	
Root mean squared error	0.3311	
Relative absolute error	58.5788 %	
Root relative squared error	76.5368 %	
Total Number of Instances	8068	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.827	0.115	0.738	0.827	0.780	0.689	0.935	0.827	D
	0.650	0.114	0.649	0.650	0.650	0.536	0.870	0.702	A
	0.520	0.083	0.653	0.520	0.579	0.476	0.849	0.653	B
	0.718	0.108	0.682	0.718	0.699	0.599	0.891	0.698	C
Weighted Avg.	0.686	0.105	0.683	0.686	0.682	0.581	0.889	0.725	

=== Confusion Matrix ===

	a	b	c	d	<-- classified as
1875	252	82	59		a = D
290	1282	226	174		b = A
183	282	966	427		c = B
192	158	206	1414		d = C

We can see that the **correctly** classified instance is: 68.6292%

We can see that the **incorrectly** classified instance is: 31.3708%



## # Classification using Naïve Bayes:

Classify → choose → bayes → NaiveBayes

The screenshot shows the Weka Explorer interface with the NaiveBayes classifier selected. The 'Test options' section on the left has 'Use training set' selected. The 'Classifier output' section on the right displays the following results:

```
=== Evaluation on training set ===  
Time taken to test model on training data: 0.03 seconds  
  
=== Summary ===  
Correctly Classified Instances 4088 50.6693 %  
Incorrectly Classified Instances 3980 49.3307 %  
Kappa statistic 0.3387  
Mean absolute error 0.2832  
Root mean squared error 0.3989  
Relative absolute error 75.6721 %  
Root relative squared error 92.2017 %  
Total Number of Instances 8068  
  
=== Detailed Accuracy By Class ===  


|               | TP Rate | FP Rate | Precision | Recall | F-Measure | MCC   | ROC Area | PRC Area | Class |
|---------------|---------|---------|-----------|--------|-----------|-------|----------|----------|-------|
|               | 0.702   | 0.164   | 0.627     | 0.702  | 0.662     | 0.521 | 0.846    | 0.689    | D     |
|               | 0.434   | 0.185   | 0.431     | 0.434  | 0.432     | 0.248 | 0.720    | 0.415    | A     |
|               | 0.207   | 0.106   | 0.369     | 0.207  | 0.266     | 0.127 | 0.678    | 0.357    | B     |
|               | 0.638   | 0.204   | 0.502     | 0.638  | 0.562     | 0.403 | 0.772    | 0.555    | C     |
| Weighted Avg. | 0.507   | 0.165   | 0.489     | 0.507  | 0.490     | 0.335 | 0.759    | 0.513    |       |

  
=== Confusion Matrix ===  


|      | a   | b   | c    | d | <-- classified as |
|------|-----|-----|------|---|-------------------|
| 1592 | 417 | 157 | 102  |   | a = D             |
| 432  | 855 | 300 | 385  |   | b = A             |
| 220  | 495 | 385 | 758  |   | c = B             |
| 297  | 217 | 200 | 1256 |   | d = C             |


```

We can see that the correctly classified instance is: 50.6693%

We can see that the incorrectly classified instance is: 49.3307%

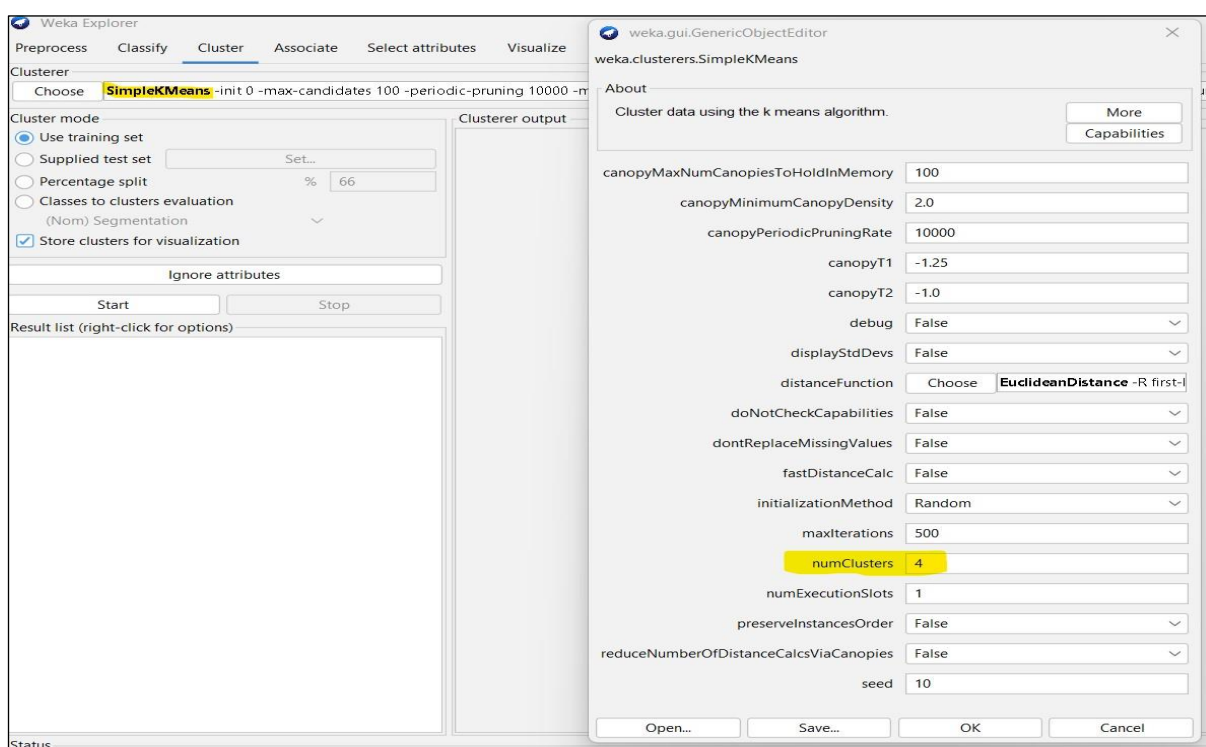
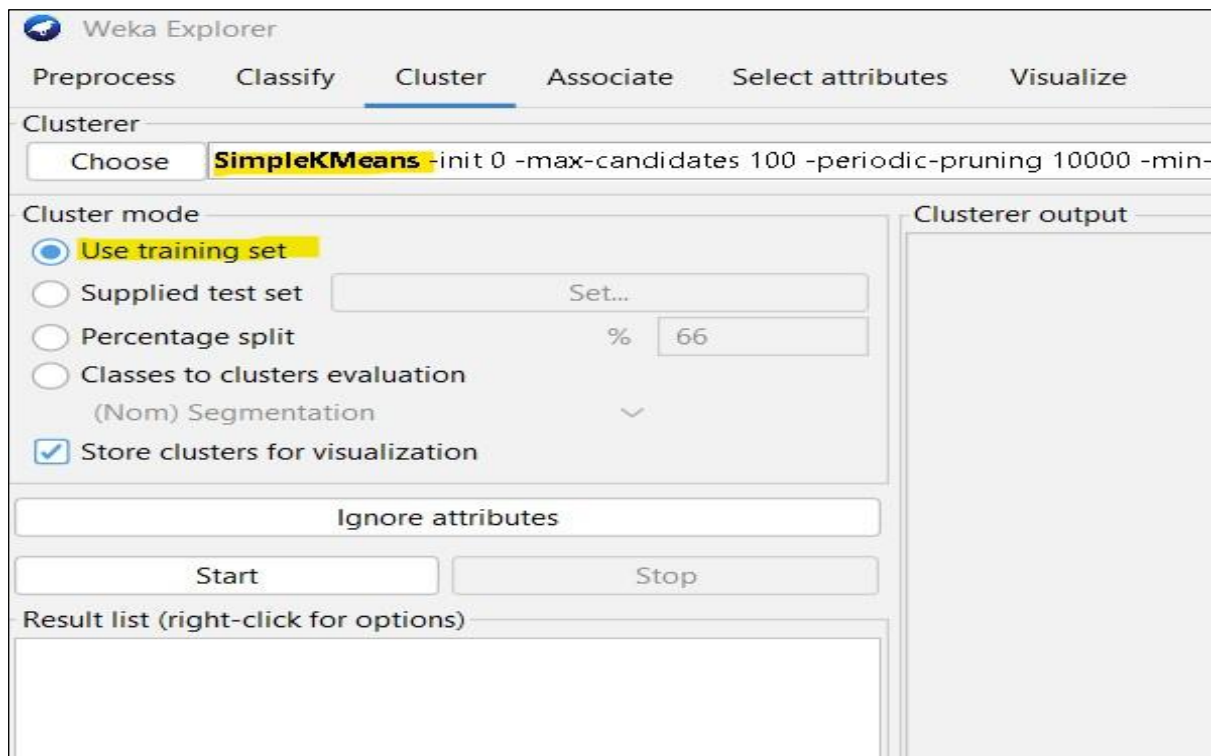


We can see that classification by using the first model (j84 Decision Tree) more accurate than classification by using the second model (naïve Bayes).

So, we chose the j48 Decision Tree for Classification

## # Clustering:

### Simple K-Means



Clusterer

ChooseSimpleKMeans-init 0 -max-candidates 100 -periodic-pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -1.0 -N 4 -A "weka.core.EuclideanDistance -R first-last" -l 500 -num-slots 1 -S 10

Cluster mode

☒ Use training set

☐ Supplied test set

Set...

☐ Percentage split

%66

☐ Classes to clusters evaluation

(Nom) Segmentation

☒ Store clusters for visualization

Ignore attributes

StartStop

Result list (right-click for options)

23:32:45 - SimpleKMeans

Clusterer output

Cluster 1: Male,No,0.211268,Yes,Doctor,0,Low,0.25,Cat\_6,C  
Cluster 2: Male,No,0.15493,Yes,Entertainment,0.571429,Low,0.25,Cat\_6,D  
Cluster 3: Female,No,0.126761,Yes,Engineer,0.285714,Low,0.125,Cat\_6,D

Missing values globally replaced with mean/mode

Final cluster centroids:

Attribute	Full Data (8068.0)	Cluster# 0 (1593.0)	1 (3150.0)	2 (1134.0)	3 (2191.0)
Gender	Male	Male	Male	Male	Female
Ever_Married	Yes	Yes	Yes	No	No
Age	0.3587	0.2545	0.4992	0.256	0.2856
Graduated	Yes	No	Yes	Yes	Yes
Profession	Artist	Healthcare	Artist	Entertainment	Artist
Work_Experience	0.1766	0.1371	0.1473	0.2649	0.2018
Spending_Score	Low	Low	Average	Low	Low
Family_Size	0.232	0.3412	0.2237	0.2184	0.1718
Var_1	Cat_6	Cat_4	Cat_6	Cat_6	Cat_6
Segmentation	D	D	C	D	D

Time taken to build model (full training data) : 0.06 seconds

=== Model and evaluation on training set ===

Clustered Instances

0	1593 ( 20%)
1	3150 ( 39%)
2	1134 ( 14%)
3	2191 ( 27%)

## Decision

In this project we went through all the process from defining the business objective, knowing that dataset, clean and preparing the data exploring features and distributions, data modelling and presenting different algorithms to select the best to predict the Customer Segmentation, what will help the business adopt the best marketing strategies to each of them and bring more market share and revenue to the company.

The chosen model was j84 model since it's the most accurate, although it doesn't present a high accuracy. We could reach a more accurate model having more data about customers, it's something to explore and go deeper in the organization with the business team and the data engineer in order to explore if more relevant features are available.

## The End