

VNUHCM – UNIVERSITY OF SCIENCE
FACULTY OF INFORMATION TECHNOLOGY



LAB 01

SUBJECT : DATA MINING

TEACHER : LÊ HOÀI BẮC
NGUYỄN THỊ THU HẰNG
NGUYỄN BẢO LONG

HỒ CHÍ MINH – 2023

I. Information:

Full Name	MSSV
Bùi Duy Bảo	20127444
Nguyễn Thái Bảo	20127448

II. Task:

3.1 Install WEKA (0.5 points)

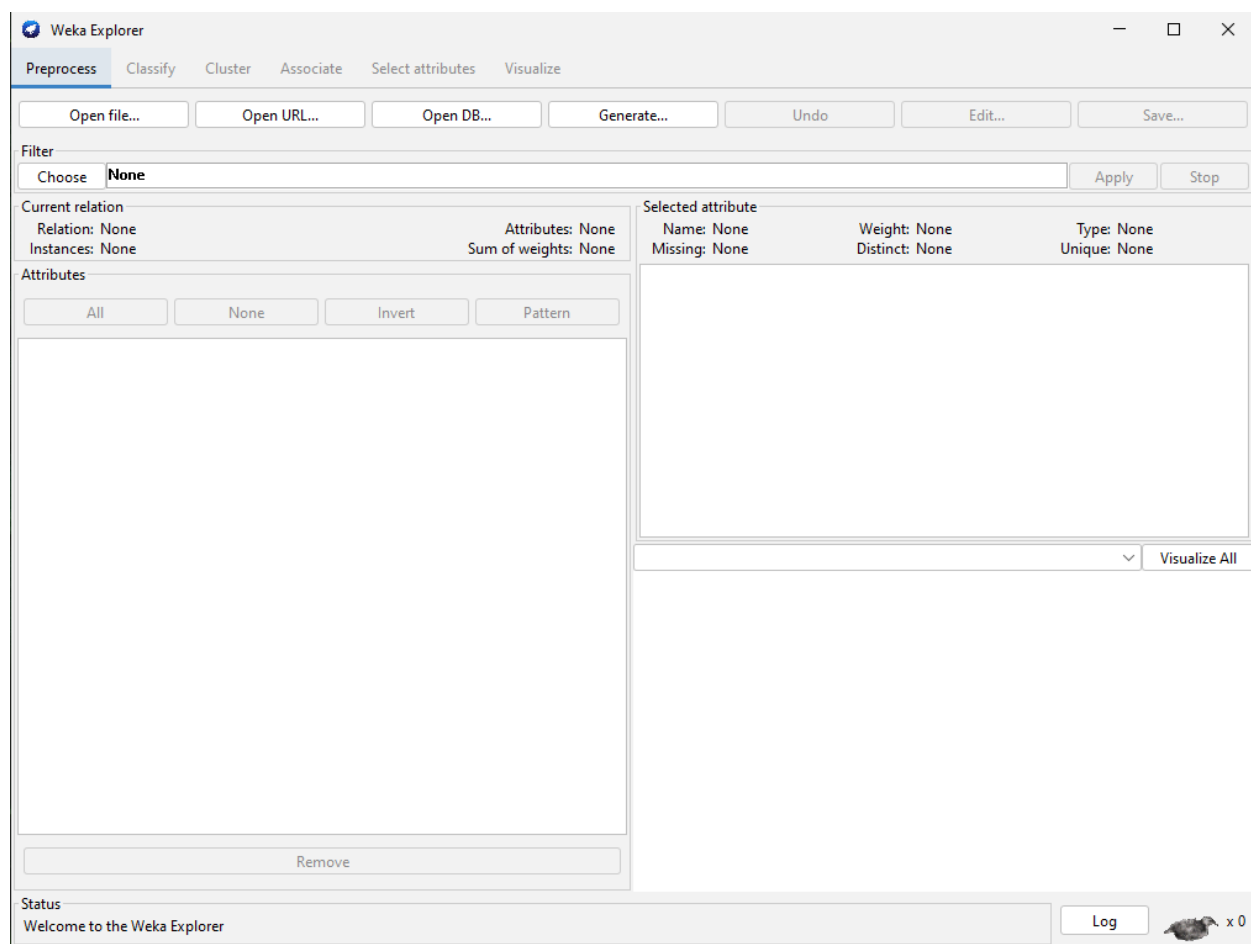


Image 1: Weka interface

3.2 Getting Acquainted With WEKA (4.5 points)

3.2.1 Exploring Breast Cancer data set

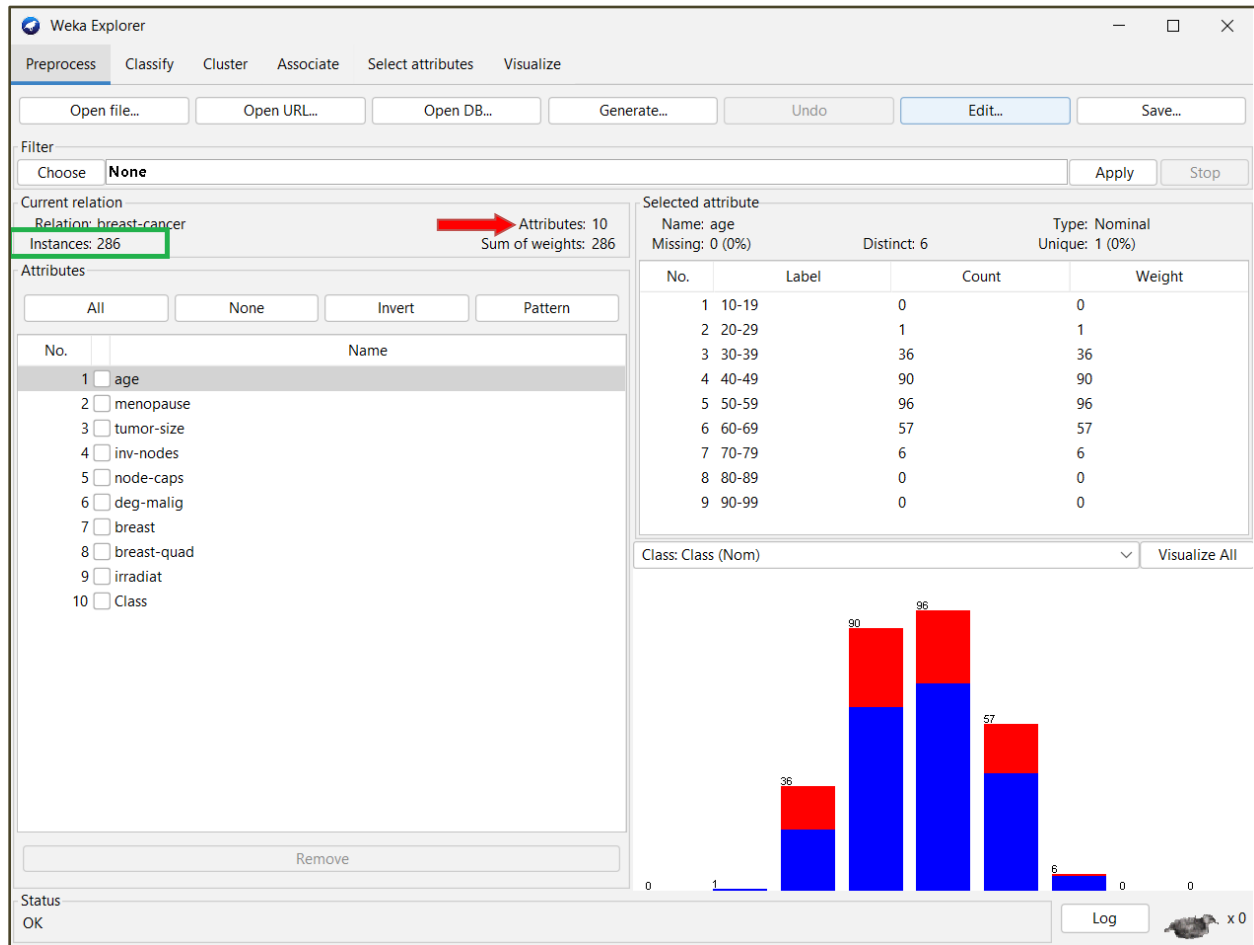


Image 2: basic information

- How many instances does this data set have?
 - There are 286 instances
- How many attributes does this data set have?
 - This data set has 10 attributes
- Which attribute is used for the label? Can it be changed? How?

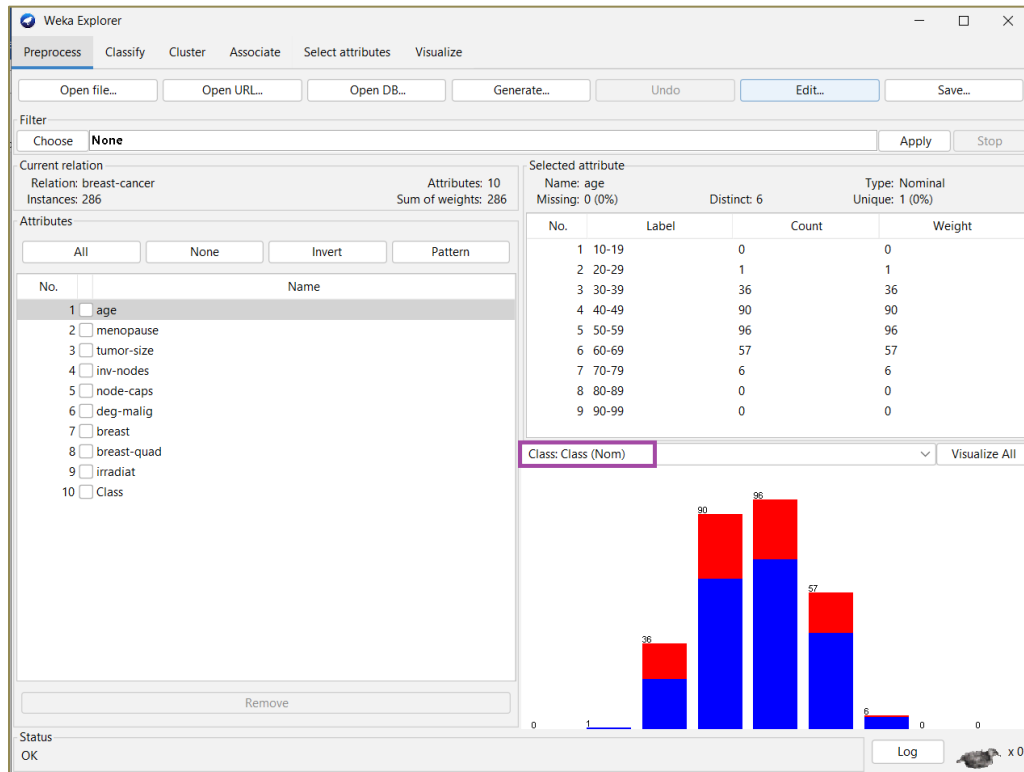


Image 3

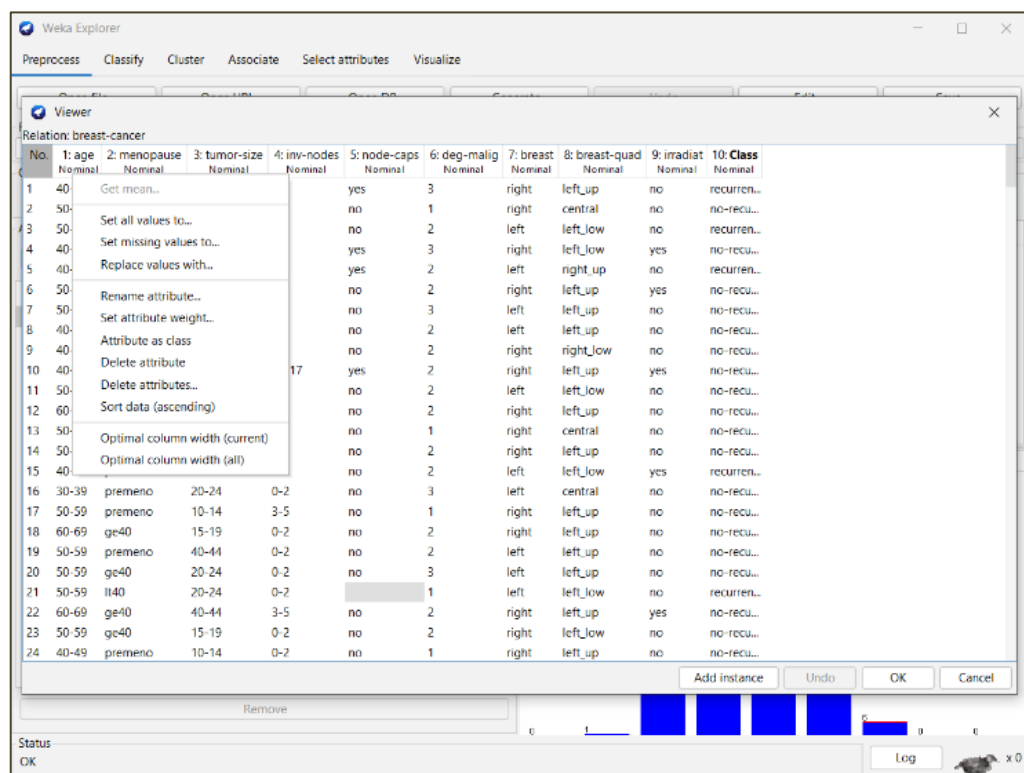


Image 4

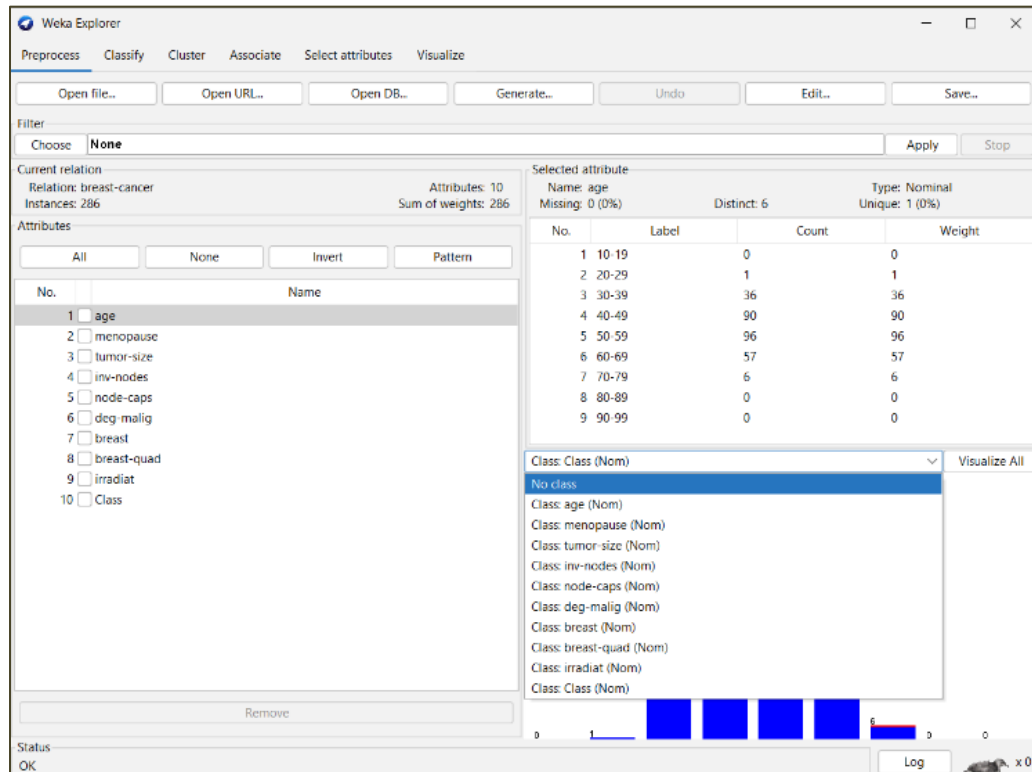


Image 5

- Attribute "Class" (Image 3) which is used to classify the data. The Class attribute distinguishes between benign (non-cancerous) and malignant (cancerous) tumors in breast cancer patients. It is possible to change the Class attribute in WEKA by editing the ARFF file that contains the dataset.
 - C1: Edit -> Choose attribute is used for label -> Attribute as class
 - C2: Choose in main screen
- What is the meaning of each attribute?
 - **Class:** This attribute is the class label, which indicates the diagnosis of the breast cancer patient. It has two possible values: "benign" for non-cancerous tumors and "malignant" for cancerous tumors.
 - **Age:** This attribute is the age of the patient in years at the time of diagnosis.
 - **Menopause:** This attribute indicates whether the patient has experienced menopause or not at the time of diagnosis. It has three possible values: "premeno" for premenopausal patients, "lt40" for patients with menopause at age less than 40, and "ge40" for patients with menopause at age greater than or equal to 40.
 - **Tumor-size:** This attribute is the size of the tumor in millimeters at the time of diagnosis.

- **Inv-nodes:** This attribute is the number of axillary lymph nodes that contain cancer at the time of diagnosis.
 - **Nodecaps:** This attribute indicates whether there is evidence of cancer cells in the lymph node capsules or not. It has two possible values: "yes" for cancer cells detected and "no" for cancer cells not detected.
 - **Deg-malig:** This attribute is the degree of malignancy of the tumor, which is determined by examining the tumor cells under a microscope. It has three possible values: 1 for mild malignancy, 2 for moderate malignancy, and 3 for severe malignancy.
 - **Breast:** This attribute indicates which breast the tumor was detected in. It has two possible values: "left" for the left breast and "right" for the right breast.
 - **Breast-quad:** This attribute is the location of the tumor within the breast. It is divided into four quadrants: "left-up", "left-low", "right-up", and "right-low".
 - **Irradiat:** This attribute indicates whether the patient received radiation therapy after surgery or not. It has two possible values: "yes" for radiation therapy received and "no" for radiation therapy not received.
- *Let's investigate the missing value status in each attribute and describe in general ways to solve the problem of missing values.*

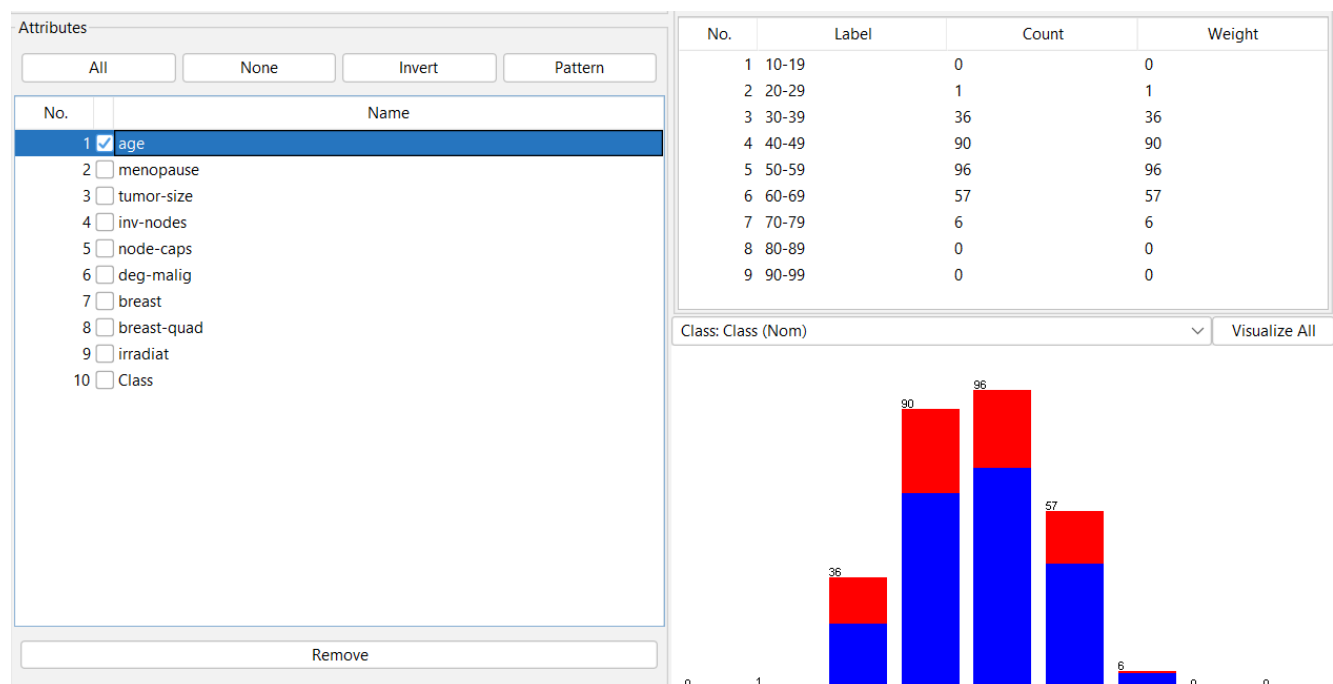
Selected attribute Name: age Missing: 0 (0%)	Distinct: 6	Type: Nominal Unique: 1 (0%)
Selected attribute Name: menopause Missing: 0 (0%)	Distinct: 3	Type: Nominal Unique: 0 (0%)
Selected attribute Name: tumor-size Missing: 0 (0%)	Distinct: 11	Type: Nominal Unique: 0 (0%)
Selected attribute Name: inv-nodes Missing: 0 (0%)	Distinct: 7	Type: Nominal Unique: 1 (0%)
Selected attribute Name: node-caps Missing: 8 (3%)	Distinct: 2	Type: Nominal Unique: 0 (0%)
Selected attribute Name: deg-malig Missing: 0 (0%)	Distinct: 3	Type: Nominal Unique: 0 (0%)

- There are several ways to handle missing values in a dataset, including:
 - **Delete rows containing missing values:** This is the simplest method, but it can lead to loss of important data and degrade the quality of the data file
 - **Use mean:** If the attribute is numeric, you can use the average of the attribute to fill in the missing values. This method can help retain the distribution of the original data, but it can be affected by outliers.
 - **Use the median:** Similar to the mean, if the attribute is numeric, you can use the median of the attribute to fill in the missing values. This method can help retain the distribution of the original data, while minimizing the effect of

outliers.

- **Use the mode:** If the attribute is categorical, you can use the attribute's most common value to fill in the missing values. This method can help to retain the categorization of the original data.
- **Use predictive models:** You can use predictive models to predict missing values based on other attributes.

- *Let's propose solutions to the problem of missing values in the specific attribute.*
 - There are two attribute missing value : **Node-caps** with 8 value missing and **Breast-quad** with 1 value missing
 - Some solutions like :
 - Remove row have missing value
 - Missing value can be change by value median , mean or mode with condition this column is numeric
 - With Columns have category type we can change by the value with the highest frequency in the column, if the number of missing values is high, all can be classified into a new class.
 - Or we can use KNN , Naïve Bayes algorithm to fill in missing value places
- *Let's explain the meaning of the chart in the WEKA Explorer. Setting the title for it and describing its legend.*

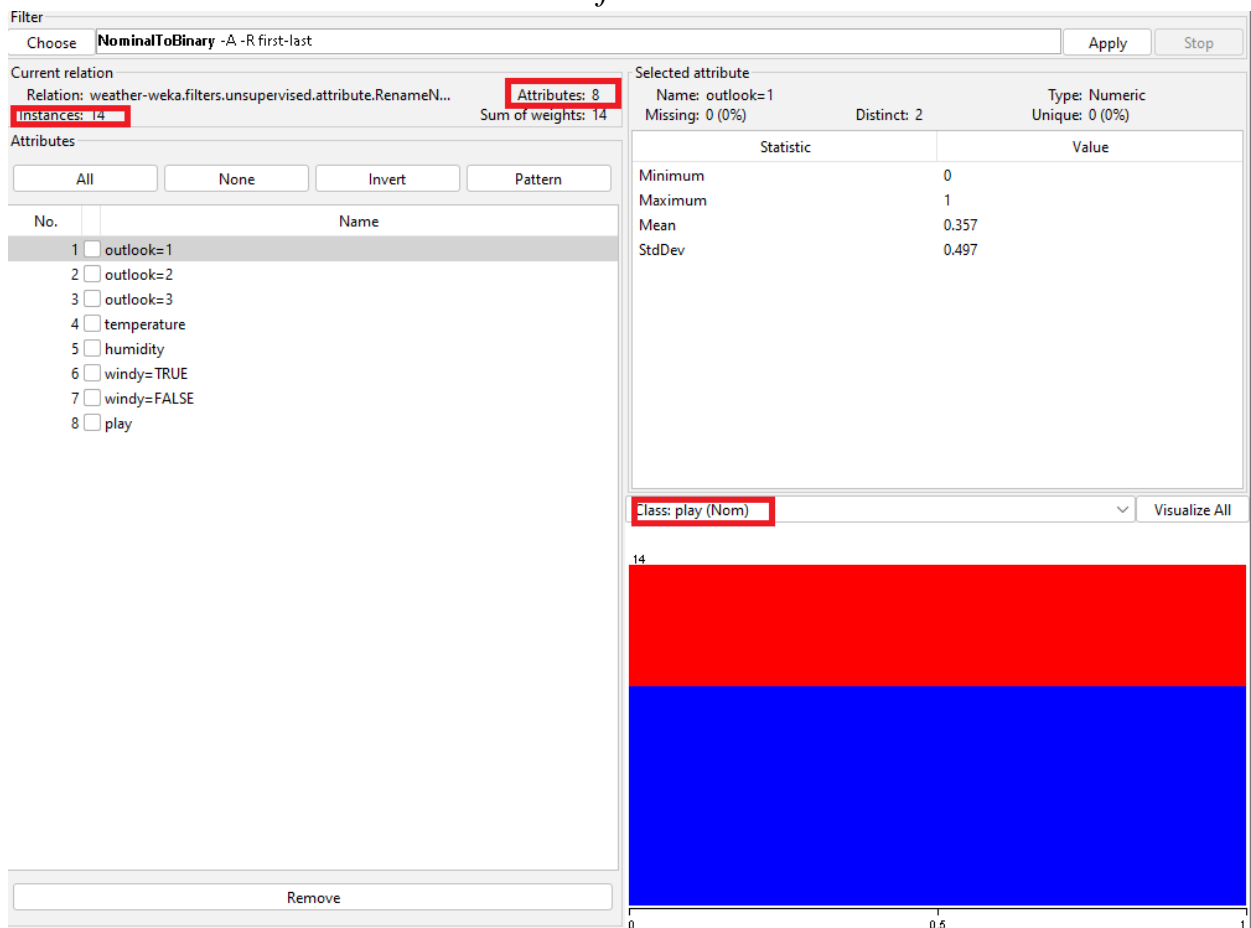


- ⇒ Taking the representative graph when we select the age attribute in Attributes, we get the following

- This graph shows the total number of people with breast cancer surveyed in different age groups
- It also shows the total number of relapsers (shown in red) and the total number of people who do not relapse (shown in blue) by age, as well as the correlation between these quantities.
- After labeling the x-axis as age groups, label the y-axis as quantity, set the legend for the graph; We can name it: "Recurrence status of breast cancer patients in age groups".

3.2.2 Exploring Weather data set

- *How many attributes does this data set have? How many samples? Which attributes have data type categorical? Which attributes have a data type that is numerical? Which attribute is used for the label?*

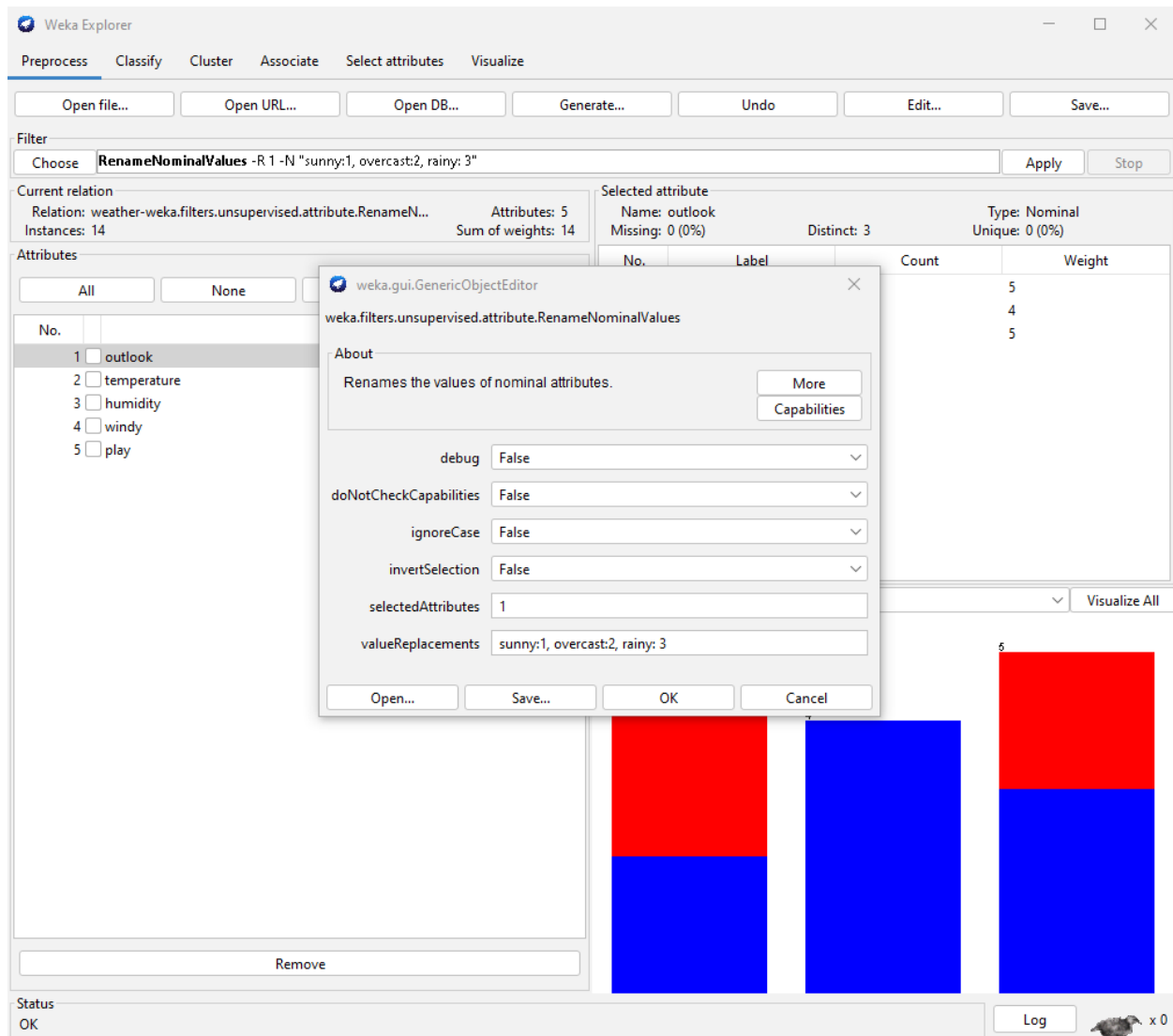


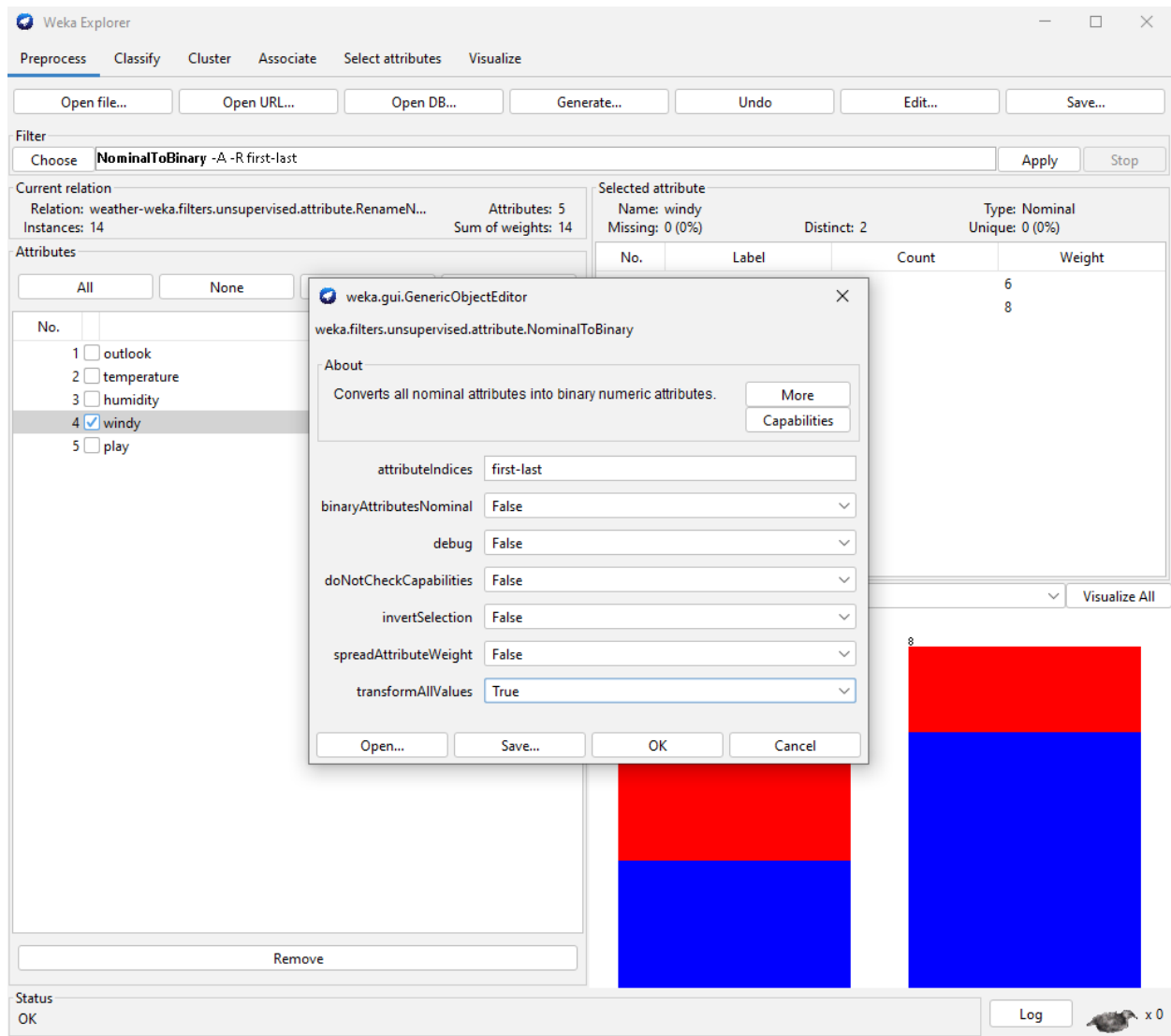
- ⇒ Number of attributes: 8
- ⇒ Number of samples: 14
- ⇒ Attributes with categorical data type: None. All attributes have numerical data

type.

⇒ Attributes with numerical data type: All attributes.

⇒ Label attribute: The last attribute (i.e., attribute 8), which is "play", is used as the label attribute.





⇒ Note that the "weather.numeric.arff" dataset is a modified version of the well-known "weather" dataset, where the original nominal values have been replaced with numerical values. Therefore, all attributes in this dataset are numerical, even though they may represent categories such as "sunny", "overcast", and "rainy" in the original dataset. how to replace

Viewer

Relation: weather-weka.filters.unsupervised.attribute.RenameNominalValues-R1-Nsunny:1, overcast:2, rainy:3-weka.filters.unsupervised.attribute.NominalToBinary-A-Rfirst-last

No.	1: outlook=1 Numeric	2: outlook=2 Numeric	3: outlook=3 Numeric	4: temperature Numeric	5: humidity Numeric	6: windy=TRUE Numeric	7: windy=FALSE Numeric	8: play Nominal
1	1.0	0.0	0.0	85.0	85.0	0.0	1.0	no
2	1.0	0.0	0.0	80.0	90.0	1.0	0.0	no
3	0.0	1.0	0.0	83.0	86.0	0.0	1.0	yes
4	0.0	0.0	1.0	70.0	96.0	0.0	1.0	yes
5	0.0	0.0	1.0	68.0	80.0	0.0	1.0	yes
6	0.0	0.0	1.0	65.0	70.0	1.0	0.0	no
7	0.0	1.0	0.0	64.0	65.0	1.0	0.0	yes
8	1.0	0.0	0.0	72.0	95.0	0.0	1.0	no
9	1.0	0.0	0.0	69.0	70.0	0.0	1.0	yes
10	0.0	0.0	1.0	75.0	80.0	0.0	1.0	yes
11	1.0	0.0	0.0	75.0	70.0	1.0	0.0	yes
12	0.0	1.0	0.0	72.0	90.0	1.0	0.0	yes
13	0.0	1.0	0.0	81.0	75.0	0.0	1.0	yes
14	0.0	0.0	1.0	71.0	91.0	1.0	0.0	no

Add instance Undo OK Cancel

- Let's list five-number summary of two attributes temperature and humidity. Does WEKA provide these values?

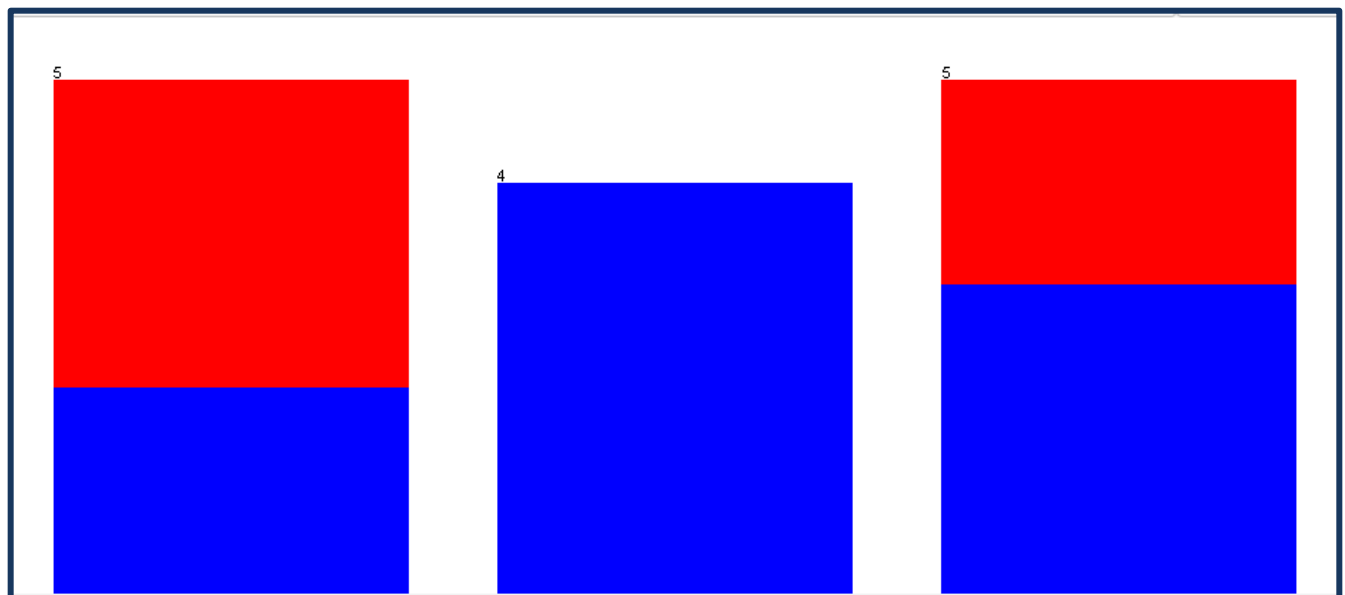
- ⇒ Weka provides four -number summary of the attributes in a dataset, including temperature and humidity
- ⇒ Attribute temperature :

Selected attribute		
Name: temperature		Type: Numeric
Missing: 0 (0%)		Unique: 10 (71%)
Distinct: 12		
Statistic	Value	
Minimum	64	
Maximum	85	
Mean	73.571	
StdDev	6.572	

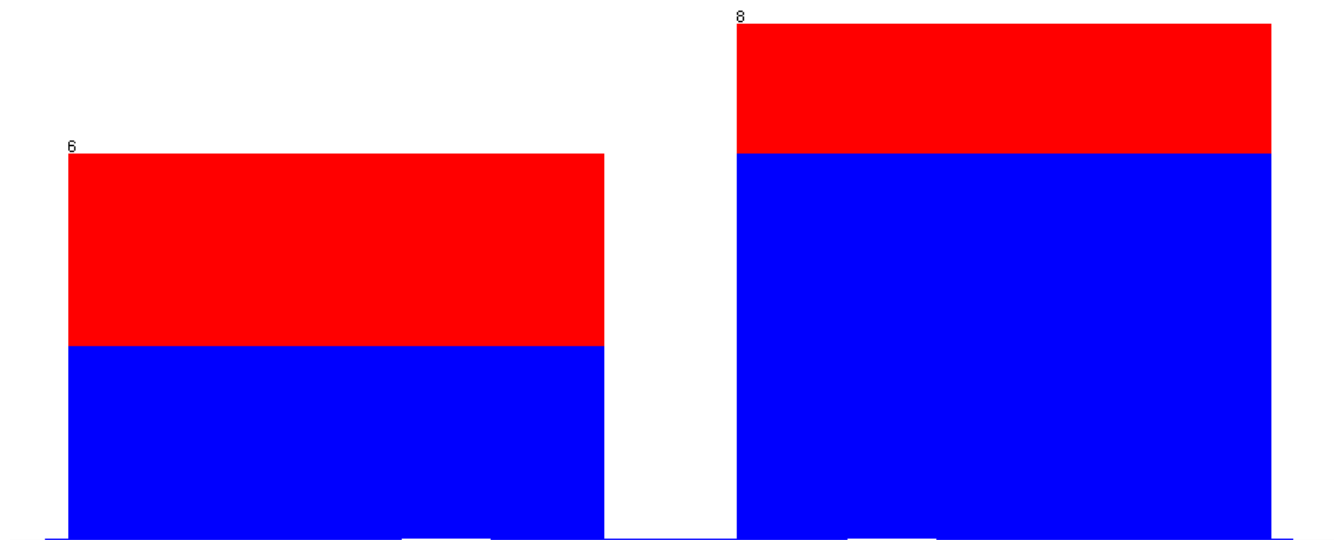
- ⇒ Attribute Humidity :

Selected attribute		
Name: humidity		Type: Numeric
Missing: 0 (0%)	Distinct: 10	Unique: 7 (50%)
Statistic	Value	
Minimum	65	
Maximum	96	
Mean	81.643	
StdDev	10.285	

- Let's explain the meaning of all charts in the WEKA Explorer. Setting the title for it and describing its legend.
- In the following graphs with other attributes, the y-axis label is quantity, blue is for deciding to hang out, red is not.
- 1. Attribute outlook : the peeling turns are 'sunny', 'overcast', 'rainy'
 - ⇒ I see, everyone chooses to go out if it's overcast. When it's sunny, tending is not to go, when it's raining, tending is to go out

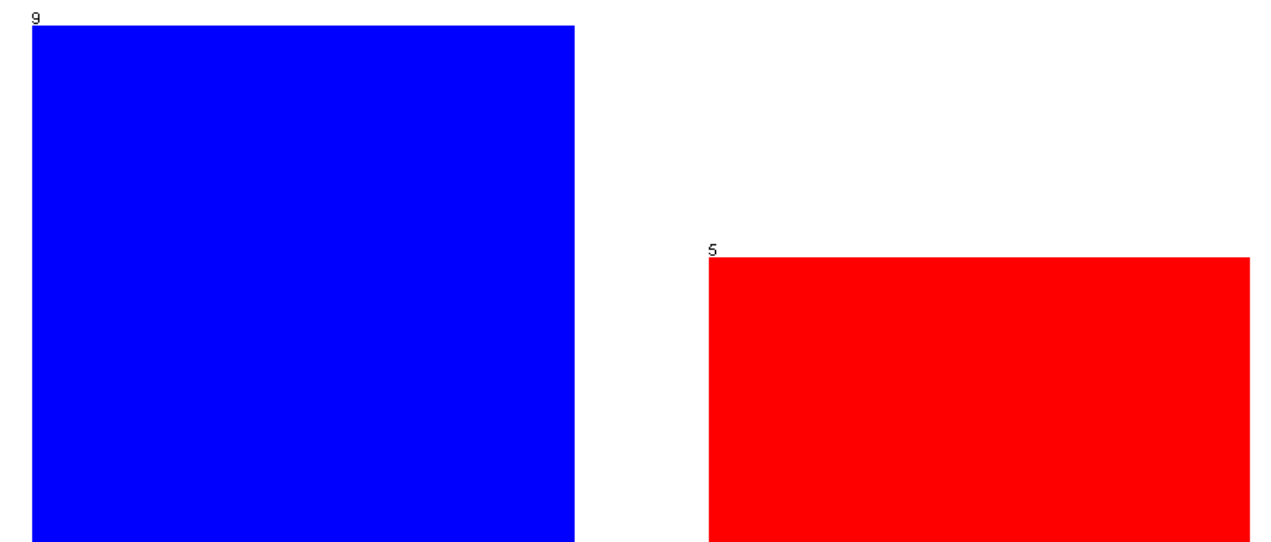


2. Attribute windy : label of x is TRUE AND FALSE
 - ⇒ The graph shows that people enjoy going out when it's not windy, when it's windy, half of them go, half don't.



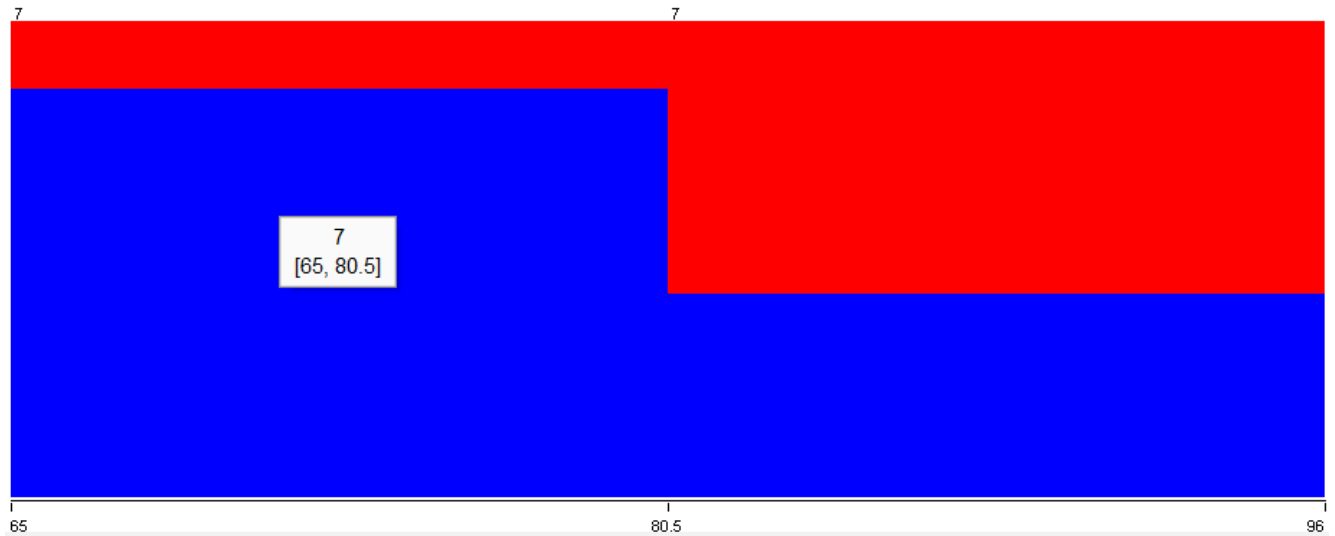
3. Attribute play : label of x is True and False

⇒ The graph show everyone Agrees to go out and play almost twice as much Disagree



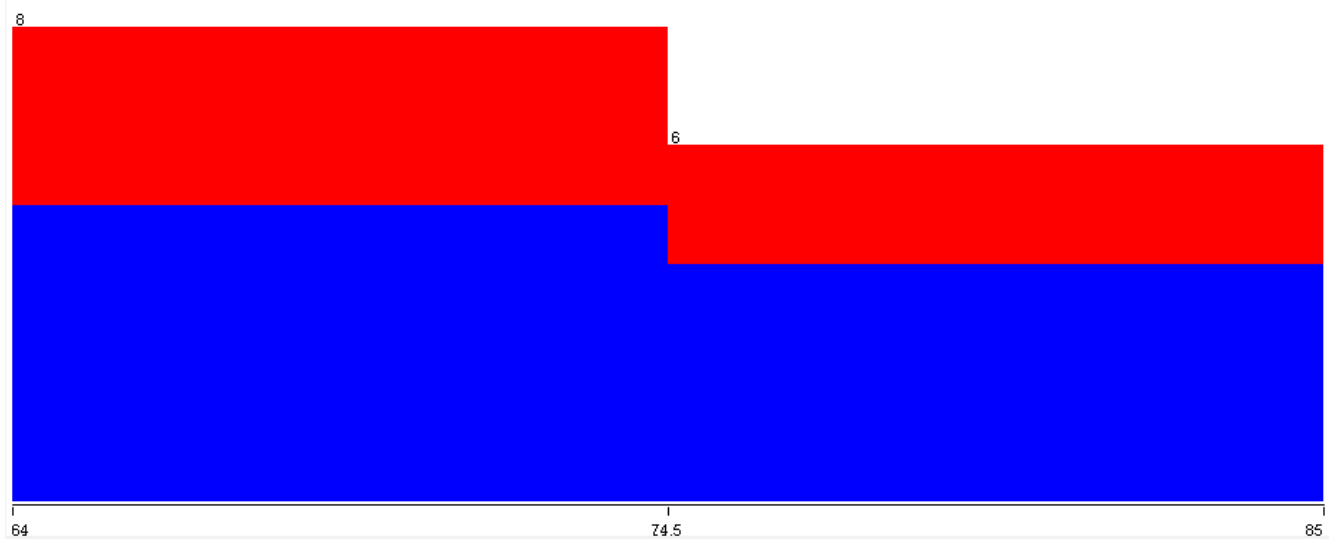
4. Attribute humidity :

⇒ The graph show area value of humidity , most of people agree to go out when humidity is less than mean (80.5) and the ratio of disagreeing to go out and agreeing to go out when the humidity is greater than mean is not much different



5. Attribute temperature :

⇒ The graph show ratio people agree to go out always greater than disagree for all temperature



⇒ *The title for all chat is “Ratio go out with different weather conditions”*

- *Let’s move to the Visualize tag. What’s the name of this chart? Do you think there are any pairs of different attributes that have correlated?*

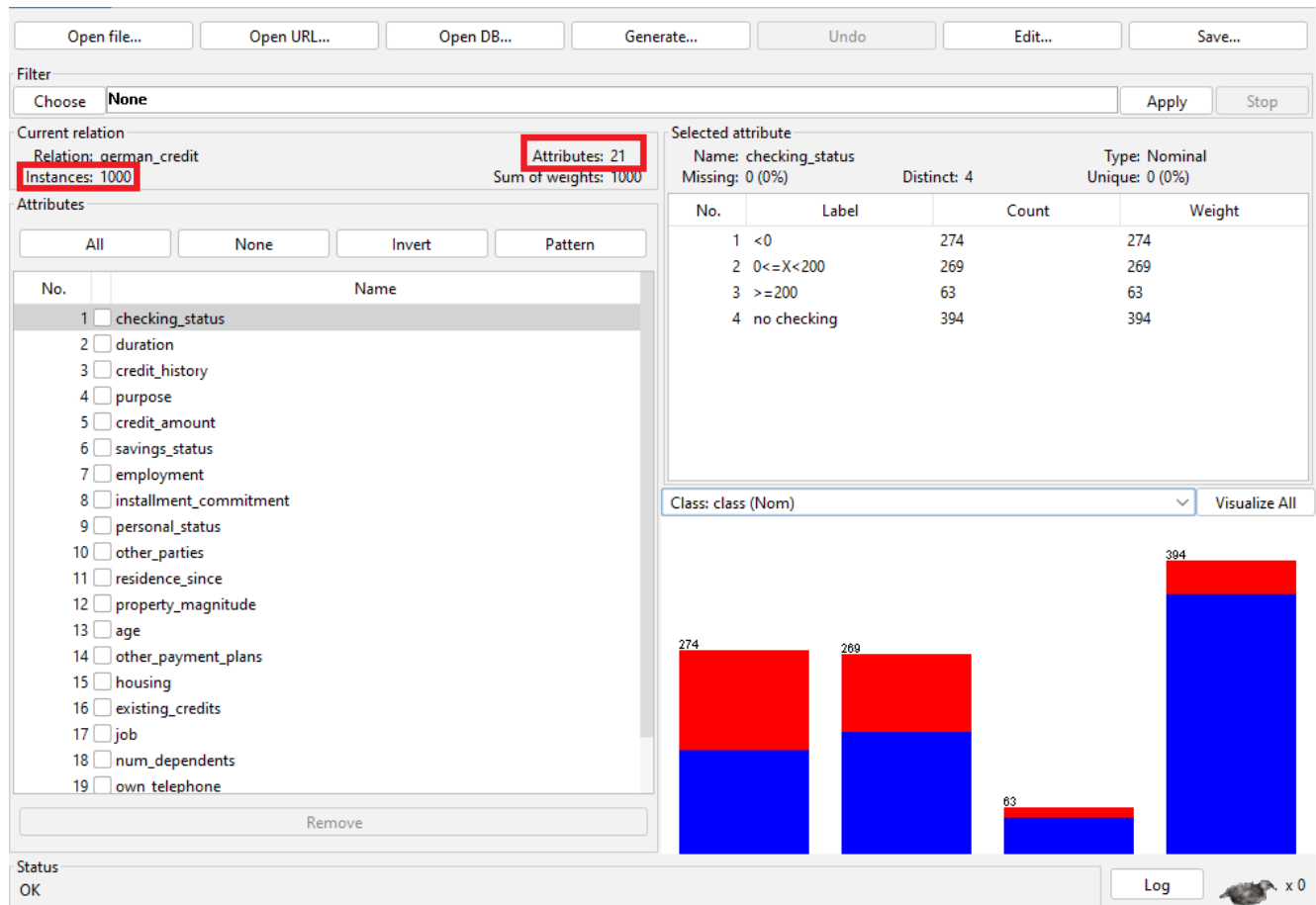
⇒ The graphs here are the scatterplot matrix. Doesn't seem to be correlated, can't see positive correlation and negative correlation



The subgraphs in the red area do not show correlation

3.2.3 Exploring Credit in Germany data set

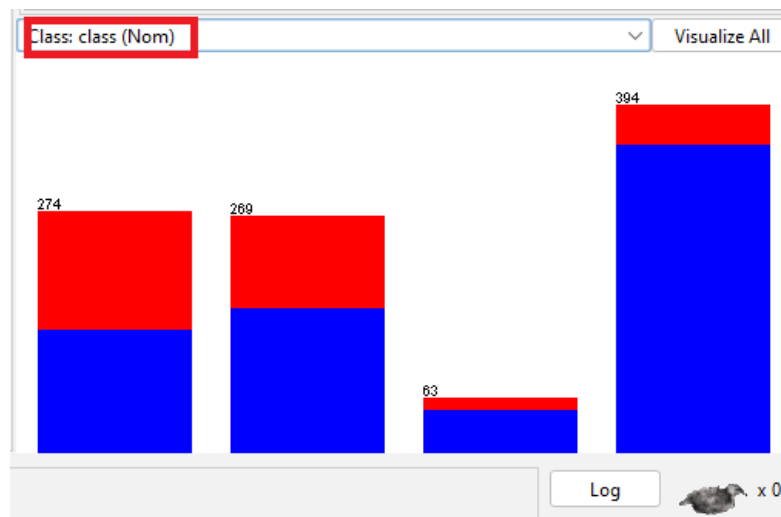
- *What is the content of the comments section in credit-g.arff (when opened with any text editor) about? How many samples does the data set have? How many attributes? Describe any five attributes (must have both discrete and continuous attributes).*



- ⇒ The content of the comments section in credit-g.arff is that the data was collected from a German bank and that the goal is to predict whether an applicant is a good or bad credit risk based on a set of attributes.
- ⇒ The dataset contains 1000 samples or instances.
- ⇒ There are 21 attributes in total
- ⇒ Here are brief descriptions of five attributes, including both discrete and continuous types:
 - **Age**: a continuous attribute representing the age of the applicant in years.
 - **Checking_account**: a categorical attribute with four possible values indicating the status of the applicant's checking account: "no checking", "less than 0 DM", "0 to 200 DM", or "greater than 200 DM".
 - **Credit_history**: a categorical attribute with five possible values indicating the credit history of the applicant: "no credit history", "all loans at bank paid back", "existing loans paid back", "delay in paying off previous loans", or "critical account/other credits existing".
 - **Credit_amount**: a continuous attribute representing the amount of the requested credit in DM.
 - **Employment_duration**: a categorical attribute with five possible values indicating the length of the applicant's current employment: "unemployed", "less than 1 year",

"1 to 4 years", "4 to 7 years", or "greater than 7 years".

- Which attribute is used for the label?



⇒ The attribute "class" is used for the label

- Let's describe the distribution of continuous attributes? (Left skewed or right skewed ?)
 - A distribution with a positive skew is said to be right-skewed, meaning that the tail of the distribution extends more to the right, while a distribution with a negative skew is left-skewed, meaning that the tail of the distribution extends more to the left.
 - Here are the skewness values for the continuous attributes in the **credit-g.arff** dataset:
 - Age: right-skewed
 - Credit_amount: right-skewed
 - Duration: right-skewed

Current relation

Relation: german_credit
Instances: 1000

Attributes: 21
Sum of weights: 1000

Attributes

All None Invert Pattern

No.	Name
1	<input type="checkbox"/> checking_status
2	<input checked="" type="checkbox"/> duration
3	<input type="checkbox"/> credit_history
4	<input type="checkbox"/> purpose
5	<input type="checkbox"/> credit_amount
6	<input type="checkbox"/> savings_status
7	<input type="checkbox"/> employment
8	<input type="checkbox"/> installment_commitment
9	<input type="checkbox"/> personal_status
10	<input type="checkbox"/> other_parties
11	<input type="checkbox"/> residence_since
12	<input type="checkbox"/> property_magnitude
13	<input type="checkbox"/> age
14	<input type="checkbox"/> other_payment_plans
15	<input type="checkbox"/> housing
16	<input type="checkbox"/> existing_credits
17	<input type="checkbox"/> job
18	<input type="checkbox"/> num_dependents
19	<input type="checkbox"/> own_telephone

Remove

Selected attribute

Name: duration
Missing: 0 (0%)

Distinct: 33

Type: Numeric
Unique: 5 (1%)

Statistic	Value
Minimum	4
Maximum	72
Mean	20.903
StdDev	12.059

Class: class (Nom) Visualize All

Open file... Open URL... Open DB... Generate... Undo Edit... Save...

Filter

Choose None Apply Stop

Current relation

Relation: german_credit
Instances: 1000

Attributes: 21
Sum of weights: 1000

Attributes

All None Invert Pattern

No.	Name
1	<input type="checkbox"/> checking_status
2	<input type="checkbox"/> duration
3	<input type="checkbox"/> credit_history
4	<input type="checkbox"/> purpose
5	<input type="checkbox"/> credit_amount
6	<input type="checkbox"/> savings_status
7	<input type="checkbox"/> employment
8	<input type="checkbox"/> installment_commitment
9	<input type="checkbox"/> personal_status
10	<input type="checkbox"/> other_parties
11	<input type="checkbox"/> residence_since
12	<input type="checkbox"/> property_magnitude
13	<input checked="" type="checkbox"/> age
14	<input type="checkbox"/> other_payment_plans
15	<input type="checkbox"/> housing
16	<input type="checkbox"/> existing_credits
17	<input type="checkbox"/> job
18	<input type="checkbox"/> num_dependents
19	<input type="checkbox"/> own_telephone

Remove

Selected attribute

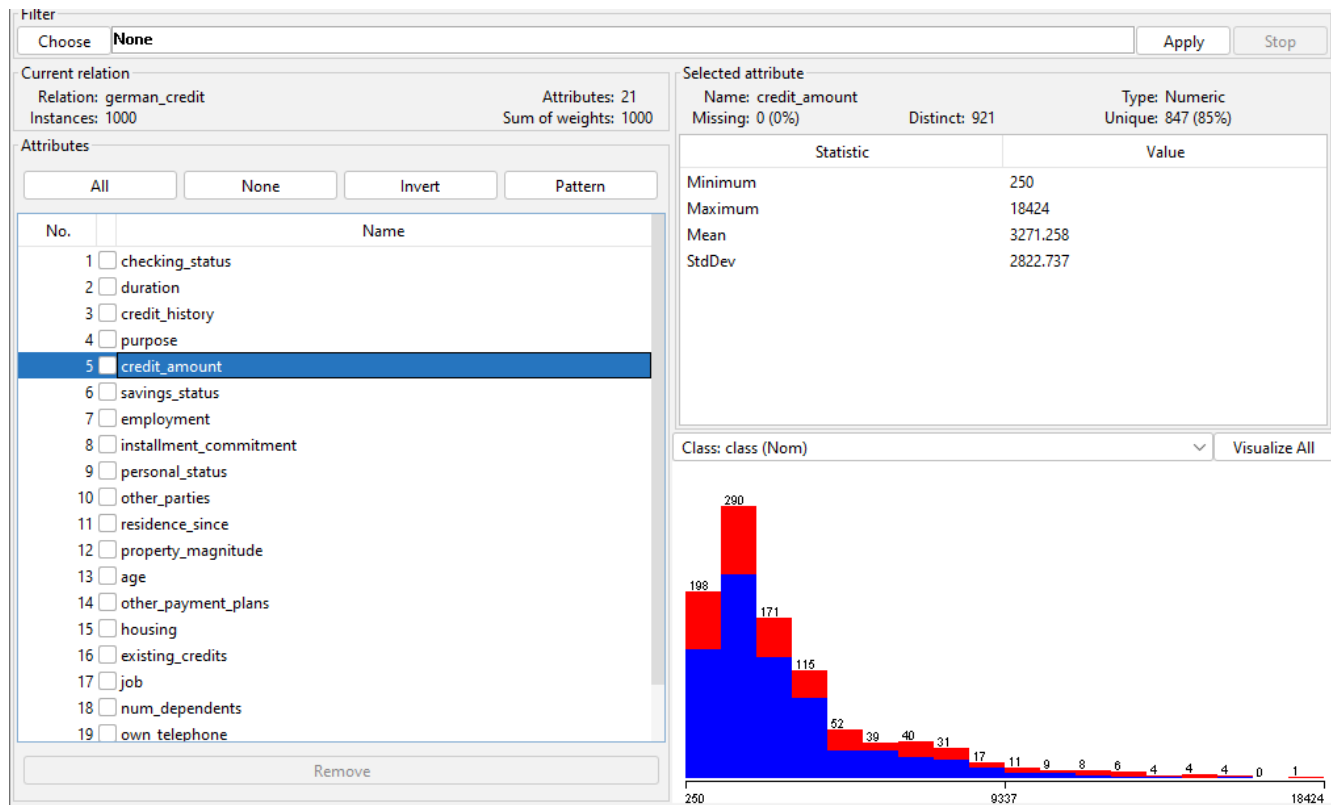
Name: age
Missing: 0 (0%)

Distinct: 53

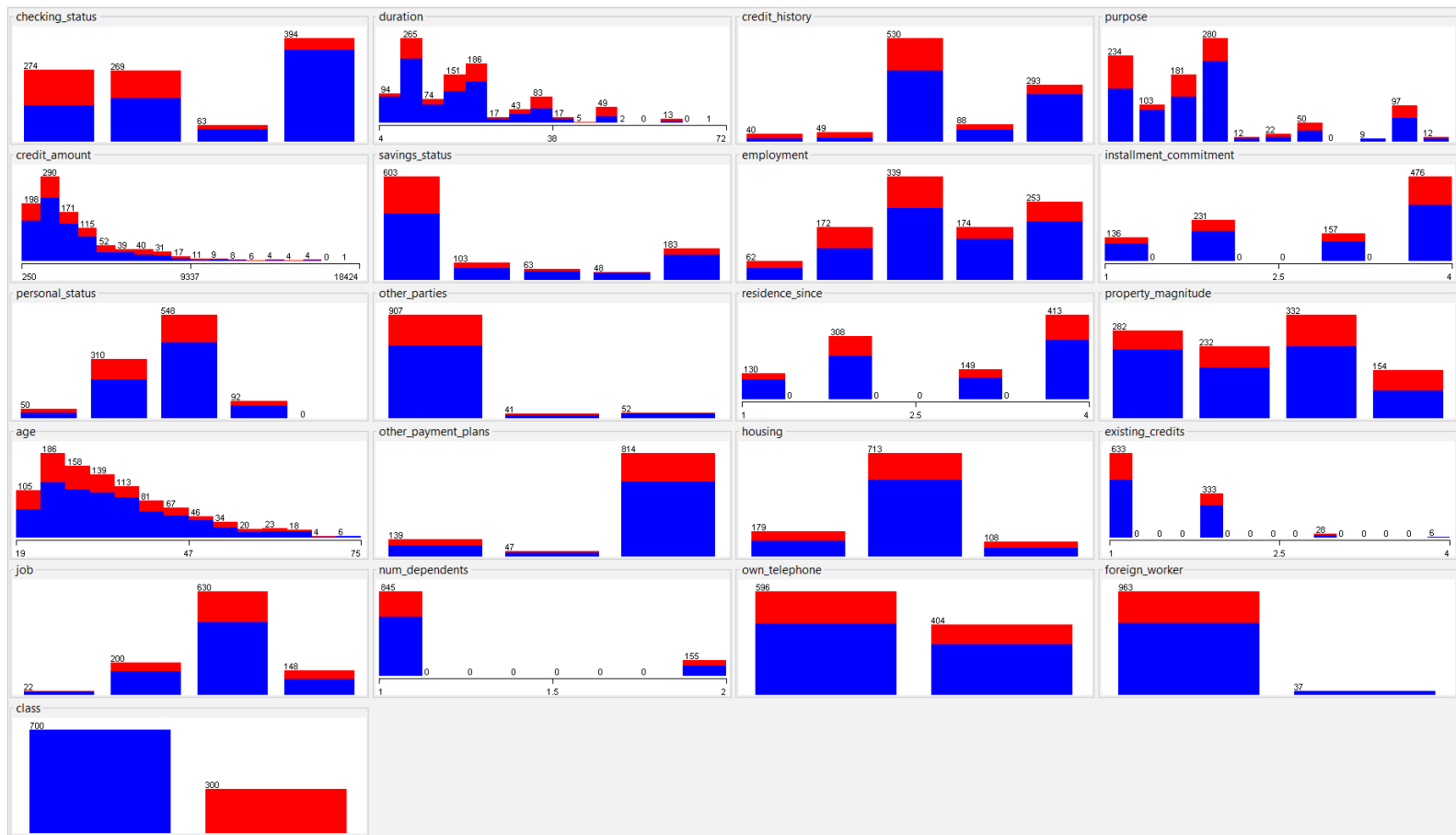
Type: Numeric
Unique: 1 (0%)

Statistic	Value
Minimum	19
Maximum	75
Mean	35.546
StdDev	11.375

Class: class (Nom) Visualize All



- Let's explain the meaning of all charts in the WEKA Explorer. Setting the title for it and describing its legend.

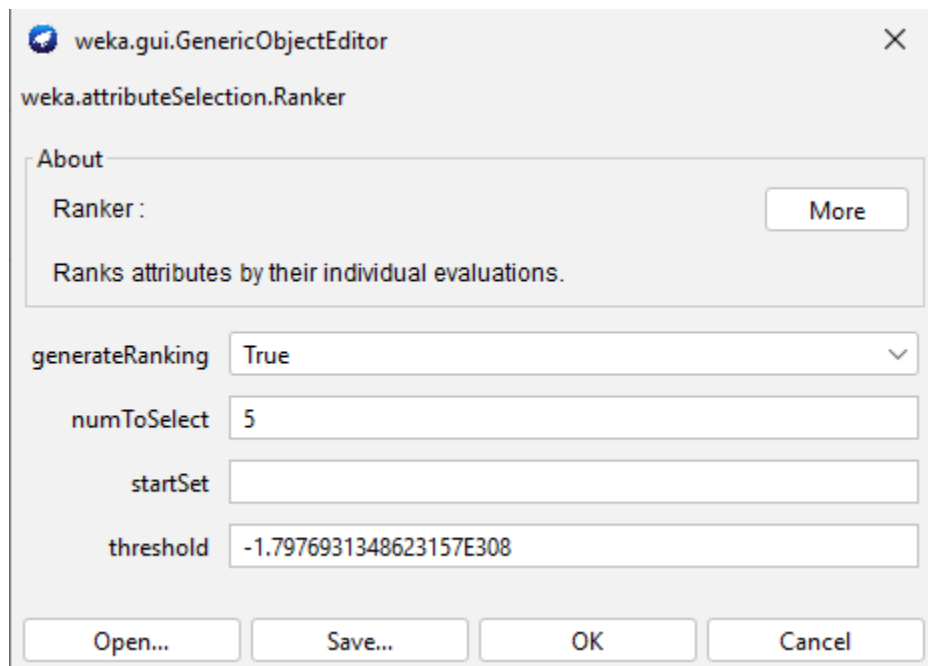
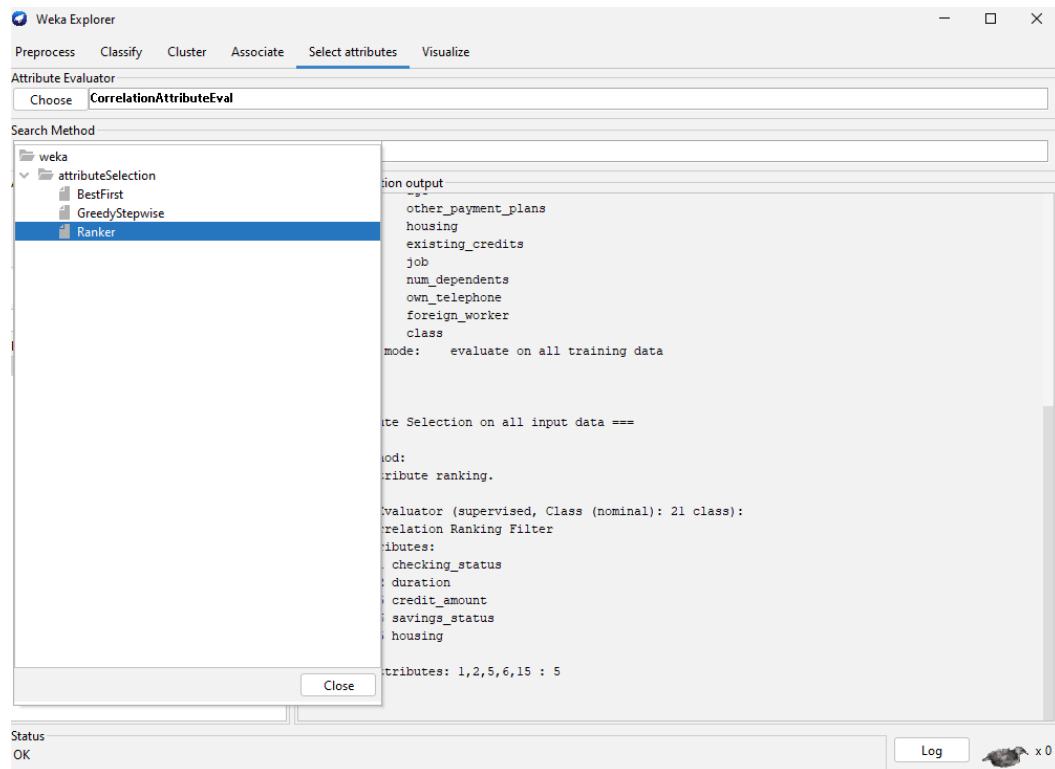


⇒ *The title for this char : “Ratio using credit in German”*

- *Let’s move to the Select attributes tag. Describe all of the options for attribute selection. Which options should be used to select the 5 attributes with the highest correlation?(Step-by-step description, with step-by-step photos and final results)*

⇒ To select the 5 attributes with the highest correlation in WEKA, we can use the "AttributeSelection" filter. Here are the steps:

1. Open WEKA and load the dataset you want to analyze.
2. Click on the "Select attribute”
3. In the "Attribute Evaluator", select "CorrelationAttributeEval"
4. In the "Search Method", select the "Ranker"
5. Set the number of attributes to 5
6. Click on the "Start" button to apply the filter and wait for WEKA to generate the output.



```
Attribute Evaluator (supervised, Class (nominal): 21 class):
Correlation Ranking Filter
Ranked attributes:
0.233    1 checking_status
0.215    2 duration
0.155    5 credit_amount
0.132    6 savings_status
0.121   15 housing

Selected attributes: 1,2,5,6,15 : 5
```

3.3 Preprocessing Data in Python (5 points)

⇒ *Some conventions*

- ❖ python Main.py <name function> <input file> <another options >
- ❖ We have 8 main functions : list_missing, count_missing, impute, remove_col, remove_row, remove_duplicate, normalize, calculate

I. *Extract columns with missing values*

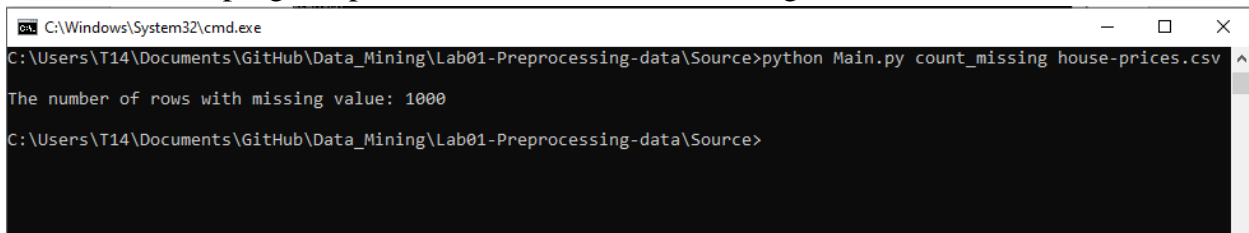
- Syntax : python Main.py list_missing house-prices.csv
- ⇒ The program lists the missing column

```
C:\Windows\System32\cmd.exe
C:\Users\T14\Documents\GitHub\Data_Mining\Lab01-Preprocessing-data\Source>python Main.py list_missing house-prices.csv
There are missing attributes in these following columns:
LotFrontage
Alley
MasVnrType
MasVnrArea
BsmtQual
BsmtCond
BsmtExposure
BsmtFinType1
BsmtFinType2
FireplaceQu
GarageType
GarageYrBlt
GarageFinish
GarageQual
GarageCond
PoolQC
Fence
MiscFeature
C:\Users\T14\Documents\GitHub\Data_Mining\Lab01-Preprocessing-data\Source>
```

II. *Count the number of lines with missing data.*

- Syntax : python Main.py count_missing house-prices.csv

- The program prints out the number of missing values

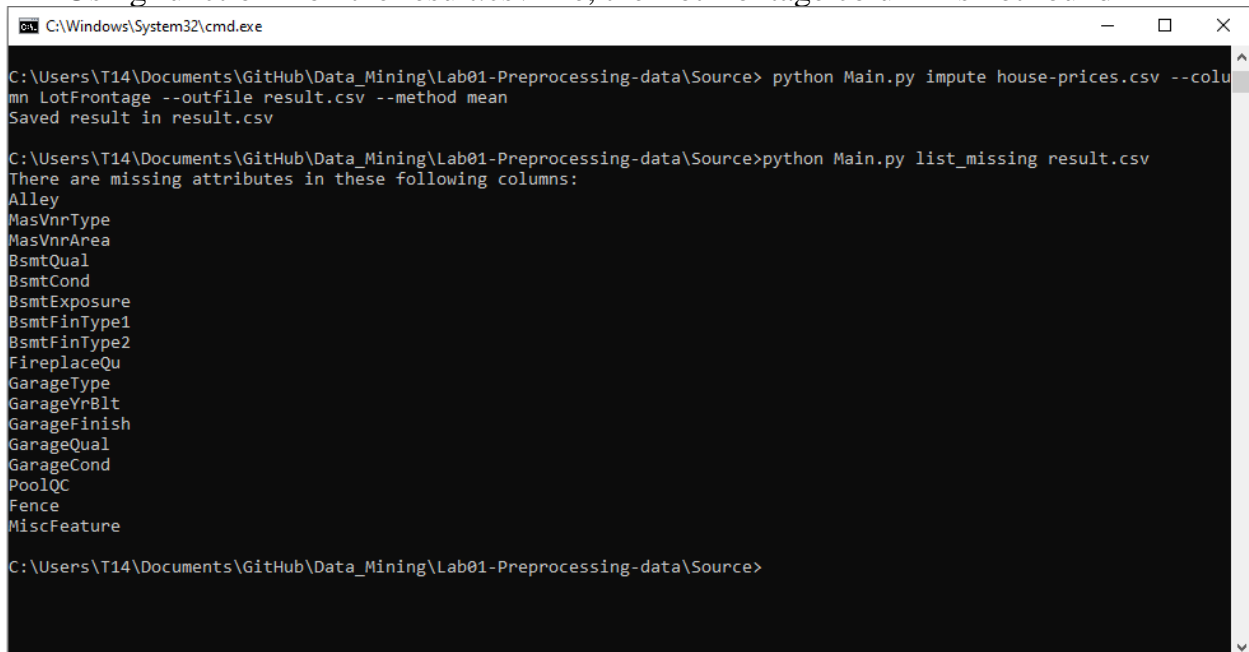


```

C:\Windows\System32\cmd.exe
C:\Users\T14\Documents\GitHub\Data_Mining\Lab01-Preprocessing-data\Source>python Main.py count_missing house-prices.csv
The number of rows with missing value: 1000
C:\Users\T14\Documents\GitHub\Data_Mining\Lab01-Preprocessing-data\Source>
  
```

III. Fill in the missing value using mean, median (for numeric properties) and mode (for the categorical attribute).

- In function3 use 3 options: column, method, outfile :
- In there :
 - Column can take multiple parameters to indicate the columns to be filled in. If not specified, all columns will be filled by default. Columns with invalid names will be ignored.
 - Method indicates the filling method for the numeric attribute: mean median and categorical: mode
 - Outfile indicates the name of the file to be saved
- Syntax : `python Main.py impute house-prices.csv --column LotFrontage --outfile result.csv --method mean`
- All missing LotFrontage values are replaced by its average
- Using function 1 on the result.csv file, the LotFrontage column is not found

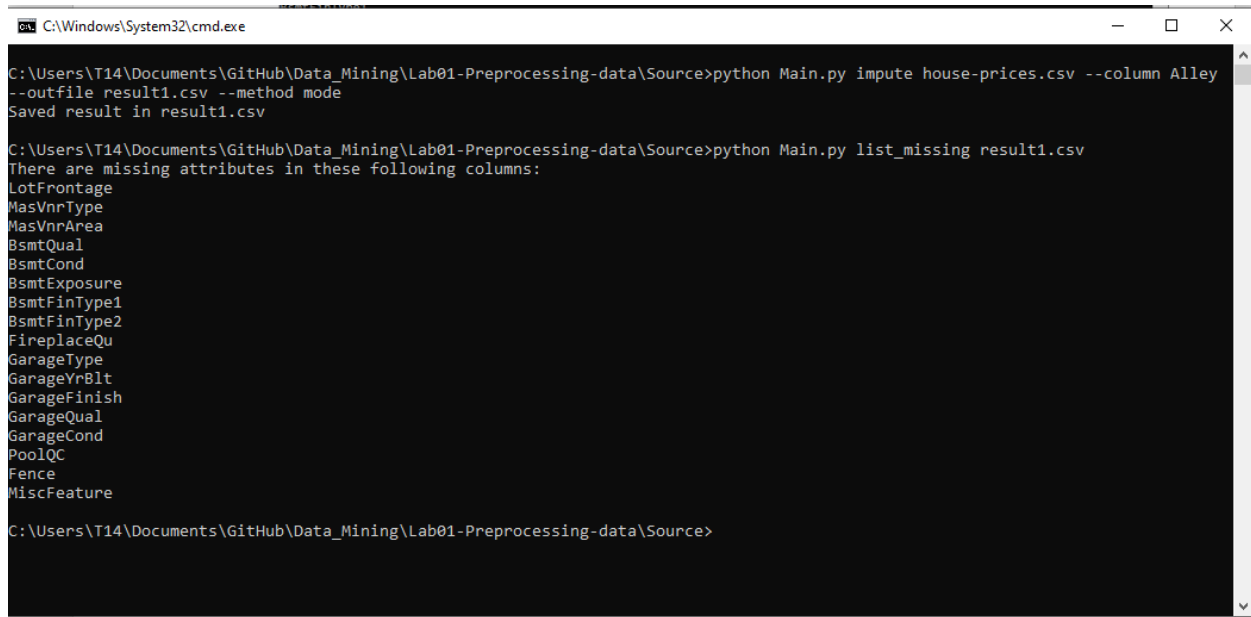


```

C:\Windows\System32\cmd.exe
C:\Users\T14\Documents\GitHub\Data_Mining\Lab01-Preprocessing-data\Source> python Main.py impute house-prices.csv --column LotFrontage --outfile result.csv --method mean
Saved result in result.csv

C:\Users\T14\Documents\GitHub\Data_Mining\Lab01-Preprocessing-data\Source>python Main.py list_missing result.csv
There are missing attributes in these following columns:
Alley
MasVnrType
MasVnrArea
BsmtQual
BsmtCond
BsmtExposure
BsmtFinType1
BsmtFinType2
FireplaceQu
GarageType
GarageYrBlt
GarageFinish
GarageQual
GarageCond
PoolQC
Fence
MiscFeature
C:\Users\T14\Documents\GitHub\Data_Mining\Lab01-Preprocessing-data\Source>
  
```

- Syntax : `python Main.py impute house-prices.csv --column Alley --outfile result1.csv --method mode`
- All missing LotFrontage values are replaced by its mode
- Using function 1 on file result1.csv, the column Alley is not visible



```
C:\Windows\System32\cmd.exe

C:\Users\T14\Documents\GitHub\Data_Mining\Lab01-Preprocessing-data\Source>python Main.py impute house-prices.csv --column Alley
--outfile result1.csv --method mode
Saved result in result1.csv

C:\Users\T14\Documents\GitHub\Data_Mining\Lab01-Preprocessing-data\Source>python Main.py list_missing result1.csv
There are missing attributes in these following columns:
LotFrontage
MasVnrType
MasVnrArea
BsmtQual
BsmtCond
BsmtExposure
BsmtFinType1
BsmtFinType2
FireplaceQu
GarageType
GarageYrBlt
GarageFinish
GarageQual
GarageCond
PoolQC
Fence
MiscFeature

C:\Users\T14\Documents\GitHub\Data_Mining\Lab01-Preprocessing-data\Source>
```

- Syntax : `python Main.py impute house-prices.csv -c LotFrontage Alley -outfile result2.csv -m median mode`
- All missing LotFrontage values are replaced by its median
- All missing values of Alley are replaced by its mode
- Using function 1 on file result2.csv, the column LotFrontage, Alley is not visible


```
C:\Windows\System32\cmd.exe

C:\Users\T14\Documents\GitHub\Data_Mining\Lab01-Preprocessing-data\Source> python Main.py impute house-prices.csv -c LotFrontage Alley
--outfile result2.csv -m median mode
Saved result in result2.csv

C:\Users\T14\Documents\GitHub\Data_Mining\Lab01-Preprocessing-data\Source>python Main.py list_missing result2.csv
There are missing attributes in these following columns:
MasVnrType
MasVnrArea
BsmtQual
BsmtCond
BsmtExposure
BsmtFinType1
BsmtFinType2
FireplaceQu
GarageType
GarageYrBlt
GarageFinish
GarageQual
GarageCond
PoolQC
Fence
MiscFeature

C:\Users\T14\Documents\GitHub\Data_Mining\Lab01-Preprocessing-data\Source>
```

- Syntax : `python Main.py impute house-prices.csv --outfile result3.csv`
- All missing values in the table have been filled in
- Save to the file `result3.csv`
- Using function 1,2 does not detect missing values

```
C:\Windows\System32\cmd.exe

C:\Users\T14\Documents\GitHub\Data_Mining\Lab01-Preprocessing-data\Source>python Main.py impute house-prices.csv --outfile result3.csv
Saved result in result3.csv

C:\Users\T14\Documents\GitHub\Data_Mining\Lab01-Preprocessing-data\Source>python Main.py list_missing result3.csv
There are missing attributes in these following columns:

C:\Users\T14\Documents\GitHub\Data_Mining\Lab01-Preprocessing-data\Source>python Main.py count_missing result3.csv
The number of rows with missing value: 0

C:\Users\T14\Documents\GitHub\Data_Mining\Lab01-Preprocessing-data\Source>
```

IV. *Deleting rows containing more than a particular number of missing values (Example: delete rows with the number of missing values is more than 50% of the number of attributes).*

- Function 4 needs 2 parameters, threshold, outfile
 - o Threshold is the erase threshold, in percent, default is 50
 - o Outfile indicates the name of the file to be saved
- Syntax : `python Main.py remove_row house-prices.csv -o result4.csv --threshold 10`

```
C:\Windows\System32\cmd.exe

C:\Users\T14\Documents\GitHub\Data_Mining\Lab01-Preprocessing-data\Source>python Main.py remove_row house-prices.csv -o result4.csv --threshold 10
Saved result in result4.csv

C:\Users\T14\Documents\GitHub\Data_Mining\Lab01-Preprocessing-data\Source>python Main.py count_missing result4.csv
The number of rows with missing value: 920

C:\Users\T14\Documents\GitHub\Data_Mining\Lab01-Preprocessing-data\Source>
```

V. *Deleting columns containing more than a particular number of missing values (Example: delete columns with the number of missing values is more than 50% of the number of samples).*

- Function 4 needs 2 parameters, threshold, outfile
 - o Threshold is the erase threshold, in percent, default is 50
 - o Outfile indicates the name of the file to be saved
- Syntax : `python Main.py remove_col house-prices.csv -o result5.csv --threshold 10`

```
C:\Windows\System32\cmd.exe

C:\Users\T14\Documents\GitHub\Data_Mining\Lab01-Preprocessing-data\Source>python Main.py remove_col house-prices.csv -o result5.csv --threshold 10
Saved result in result5.csv

C:\Users\T14\Documents\GitHub\Data_Mining\Lab01-Preprocessing-data\Source>python Main.py list_missing result5.csv
There are missing attributes in these following columns:
MasVnrArea
BsmtQual
BsmtCond
BsmtExposure
BsmtFinType1
BsmtFinType2
GarageType
GarageYrBlt
GarageFinish
GarageQual
GarageCond

C:\Users\T14\Documents\GitHub\Data_Mining\Lab01-Preprocessing-data\Source>
```

VI. *Delete duplicate samples.*

- Function 6 needs 1 outfile parameter.
- Syntax : `python Main.py remove_duplicate house-prices.csv -o result6.csv`

```
C:\Windows\System32\cmd.exe

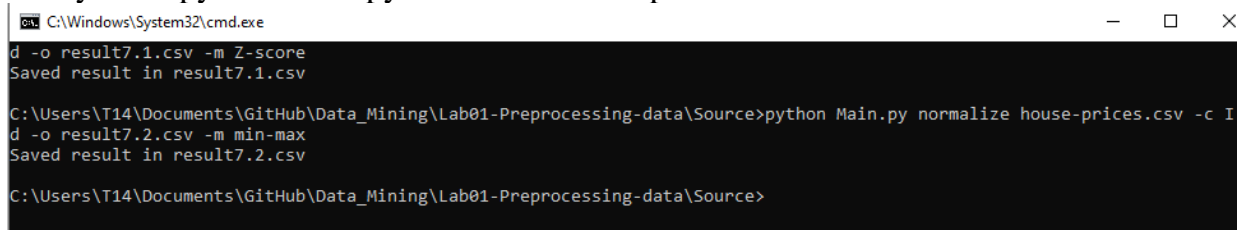
C:\Users\T14\Documents\GitHub\Data_Mining\Lab01-Preprocessing-data\Source>python Main.py remove_duplicate house-prices.csv -o result6.csv
Saved result in result6.csv

C:\Users\T14\Documents\GitHub\Data_Mining\Lab01-Preprocessing-data\Source>python Main.py count_missing result6.csv
The number of rows with missing value: 716

C:\Users\T14\Documents\GitHub\Data_Mining\Lab01-Preprocessing-data\Source>
```

VII. *Normalize a numeric attribute using min-max and Z-score methods.*

- In function 7 use 3 options: column, method, outfile.
- In there:
 - Column can take multiple parameters to indicate columns to normalize. If not specified, all columns are normalized by default. Categorical columns are ignored, invalid name columns are ignored
 - Method indicates normalization method for numeric attribute, supports min-max and Z-score, default is min-max.
 - Outfile indicates the name of the file to be saved, default is the current file. With the following two commands, we have normalized the Id property in two ways:
- Syntax : `python Main.py normalize house-prices.csv -c Id -o result7.1.csv -m Z-score`
- Syntax : `python Main.py normalize house-prices.csv -c Id -o result7.2.csv -m min-max`



```

C:\Windows\System32\cmd.exe
d -o result7.1.csv -m Z-score
Saved result in result7.1.csv

C:\Users\T14\Documents\GitHub\Data_Mining\Lab01-Preprocessing-data\Source>python Main.py normalize house-prices.csv -c Id
d -o result7.2.csv -m min-max
Saved result in result7.2.csv

C:\Users\T14\Documents\GitHub\Data_Mining\Lab01-Preprocessing-data\Source>
  
```

VIII. *Performing addition, subtraction, multiplication, and division between two numerical attributes*

- Function 4 needs 2 parameters, formula, outfile.
 - Formula is formatted as =. For complex operations involving round brackets, the formula (or just the part after the equal sign) must be enclosed in double quotes.
 - outfile same function as above.
- Syntax :
 - `python Main.py calculate house-prices.csv -o result8.1.csv --formula MyCol=Id+MSSubClass`
 - `python Main.py calculate house-prices.csv -o result8.1.csv --formula MyCol=Id-MSSubClass`
 - `python Main.py calculate house-prices.csv -o result8.1.csv --formula MyCol=Id*MSSubClass`
 - `python Main.py calculate house-prices.csv -o result8.1.csv --formula MyCol=Id/MSSubClass`

```
C:\Windows\System32\cmd.exe

C:\Users\T14\Documents\GitHub\Data_Mining\Lab01-Preprocessing-data\Source>python Main.py calculate house-prices.csv -o result8.1.csv --formula MyCol=Id+MSSubClass
Saved result in result8.1.csv

C:\Users\T14\Documents\GitHub\Data_Mining\Lab01-Preprocessing-data\Source>python Main.py calculate house-prices.csv -o result8.2.csv --formula MyCol=Id-MSSubClass
Saved result in result8.2.csv

C:\Users\T14\Documents\GitHub\Data_Mining\Lab01-Preprocessing-data\Source>python Main.py calculate house-prices.csv -o result8.3.csv --formula MyCol=Id*MSSubClass
Saved result in result8.3.csv

C:\Users\T14\Documents\GitHub\Data_Mining\Lab01-Preprocessing-data\Source>python Main.py calculate house-prices.csv -o result8.4.csv --formula MyCol=Id/MSSubClass
Saved result in result8.4.csv

C:\Users\T14\Documents\GitHub\Data_Mining\Lab01-Preprocessing-data\Source>
```

III. Requirement

1. **Completion level:** for us perspective ,we just completed 98%
2. **Division of tasks :**
 - 20127448 :
 - 3.2.1 Exploring Breast Cancer data set
 - 3.3 Preprocessing Data in Python
 - ⇒ Task : 1, 3, 5, 7
 - 20127444 :
 - 3.2.2 Exploring Weather data set
 - 3.3 Preprocessing Data in Python
 - ⇒ Task : 2, 4, 6, 8