Data mining Lab 01 – Data preprocessing

Contents

Student information 1
Self-evaluation 1
Requirement 1 – Weka installation 2
Requirement 2 – Begin with Weka 4
Requirement 2.1 – Reading data 4
Requirement 2.2 – Exploring weather dataset 8
Requirement 2.3 – Exploring German credit dataset 11
Requirement 3 – Preprocessing implementation 20
List missing 20
Imputing 22
Remove sample and attribute 24
Remove sample with threshold 24
Remove attribute with threshold 25
Remove duplicated samples 26
Normalizing 26
Add new attribute with given expression 28
References 30

Student information

In this lab, we work in the group of two whose information is shown in the following table:

Class	19KHMT		
No	**Student name**	**Student ID**	**Contribution**
1	Do Vuong Phuc	19127242	50%
2	Bui Dang Khoa	19127645	50%

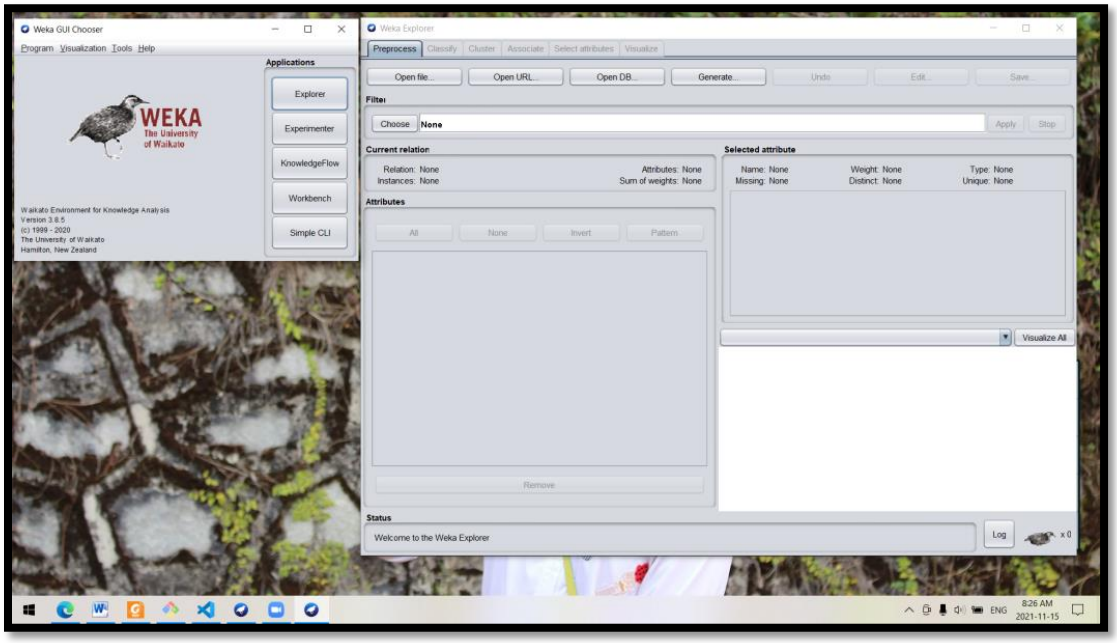
Self-evaluation

We have done all the given task from installation to the implementation with detail document.

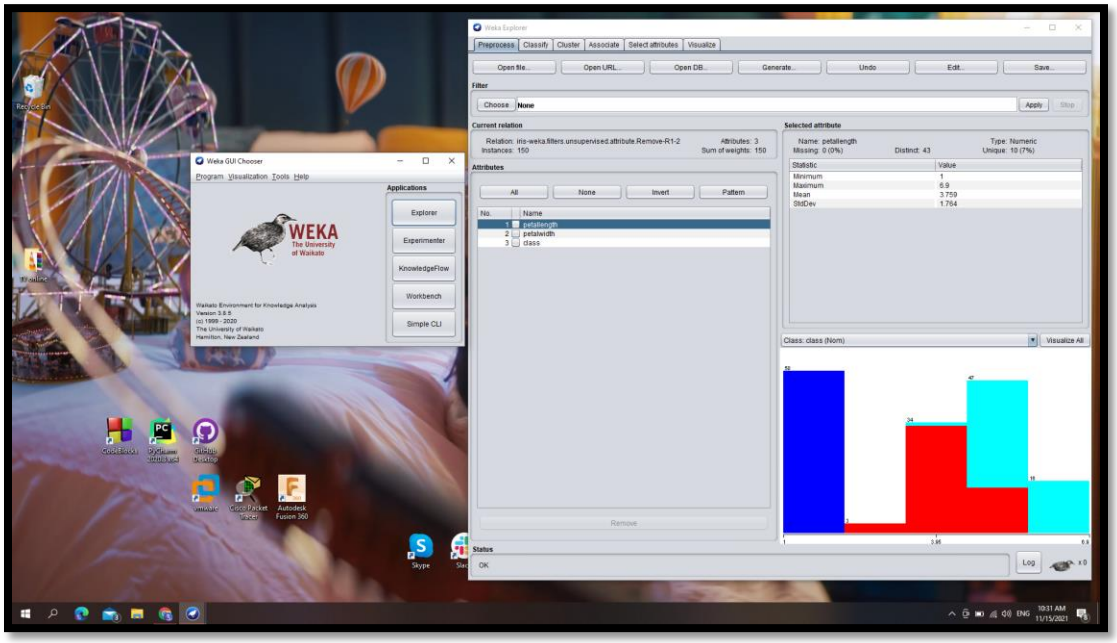
Requirement 1 – Weka installation

1. **Screenshot of Weka with desktop**

Do Vuong Phuc's screenshot:



Bui Dang Khoa's screenshot:



2. Explanation for groups of preprocess tab:

- **Filter:** this tab let us filter the data in many way (such as discrete attributes, select the suitable attribute, etc)
- **Current relation:** provides a detailed and intuitive view of the data
    - Relation: show the name of the relationship
    - Instance: number of record in the data (row)
    - Attributes: number of attribute in the data (column)
- **Attributes:** provides the attributes of the relation (has 3 columns)
    - Column 1 (No.): show the numbering of attributes
    - Column 2 (Selected cell): allow user decide which attribute to interact with (such as remove).
    - Name: display name of attribute
    - Instead of selecting one by one attribute, we can use All/None/Invert/Pattern to select attribute
    - We can use "Remove" button to remove the attribute out of the relation
- **Selected attribute:** display the feature and information of each selected attribute
    - Name: display name of selected attribute
    - Type: the type of attribute (Nominal or Numeric)
    - Missing: number and rate of missing value of selected attribute.
    - Distinct: number of distinct value of selected attribute.
    - Unique: number and rate of unique value of selected attribute.
    - Underneath these information, we can see the properties of each value for the selected attribute (including: numbering, label (value), counting sample and its total weight)
    - Moreover, at the bottom, you can see the visualize of the **class** values. If you click, visualize all, it gives you the visualization by each attribute
- **Status:**
    - Log: show history of work

3. Explanation tabs:

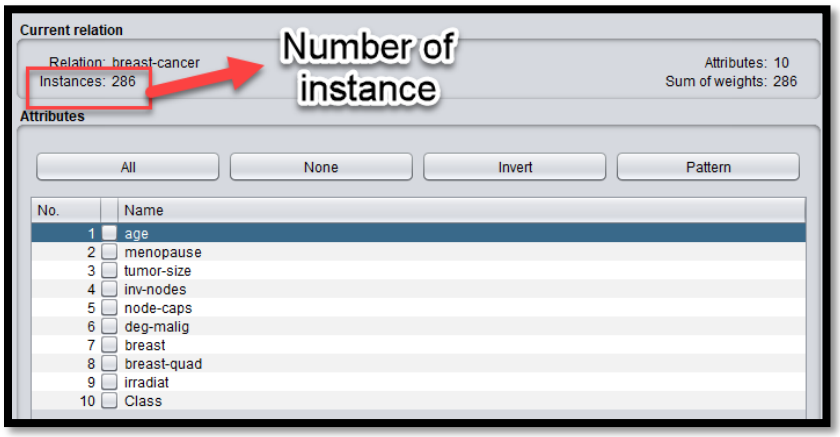
- Preprocess: manipulate the data before processing data.
- Classify: train and valid the classification model or regression model
- Cluster: clustering the data or divide data into clusters
- Associate: generate association rules from data
- Select attributes: choose the most suitable attribute base on the metric
- Visualize: graphical representation of information and data

Requirement 2 – Begin with Weka

Requirement 2.1 – Reading data

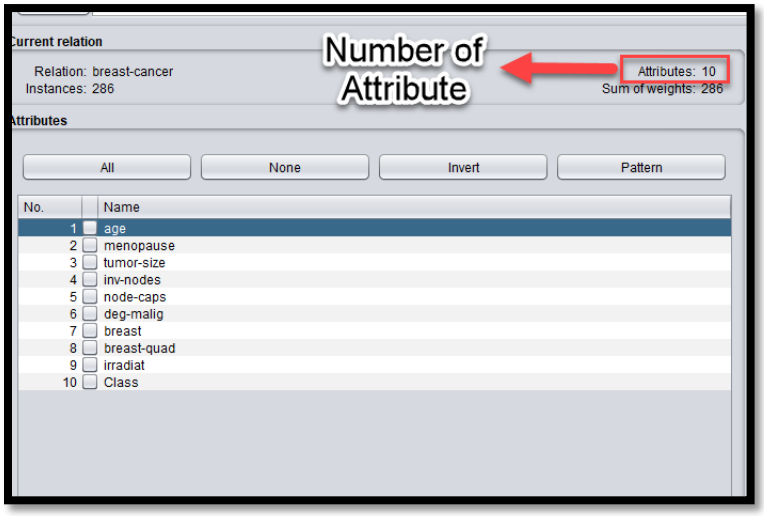
1. **How many instance are there?**

   Number of instance: 286 instances



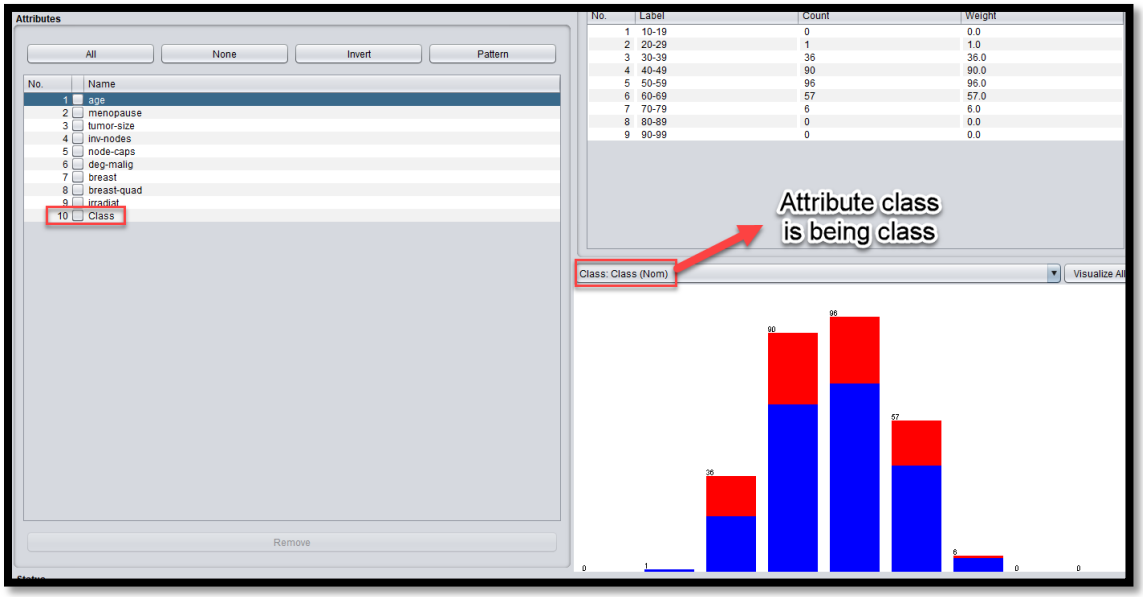
2. **How many attributes does an instance have?**

   Number of attributes: 10 attributes

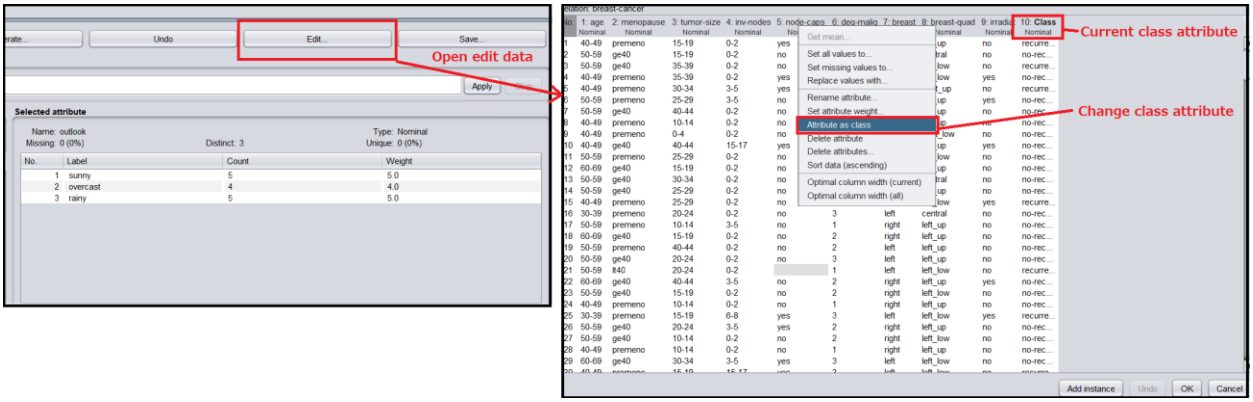


3. **Which attribute represents for the class? Can we change the class attribute and how?**

Attribute **"Class"** is being class. To change the class attribute, we select the drop down list whose label "Class" above the graph, then choose the attribute we want.



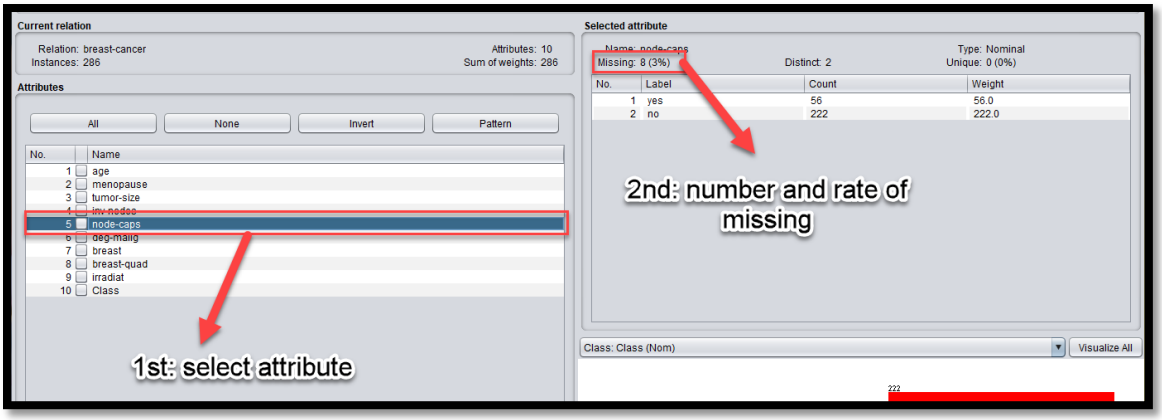
Moreover, we can see and change the attribute class by editing the data:



4. **For each attribute, how many missing values are there? Which attribute is the most (and the least) missing? How can we deal with the missing problem?**
    - The most missing: "node-caps" attribute (8 samples, 3%)
    - The least missing: Age, Menopause, Tumor-size, Inv-nodes, Deg-malig, Breast, Irradiat, Class attributes (0 samples, 0%)
    - The following table show the number of missing sample for each attribute:

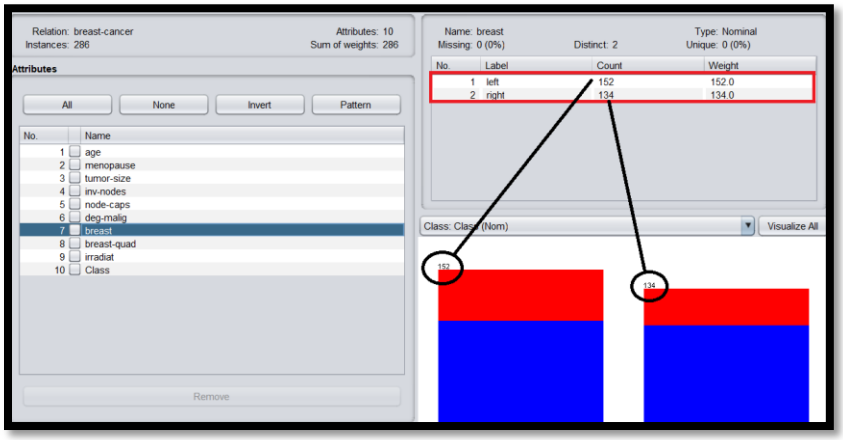
Attribute	Missing
Age	0 (0%)
Menopause	0 (0%)
Tumor-size	0 (0%)
Inv-nodes	0 (0%)
Node-caps	8 (3%)
Deg-malig	0 (0%)
Breast	0 (0%)
Breast-quad	1 (0%)
Irradiat	0 (0%)
Class	0 (0%)

    - To deal with the missing problem, we can:
        - Delete the attribute whose sample is missing
        - Delete the samples whose attribute is missing
        - Data imputation: Fill the missing with mean, median, mode; or using K-NN, Linear regression; or fill with -1, -99, -999



5. **Explain the meaning of graphs of Explorer window. What name is the most suitable for the graph if you have to assign to it? What is represented for the red and blue color? What does the graph represent for?**

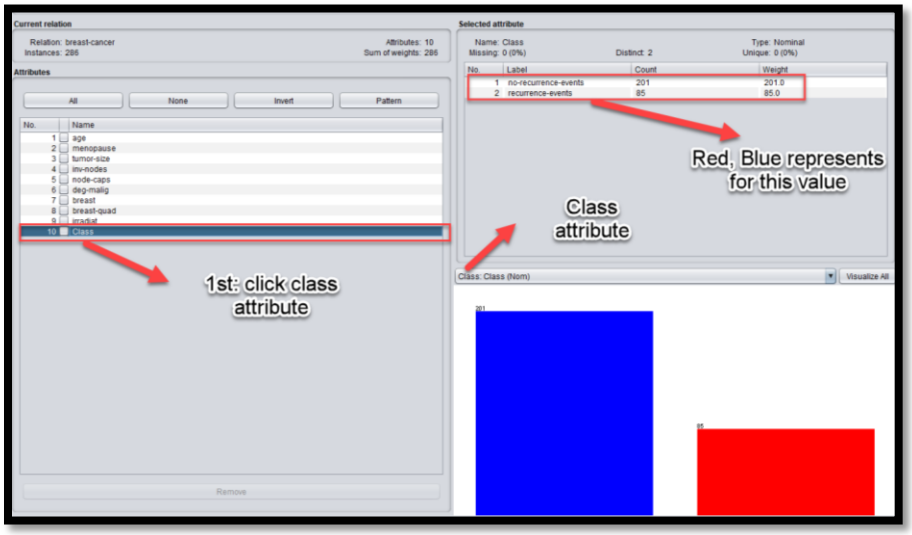
The graph shows the distribution of the selected attribute. Each column is represented for a value of the attribute and the colors stand for the class attribute. For instance, the picture below shows that there are 152 value of left and 134 value of right one.



Each color represents for a value of class attribute, in this case:

- Red: recurrence-events
- Blue: no-Recurrence-events

We will know this by clicking on the class attribute and observe its color.



So, we can name the graph as “the split chart that compares the number of samples between two classes in terms of the selected attribute”. For example, while choosing the “breast” attribute, the graph will be named “the split chart that compares the number of samples between no-recurrence-events and recurrence-events in terms of breast”.

Requirement 2.2 – Exploring weather dataset

1. **How many attributes are there? How many samples? What kind of datatype for each attribute? Which attribute is class attribute?**

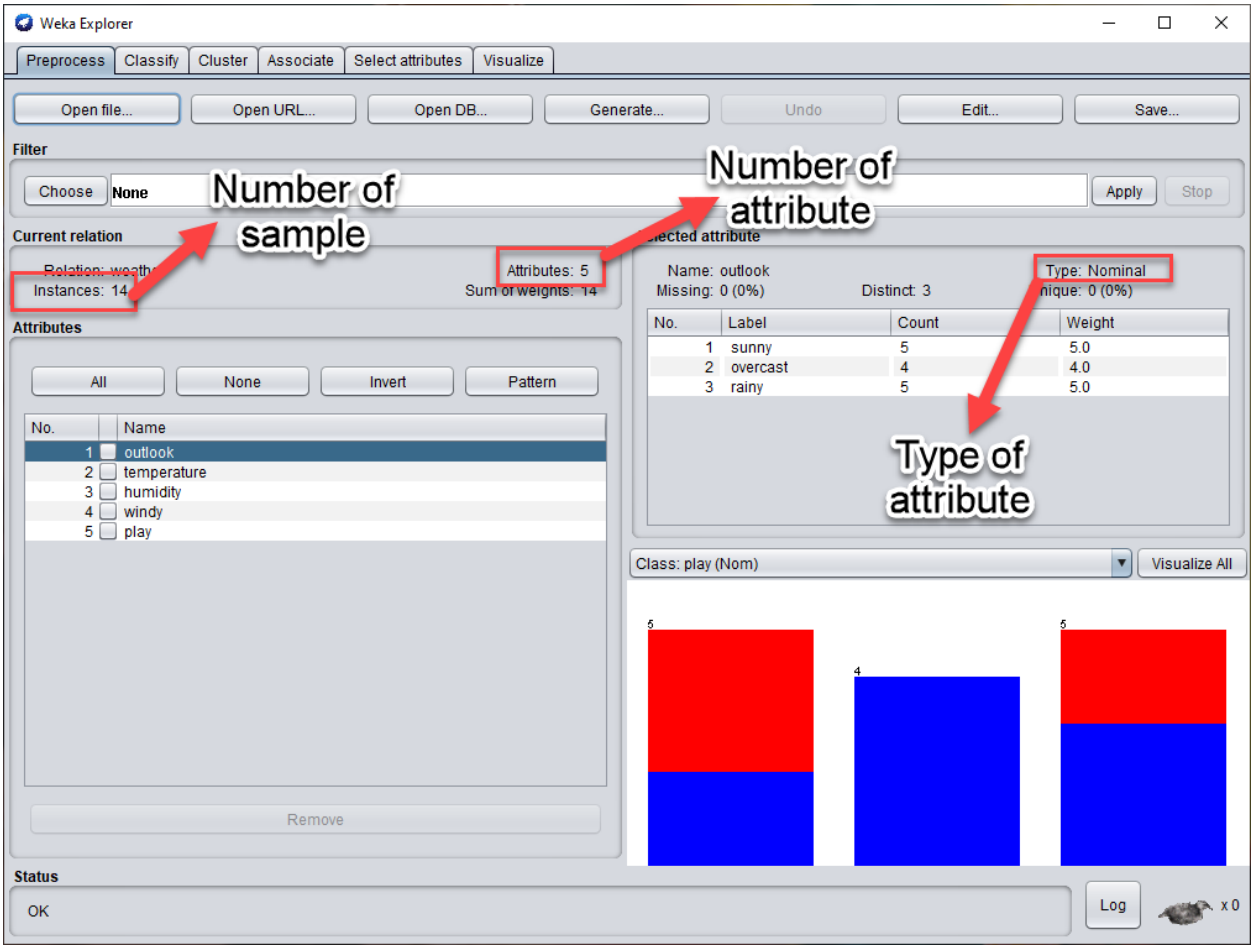
   Number of sample: 14 samples

   Number of attribute: 5 attributes

   In "selected attribute" tab, we can see the "Type" of each attribute. The datatypes of each attribute are:

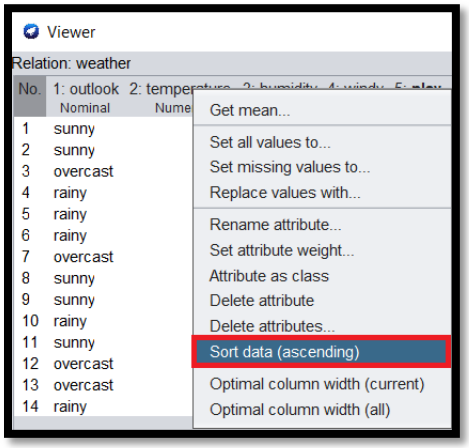
   - Nominal: outlook, windy, play
   - Numeric: temperature, humidity

   Class attribute: play



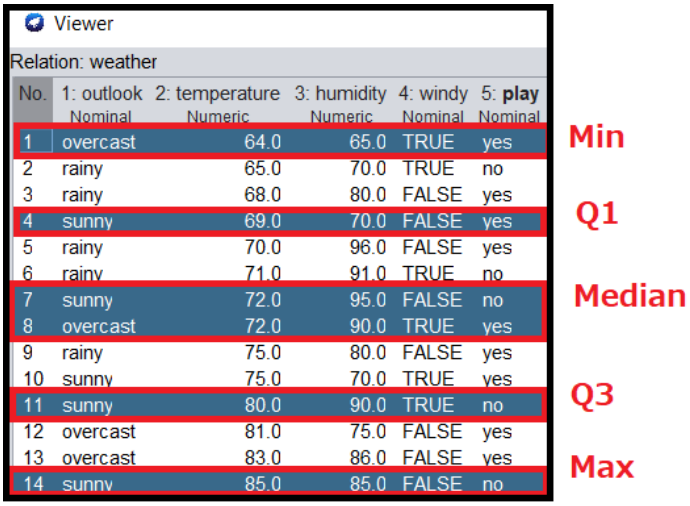
2. **List down five-number summary of tempurature and humidity. Does Weka provides any feature to calculate this?**

   Weka does not provide any feature that have us the calculate the five-number summary of the attribute. To find these summary, we sort the data ascendingly using the "edit" feature:



For the tempurature:

- Min (0%): 64
- Q1 (25%): 69
- Median (50%): 72
- Q3 (75%): 80
- Max (100%): 85



For the humidity:

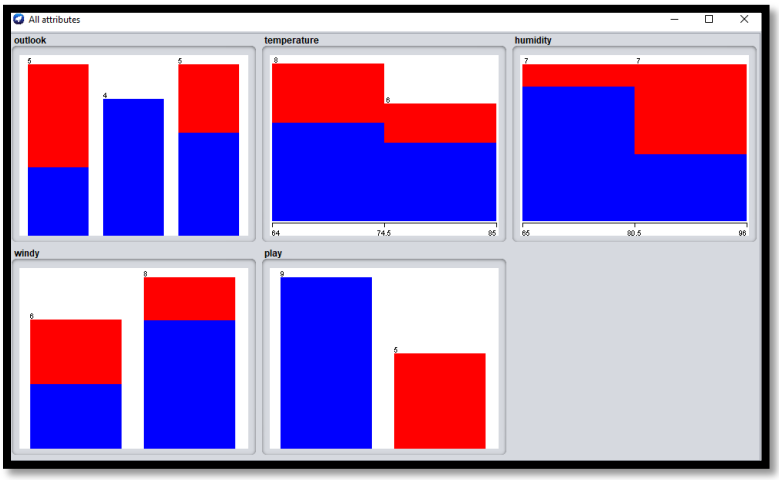
- Min (0%): 65
- Q1 (25%): 70
- Median (50%): 82.5
- Q3 (75%): 90
- Max (100%): 96

Viewer

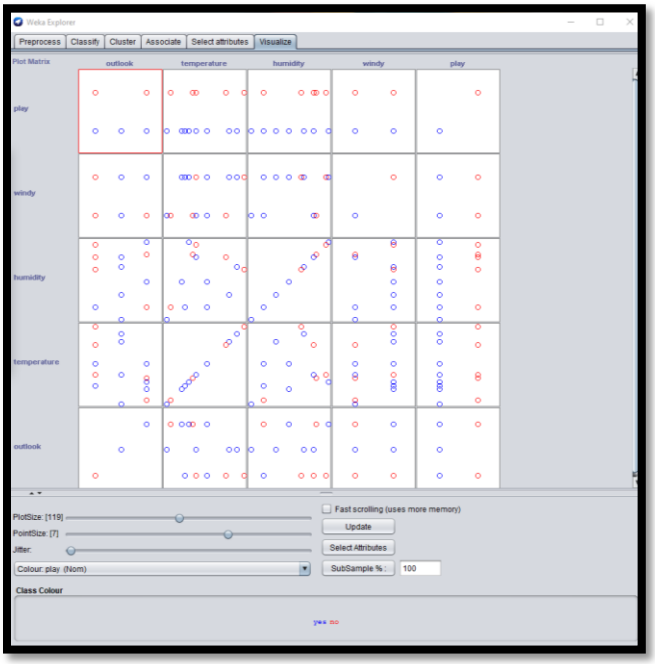
Relation: weather

No.	1: outlook Nominal	2: temperature Numeric	3: humidity Numeric	4: windy Nominal	5: play Nomina	
1	overcast	64.0	65.0	TRUE	yes	Min
2	rainy	65.0	70.0	TRUE	no	
3	sunny	69.0	70.0	FALSE	yes	
4	sunny	75.0	70.0	TRUE	yes	Q1
5	overcast	81.0	75.0	FALSE	yes	
6	rainy	68.0	80.0	FALSE	yes	
7	rainy	75.0	80.0	FALSE	yes	Median
8	sunny	85.0	85.0	FALSE	no	
9	overcast	83.0	86.0	FALSE	yes	
10	overcast	72.0	90.0	TRUE	yes	
11	sunny	80.0	90.0	TRUE	no	Q3
12	rainy	71.0	91.0	TRUE	no	
13	sunny	72.0	95.0	FALSE	no	
14	rainy	70.0	96.0	FALSE	yes	Max

3. **Alternatively look at the graph of each attribute and take screenshot?**
   Click "Visualize All" to look at the graph of each attribute.



4. **What is the terminology we call the graph in "Visualize" tab? Choose the appropriate jitter to see the distribution better? In your opinion, which is the pair of different attributes that one attribute correlate with the other one?**



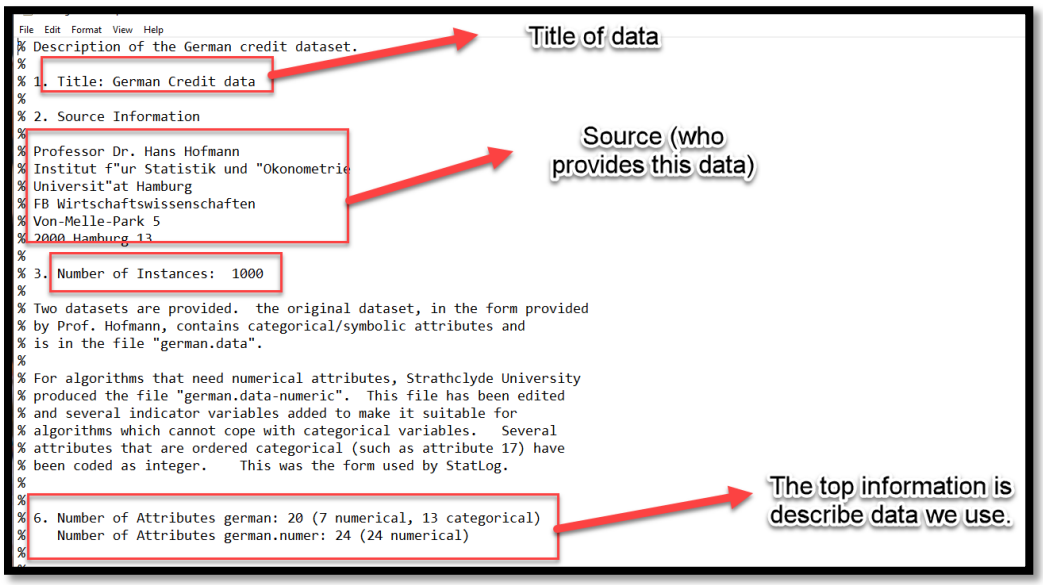
The terminology for this graph is called "pairwise scatterplot matrix". In our opinion, the pair of attributes which have the most correlative is "Outlook" and "Tempurature" since the tempurature is low whenever it rains; and high as sunny.

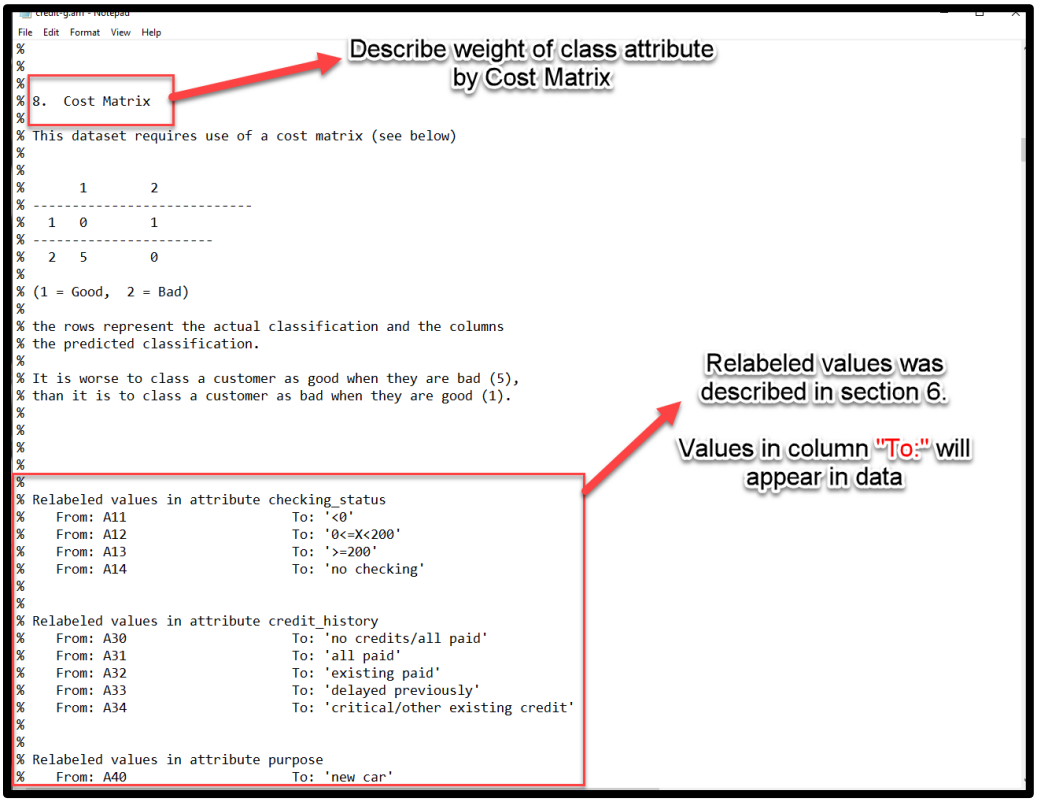
Requirement 2.3 – Exploring German credit dataset

1. **What did the comment of the dataset say? How many instances and attributes are there? Describe 5 attribute arbitrarily (must have both discrete and continuous)**

   Comment of the dataset said:

   - Section 1: title of data (German credit data)
   - Section 2: source information (Dr. Hans Hofmann, etc)
   - Section 3: number of instance which is 1000 instances
   - There are 2 dataset (categorical and numerical)
   - Section 6: the number of attribute. We use data "credit-g.arff", don't use "credit-g.numer.arff"
     - File "german": 20 attributes (7 numerical, 13 categorical) without class attribute
     - File "german.numer": 24 attributes (24 numerical) without class attribute
   - Section 7: describes the meaning of attributes for "german" (credit-g.arff) file
   - Section 8: provide the cost matrix for class attribute; and value for attributes
   - Notice that the section 7 only provides the meaning of attribute, not their values. Their values is relabeled in section 8.

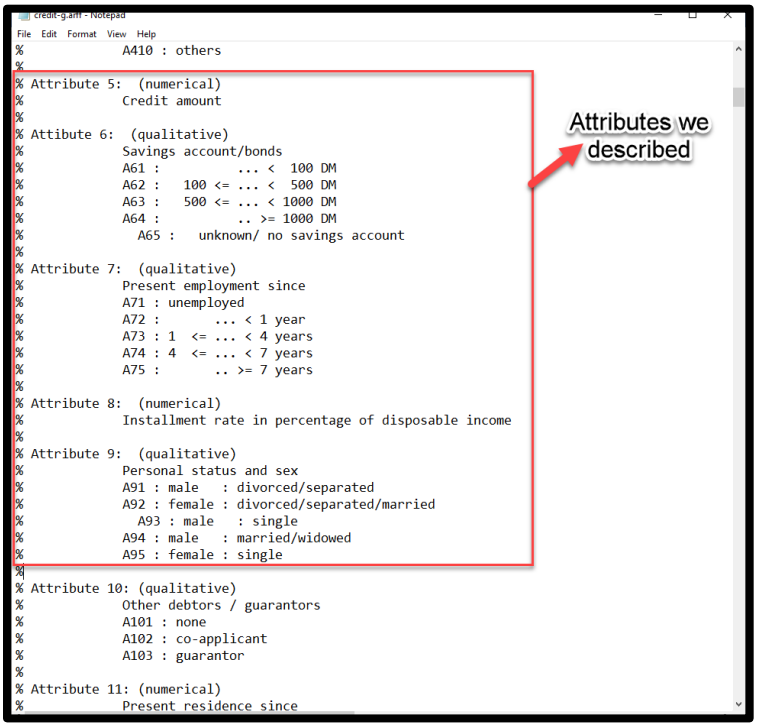




Describe 5 attributes:

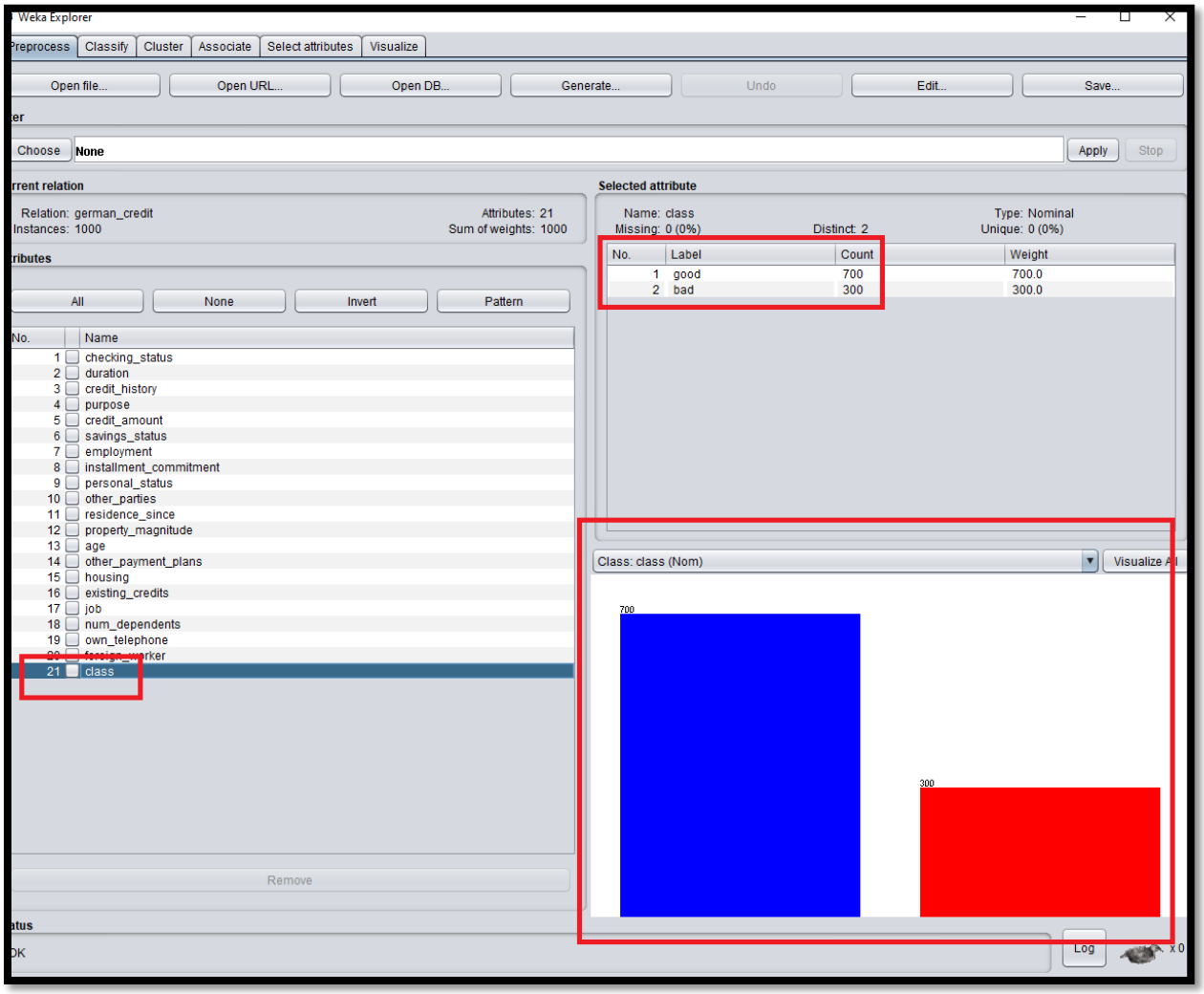
- Attribute 5 (numerical):
  - Credit Amount
- Attribute 6 (qualitative):
  - Saving account/bonds
  - Have 5 values:

        - Value 1 (A61): less than 100 DM
        - Value 2 (A62): from 100 DM to 500 DM
        - Value 3 (A63): from 500 DM to 1000 DM
        - Value 4 (A64): More than 1000 DM
        - Value 5 (A65): unknown/no savings account
- Attribute 7: (Qualitative)
    - Present employment since a period of time
    - Have 5 values:
        - Value 1 (A71): unemployed
        - Value 2 (A72): less than 1 year
        - Value 3 (A73): from 1 year to 4 years
        - Value 4 (A74): from 4 years to 7 years
        - Value 5 (A75): more than 7 years
- Attribute 8: (numerical)
    - Installment rate in percentage of disposable income
- Attribute 9: (qualitative)
    - Personal status and sex of the sample
    - Have 5 values:
        - Value 1 (A91): male and divorced/separated
        - Value 2 (A92): female and divorced/separated/married
        - Value 3 (A93): male and single
        - Value 4 (A94): male and married/widowed
        - Value 5 (A95): female and single



2. **Which attribute is the class attribute? Evaluate the distribution of classes, means that is it balance or being skew?**

   "Class" is name of the class attribute and the distribution of classes is **skew to "good"** value.



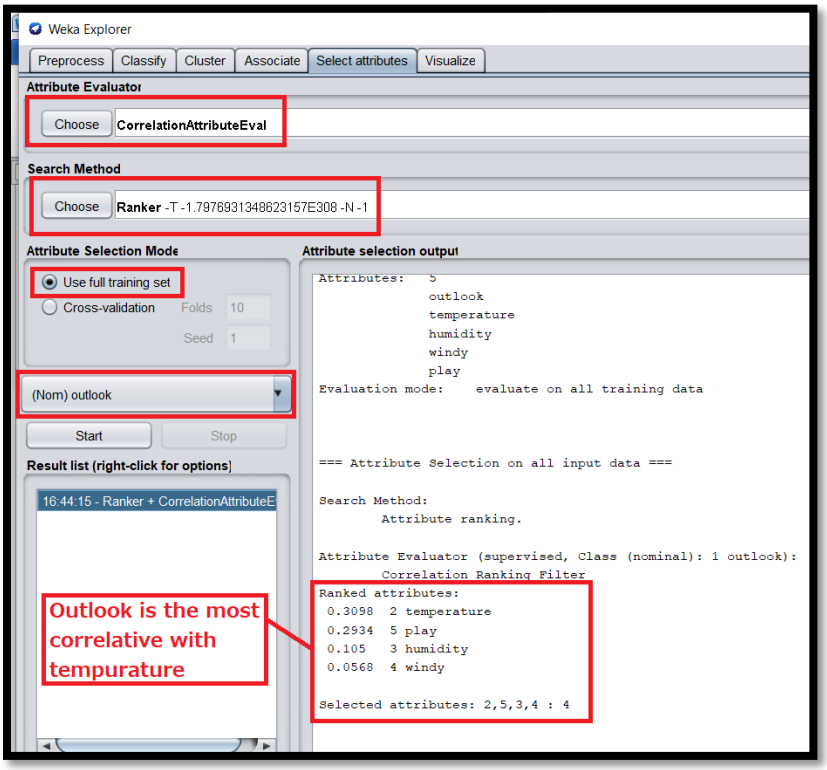
3. Using the "Select attribute" tab, list down and describe briefly for each method

a. Correlation Based Feature Selection (CorrelationAttributeEval):

Calculate the correlation (Pearson's) between each attribute and the class attribute. There are three kinds of Pearson's correlation: Positive, Neural and Negative.

After using this technique, we can choose the moderate to high positive or negative correlation and drop the neural (low correlation) one.

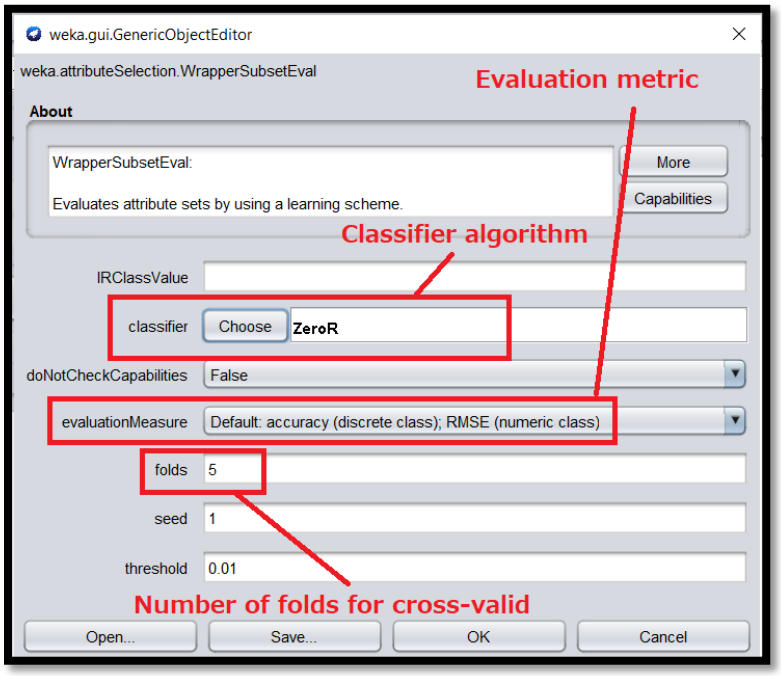
For instance, we tested this feature on the weather dataset for the "outlook" class and achieved that the most correlative attribute is "tempurature".



b. Information Gain Based Feature Selection (InfoGainAttributeEval):

Calculate the information gain (also called entropy) for each attribute for the output variable. The value is in range from 0 (no information) to 1 (maximum information). Those attributes that contribute more information will have a higher information gain value which can be selected. On the other hand, the low information gain attribute can be removed. This also used to execute the ID3 algorithm (Decision tree).

c. Learner Based Feature Selection (WrapperSubsetEval):
The subset that results in the best performance is taken as the selected subset one. It runs the classifier with cross-validation and choose based on the picked metric.



d. Correlation Based Feature Selection Subset (CfsSubsetEval):
This method evalutes how correlative that a subset (of attributes) to the class attribute. The subsets whose element are less intercorrelative, but the subset highly correlated to the target class are preferred (elements should as independent as possible).

e. Gain Ratio Attribute evaluation (GainRatioAttributeEval):
This method measures the significance of attributes with respect to target class on the basis of gain ratio. It can be calculated by the following formula:

$$GainR(Class, Attribute) = \frac{H(Class) - H(Class \mid Attribute)}{H(Attribute)}$$

Where H represents for the Entropy function

f. Classifier Attribute Evaluation (ClassifierAttributeEval):
Evaluates the worth of an attribute by using a user-specified classifier.

g. Classifier Subset Evaluator (ClassifierSubsetEval):
Evaluates attribute subsets on training data or a separate hold out testing set. After that it uses the given classifier (by user) to estimate the set of attribute

h. One R Attribute Evaluation (OneRAttributeEval) :
Evaluates the worth of an attribute by using the OneR classifier.

i. Principal components:
Performs a principal components analysis (PCA) and transformation of the data

j. Relief F Attribute Evaluation (ReliefFAttributeEval):
Evaluates the worth of an attribute by repeatedly sampling an instance and considering the value of the given attribute for the nearest instance of the same and different class.

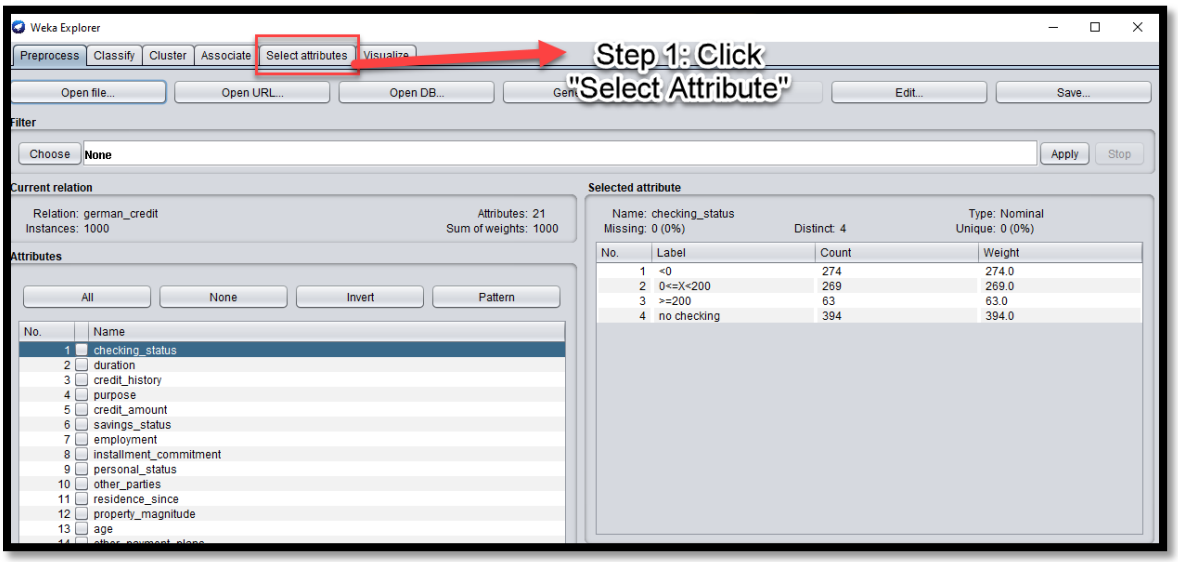
k. Symmetrical Uncert Attribute Evaluation (SymmetricalUncertAttributeEval) :
Evaluates the worth of an attribute by measuring the symmetrical uncertainty with respect to the class.

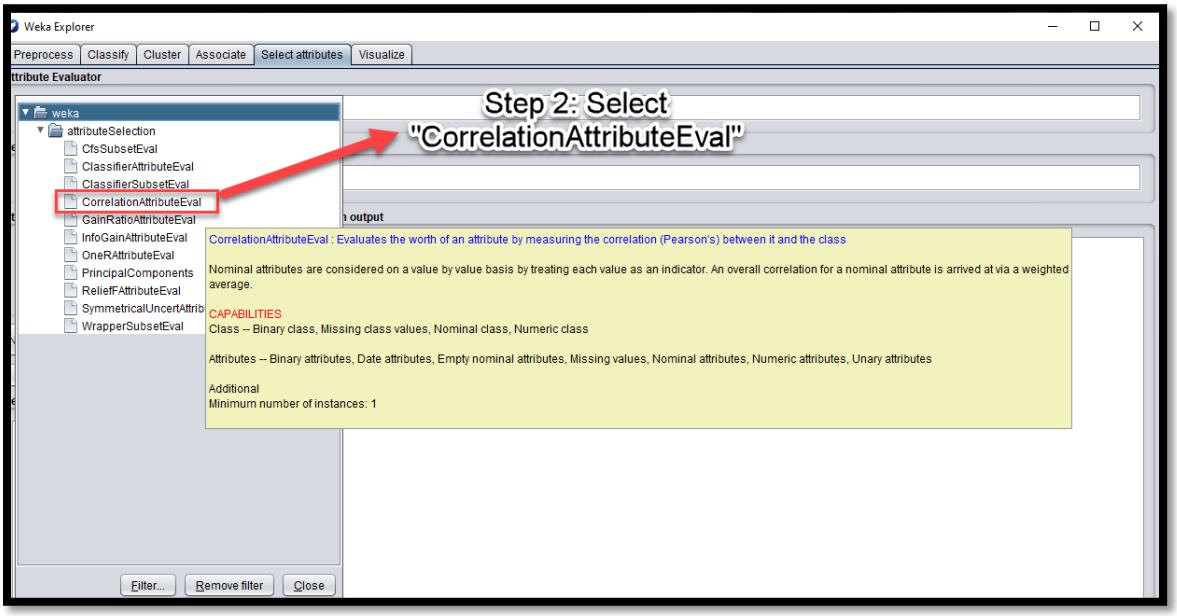
4. Which filter should we use to find out 5 most correlative attributes (to the class attribute)? Describe step-by-step and include the screenshot

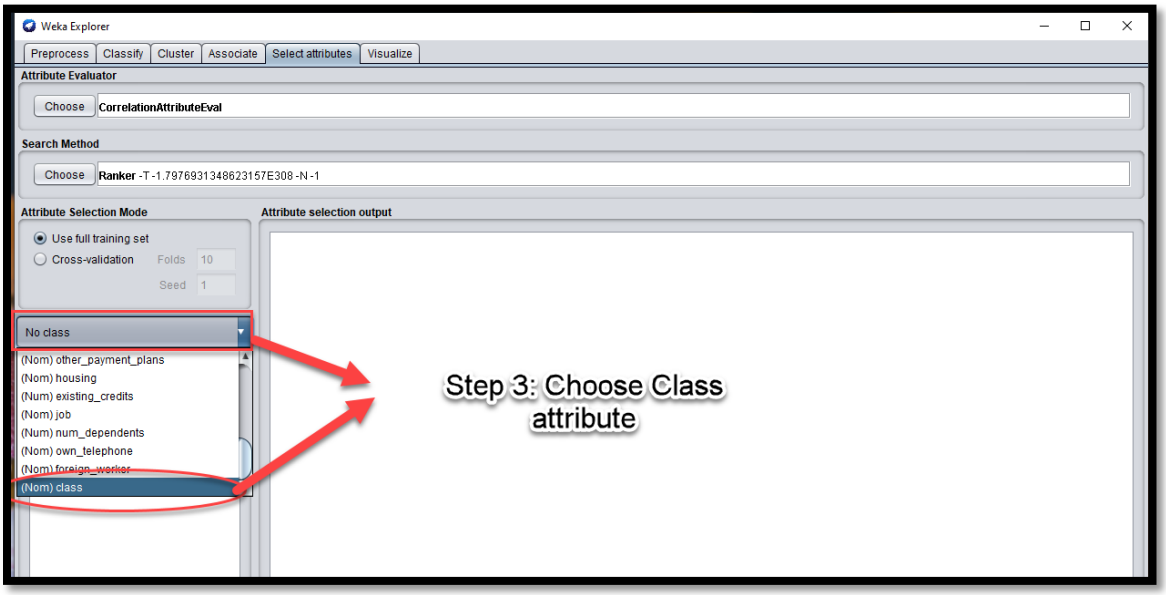
To find the most correlative attributes, we do the following step:

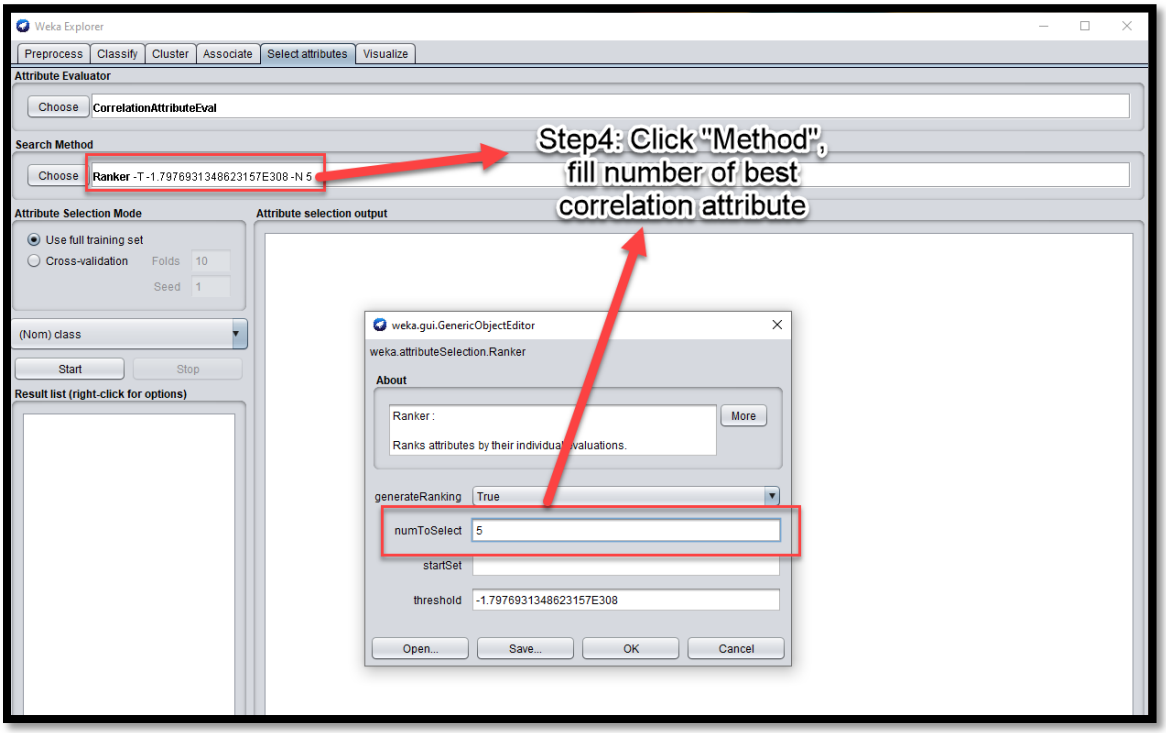
- Step 1: Click “Select Attribute” tab
- Step 2: Choose Attribute Evaluator “Correlation Attribute Eval”
- Step 3: Choose class attribute which is “class”
- Step 4: Choose Search Method “Ranker” with number to select is Top-5
- Step 5: Click Start

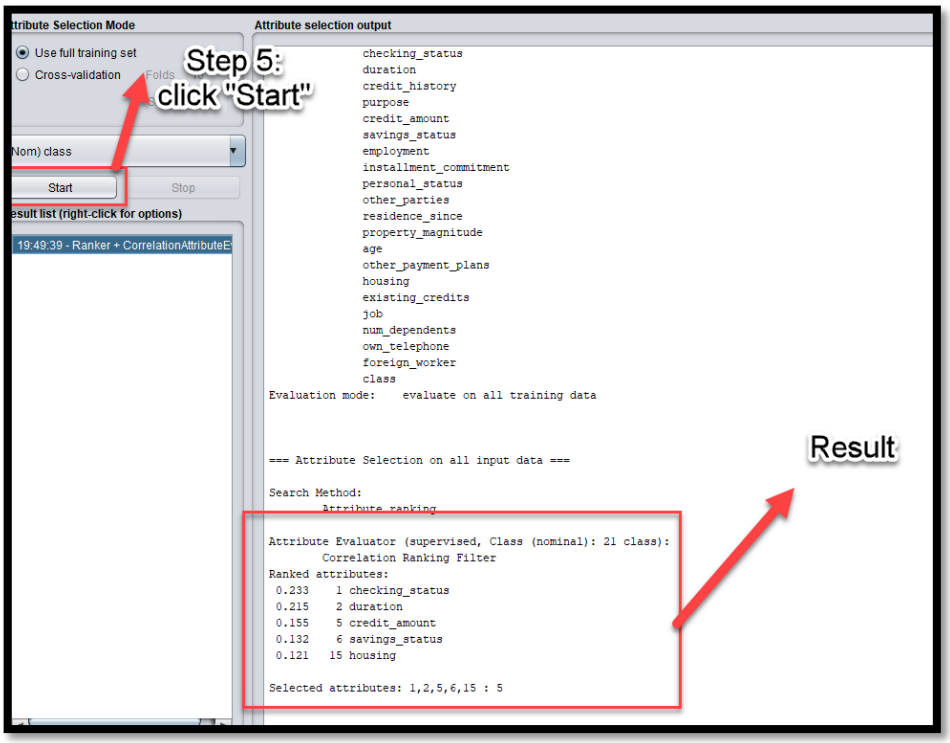
And the result show top-5 attributes are: checking_status, duration, credit_amount, savings_status and housing.




Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Step 2: Select "CorrelationAttributeEval"
weka
attributeSelection
CfsSubsetEval
ClassifierAttributeEval
ClassifierSubsetEval
CorrelationAttributeEval
GainRatioAttributeEval
InfoGainAttributeEval
OneRAttributeEval
PrincipalComponents
ReliefFAttributeEval
SymmetricalUncertAttrib
WrapperSubsetEval
CorrelationAttributeEval : Evaluates the worth of an attribute by measuring the correlation (Pearson's) between it and the class
Nominal attributes are considered on a value by value basis by treating each value as an indicator. An overall correlation for a nominal attribute is arrived at via a weighted average.
CAPABILITIES
Class -- Binary class, Missing class values, Nominal class, Numeric class
Attributes -- Binary attributes, Date attributes, Empty nominal attributes, Missing values, Nominal attributes, Numeric attributes, Unary attributes
Additional
Minimum number of instances: 1
Filter...
Remove filter
Close


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Attribute Evaluator
Choose
CorrelationAttributeEval
Search Method
Choose
Ranker -T -1.7976931348623157E308 -N -1
Attribute Selection Mode
Use full training set
Cross-validation
Folds
10
Seed
1
Attribute selection output
No class
(Nom) other_payment_plans
(Nom) housing
(Num) existing_credits
(Nom) job
(Num) num_dependents
(Nom) own_telephone
(Nom) foreign_worker
(Nom) class
Step 3: Choose Class attribute


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Attribute Evaluator
Choose
CorrelationAttributeEval
Search Method
Choose
Ranker -T -1.7976931348623157E308 -N 5
Step4: Click "Method", fill number of best correlation attribute
Attribute Selection Mode
Use full training set
Cross-validation
Folds
10
Seed
1
(Nom) class
Start
Stop
Result list (right-click for options)
Attribute selection output
weka.gui.GenericObjectEditor
weka.attributeSelection.Ranker
About
Ranker :
Ranks attributes by their individual evaluations.
More
generateRanking
True
numToSelect
5
startSet
threshold
-1.7976931348623157E308
Open...
Save...
OK
Cancel


Attribute Selection Mode
Use full training set
Cross-validation
(Nom) class
Start
Stop
Result list (right-click for options)
19:49:39 - Ranker + CorrelationAttributeE
Step 5: click "Start"
Attribute selection output
checking_status
duration
credit_history
purpose
credit_amount
savings_status
employment
installment_commitment
personal_status
other_parties
residence_since
property_magnitude
age
other_payment_plans
housing
existing_credits
job
num_dependents
own_telephone
foreign_worker
class
Evaluation mode: evaluate on all training data
=== Attribute Selection on all input data ===
Search Method:
Attribute ranking.
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
Result

Requirement 3 – Preprocessing implementation

Firstly, in our solution, we have written a readme document ("README.md") to instruct the user how to use the source code. Moreover, in readme document, we also jot down the implementation document. In this task, we divide our repository into folder:

- data: contains input and output data
- src: stores the main solution
- readme.md: document for usage and implementation

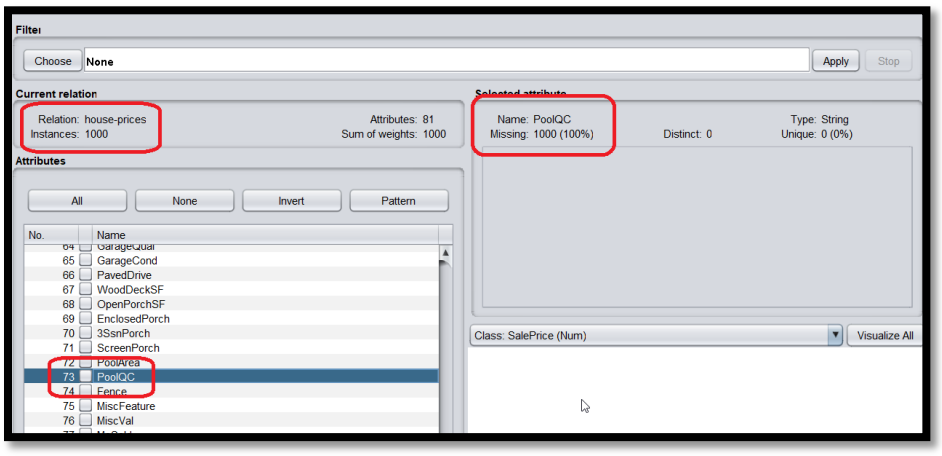
In our testing phase, we use the given data "house-prices.csv" for testing and compare the result with the one given by WEKA and excel.

List missing

Firstly, we list out the missing feature of the house-prices data and achieve this result.

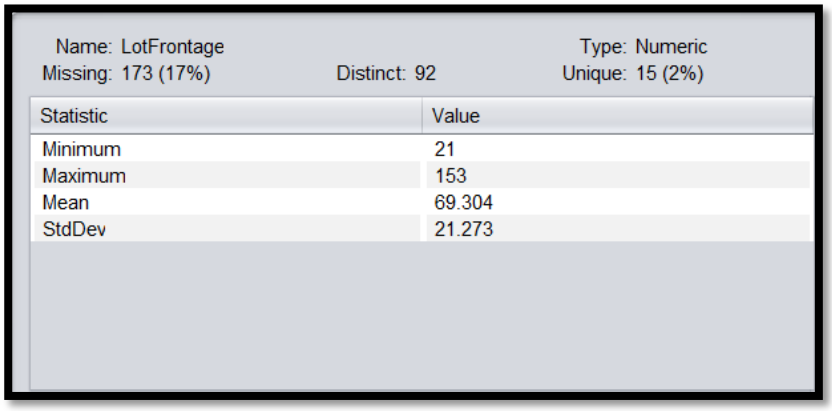
```
C:\Users\VuongPhuc\Desktop\DataMining-Preprocessing\src>list-missing.py ../data/house-prices.csv
Number of samples: 1000
Number of missing samples: 1000
The missing attribute:
        LotFrontage: 173
        Alley: 941
        MasVnrType: 593
        MasVnrArea: 10
        BsmtQual: 27
        BsmtCond: 27
        BsmtExposure: 28
        BsmtFinType1: 27
        BsmtFinType2: 29
        FireplaceQu: 501
        GarageType: 60
        GarageYrBlt: 60
        GarageFinish: 60
        GarageQual: 60
        GarageCond: 60
        PoolQC: 1000
        Fence: 815
        MiscFeature: 963
```

Comparing to WEKA, we have the same number of samples and number of missing sample. Moreover, the number of missing sample is 1000 as all values of attribute PoolQC are missing.



And we check for some attributes which is missing using Weka:

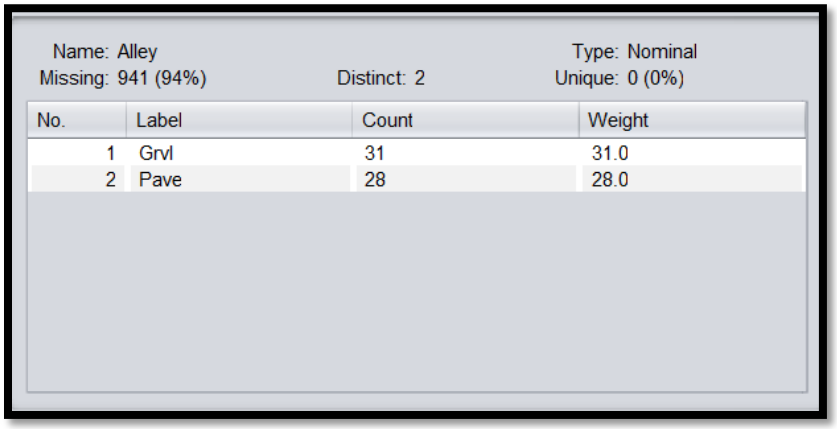
- LotFrontage: 173 missing samples



Name: LotFrontage Type: Numeric
Missing: 173 (17%) Distinct: 92 Unique: 15 (2%)

Statistic	Value
Minimum	21
Maximum	153
Mean	69.304
StdDev	21.273

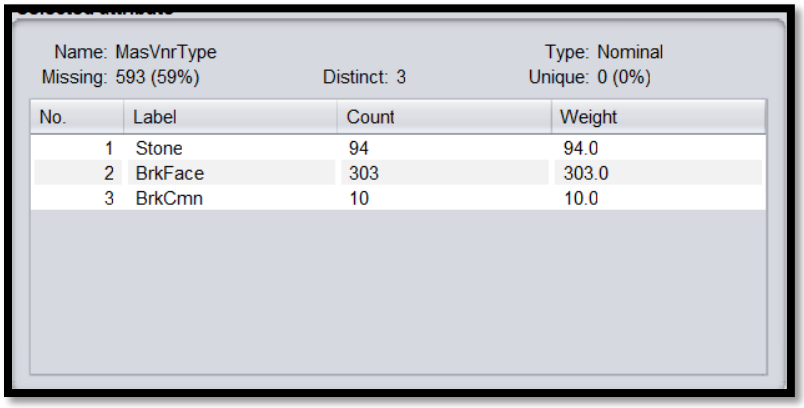
- Alley: 941 missing samples



Name: Alley Type: Nominal
Missing: 941 (94%) Distinct: 2 Unique: 0 (0%)

No.	Label	Count	Weight
1	Grvl	31	31.0
2	Pave	28	28.0

- MasVrnType: 593 missing samples



Name: MasVnrType Type: Nominal
Missing: 593 (59%) Distinct: 3 Unique: 0 (0%)

No.	Label	Count	Weight
1	Stone	94	94.0
2	BrkFace	303	303.0
3	BrkCmn	10	10.0

Imputing

Now, we doing the imputation for missing data. First thing first, we test that the source code must notify whenever the user pick the wrong method. In particularly, we try to impute the Alley attribute which is a nominal with the median method:

```
C:\Users\VuongPhuc\Desktop\DataMining-Preprocessing\src>impute.py ../data/house-prices.csv
 --method=median --columns Alley --out=../data/output.csv
Method for nominal attribute must be method="MODE"
```

As the Alley attribute is nominal, we use the mode method to impute this attribute and using mean method for the LotFrontage attribute.

Viewer

Relation: house-prices

No.	1: Id Numeric	2: MSSubClass Numeric	3: MSZoning Nominal	4: LotFrontage Numeric	5: LotArea Numeric	6: Street Nominal	7: Alley Nominal
1	1242.0	20.0	RL	83.0	9849.0	Pave	
2	1233.0	90.0	RL	70.0	9842.0	Pave	
3	1401.0	50.0	RM	50.0	6000.0	Pave	
4	1377.0	30.0	RL	52.0	6292.0	Pave	
5	208.0	20.0	RL		12493.0	Pave	
6	1392.0	90.0	RL	65.0	8944.0	Pave	
7	980.0	20.0	RL	80.0	8816.0	Pave	
8	484.0	120.0	RM	32.0	4500.0	Pave	
9	392.0	60.0	RL	71.0	12209.0	Pave	
10	730.0	30.0	RM	52.0	6240.0	Pave	Grvl
11	255.0	20.0	RL	70.0	8400.0	Pave	
12	1094.0	20.0	RL	71.0	9230.0	Pave	
13	1021.0	20.0	RL	60.0	7024.0	Pave	
14	1341.0	20.0	RL	70.0	8294.0	Pave	
15	1025.0	20.0	RL		15498.0	Pave	
16	848.0	20.0	RL	36.0	15523.0	Pave	
17	457.0	70.0	RM	34.0	4571.0	Pave	Grvl
18	1266.0	160.0	FV	35.0	3735.0	Pave	
19	695.0	50.0	RM	51.0	6120.0	Pave	
20	24.0	120.0	RM	44.0	4224.0	Pave	
21	1314.0	60.0	RL	108.0	14774.0	Pave	
22	514.0	20.0	RL	71.0	9187.0	Pave	
23	1068.0	60.0	RL	80.0	9760.0	Pave	
24	1423.0	120.0	RM	37.0	4435.0	Pave	
25	1258.0	30.0	RL	56.0	4060.0	Pave	
26	620.0	60.0	RL	85.0	12244.0	Pave	
27	1213.0	30.0	RL	50.0	9340.0	Pave	
28	71.0	20.0	RL	95.0	13651.0	Pave	

Now, we create the file “output1.csv” after impute the Alley attribute and continue with that file to impute for the LotFrontage attribute:

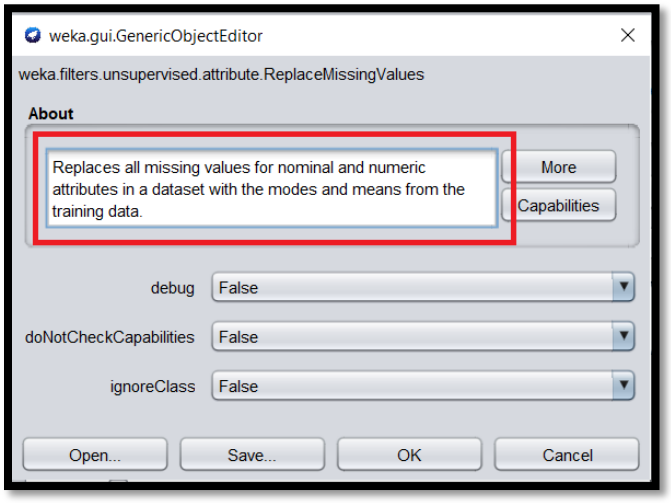
```
:\Users\VuongPhuc\Desktop\DataMining-Preprocessing\src>impute.py ../data/house-prices.csv
 --method=mode --columns Alley --out=../data/output1.csv

:\Users\VuongPhuc\Desktop\DataMining-Preprocessing\src>impute.py ../data/output1.csv
 --method=mean --columns LotFrontage --out=../data/output2.csv
```

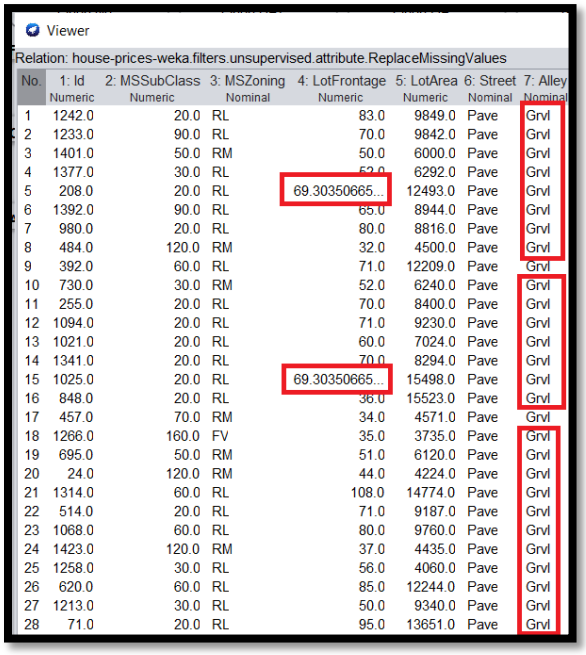
And this is the data after being imputed:

	A	B	C	D	E	F	G
1	Id	MSSubClas	MSZoning	LotFrontag	LotArea	Street	Alley
2	1242	20	RL	83	9849	Pave	Grvl
3	1233	90	RL	70	9842	Pave	Grvl
4	1401	50	RM	50	6000	Pave	Grvl
5	1377	30	RL	52	6292	Pave	Grvl
6	208	20	RL	69.30351	12493	Pave	Grvl
7	1392	90	RL	65	8944	Pave	Grvl
8	980	20	RL	80	8816	Pave	Grvl
9	484	120	RM	32	4500	Pave	Grvl
10	392	60	RL	71	12209	Pave	Grvl
11	730	30	RM	52	6240	Pave	Grvl
12	255	20	RL	70	8400	Pave	Grvl
13	1094	20	RL	71	9230	Pave	Grvl
14	1021	20	RL	60	7024	Pave	Grvl
15	1341	20	RL	70	8294	Pave	Grvl
16	1025	20	RL	69.30351	15498	Pave	Grvl
17	848	20	RL	36	15523	Pave	Grvl
18	457	70	RM	34	4571	Pave	Grvl
19	1266	160	FV	35	3735	Pave	Grvl
20	695	50	RM	51	6120	Pave	Grvl
21	24	120	RM	44	4224	Pave	Grvl
22	1314	60	RL	108	14774	Pave	Grvl
23	514	20	RL	71	9187	Pave	Grvl
24	1068	60	RL	80	9760	Pave	Grvl
25	1423	120	RM	37	4435	Pave	Grvl
26	1258	30	RL	56	4060	Pave	Grvl

To do the imputation in weka, we use the unsupervised filter which named “ReplaceMissingValue”. This filter, will replace all the missing nominal value with mode method, and missing numeric value with mean method.



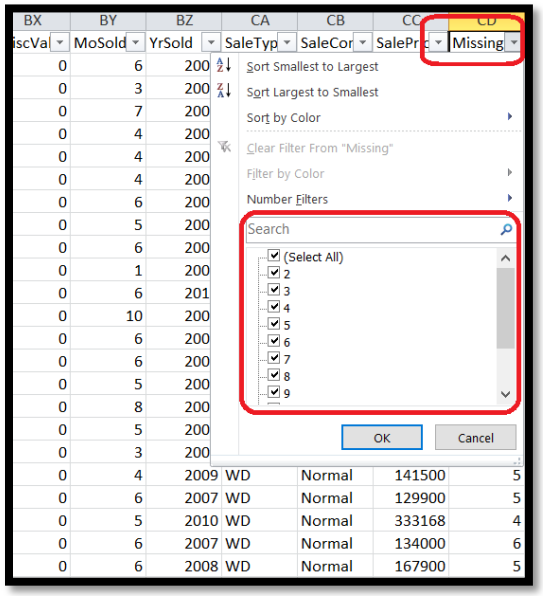
Finally, we observed that two result are the same:



Remove sample and attribute

Remove sample with threshold

In this section, for testing, we use excel to calculate the number of missing value for one-by-one sample (using COUNTBLANK). And then, we observed that the number of missing value is in range of [2,16]. So we testing for remove the sample whose number of missing value is greater than 3 (which means we only take the samples missing 2 and 3 values).



To count the number of samples whose number of missing value is 2, or 3; we use countif formula of excel. After counting, we see that it has **34 samples**. Lastly, we calculate the threshold which is $\frac{3}{81} \approx 0.03703$, since we only remove sample whose rate is greater than threshold, we choose 0.038 to be our threshold.

```
C:\Users\VuongPhuc\Desktop\DataMining-Preprocessing\src>remove-threshold.py ../data/house-prices.csv
 --axis=sample --threshold=0.038 --out=../data/output.csv

C:\Users\VuongPhuc\Desktop\DataMining-Preprocessing\src>list-missing.py ../data/output.csv
Number of samples: 34
Number of missing samples: 34
The missing attribute:
        LotFrontage: 3
        Alley: 29
        MasVnrType: 9
        PoolQC: 34
        Fence: 3
        MiscFeature: 20
```

Remove attribute with threshold

From the list of missing attribute, we can calculate the threshold to remove attributes. For instance, we try to remove attributes: Alley, MiscFeature and PoolQC. The minimum number of missing sample is 941 samples (corresponding to Alley). So the threshold we need is $threshold = \frac{941}{1000}$, however this feature only remove whenever the rate is **greater**. In conclude, the threshold must be $\frac{940}{1000} = 0.94$.

```
C:\WINDOWS\system32\cmd.exe
C:\Users\VuongPhuc\Desktop\DataMining-Preprocessing\src>remove-threshold.py ../data/house-prices.csv
--axis=attribute --threshold=0.94 --out=../data/output.csv

C:\Users\VuongPhuc\Desktop\DataMining-Preprocessing\src>list-missing.py ../data/output.csv
Number of samples: 1000
Number of missing samples: 980
The missing attribute:
        LotFrontage: 173
        MasVnrType: 593
        MasVnrArea: 10
        BsmtQual: 27
        BsmtCond: 27
        BsmtExposure: 28
        BsmtFinType1: 27
        BsmtFinType2: 29
        FireplaceQu: 501
        GarageType: 60
        GarageYrBlt: 60
        GarageFinish: 60
        GarageQual: 60
        GarageCond: 60
        Fence: 815
```

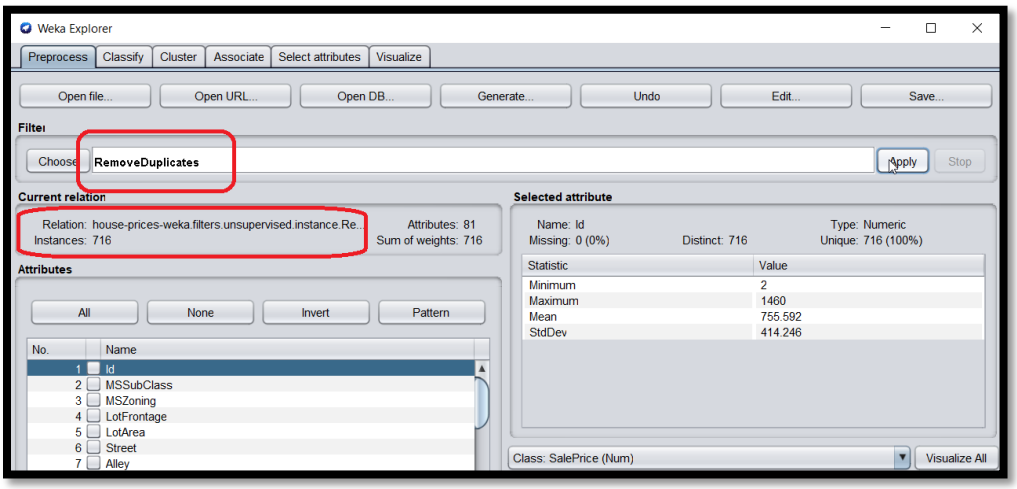
Remove duplicated samples

We can easily test this feature by removing and listing down the number of rest samples. After removed, we only have 716 samples left.

```
C:\Users\VuongPhuc\Desktop\DataMining-Preprocessing\src>remove-duplicated.py ../data/house-prices.csv
 --out=../data/output.csv

C:\Users\VuongPhuc\Desktop\DataMining-Preprocessing\src>list-missing.py ../data/output.csv
Number of samples: 716
Number of missing samples: 716
The missing attribute:
        LotFrontage: 121
        Alley: 673
        MasVnrType: 429
        MasVnrArea: 7
        BsmtQual: 19
        BsmtCond: 19
        BsmtExposure: 20
        BsmtFinType1: 19
        BsmtFinType2: 20
        FireplaceQu: 352
        GarageType: 43
        GarageYrBlt: 43
        GarageFinish: 43
        GarageQual: 43
        GarageCond: 43
        PoolQC: 716
```

In Weka, to remove duplicated samples, we use RemoveDuplicates filter in unsupervised/instance:



Normalizing

We used excel to normalize the “SalePrice” attribute by finding its properties:

- Min: using min formula
- Max: using max formula
- Mean: using average formula
- Std: using stdevpa formula

Then we normalize for both methods:

- Z-score: using standardize formula
- Min-max: doing manually

For the normalization, we achieve some head samples:

SalePrice	Mean	Std	Z-score	Min-max
248328	178116	80133.9	0.876183	0.3696
101800	Min	Max	-0.952357	0.11536
120000	35311	611657	-0.725237	0.14694
91000			-1.087131	0.09662
141000			-0.463175	0.18338
124000			-0.675320	0.15388
139000			-0.488134	0.17991
164000			-0.176156	0.22328
215000			0.460279	0.31177
103000			-0.937382	0.11745
145000			-0.413259	0.19032
146000			-0.400780	0.19205
176000			-0.026406	0.24411
123000			-0.687799	0.15215
287000			1.358775	0.4367
133500			-0.556769	0.17036
98000			-0.999777	0.10877

Then we use our source code to do the normalization for each method:

```
C:\Users\VuongPhuc\Desktop\DataMining-Preprocessing\src>normalize ../data/house-prices.csv
--method=z-score --columns SalePrice --out=../data/z_score.csv

C:\Users\VuongPhuc\Desktop\DataMining-Preprocessing\src>normalize ../data/house-prices.csv
--method=min-max --columns SalePrice --out=../data/min_max.csv
```

Comparing two results, we see that they are matched:

SalePrice
0.87618
-0.95236
-0.72524
-1.08713
-0.46318
-0.67532
-0.48813
-0.17616
0.46028
-0.93738
-0.41326
-0.40078
-0.02641
-0.6878
1.35878

Z-score

SalePrice
0.3696
0.11536
0.14694
0.09662
0.18338
0.15388
0.17991
0.22328
0.31177
0.11745
0.19032
0.19205
0.24411
0.15215
0.4367

Min-max

Add new attribute with given expression

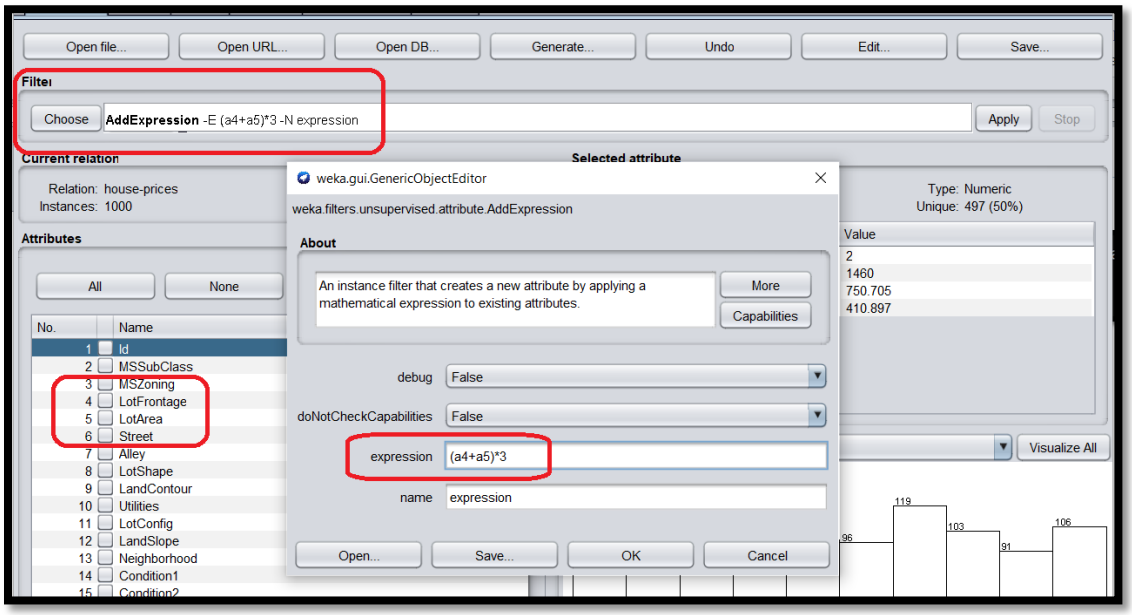
In this example, we pick two attributes which have some missing value: LotFrontage and LotArea and using the expresison "(LotFrontage+LotArea)*3" for testing:

```
C:\Users\VuongPhuc\Desktop\DataMining-Preprocessing\src>evaluate.py
../data/house-prices.csv --expression=(LotFrontage+LotArea)*3
--out=../data/output.csv
```

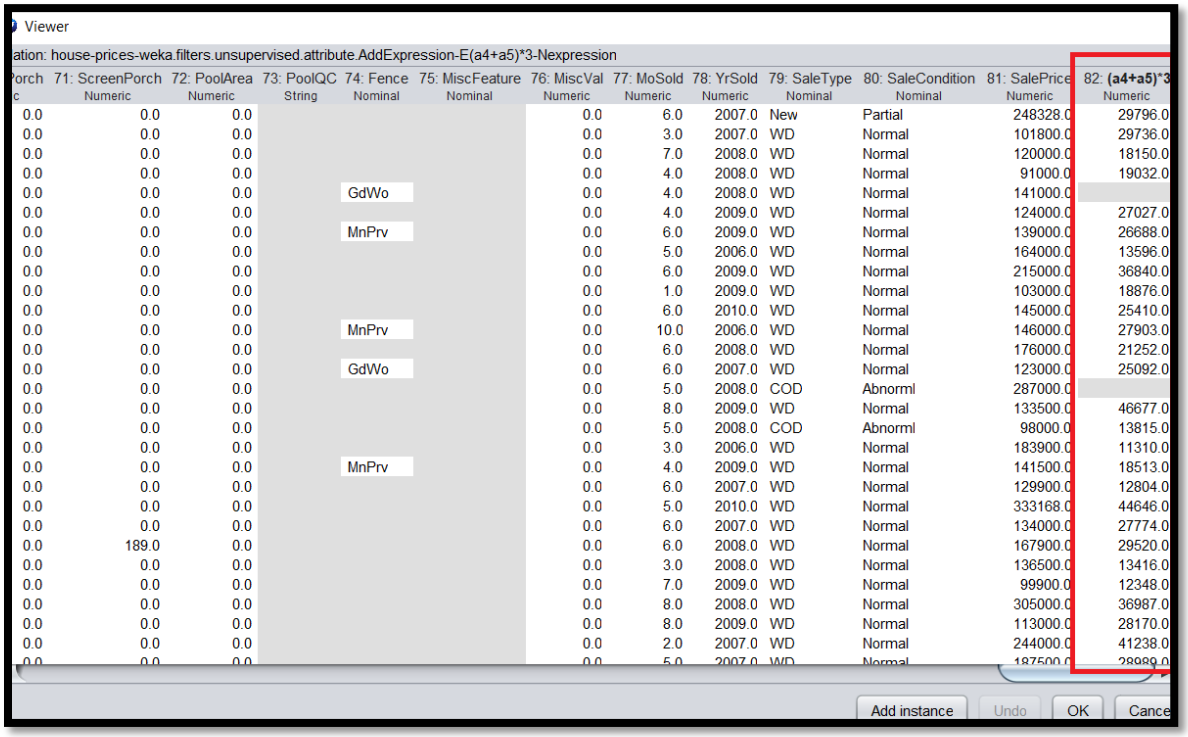
And this is our result:

CB	CC	CD
SaleCondit	SalePrice	(LotFrontage+LotArea)*3
Partial	248328	29796
Normal	101800	29736
Normal	120000	18150
Normal	91000	19032
Normal	141000	
Normal	124000	27027
Normal	139000	26688
Normal	164000	13596
Normal	215000	36840
Normal	103000	18876
Normal	145000	25410
Normal	146000	27903
Normal	176000	21252
Normal	123000	25092
Abnorml	287000	
Normal	133500	46677
Abnorml	98000	13815
Normal	183900	11310
Normal	141500	18513
Normal	129900	12804

In weka, we can add an expression using AddExpression filter. We can see that LotFrontage is the 4th attribute (a4) and LotArea is the 5th one (a5), so our expression need to fill in is "(a4+a5)*3"



Easily, we can see the result provided by Weka match ours:



orch c	71: ScreenPorch Numeric	72: PoolArea Numeric	73: PoolQC String	74: Fence Nominal	75: MiscFeature Nominal	76: MiscVal Numeric	77: MoSold Numeric	78: YrSold Numeric	79: SaleType Nominal	80: SaleCondition Nominal	81: SalePrice Numeric	82: (a4+a5)*3 Numeric
0.0	0.0	0.0				0.0	6.0	2007.0	New	Partial	248328.0	29796.0
0.0	0.0	0.0				0.0	3.0	2007.0	WD	Normal	101800.0	29736.0
0.0	0.0	0.0				0.0	7.0	2008.0	WD	Normal	120000.0	18150.0
0.0	0.0	0.0				0.0	4.0	2008.0	WD	Normal	91000.0	19032.0
0.0	0.0	0.0		GdWo		0.0	4.0	2008.0	WD	Normal	141000.0	
0.0	0.0	0.0				0.0	4.0	2009.0	WD	Normal	124000.0	27027.0
0.0	0.0	0.0		MnPrv		0.0	6.0	2009.0	WD	Normal	139000.0	26688.0
0.0	0.0	0.0				0.0	5.0	2006.0	WD	Normal	164000.0	13596.0
0.0	0.0	0.0				0.0	6.0	2009.0	WD	Normal	215000.0	36840.0
0.0	0.0	0.0				0.0	1.0	2009.0	WD	Normal	103000.0	18876.0
0.0	0.0	0.0				0.0	6.0	2010.0	WD	Normal	145000.0	25410.0
0.0	0.0	0.0		MnPrv		0.0	10.0	2006.0	WD	Normal	146000.0	27903.0
0.0	0.0	0.0				0.0	6.0	2008.0	WD	Normal	176000.0	21252.0
0.0	0.0	0.0		GdWo		0.0	6.0	2007.0	WD	Normal	123000.0	25092.0
0.0	0.0	0.0				0.0	5.0	2008.0	COD	Abnorml	287000.0	
0.0	0.0	0.0				0.0	8.0	2009.0	WD	Normal	133500.0	46677.0
0.0	0.0	0.0				0.0	5.0	2008.0	COD	Abnorml	98000.0	13815.0
0.0	0.0	0.0				0.0	3.0	2006.0	WD	Normal	183900.0	11310.0
0.0	0.0	0.0		MnPrv		0.0	4.0	2009.0	WD	Normal	141500.0	18513.0
0.0	0.0	0.0				0.0	6.0	2007.0	WD	Normal	129900.0	12804.0
0.0	0.0	0.0				0.0	5.0	2010.0	WD	Normal	333168.0	44646.0
0.0	0.0	0.0				0.0	6.0	2007.0	WD	Normal	134000.0	27774.0
0.0	189.0	0.0				0.0	6.0	2008.0	WD	Normal	167900.0	29520.0
0.0	0.0	0.0				0.0	3.0	2008.0	WD	Normal	136500.0	13416.0
0.0	0.0	0.0				0.0	7.0	2009.0	WD	Normal	99900.0	12348.0
0.0	0.0	0.0				0.0	8.0	2008.0	WD	Normal	305000.0	36987.0
0.0	0.0	0.0				0.0	8.0	2009.0	WD	Normal	113000.0	28170.0
0.0	0.0	0.0				0.0	2.0	2007.0	WD	Normal	244000.0	41238.0
0.0	0.0	0.0				0.0	5.0	2007.0	WD	Normal	187500.0	28988.0

References

1. Classification Algorithms with Attribute Selection: an evaluation study using WEKA – Dr. S.Gnamambal et al – 04th Apr, 2018
2. Perform feature selection Machine learning data WEKA – Machine Learning Mastery – Jason Brownlee – 13th July, 2018
3. Machine Learning with WEKA WEKA Explorer Tutorial - S. Aksenova - California State University – 2004
4. Quartiles – Math is fun – 2019
5. A Tour of the Weka Machine Learning Workbench – Jason Brownlee – 21st Jun, 2016
6. Attribute-Relation File Format (ARFF) – University of Waikato – 1st Nov, 2008
7. How to Calculate Correlation Between Variables in Python – Machine Learning Mastery – Jason Brown, 27th Apr, 2018