EXERCISE -4

4.1 Problem Statement:

Generate strong Association rules by using Apriori algorithm on 'German_credit' dataset with Min_Sup=60% and Min_Conf=80%.

4.2 Description:

About Dataset

The objective of the German Credit Data is to minimize the chances of issuing risky loans to applicants while maximizing the chances of profiting from good loans. An applicant's demographic and socio-economic profiles are considered by loan managers before a decision is taken regarding his/her loan application.

The German Credit data set is a publically available data set downloaded from the UCI Machine Learning Repository. The German Credit Data contains data on 20 variables and the classification of whether an applicant is considered a Good or Bad credit risk for 1000 loan applicants. The task requires exploring the data and building a predictive model to provide a bank manager guidance for making a decision on whether to approve a loan to a prospective applicant based on his/her profile.

the original dataset that only has 19 variables:

1. Checking_Status
2. Credit_history
3. Purpose
4. Savings_status
5. Employment
6. Personal_status
7. Other_parties
8. Property_Magnitude

9. Other_payment_plans
10. Housing
11. Job
12. Own_telephone
13. Foreign_worker
14. Duration
15. Credit_amout
16. Installment_Commitment
17. Residence_since
18. Age
19. Existing_credits

About Arff:

An ARFF (Attribute-Relation File Format) file is an ASCII text file that describes a list of instances sharing a set of attributes.

ARFF files have two distinct sections. The first section is the **Header** information, which is followed the **Data** information.

The **Header** of the ARFF file contains the name of the relation, a list of the attributes (the columns in the data), and their types.

Lines that begin with a % are comments. The **@RELATION**, **@ATTRIBUTE** and **@DATA** declarations are case insensitive.

About CSV:

Files with .csv (Comma Separated Values) extension represent plain text files that contain records of data with comma separated values. Each line in a CSV file is a new record from the set of records contained in the file. Such files are generated when data transfer is intended from one storage system to another. Since all applications can recognize records separated by comma, import of such data files to database is done very conveniently. Almost all spreadsheet applications such as Microsoft Excel or OpenOffice Calc can import CSV without much effort. Data imported from such files is arranged in cells of a spreadsheet for representation to user.

Datatypes that are supported by Weka:

- numeric
- integer is treated as numeric
- real is treated as numeric
- [nominal-specification]
- string
- date [date-format]
- relational for multi-instance data (for future use)

where [nominal-specification] and [date-format] are defined below. The keywords **numeric**, **real**, **integer**, **string** and **date** are case insensitive.

Numeric attributes
Numeric attributes can be real or integer numbers.

Nominal attributes
Nominal values are defined by providing an [nominal-specification] listing the possible values: {[nominal-name1], [nominal-name2], [nominal-name3], ...}

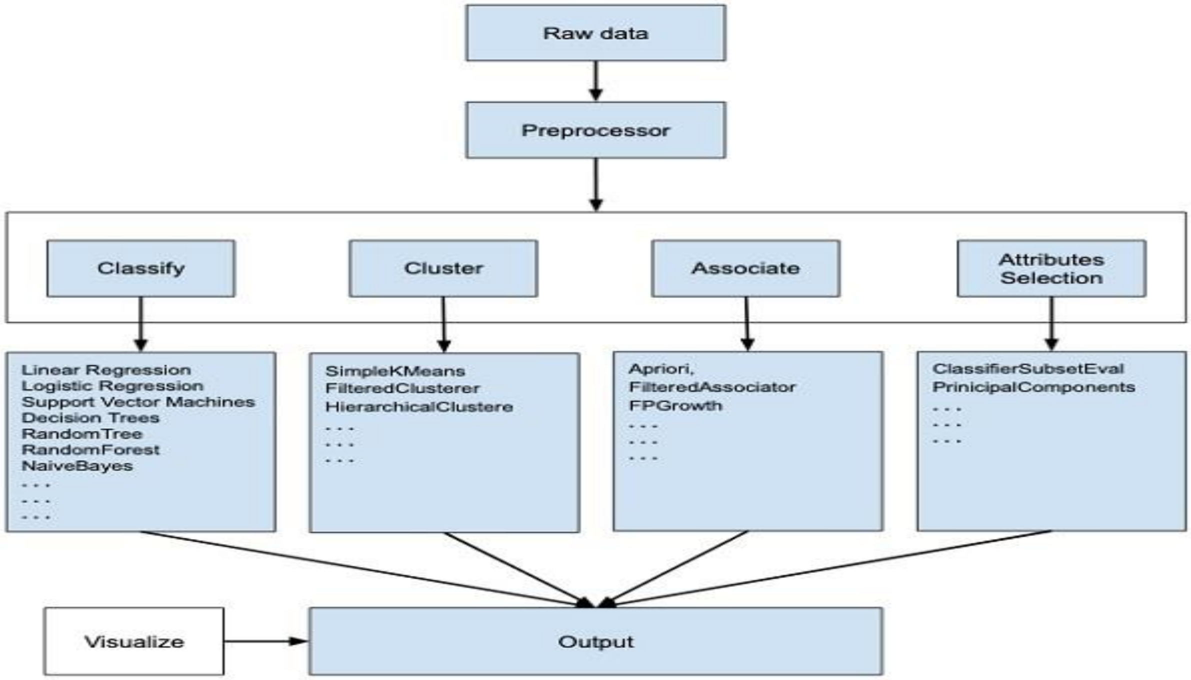
For example, the class value of the Iris dataset can be defined as follows:

```
@ATTRIBUTE class        {Iris-setosa,Iris-versicolor,Iris-virginica}
```

Values that contain spaces must be quoted.

About WEKA Software:

WEKA - an opensource software provides tools for data preprocessing, implementation of several Machine Learning algorithms, and visualization tools so that you can develop machine learning techniques and apply them to real-world data mining problems. What WEKA offers is summarized in the following diagram −



If you observe the beginning of the flow of the image, you will understand that there are many stages in dealing with Big Data to make it suitable for machine learning −

First, you will start with the raw data collected from the field. This data may contain several null values and irrelevant fields. You use the data preprocessing tools provided in WEKA to cleanse the data.

Then, you would save the preprocessed data in your local storage for applying ML algorithms.

Next, depending on the kind of ML model that you are trying to develop you would select one of the options such as **Classify, Cluster**, or **Associate**. The **Attributes Selection** allows the automatic selection of features to create a reduced dataset.

Note that under each category, WEKA provides the implementation of several algorithms. You would select an algorithm of your choice, set the desired parameters and run it on the dataset.

Then, WEKA would give you the statistical output of the model processing. It provides you a visualization tool to inspect the data.

The various models can be applied on the same dataset. You can then compare the outputs of different models and select the best that meets your purpose.

Thus, the use of WEKA results in a quicker development of machine learning models on the whole.

Now that we have seen what WEKA is and what it does, in the next chapter let us learn how to install WEKA on your local computer

4.3 Apriori Algorithm:

Apriori algorithm refers to the algorithm which is used to calculate the association rules between objects. It means how two or more objects are related to one another. In other words, we can say that the apriori algorithm is an association rule leaning that analyzes that people who bought product A also bought product B.

The primary objective of the apriori algorithm is to create the association rule between different objects. The association rule describes how two or more objects are related to one another. Apriori algorithm is also called frequent pattern mining. Generally, you operate the Apriori algorithm on a database that consists of a huge number of transactions.

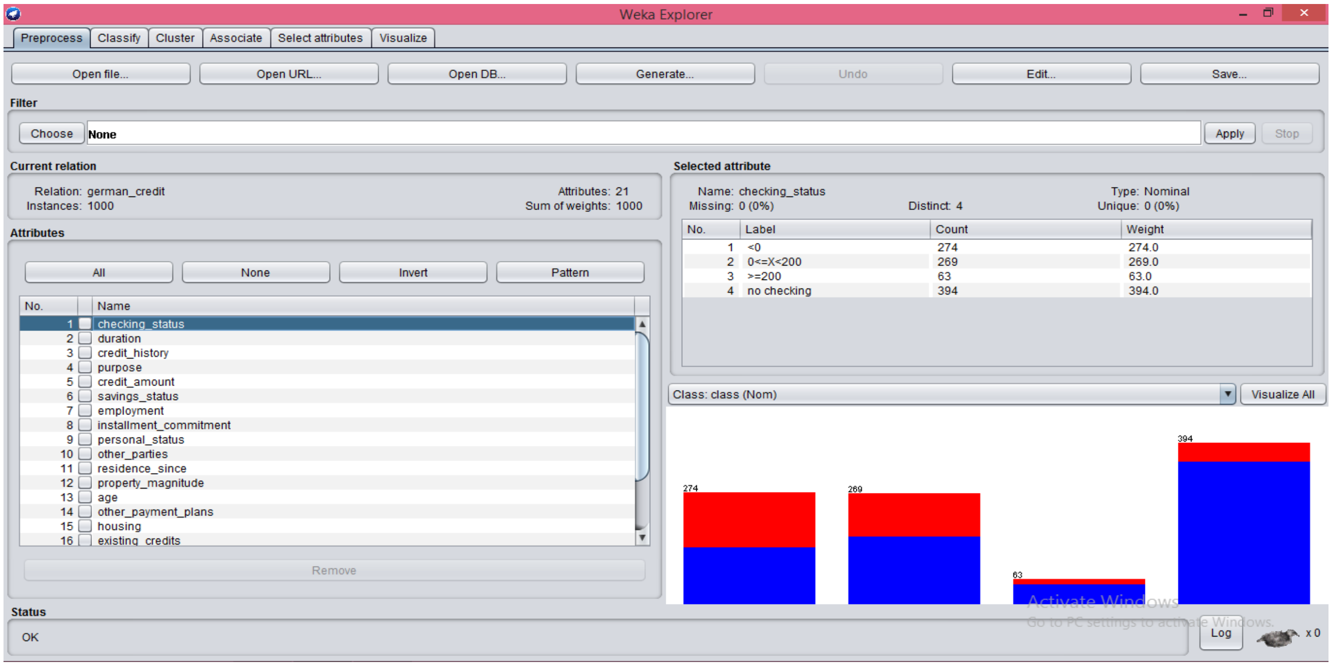
COMPONENTS OF APRIORI ALGORITHM

The given three components comprise the apriori algorithm.

1. Support
2. Confidence
3. Lift

4.4 procedure:

1. To generate association rules by using apriori algorithm on 'German credit' data set first upload the German credit dataset into weka software by choosing open file option and selecting German Credit arff file.



Filter: None

Relation: german_credit
Instances: 1000
Attributes: 21
Sum of weights: 1000

Name: checking_status
Missing: 0 (0%)
Distinct: 4
Type: Nominal
Unique: 0 (0%)

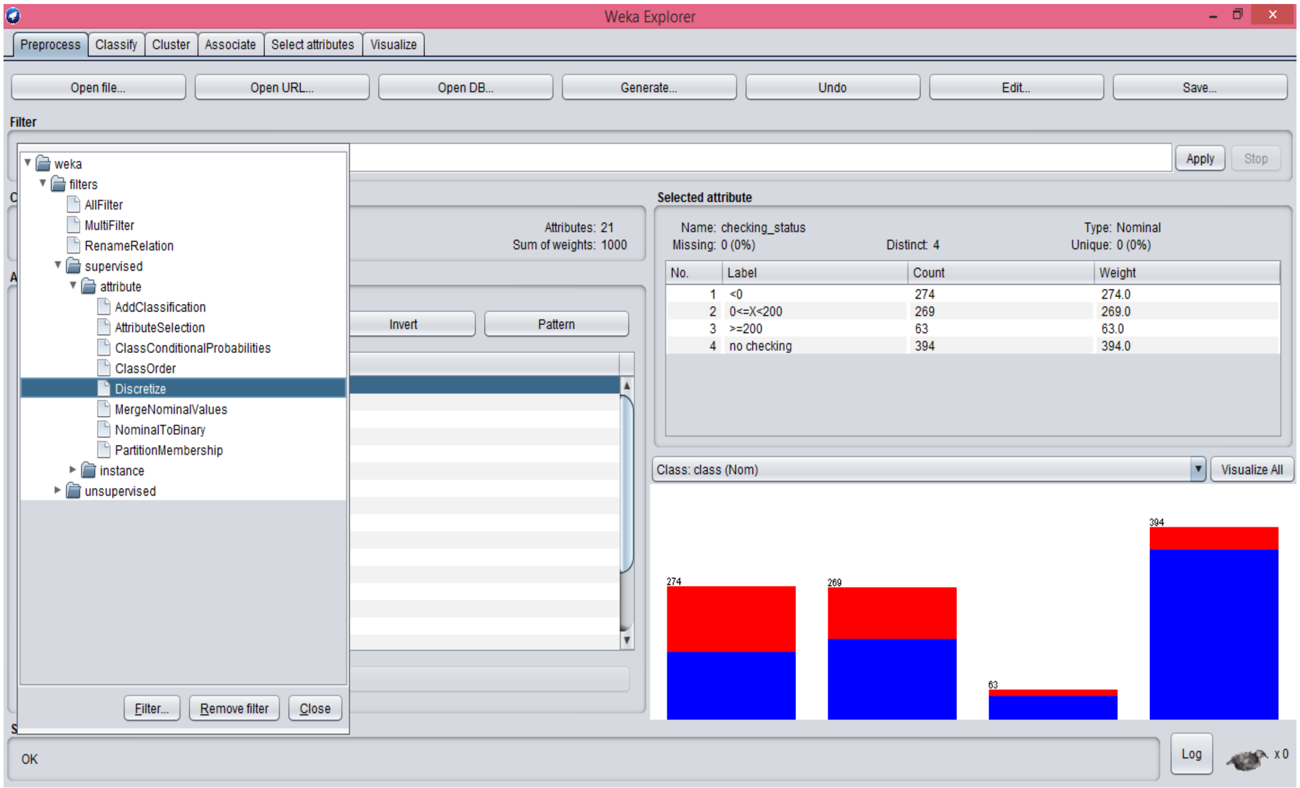
No.	Label	Count	Weight
1	<0	274	274.0
2	0<=X<200	269	269.0
3	>=200	63	63.0
4	no checking	394	394.0

No.	Name
1	checking_status
2	duration
3	credit_history
4	purpose
5	credit_amount
6	savings_status
7	employment
8	installment_commitment
9	personal_status
10	other_parties
11	residence_since
12	property_magnitude
13	age
14	other_payment_plans
15	housing
16	existing_credits

Class: class (Nom)

OK

2. Now perform discretization on the dataset. For this go to choose and select filter. Under filter select the attribute under supervised. Choose the discretize option.



- weka
  - filters
    - AllFilter
    - MultiFilter
    - RenameRelation
    - supervised
      - attribute
        - AddClassification
        - AttributeSelection
        - ClassConditionalProbabilities
        - ClassOrder
        - Discretize
        - MergeNominalValues
        - NominalToBinary
        - PartitionMembership
      - instance
    - unsupervised

Attributes: 21
Sum of weights: 1000

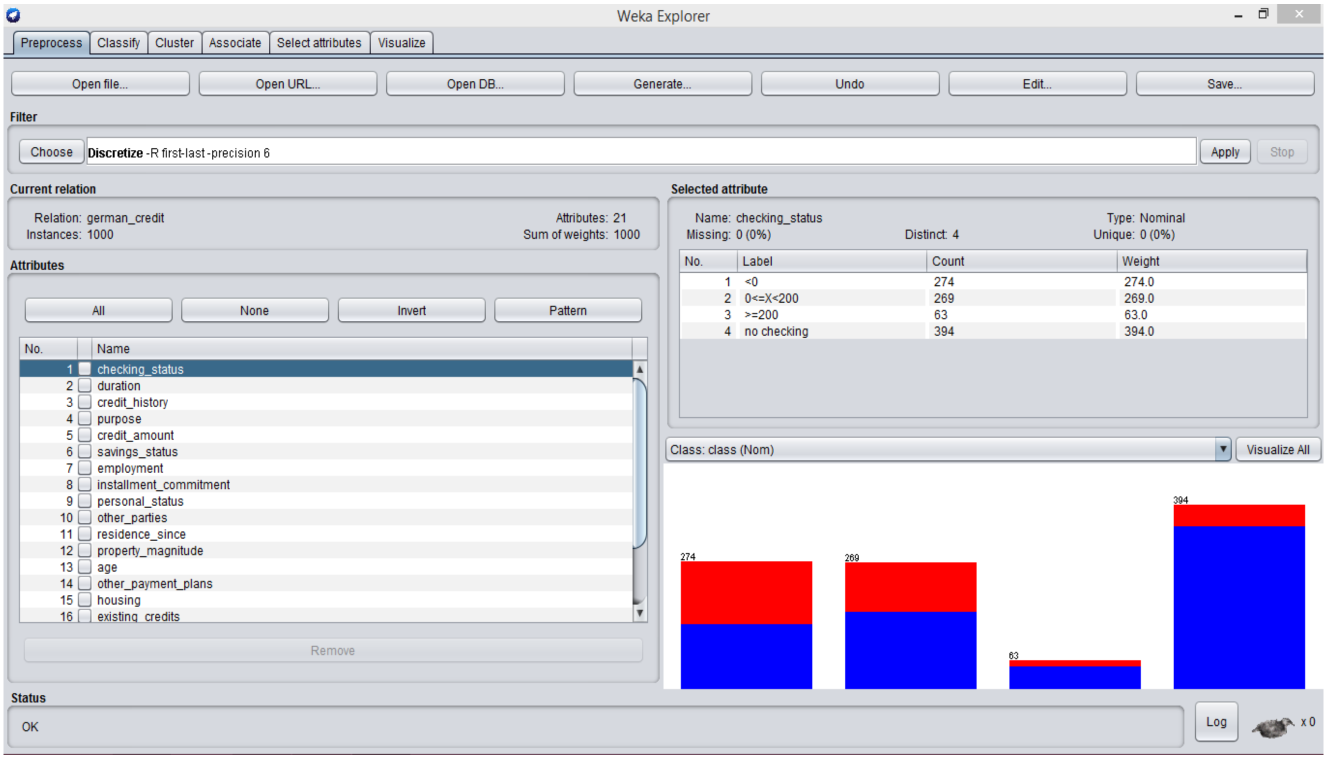
Name: checking_status
Missing: 0 (0%)
Distinct: 4
Type: Nominal
Unique: 0 (0%)

No.	Label	Count	Weight
1	<0	274	274.0
2	0<=X<200	269	269.0
3	>=200	63	63.0
4	no checking	394	394.0

Class: class (Nom)

OK

3. After choosing discretize option click on apply to apply the filter to the dataset.



Filter: **Discretize** -R first-last -precision 6

Current relation

Relation: german_credit
Instances: 1000
Attributes: 21
Sum of weights: 1000

Selected attribute

Name: checking_status
Missing: 0 (0%)
Distinct: 4
Type: Nominal
Unique: 0 (0%)

No.	Label	Count	Weight
1	<0	274	274.0
2	0<=X<200	269	269.0
3	>=200	63	63.0
4	no checking	394	394.0

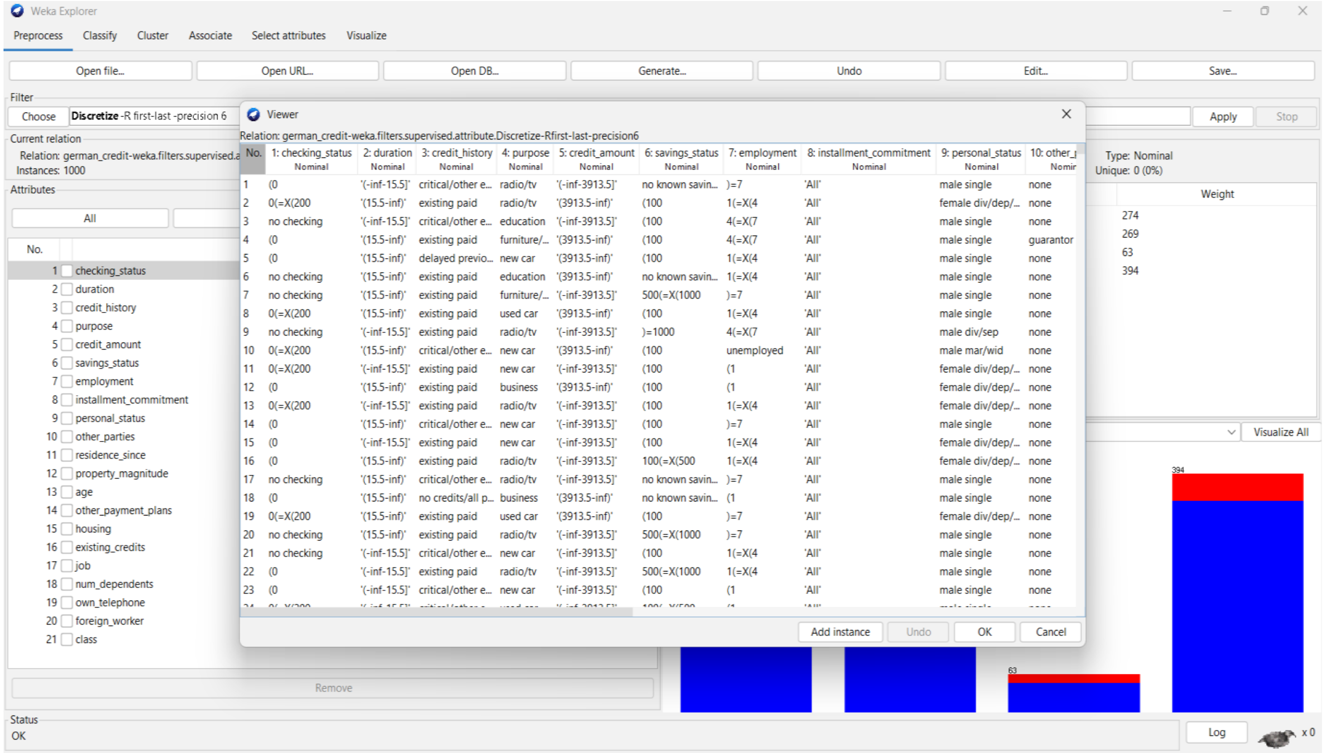
Attributes

No.	Name
1	checking_status
2	duration
3	credit_history
4	purpose
5	credit_amount
6	savings_status
7	employment
8	installment_commitment
9	personal_status
10	other_parties
11	residence_since
12	property_magnitude
13	age
14	other_payment_plans
15	housing
16	existing_credits

Class: class (Nom)

Status
OK

4. Now to observe the dataset in discretization click on Edit option.



Filter: **Discretize** -R first-last -precision 6

Current relation
Relation: german_credit-weka.filters.supervised.a
Instances: 1000

Type: Nominal
Unique: 0 (0%)

Weight
274
269
63
394

Attributes

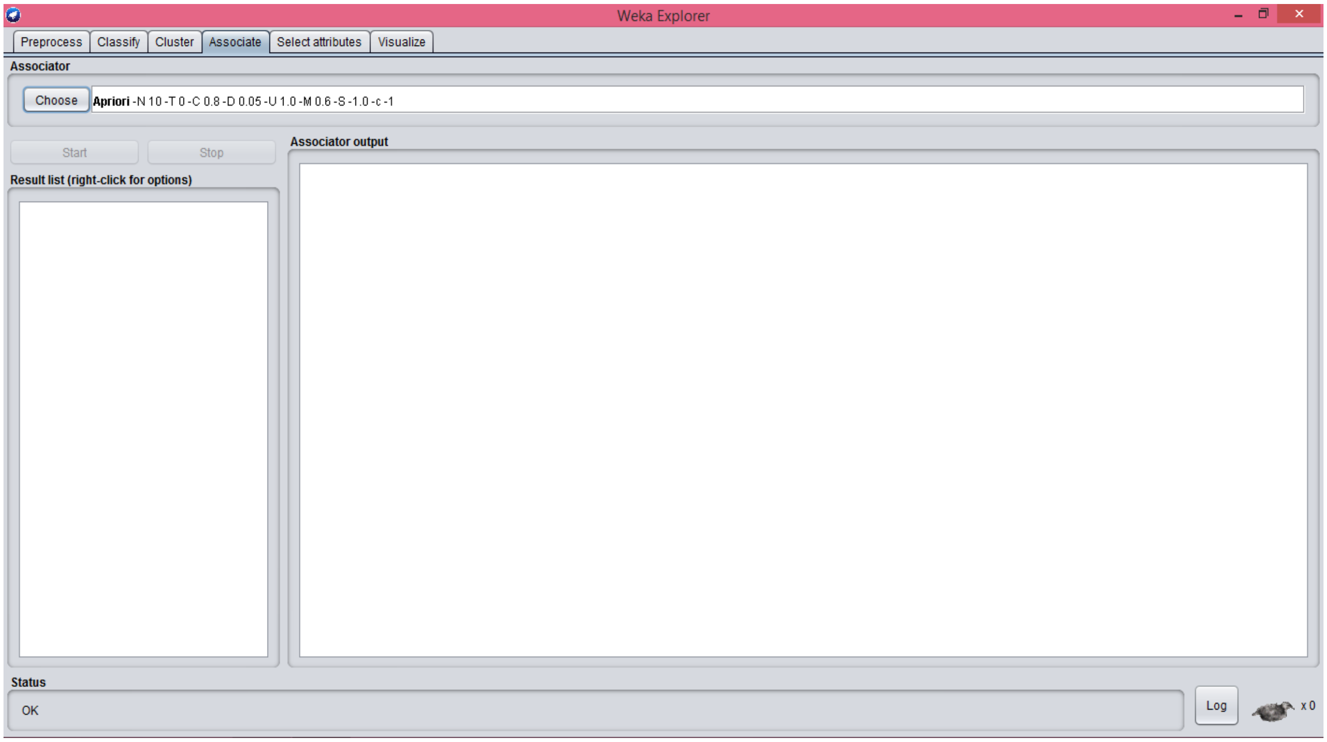
No.	Name
1	checking_status
2	duration
3	credit_history
4	purpose
5	credit_amount
6	savings_status
7	employment
8	installment_commitment
9	personal_status
10	other_parties
11	residence_since
12	property_magnitude
13	age
14	other_payment_plans
15	housing
16	existing_credits
17	job
18	num_dependents
19	own_telephone
20	foreign_worker
21	class

Relation: german_credit-weka.filters.supervised.attribute.Discretize-Rfirst-last-precision6

No.	1: checking_status Nominal	2: duration Nominal	3: credit_history Nominal	4: purpose Nominal	5: credit_amount Nominal	6: savings_status Nominal	7: employment Nominal	8: installment_commitment Nominal	9: personal_status Nominal	10: other_ Nomir
1	(0	'(-inf-15.5]'	critical/other e...	radio/tv	'(-inf-3913.5]'	no known savin...	)=7	'All'	male single	none
2	0(=X(200	'(15.5-inf)'	existing paid	radio/tv	'(3913.5-inf)'	(100	1(=X(4	'All'	female div/dep/...	none
3	no checking	'(-inf-15.5]'	critical/other e...	education	'(-inf-3913.5]'	(100	4(=X(7	'All'	male single	none
4	(0	'(15.5-inf)'	existing paid	furniture/...	'(3913.5-inf)'	(100	4(=X(7	'All'	male single	guarantor
5	(0	'(15.5-inf)'	delayed previo...	new car	'(3913.5-inf)'	(100	1(=X(4	'All'	male single	none
6	no checking	'(15.5-inf)'	existing paid	education	'(3913.5-inf)'	no known savin...	1(=X(4	'All'	male single	none
7	no checking	'(15.5-inf)'	existing paid	furniture/...	'(-inf-3913.5]'	500(=X(1000	)=7	'All'	male single	none
8	0(=X(200	'(15.5-inf)'	existing paid	used car	'(3913.5-inf)'	(100	1(=X(4	'All'	male single	none
9	no checking	'(-inf-15.5]'	existing paid	radio/tv	'(-inf-3913.5]'	)=1000	4(=X(7	'All'	male div/sep	none
10	0(=X(200	'(15.5-inf)'	critical/other e...	new car	'(3913.5-inf)'	(100	unemployed	'All'	male mar/wid	none
11	0(=X(200	'(-inf-15.5]'	existing paid	new car	'(-inf-3913.5]'	(100	(1	'All'	female div/dep/...	none
12	(0	'(15.5-inf)'	existing paid	business	'(3913.5-inf)'	(100	(1	'All'	female div/dep/...	none
13	0(=X(200	'(-inf-15.5]'	existing paid	radio/tv	'(-inf-3913.5]'	(100	1(=X(4	'All'	female div/dep/...	none
14	(0	'(15.5-inf)'	critical/other e...	new car	'(-inf-3913.5]'	(100	)=7	'All'	male single	none
15	(0	'(-inf-15.5]'	existing paid	new car	'(-inf-3913.5]'	(100	1(=X(4	'All'	female div/dep/...	none
16	(0	'(15.5-inf)'	existing paid	radio/tv	'(-inf-3913.5]'	100(=X(500	1(=X(4	'All'	female div/dep/...	none
17	no checking	'(15.5-inf)'	critical/other e...	radio/tv	'(-inf-3913.5]'	no known savin...	)=7	'All'	male single	none
18	(0	'(15.5-inf)'	no credits/all p...	business	'(3913.5-inf)'	no known savin...	(1	'All'	male single	none
19	0(=X(200	'(15.5-inf)'	existing paid	used car	'(3913.5-inf)'	(100	)=7	'All'	female div/dep/...	none
20	no checking	'(15.5-inf)'	existing paid	radio/tv	'(-inf-3913.5]'	500(=X(1000	)=7	'All'	male single	none
21	no checking	'(-inf-15.5]'	critical/other e...	new car	'(-inf-3913.5]'	(100	1(=X(4	'All'	male single	none
22	(0	'(-inf-15.5]'	existing paid	radio/tv	'(-inf-3913.5]'	500(=X(1000	1(=X(4	'All'	male single	none
23	(0	'(-inf-15.5]'	critical/other e...	new car	'(-inf-3913.5]'	(100	(1	'All'	male single	none

Status
OK

5. Now to apply Apriori algorithm on German credit dataset select and open associate option.

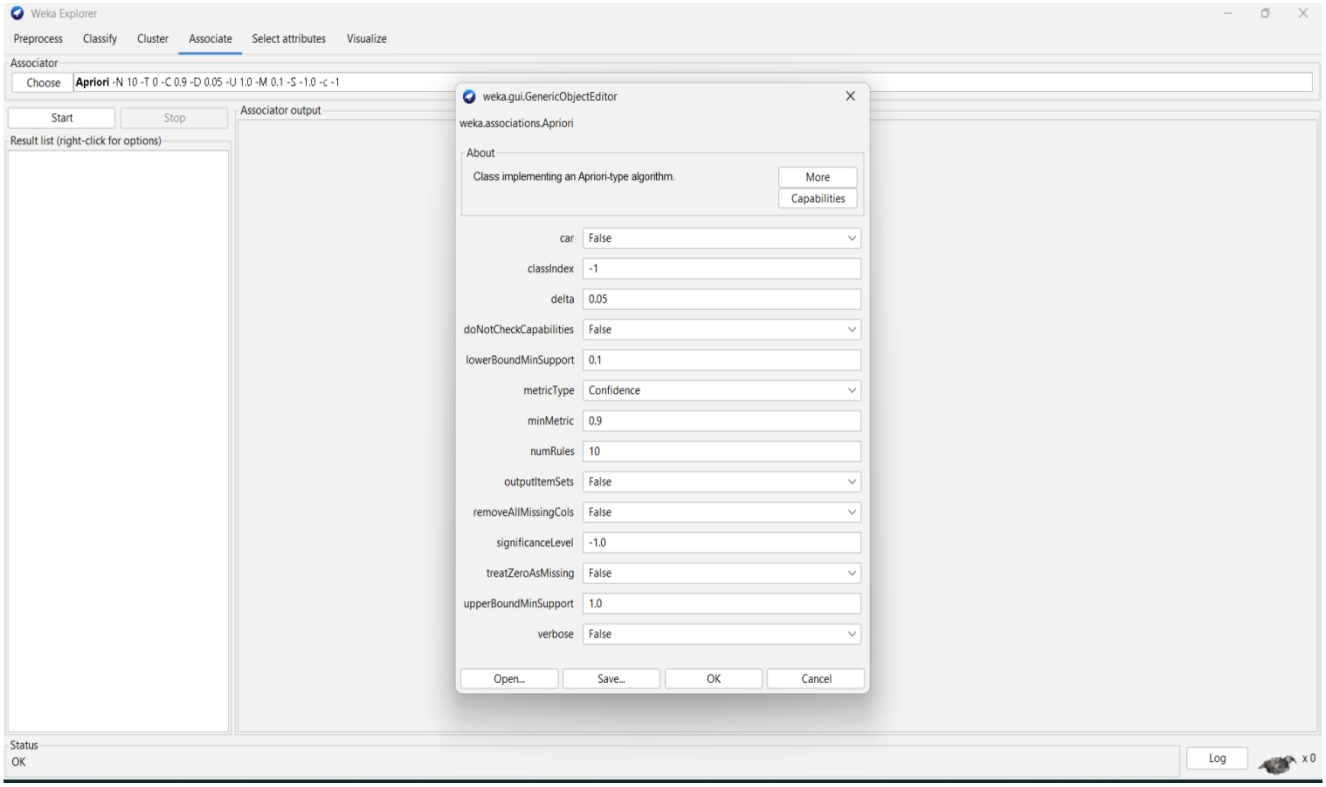


Apriori -N 10 -T 0 -C 0.8 -D 0.05 -U 1.0 -M 0.6 -S -1.0 -c -1

OK

6. According to our problem select and choose Apriori from the available list of associators.

7. Before clicking start click on apriori and an editor box will appear.



Apriori -N 10 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.1 -S -1.0 -c -1

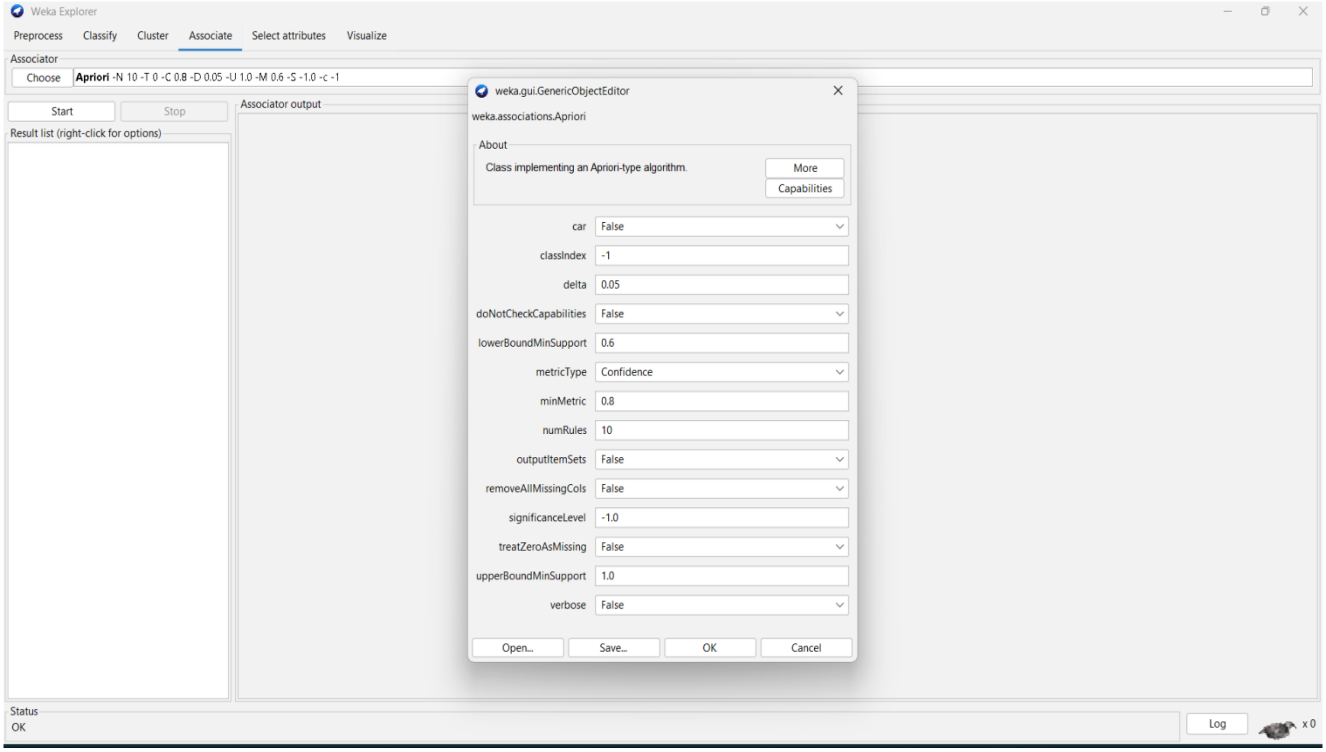
weka.associations.Apriori

Class implementing an Apriori-type algorithm.

car	False
classIndex	-1
delta	0.05
doNotCheckCapabilities	False
lowerBoundMinSupport	0.1
metricType	Confidence
minMetric	0.9
numRules	10
outputItemSets	False
removeAllMissingCols	False
significanceLevel	-1.0
treatZeroAsMissing	False
upperBoundMinSupport	1.0
verbose	False

OK

8. Change the values of min_support to 0.6 which is 60% and min_confident to 0.8 which is 80%. Click on ok.



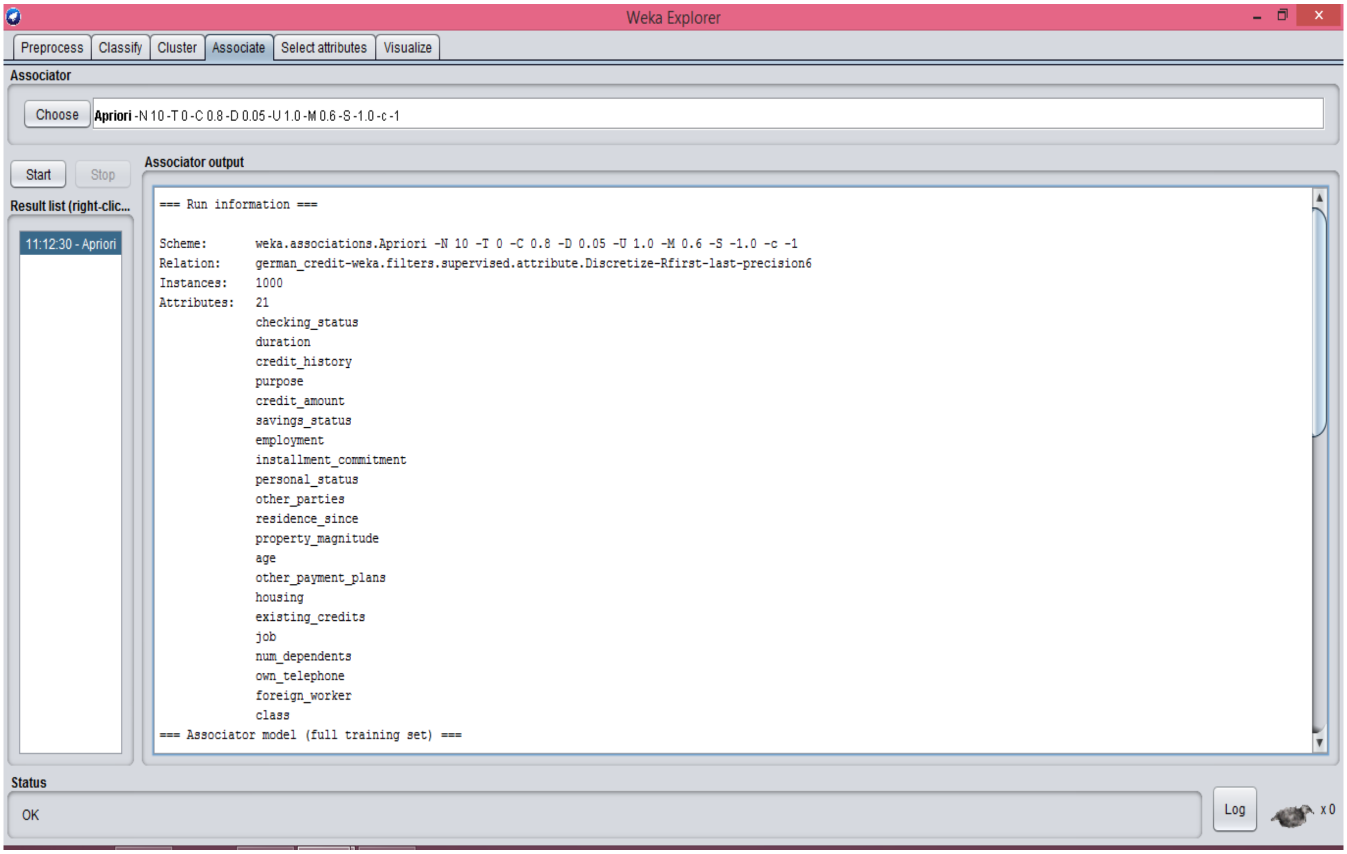
Apriori -N 10 -T 0 -C 0.8 -D 0.05 -U 1.0 -M 0.6 -S -1.0 -c -1

weka.associations.Apriori

Class implementing an Apriori-type algorithm.

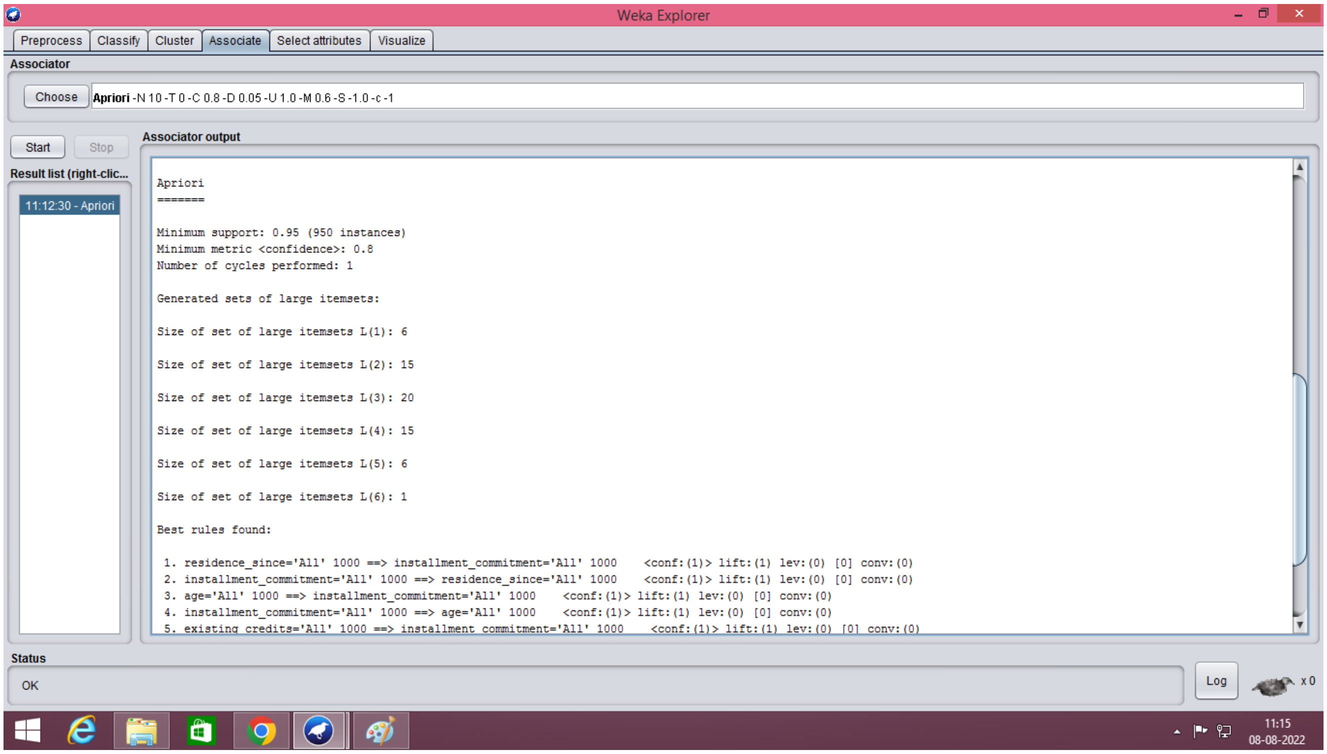
car	False
classIndex	-1
delta	0.05
doNotCheckCapabilities	False
lowerBoundMinSupport	0.6
metricType	Confidence
minMetric	0.8
numRules	10
outputItemSets	False
removeAllMissingCols	False
significanceLevel	-1.0
treatZeroAsMissing	False
upperBoundMinSupport	1.0
verbose	False

9. Click on start and observe the association rules that are generated.



Apriori -N 10 -T 0 -C 0.8 -D 0.05 -U 1.0 -M 0.6 -S -1.0 -c -1

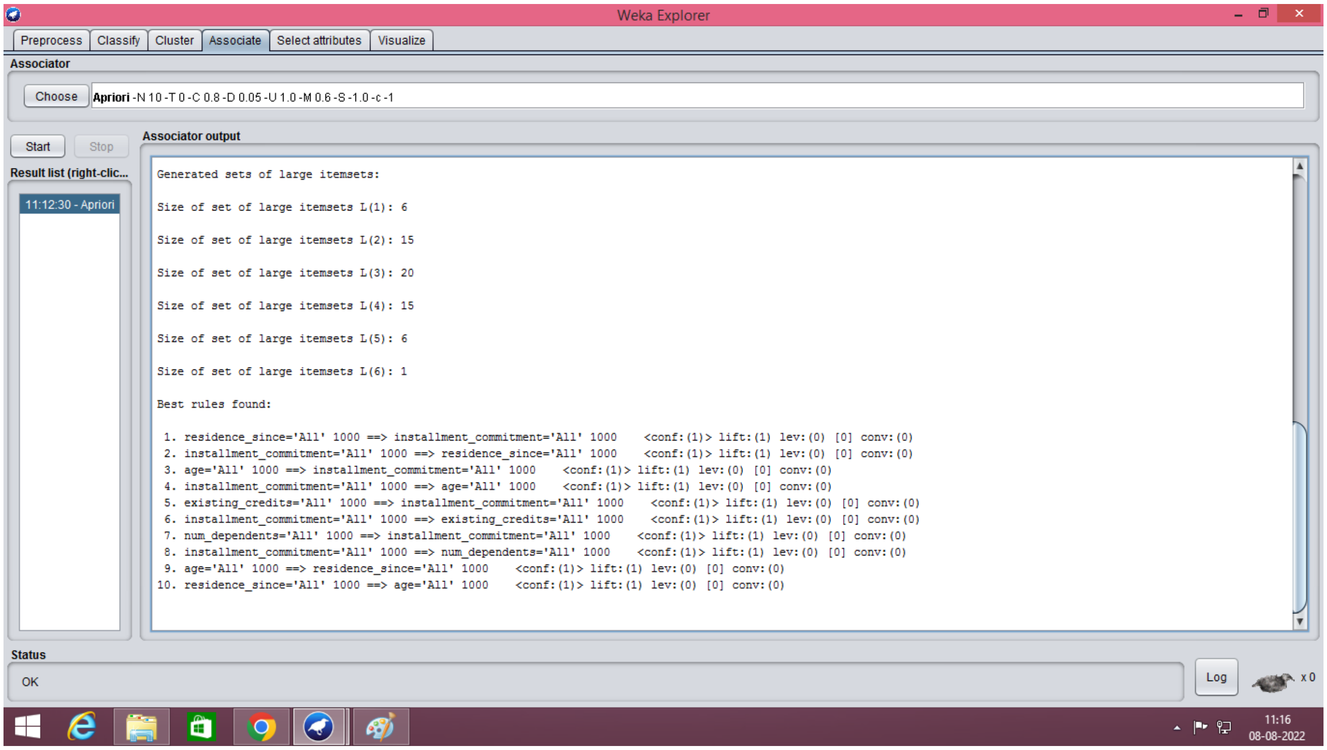
```
=== Run information ===

Scheme:       weka.associations.Apriori -N 10 -T 0 -C 0.8 -D 0.05 -U 1.0 -M 0.6 -S -1.0 -c -1
Relation:     german_credit-weka.filters.supervised.attribute.Discretize-Rfirst-last-precision6
Instances:    1000
Attributes:   21
              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
=== Associator model (full training set) ===
```



Apriori -N 10 -T 0 -C 0.8 -D 0.05 -U 1.0 -M 0.6 -S -1.0 -c -1

```
Apriori
=======

Minimum support: 0.95 (950 instances)
Minimum metric <confidence>: 0.8
Number of cycles performed: 1

Generated sets of large itemsets:

Size of set of large itemsets L(1): 6

Size of set of large itemsets L(2): 15

Size of set of large itemsets L(3): 20

Size of set of large itemsets L(4): 15

Size of set of large itemsets L(5): 6

Size of set of large itemsets L(6): 1

Best rules found:

 1. residence_since='All' 1000 ==> installment_commitment='All' 1000    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
 2. installment_commitment='All' 1000 ==> residence_since='All' 1000    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
 3. age='All' 1000 ==> installment_commitment='All' 1000    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
 4. installment_commitment='All' 1000 ==> age='All' 1000    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
 5. existing_credits='All' 1000 ==> installment_commitment='All' 1000    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
```

Status: OK



Apriori -N 10 -T 0 -C 0.8 -D 0.05 -U 1.0 -M 0.6 -S -1.0 -c -1

```
Generated sets of large itemsets:

Size of set of large itemsets L(1): 6

Size of set of large itemsets L(2): 15

Size of set of large itemsets L(3): 20

Size of set of large itemsets L(4): 15

Size of set of large itemsets L(5): 6

Size of set of large itemsets L(6): 1

Best rules found:

 1. residence_since='All' 1000 ==> installment_commitment='All' 1000    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
 2. installment_commitment='All' 1000 ==> residence_since='All' 1000    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
 3. age='All' 1000 ==> installment_commitment='All' 1000    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
 4. installment_commitment='All' 1000 ==> age='All' 1000    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
 5. existing_credits='All' 1000 ==> installment_commitment='All' 1000    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
 6. installment_commitment='All' 1000 ==> existing_credits='All' 1000    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
 7. num_dependents='All' 1000 ==> installment_commitment='All' 1000    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
 8. installment_commitment='All' 1000 ==> num_dependents='All' 1000    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
 9. age='All' 1000 ==> residence_since='All' 1000    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
10. residence_since='All' 1000 ==> age='All' 1000    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
```

Status: OK

4.5 Results and Discussion:

By using Apriori algorithm on 'German credit' data set with Min_Sup=60% and Min_Conf=80% we generated strong associate rules successfully.