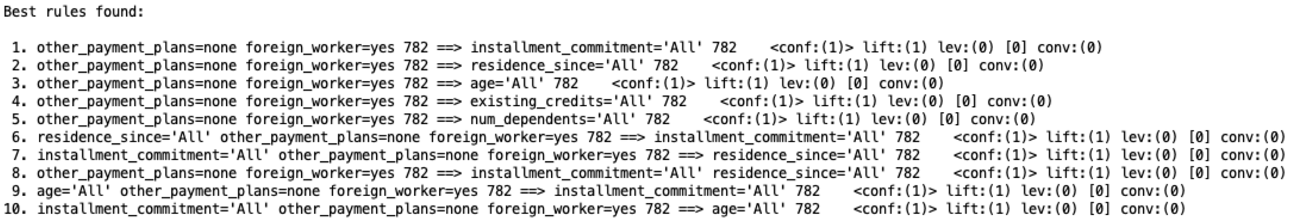
Weka Lab 6 Preprocessing Association and Clustering using Weka

1. Load in the ./weka-3.8/data/credit-g.arff dataset
2. What types of features are in the dataset?

The dataset contains features of each users and their personal data, that is related to their employment, age, credit history and property details.

3. How should you pre-process the dataset before applying association rule mining?

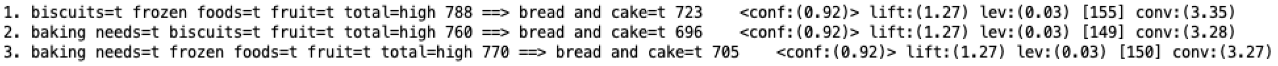
A filter is required to be applied on the dataset's attribute before we apply association rule mining. We apply a discretizer



```
Best rules found:

 1. other_payment_plans=none foreign_worker=yes 782 ==> installment_commitment='All' 782    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
 2. other_payment_plans=none foreign_worker=yes 782 ==> residence_since='All' 782    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
 3. other_payment_plans=none foreign_worker=yes 782 ==> age='All' 782    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
 4. other_payment_plans=none foreign_worker=yes 782 ==> existing_credits='All' 782    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
 5. other_payment_plans=none foreign_worker=yes 782 ==> num_dependents='All' 782    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
 6. residence_since='All' other_payment_plans=none foreign_worker=yes 782 ==> installment_commitment='All' 782    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
 7. installment_commitment='All' other_payment_plans=none foreign_worker=yes 782 ==> residence_since='All' 782    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
 8. other_payment_plans=none foreign_worker=yes 782 ==> installment_commitment='All' residence_since='All' 782    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
 9. age='All' other_payment_plans=none foreign_worker=yes 782 ==> installment_commitment='All' 782    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
10. installment_commitment='All' other_payment_plans=none foreign_worker=yes 782 ==> age='All' 782    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
```

4. With a minSup=0.8 threshold, identify the top 10 association rules (based on confidence scores)? ○ What do you observe? How can you obtain the top 10 rules?

5. Now load in the ./weka-3.8/data/supermarket.arff dataset ○ With a minSup=0.5 threshold, what are the frequent 2-itemsets? ○ What are the top 3 association rules based on confidence?

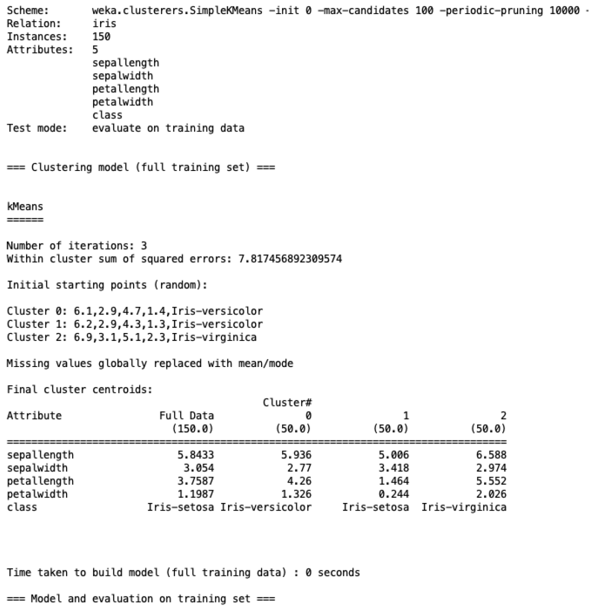


```
1. biscuits=t frozen foods=t fruit=t total=high 788 ==> bread and cake=t 723    <conf:(0.92)> lift:(1.27) lev:(0.03) [155] conv:(3.35)
2. baking needs=t biscuits=t fruit=t total=high 760 ==> bread and cake=t 696    <conf:(0.92)> lift:(1.27) lev:(0.03) [149] conv:(3.28)
3. baking needs=t frozen foods=t fruit=t total=high 770 ==> bread and cake=t 705    <conf:(0.92)> lift:(1.27) lev:(0.03) [150] conv:(3.27)
```

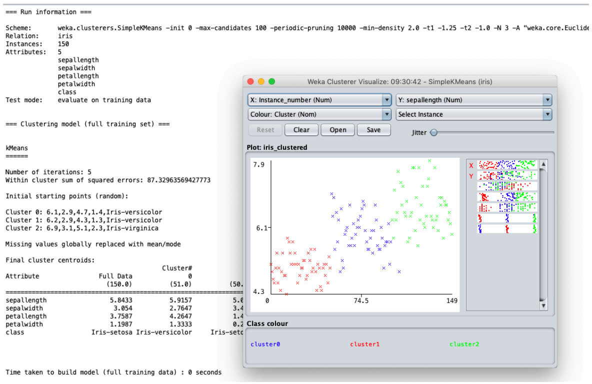
1. Load in the ./weka-3.8/data/iris.arff dataset

2. Run the k-means (SimpleKMeans) algorithm multiple times with k=3 and observe the sum of squared errors (SSE) values. ○ K-means typically return different clusters with each run, why do you observe in terms of SSE and why is this so?

3. Run k-means again, with feature normalization and without. ○ What do you observe now in terms of SSE?



```
Scheme:       weka.clusterers.SimpleKMeans -init 0 -max-candidates 100 -periodic-pruning 10000 -
Relation:     iris
Instances:    150
Attributes:   5
              sepallength
              sepalwidth
              petallength
              petalwidth
              class
Test mode:    evaluate on training data


=== Clustering model (full training set) ===


kMeans
======

Number of iterations: 3
Within cluster sum of squared errors: 7.817456892309574

Initial starting points (random):

Cluster 0: 6.1,2.9,4.7,1.4,Iris-versicolor
Cluster 1: 6.2,2.9,4.3,1.3,Iris-versicolor
Cluster 2: 6.9,3.1,5.1,2.3,Iris-virginica

Missing values globally replaced with mean/mode

Final cluster centroids:
                                    Cluster#
Attribute               Full Data          0           1           2
                          (150.0)     (50.0)      (50.0)      (50.0)
=====================================================================
sepallength                5.8433      5.936       5.006       6.588
sepalwidth                  3.054       2.77       3.418       2.974
petallength                3.7587       4.26       1.464       5.552
petalwidth                 1.1987      1.326       0.244       2.026
class                 Iris-setosa Iris-versicolor Iris-setosa Iris-virginica



Time taken to build model (full training data) : 0 seconds

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



```
=== Run information ===

Scheme:       weka.clusterers.SimpleKMeans -init 0 -max-candidates 100 -periodic-pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -1.0 -N 3 -A "weka.core.Euclide
Relation:     iris
Instances:    150
Attributes:   5
              sepallength
              sepalwidth
              petallength
              petalwidth
              class
Test mode:    evaluate on training data


=== Clustering model (full training set) ===


kMeans
======

Number of iterations: 5
Within cluster sum of squared errors: 87.32963569427773

Initial starting points (random):

Cluster 0: 6.1,2.9,4.7,1.4,Iris-versicolor
Cluster 1: 6.2,2.9,4.3,1.3,Iris-versicolor
Cluster 2: 6.9,3.1,5.1,2.3,Iris-virginica

Missing values globally replaced with mean/mode

Final cluster centroids:
                                    Cluster#
Attribute               Full Data          0
                          (150.0)     (51.0)      (50.
==============================================================
sepallength                5.8433     5.9157      5.0
sepalwidth                  3.054     2.7647      3.4
petallength                3.7587     4.2647      1.4
petalwidth                 1.1987     1.3333      0.2
class                 Iris-setosa Iris-versicolor Iris-setc



Time taken to build model (full training data) : 0 seconds
```