Action Rules

Group 3
Jagadeesh Sudhakariah
Karthik Rangaraj
Katy Mitchell
Aravind Telidevara
Keerthana Sridhar
Krupa Chand

What Are Action Rules?

- An action rule is a rule extracted from an information system that describes a transition of objects from one state to another with respect to a decision attribute.

  A rule can be defined as:

  $$r = [a_{1,1} \wedge a_{2,1} \wedge \ldots \wedge a_{p,1}] \wedge [b_{1,1} \wedge b_{2,1} \wedge \ldots \wedge b_{q,1}] \rightarrow d_1$$

  stable part flexible part

  Stable attribute: attribute's value does not change

  Flexible attributes: attribute's value can change

- Actions can be constructed from two rules extracted previously from the same database. These two rules describe two different decision classes. The goal is to re-classify objects from one of these classes into the other.
- Action rules describe knowledge about possible actions associated with objects which are hidden in a decision system.

Project Description

Extraction of Classification rules using LERS:

- The software required for the project was installed.
- The java code was run to obtain the classification rules using the LERS algorithm.
- The program returns the certain, possible and marked values after executing the LERS algorithm.

Reclassification of rules:

- ARAS generates all the classification rules using the LERS strategy.
- Clusters are built around the objects that are to be reclassified by the user by considering a decision attribute from the rules.

A Graphical user interface has been developed to ease the process of accepting input, producing the outputs, setting the support, confidence, initial, end values and selecting the decision attribute using Java.

Dataset

- **Car**: The car dataset consists of 2 files, the data file and the attribute file. The car_data.txt file contains the data on which the algorithm runs in order to calculate rules whereas the car_names.txt file contains the attribute names of the data present in car_data.txt file. The contents of these files can be seen below:

car_data.txt

```
vhigh,vhigh,2,2,small,low,unacc
vhigh,vhigh,2,2,small,med,unacc
vhigh,vhigh,2,2,small,high,unacc
```

car_names.txt

```
buying
maint
doors
persons
lug_boot
safety
class
```

- **Mammographic**: Similarly, the mammographic dataset has 2 files whose contents can be seen below:

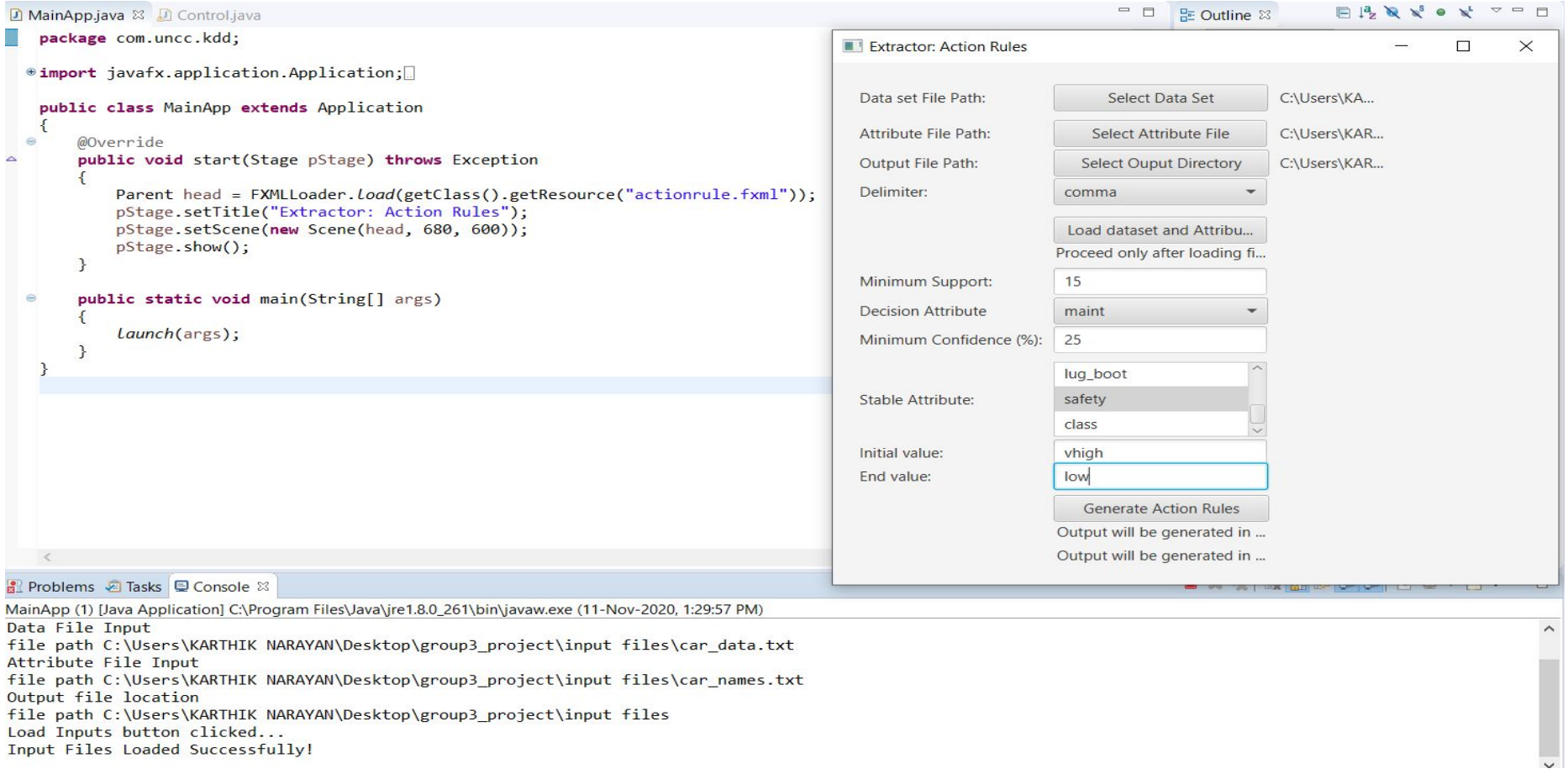
mam_data.txt

```
5,67,3,5,3,1
4,43,1,1,?,1
5,58,4,5,3,1
```

mam_names.txt

```
BI-RADS assessment
Age
Shape
Margin
Density
Severity
```

Car Dataset Evaluation



Analysis/Results

MainApp.java | Control.java

```
package com.uncc.kdd;

import javafx.application.Application;

public class MainApp extends Application
{
    @Override
    public void start(Stage pStage) throws Exception
    {
        Parent head = FXMLLoader.load(getClass().getResource("actionrule.fxml"));
        pStage.setTitle("Extractor: Action Rules");
        pStage.setScene(new Scene(head, 680, 600));
        pStage.show();
    }

    public static void main(String[] args)
    {
        launch(args);
```

Outline

- com.uncc.kdd
- MainApp
  - start(Stage) : void
  - main(String[]) : void

Problems | Tasks | Console

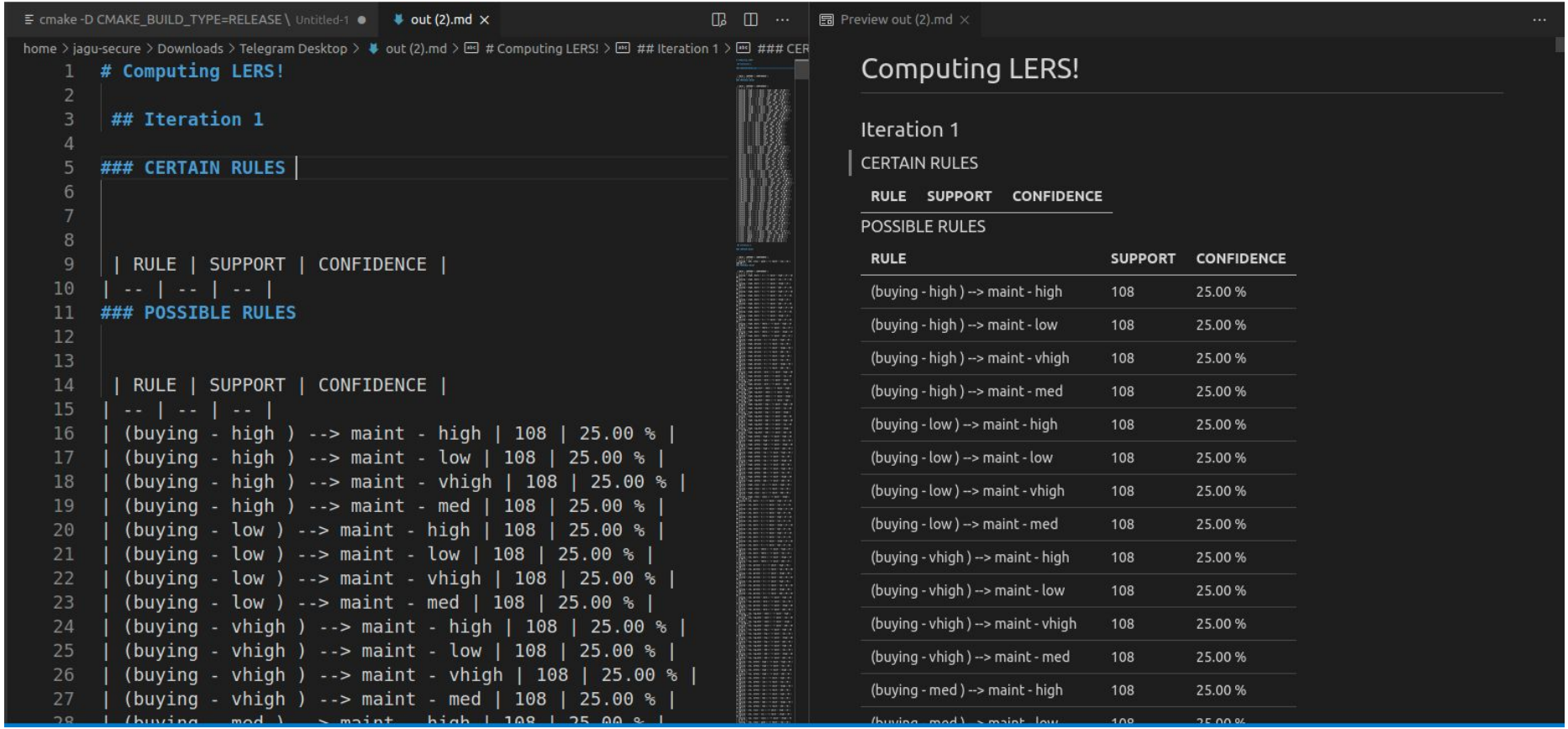
<terminated> MainApp (1) [Java Application] C:\Program Files\Java\jre1.8.0_261\bin\javaw.exe (11-Nov-2020, 1:29:57 PM)

```
$$ Computing Action Rules! $$


$$ ACTION RULES $$


(class: acc --> good ) --> (maint: vhigh --> low) #### SUPPORT: 36 #### CONFIDENCE: 26.97 percent
(safety = high, class: acc --> good ) --> (maint: vhigh --> low) #### SUPPORT: 36 #### CONFIDENCE: 26.97 percent
(safety = med, class: acc --> good ) --> (maint: vhigh --> low) #### SUPPORT: 36 #### CONFIDENCE: 26.97 percent
(safety = med, class: unacc --> good ) --> (maint: vhigh --> low) #### SUPPORT: 36 #### CONFIDENCE: 26.67 percent
(lug_boot = big, safety = med, class: unacc --> good ) --> (maint: vhigh --> low) #### SUPPORT: 36 #### CONFIDENCE: 25.00 percent
(buying: low --> vhigh, class = acc ) --> (maint: vhigh --> low) #### SUPPORT: 16 #### CONFIDENCE: 33.33 percent
(buying: low --> vhigh, lug_boot = big, class = acc ) --> (maint: vhigh --> low) #### SUPPORT: 16 #### CONFIDENCE: 33.33 percent
(buying: low --> vhigh, safety = high, class = acc ) --> (maint: vhigh --> low) #### SUPPORT: 16 #### CONFIDENCE: 33.33 percent
(lug_boot = big, safety = high, class: acc --> vgood ) --> (maint: vhigh --> low) #### SUPPORT: 16 #### CONFIDENCE: 26.67 percent
(lug_boot = big, safety = med, class: acc --> good ) --> (maint: vhigh --> low) #### SUPPORT: 16 #### CONFIDENCE: 44.44 percent
(safety = high, class: acc --> vgood ) --> (maint: vhigh --> low) #### SUPPORT: 23 #### CONFIDENCE: 27.88 percent
(safety = high, class: unacc --> vgood ) --> (maint: vhigh --> low) #### SUPPORT: 24 #### CONFIDENCE: 26.67 percent
```

Results generated as out.md



Conclusion

- LERS-type algorithm was implemented in order to construct action rules from a single classification rule.
- Relations representing rules produced by LERS strategy are marked.
- The overall complexity of the algorithm was decreased when LERS is used as the pre-processing module for ARAS .
- A Graphical user interface was created and operations were performed on it to obtain the action rules output in a text and md file.

References

1. Raś, Zbigniew W., Elżbieta Wyrzykowska, and Hanna Wasyluk. "ARAS: Action rules discovery based on agglomerative strategy." International Workshop on Mining Complex Data. Springer, Berlin, Heidelberg, 2007.
2. Im, Seunghyun, and Zbigniew W. Raś. "Action rule extraction from a decision table: ARED." International Symposium on Methodologies for Intelligent Systems. Springer, Berlin, Heidelberg, 2008.
3. Tzacheva, Angelina A., and Zbigniew W. Raś. "Action rules mining." International Journal of Intelligent Systems 20.7 (2005): 719-736.

Thank you