WPI - CS4445 Data Mining and Knowledge Discovery in Databases

Project 0: Data Pre-processing, Mining, and Evaluation of Patterns – SAMPLE SOLUTIONS

Amro Khasawneh

Part I [65 pts]

1.2.2) [5 pts] The headers of the iris.arff dataset:

```
@RELATION iris
@ATTRIBUTE sepallength  REAL
@ATTRIBUTE sepalwidth   REAL
@ATTRIBUTE petallength  REAL
@ATTRIBUTE petalwidth   REAL
@ATTRIBUTE class        {Iris-setosa, Iris-versicolor, Iris-virginica}

@DATA
…
```

1.3.3) [2 pts] The min/max of numeric attributes:

Attribute name	Min	Max
sepallength	4.3	7.9
sepalwidth	2	4.4
petallength	1	6.9
petalwidth	0.1	2.5

2.3.1) [2 pts] ZeroR predicts the majority class. Since all classes are evenly distributed, ZeroR just returns the first attribute listed “Iris-setosa” as the target class.

2.3.2) [2 pts] (Testing is using 10-fold cross validation)

```
Correctly Classified Instances         50               33.3333 %
Incorrectly Classified Instances      100               66.6667 %
Total Number of Instances             150
```

2.3.3) [5 pts] Confusion Matrix:

```
 a  b  c   <-- classified as
50  0  0 |  a = Iris-setosa
50  0  0 |  b = Iris-versicolor
50  0  0 |  c = Iris-virginica
```

Since ZeroR classification predicts the *Iris-setosa* for all test instances, we will have that column contain non-zeros in the confusion matrix, while all the others will be zeros.

2.5.1) [2 pts] OneR:

```
petallength:
        < 2.45     -> Iris-setosa
        < 4.85     -> Iris-versicolor
        >= 4.85    -> Iris-virginica
```

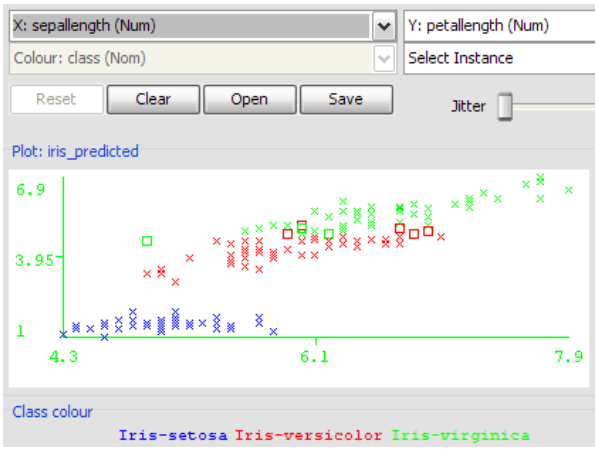
2.5.2) [2 pts] (Testing is using 10-fold cross validation)

Correctly Classified Instances	*141*	*94 %*
Incorrectly Classified Instances	*9*	*6 %*
Total Number of Instances	*150*	

2.5.3) [2 pts] Confusion Matrix:

```
 a  b  c   <-- classified as
50  0  0 |  a = Iris-setosa
 0 44  6 |  b = Iris-versicolor
 0  3 47 |  c = Iris-virginica
```

2.5.4) [10 pts]



By plotting the *petallength* attribute against any other, we can see a clear separation between the classes represented by the different colors with the same separating values found with the previous OneR rule. This fact is provided by the high 94% classification accuracy using only OneR with the mentioned attribute.

3.1.3) [4 pts] The result of applying supervised discretization:

Attribute name	Attribute value	Count
sepallength		
	(-inf-5.55]	59
	(5.55-6.15]	36
	(6.15-inf)	55
sepalwidth		
	(-inf-2.95]	57
	(2.95-3.35]	57
	(3.35-inf)	36
petallength		
	(-inf-2.45]	50
	(2.45-4.75]	45
	(4.75-inf)	55
petalwidth		
	(-inf-0.8]	50
	(0.8-1.75]	54
	(1.75-inf)	46

3.2) [7 pts] ZeroR predicts class value: Iris-setosa

Correctly Classified Instances	*50*	*33.3333 %*
Incorrectly Classified Instances	*100*	*66.6667 %*
Total Number of Instances	*150*	

```
 a  b  c   <-- classified as
50  0  0 |  a = Iris-setosa
50  0  0 |  b = Iris-versicolor
50  0  0 |  c = Iris-virginica
```

We notice that there are no differences between the result of this experiment and the result of using ZeroR on the original dataset. This is because the ZeroR classifier does not rely on any attribute but instead only returns the majority class as the result, and we know that supervised discretization never changes the class attribute.

3.2) [7 pts] OneR returns the following rule:

petalwidth:
'(-inf-0.8]' *-> Iris-setosa*
'(0.8-1.75]' *-> Iris-versicolor*
'(1.75-inf)' *-> Iris-virginica*

Correctly Classified Instances	*141*	*94 %*	*a b c <-- classified as*
Incorrectly Classified Instances	*9*	*6 %*	*50 0 0 | a = Iris-setosa*
Total Number of Instances	*150*		*0 46 4 | b = Iris-versicolor*
			0 5 45 | c = Iris-virginica

This time, the result returned by OneR when applied to discretized data is different than the one applied to the raw data. This is expected since the attributes have changed from numeric to nominal, and we know that OneR finds the attribute that best distinguishes the class value of the training data (using entropy), and in this case it happens to be *petalwidth*. One interesting remark is that the classification accuracy is still the same as before coincidently.

4) [15 pts] //CODE
-calculateCutPoints()
-calculateCutPointsByMDL()
-calculateCutPointsForSubset()
-FayyadAndIranisMDL()
…

Outline of the supervised Discretize algorithm:

> *Do for each attribute:*
> *- sort the instances by this attribute values*
> *- consider possible breakpoint (ex: breakpoint is not allowed to separate items of the same class)*
> *- calculate the entropy for each possible cut*
> *- choose the cut that minimize the entropy*
> *- repeat this process on the lower/upper parts of the range recursively, with MDL as stopping criterion.*

The minimum description length (MDL) criterion is defined as:

$$gain > \frac{\log_2(N-1)}{N} + \frac{\log_2(3^k-2) - kE + k_1 E_1 + k_2 E_2}{N}$$

N = # instances
Original set: k classes, entropy E
First subset: k_1 classes, entropy E_1
Second subset: k_2 classes, entropy E_2

Part II [35 pts]

2) [15 pts] The census-income.arff file headers:

```
@relation census-income-data

@attribute age                real
@attribute workclass          { Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov, State-gov, Without-pay, Never-worked }
@attribute fnlwgt             real
@attribute education          { Bachelors, Some-college, 11th, HS-grad, Prof-school, Assoc-acdm, Assoc-voc, 9th, 7th-8th, 12th, Masters, 1st-4th, 10th, Doctorate, 5th-6th, Preschool }
@attribute education-num      real
@attribute marital-status     { Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse }
@attribute occupation         { Tech-support, Craft-repair, Other-service, Sales, Exec-managerial, Prof-specialty, Handlers-cleaners, Machine-op-inspct, Adm-clerical, Farming-fishing, Transport-moving, Priv-house-serv, Protective-serv, Armed-Forces }
@attribute relationship       { Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried }
@attribute race               { White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black }
@attribute sex                { Female, Male }
@attribute capital-gain       real
@attribute capital-loss       real
@attribute hours-per-week     real
@attribute native-country     { United-States, Cambodia, England, Puerto-Rico, Canada, Germany, Outlying-US(Guam-USVI-etc), India, Japan, Greece, South, China, Cuba, Iran, Honduras, Philippines, Italy, Poland, Jamaica, Vietnam, Mexico, Portugal, Ireland, France, Dominican-Republic, Laos, Ecuador, Taiwan, Haiti, Columbia, Hungary, Guatemala, Nicaragua, Scotland, Thailand, Yugoslavia, El-Salvador, Trinadad&Tobago, Peru, Hong, Holand-Netherlands }
@attribute class              { >50K, <=50K }

@data
39, State-gov, 77516, Bachelors, 13, Never-married, Adm-clerical, Not-in-family, White, Male, 2174, 0, 40, United-States, <=50K
50, Self-emp-not-inc,83311,Bachelors,13,Married-civ-spouse, Exec-managerial, Husband, White, Male, 0, 0, 13, United-States, <=50K
38, Private, 215646, HS-grad, 9, Divorced, Handlers-cleaners, Not-in-family, White, Male, 0, 0, 40, United-States, <=50K
53, Private, 234721, 11th, 7, Married-civ-spouse, Handlers-cleaners, Husband, Black, Male, 0, 0, 40, United-States, <=50K
28, Private, 338409, Bachelors, 13, Married-civ-spouse, Prof-specialty, Wife, Black, Female, 0, 0, 40, Cuba, <=50K
37, Private, 284582, Masters, 14, Married-civ-spouse, Exec-managerial, Wife, White, Female, 0, 0, 40, United-States, <=50K
49, Private, 160187, 9th, 5, Married-spouse-absent, Other-service, Not-in-family, Black, Female, 0, 0, 16, Jamaica, <=50K
52, Self-emp-not-inc, 209642, HS-grad, 9, Married-civ-spouse, Exec-managerial, Husband, White, Male, 0, 0, 45, United-States, >50K
31, Private, 45781, Masters, 14, Never-married, Prof-specialty, Not-in-family, White, Female, 14084, 0, 50, United-States, >50K
42, Private, 159449, Bachelors, 13, Married-civ-spouse, Exec-managerial, Husband, White, Male, 5178, 0, 40, United-States, >50K
```

4) [10 pts] The ReplaceMissingValues filter works on each attribute in the dataset and replaces all missing values "?" with either the mean/mode of the attribute depending on the type of this attribute. This means for nominal attributes, missing values will be replaced by the majority (mode) value. Similarly for numeric attributes, missing values are replaced by the mean (average) value.

5) [10 pts] Resample filter: *Produces a random subsample of a dataset using either sampling with replacement or without replacement […].*

- *invertSelection*: works only for sampling without replacement. The resulting instances from the sampling process are the instances NOT considered in the final set of instances (instead, all the other unselected instances). Note that if the *sampleSizePercent* is used, we end up with (100 - *sampleSizePercent)%* of the original dataset size.
- *noReplacement*: choose sampling with/without replacement
- *randomSeed*: a seed passed to the random number generator. Can be used to reproduce experiments.
- *sampleSizePercent*: size of the final subsample as a percentage of the original dataset.