IST565 Data Mining
Professor Bei Yu

RWeka Tutorial

You can use the "RWeka" package to run Weka in R. Below is a step-by-step tutorial on how to run J48 on the Titanic data using RWeka.

For more details, check out the official RWeka manual at https://cran.r-project.org/web/packages/RWeka/RWeka.pdf

Step1: INSTALL R PACAKGE

```
install.packages("RWeka")
```

Note: If you are installing the RWeka package on a windows pc, you may need to set the Java environment first by execute the following script in R:

```
Sys.setenv(JAVA_HOME="C:\\Program Files\\Java\\jdk1.8.0_51\\jre")
```

Step2: READ FILES (change file path to your own file path)

```
trainset <- read.csv("/Users/byu/Documents/R/train.csv")
testset <- read.csv("/Users/byu/Documents/R/test.csv")
```

Or

```
trainset <- read.arff("/Users/byu/Documents/R/train.arff")
testset <- read.arff("/Users/byu/Documents/R/test.arff")
```

Step3: PREPROCESS DATA IN R

We can use either the built-in R functions or the RWeka filter interface, here we are introducing the RWeka interface for data preprocessing.

- **Transform data type**
  Numeric –> Nominal

NN <- make_Weka_filter("weka/filters/unsupervised/attribute/NumericToNominal") # *build a function using RWeka filter interface*

Apply the filter function to both training and test datasets.

trainset <- NN(data=trainset, control= Weka_control(R="1-3"), na.action = NULL)

testset <- NN(data=testset, control= Weka_control(R="1,3"), na.action = NULL)

- **Deal with missing values**

MS <- make_Weka_filter("weka/filters/unsupervised/attribute/ReplaceMissingValues") # *build a function using RWeka filter interface*

Apply the filter function to both training and test datasets.

trainset <-MS(data=trainset, na.action = NULL)

testset <-MS(data=testset, na.action = NULL)

- **Check data definition (optional)**

  str(trainset)

Step4: APPLY J48 Algorithm

- **Build decision tree model**

  m=J48(Survived~., data = trainset)

  m=J48(Survived~., data = trainset, control=Weka_control(U=FALSE, M=2, C=0.5))

  * View parameters with function WOW:

  i.e. WOW("J48")

```
> WOW("J48")
-U      Use unpruned tree.
-O      Do not collapse tree.
-C <pruning confidence>
        Set confidence threshold for pruning.  (default 0.25)
        Number of arguments: 1.
-M <minimum number of instances>
        Set minimum number of instances per leaf.  (default 2)
        Number of arguments: 1.
-R      Use reduced error pruning.
-N <number of folds>
        Set number of folds for reduced error pruning. One fold is used as pruning set.
        (default 3)
        Number of arguments: 1.
-B      Use binary splits only.
-S      Do not perform subtree raising.
-L      Do not clean up after the tree has been built.
-A      Laplace smoothing for predicted probabilities.
-J      Do not use MDL correction for info gain on numeric attributes.
-Q <seed>
        Seed for random data shuffling (default 1).
        Number of arguments: 1.
```

- **Use 10 fold cross-validation to evaluate the model**

  e <- evaluate_Weka_classifier(m,

  numFolds = 10,

  seed = 1, class = TRUE)

```
=== Summary ===

Correctly Classified Instances         720               80.8081 %
Incorrectly Classified Instances       171               19.1919 %
Kappa statistic                          0.5753
Mean absolute error                      0.2646
Root mean squared error                  0.3659
Relative absolute error                 55.9347 %
Root relative squared error             75.2366 %
Coverage of cases (0.95 level)          99.5511 %
Mean rel. region size (0.95 level)      91.9753 %
Total Number of Instances              891

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.918    0.368    0.800      0.918    0.855      0.587    0.827     0.846     0
                 0.632    0.082    0.828      0.632    0.716      0.587    0.827     0.796     1
Weighted Avg.    0.808    0.258    0.811      0.808    0.802      0.587    0.827     0.827

=== Confusion Matrix ===

   a   b   <-- classified as
 504  45 |   a = 0
 126 216 |   b = 1
```

- **Apply the model with test dataset**

  pred=predict (m, newdata = testset, type = c(“class”))

Predict Result:

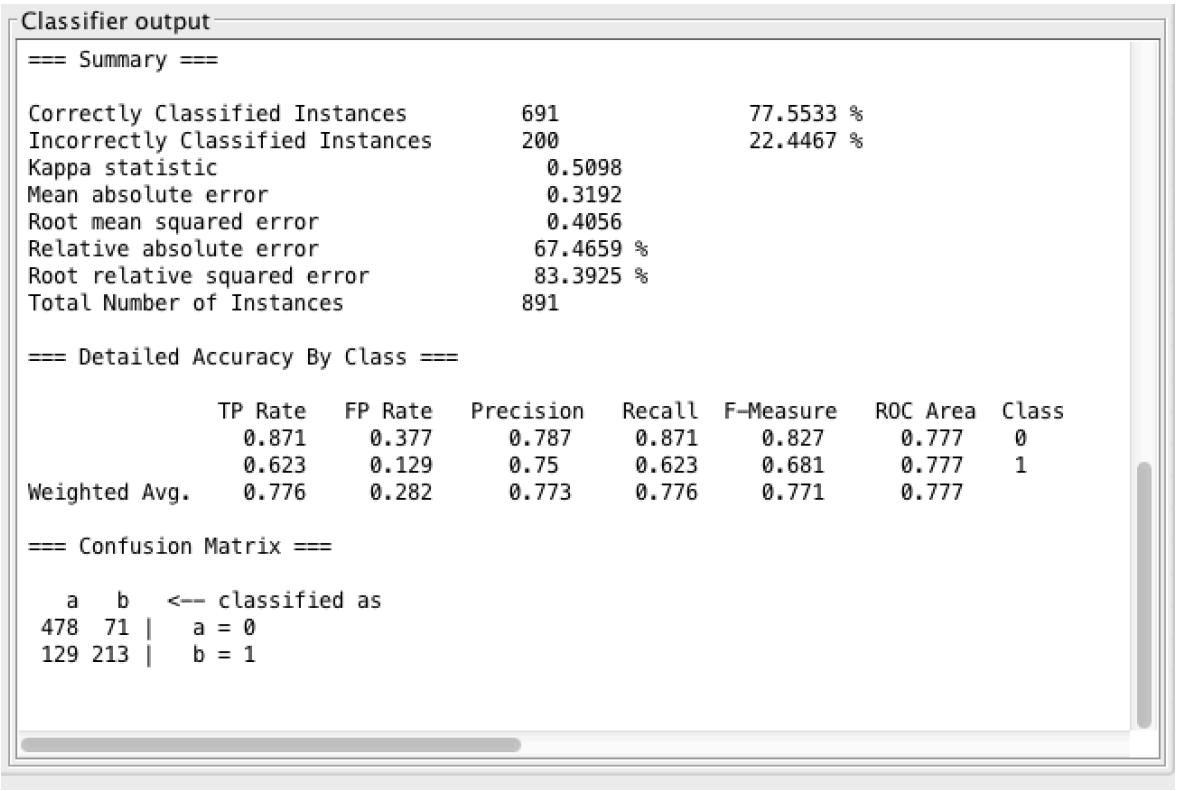
```
> predict(m, newdata = testset, type = c("class"))
  [1] 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 0 0 0 0 0 1 1 0 1 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 1
 [50] 0 0 0 1 1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 1 0 0 0 0 1 0 0 1 0 0 1 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 1 0
 [99] 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 1 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0
[148] 0 0 0 1 0 0 0 0 0 1 0 0 0 0 1 1 0 0 0 0 0 1 0 0 0 0 0 0 1 1 0 1 1 0 0 1 0 1 0 1 0 1 0 0 0 0 0 1 0
[197] 1 0 0 0 0 1 0 1 0 0 0 0 1 0 0 0 0 1 1 0 0 0 1 0 1 0 1 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 1 1 1 1 0 0 0
[246] 0 1 0 1 0 1 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0
[295] 0 0 1 0 0 0 0 0 0 0 1 1 0 1 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
[344] 1 0 0 0 0 0 1 1 0 0 0 1 0 1 0 0 0 0 1 1 0 1 1 0 0 1 0 0 1 0 0 1 1 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 1
[393] 0 0 0 1 0 1 0 0 1 0 1 0 0 0 0 0 0 0 0 1 1 0 1 0 0 0
Levels: 0 1
```

Then save the prediction to a csv file:

write.csv(pred, file="path/filename.csv")

Note that the model built by RWeka may not be exactly the same as the model built in Weka, if the pre-processing steps are different. For example, to build the above RWeka model, missing values were replaced by mean for the "age"

attribute. However, the Weka GUI's filter "replaceMissingValues" actually applies to all variables, causing slightly different model and performance.



Removing irrelevant attributes:

```
myVars=c("Pclass", "Sex", "Age", "SibSp", "Parch", "Fare", "Survived")
newtrain=trainset[myVars]
newtest=testset[myVars]
m=J48(Survived~., data = newtrain)
m=J48(Survived~., data = newtrain, control=Weka_control(U=FALSE, M=2, C=0.5))
e=evaluate_Weka_classifier(m, seed=1, numFolds=10)
pred=predict (m, newdata = newtest, type = c("class"))
myids=c("PassengerId")
id_col=testset[myids]
newpred=cbind(id_col, pred)
colnames(newpred)=c("Passengerid", "Survived")
View(newpred)
```

write.csv(newpred, file="/Users/byu/Desktop/Data/titanic-J48-pred.csv", row.names=FALSE)

InfoGainAttributeEval(Survived ~ . , data = trainset)

Benefit of RWeka:

1. **RWeka does not give incompatibility error when reading in both train.csv and test.csv. Weka GUI did.**
2. **Running Weka in R would allow you to use Weka prediction algorithms and also the rich data pre-processing tools in R.**