Data Mining in Medicine: Predicting Outcomes after the Onset of Stroke

A Project by

Patrick Granada

Isabelle Tingzon

Submitted to Sir Pros Naval

Department of Computer Science

College of Engineering

University of the Philippines

Diliman, Quezon City

2015

Contents

1 Introduction

Data mining plays an important role in predicting the outcomes of diseases in the medical field. An early and accurate prognosis for recovery in patients is important for initiation of treatment and for informing patients and relatives. One fatal disease is Cerebrovascular Accident (CVA), commonly known as stroke, which affects more that 15 million people worldwide each year. In the Philippines, stroke remains to be among the top causes of death making up almost 10% of total deaths. [1]

In this project, we used data mining techniques to build models to predict the status of the patient 6 months after the onset of stroke. We built decision tree models, generated association rules, and analyzed the accuracy of the models produced by the C5.0 algorithm [3] and the CART model [4].

2 Objectives

In this mini project, we aim to demonstrate my knowledge of classification data mining by using the C5.0 algorithm to mine association rules and generate decision trees from a given data set.The objectives of this Mini Project are as follows:

- To write R code that uses C5.0 algorithm to generate a decision tree model and a set of association rules for the The International Stroke Trial (IST) Data Set that predicts the status of the patient 6 months after the onset of stroke.
- To analyze the generated decision tree and association rules and to report the accuracy of the classification model.
- To gain a deeper understanding of data mining through classification and to apply the predictive model to a real world application, specifically in the medical field.

3 Methodology

3.1 Preliminaries

In this section, we discuss the preliminaries and background used in predictive data mining.

3.1.1 Classification and Decision Trees

One important concept in this assignment is *classification*. Classification is the task of assigning objects to a specific category. More specifically, it is the task of learning a target function f that maps each attribute x to some predefined category y. Classification has two model types: namely *descriptive model* and *predictive model*. In this project, we are interested in the predictive model, specifically a model that predicts the status of a patient 6 months after the onset of stroke.

One classification technique among many is the decision tree-based model. Decision trees are used to break down a data set into smaller subsets. The leaf nodes represent a category, decision, or *class*. All other nodes are called *features* and are used to split data (using a calculated entropy, gini index, etc.).

3.1.2 Association Rules

An association rule is an implication of the form $(A \Rightarrow B)$ where $A, B \neq \emptyset$, $A \subset I$, $B \subset I$, and $A \cap B \neq \emptyset$. Association rules are used to identify and uncover relationships among data. In this mini project, we used association rules to identify conditions and diagnoses that may lead to a certain patient

status after a 6 month follow up.

3.1.3 C5.0 Algorithm

Ross Quinlan was known for developing tree-based models (e.g. ID3 and C4.5). Quinlan continually worked on classification tree and rule-based models, and in the 1980s created C5.0, an extension of C4.5. In this project, we used the C5.0 package developed by Kuhn, Weston, and Coulter in R to build predictive decision trees and generate association rules.

To install C5.0, we run the R system and load the C5.0 package using the command:

Listing 1: Installing C5.0 Library in R

```
install.packages("C50")
```

We can then load the C5.0 library in R using the command:

Listing 2: Loading C5.0 Library in R

```
library(c50)
```

3.2 The International Stroke Trial Database

We obtained the dataset from The International Stroke Trial (IST) Database which includes data on 19,435 patients with acute stroke, with 99% complete follow-up. [2] A full description of the list of features of the dataset can be seen in the attached file ISTB_Features.pdf.

3.3 Classification with C5.0 Algorithm

3.3.1 Data Pre-processing Techniques

The data mining process may involve pre-processing steps in order to assure that the data set have the quality and the format required by the algorithms [**?**]. To process the dataset from The International Stroke Trial (IST) Database, we read the .csv file and assign the corresponding column names (see ISTB_Features.pdf) as follows:

Listing 3: Loading data for C5.0 Algorithm

```
#Load data and label features
istdb <- read.csv("IST_DATABASE.csv")

istdata <- istdb[c("SEX", "AGE",
          "DASPLT", "RVISINF", "RSBP",
          "RDEF1","RDEF2","RDEF3","RDEF3",
          "RDEF4","RDEF5","RDEF6","RDEF7",
          "STYPE", "DDIAGISC", "DDIAGHA",
          "DDIAGUN", "DNOSTRK")]
```

Since some attributes contained blank entries, we used "U" for *unknown* as a space filler. We took into consideration that this may alter the results.

Listing 4: Cleaning up data

```
levels(istdata$DNOSTRK )[1] = "U"
levels(istdata$DDIAGUN)[1] = "U"
levels(istdata$DDIAGHA)[1] = "U"
levels(istdata$DDIAGISC)[1] = "U"
```

```
levels(istdata$DASPLT)[1] = "U"
```

We then collapse the columns "DDEAD", "FDEAD", "DALIVE", "FRECOVER", "FDENNIS", "UNKNOWN" into one feature called PROG.

Listing 5: Cleaning up data

```
PROG <- matrix(0, ncol = 1, nrow = nrow(istdb))
istdb.PROG <- data.frame(PROG)
levels(istdb.PROG$PROG) <- c("DDEAD",
 "FDEAD", "DALIVE", "FRECOVER",
 "FDENNIS", "UNKNOWN")
for (i in 1:nrow(istdb)){
  if (istdb$DDEAD[i] == "Y"){
    istdb.PROG$PROG[i] <- "DDEAD"
  } else if (istdb$FDEAD[i] == "Y"){
    istdb.PROG$PROG[i] <- "DDEAD"
  } else if (istdb$DALIVE[i] == "Y"){
    istdb.PROG$PROG[i] <- "DALIVE"
  } else if (istdb$FRECOVER[i] == "Y"){
    istdb.PROG$PROG[i] <- "DALIVE"
  } else if (istdb$FDENNIS[i] == "Y"){
    istdb.PROG$PROG[i] <- "FDENNIS"
  } else {
    istdb.PROG$PROG[i] <- "UNKNOWN"
  }
}
        istdata["PROG"] <- istdb.PROG
        levels(istdata$PROG) <- c("DDEAD",
```

```
"FDEAD", "DALIVE", "FRECOVER",
"FDENNIS", "UNKNOWN")

istdata$PROG <- factor(istdata$PROG)
```

3.3.2 Data Processing with C5.0

After cleaning the data, its order was randomized such that each and every tuple of the dataset has a chance to be included in either the training or testing dataset. The following code was used to randomize the dataset.

Listing 6: Randomizing Data

```
istdata <-
        istdata[ sample( nrow( istdata ) ), ]
```

After randomizing, the data was partitioned into two groups namely training set and testing set. The training set is to be used as the input to generate the decision tree while the testing set is used to test the accuracy of the model generated. The ratio of the training and testing set is 80:20 respectively.

Listing 7: Splitting Training Set and Testing Set

```
testingSet <- istdata[15549:19435,]
trainingSet <- istdata[1:15548,]

X<-trainingSet[,1:18]
y<-trainingSet$PROG

X1<-testingSet[,1:18]
y1<-testingSet$PROG
```

The decision tree model was generated using the training set and was used to predict the labels given the attributes of testing dataset. The labels predicted are then compared with the ground truth values of the testing dataset and the accuracy of is computed by acquiring the number of correct matches over the total number of tuples.

Listing 8: Generating the Decision Tree

```
#Generate decision tree model
treeModel <- C5.0(PROG ~ ., data=trainingSet)
summary(treeModel)
```

The association rules of the generated tree were also extracted.

Listing 9: Generating the Association Rules

```
rules <- C5.0(PROG ~ ., data=istdata,
          rules = TRUE)
summary(rules)
```

3.3.3 Data Post-processing Techniques

To improve the quality of the results, we implemented some post-processing techniques on the produced results. In particular, we applied boosting techniques to achieve better accuracy through a series of trials.

To predict the labels for test data using the tree model with boosting, we invoke

Listing 10: Boosting

```
boostTreeModel <- C5.0(PROG ~ ., data
          = trainingSet, trials = 5)
```

4 Analysis and Discussion of Results

As data miners, our goal is to obtain rules and models for better decision support that may aid to benefit medical professionals. In this section, we present the analysis of the experimental results produced by the C5.0 algorithm as seen in the previous section.

4.1 Decision Trees

The details of the decision tree we obtained from the C5.0 algorithm are as follows:

Listing 11: C5.0 Decision Tree

```
Evaluation on training data (15548 cases):
    Decision Tree
  ----------------
  Size      Errors

   523  5326(34.3%)   <<


   (a)   (b)   (c)   (d)    <-classified as
  ----  ----  ----  ----
  7328   479   185     7    (a): class DALIVE
  1123  2173   159     5    (b): class DDEAD
  1856   596   661     7    (c): class FDENNIS
   791    75    43    60    (d): class UNKNOWN
```

```
Attribute usage:
  100.00% DASPLT
   78.20% RDEF4
   68.82% STYPE
   49.49% AGE
   44.15% RDEF5
   35.86% RDEF3
   35.60% RDEF2
   25.63% RDEF6
   22.53% RDEF7
   21.76% DDIAGUN
   19.99% DNOSTRK
   19.35% DDIAGHA
   19.24% RSBP
   17.17% RDEF1
   10.79% RVISINF
    9.80% DDIAGISC
    9.26% SEX
Time: 0.4 secs
```

After boosting, we obtained the a boosted tree model with the following details:

Listing 12: Boosted Decision Tree

```
Evaluation on training data (15548 cases):

Trial        Decision Tree
-----      ----------------
```

	Size	Errors	
0	523	5326(34.3%)	
1	136	5941(38.2%)	
2	149	5965(38.4%)	
3	147	5899(37.9%)	
4	252	5789(37.2%)	
boost		5397(34.7%)	<<

(a)	(b)	(**c**)	(d)	<-classified **as**
7271	572	154	2	(a): **class** DALIVE
1067	2286	107		(b): **class** DDEAD
1819	725	574	2	(**c**): **class** FDENNIS
812	98	39	20	(d): **class** UNKNOWN

Attribute usage:
100.00% DASPLT
100.00% RDEF5
99.39% STYPE
99.04% RDEF4
90.67% RDEF6
60.88% AGE
60.05% RDEF3
56.05% RDEF2
53.42% DNOSTRK

```
45.91% RDEF1
41.93% DDIAGUN
40.65% RSBP
40.08% RDEF7
35.27% DDIAGHA
25.36% DDIAGISC
22.80% SEX
20.32% RVISINF
Time: 0.6 secs
```

We also used CART (Classification and Regression Trees) to achieve a visual representation of the the data.

Listing 13: CART Modeling

```
form <- as.formula(PROG ~ .)
fit <-
rpart(form,data=istdata,
 control=rpart.control(minsplit=20,cp=0))
pfit<- prune(fit, cp=fit
          $cptable[which.min(fit
          $cptable[,"xerror"]),"CP"])
plot(pfit, uniform=TRUE)
text(pfit, use.n=TRUE, all=TRUE, cex=.5)
```

We obtained the following tree.

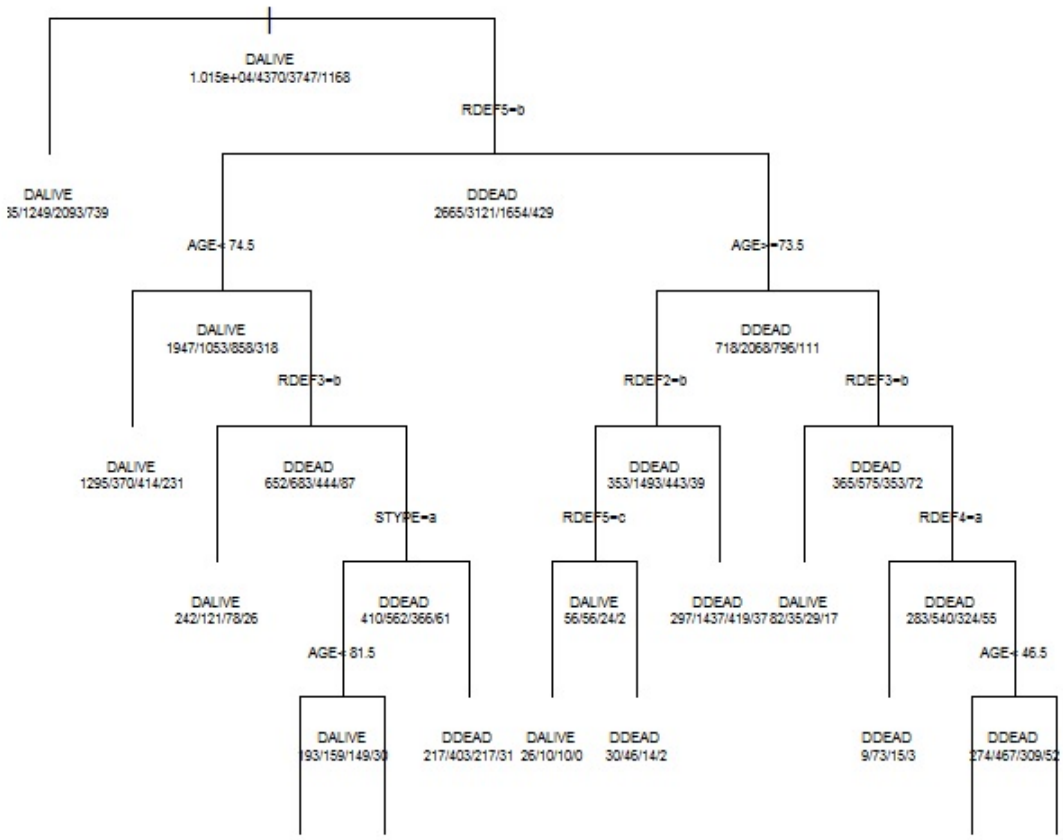

Figure 1: CART Model

4.2 Association Rules

The first 5 association rules obtained from the data using C5.0 algorithm is shown in Figure 2.

```
Rule 1: (11, lift 1.8)
  SEX = M
  AGE > 62
  AGE <= 69
  RSBP > 143
  RSBP <= 155
  RDEF1 = Y
  RDEF5 = N
  RDEF6 = Y
  ->  class DALIVE  [0.923]

Rule 2: (155/24, lift 1.6)
  DASPLT = Y
  DNOSTRK = Y
  ->  class DALIVE  [0.841]

Rule 3: (1105/196, lift 1.6)
  AGE <= 73
  RDEF1 = N
  RDEF3 = N
  RDEF4 in {N, Y}
  ->  class DALIVE  [0.822]

Rule 4: (57/10, lift 1.6)
  STYPE = OTH
  ->  class DALIVE  [0.814]

Rule 5: (137/25, lift 1.6)
  SEX = M
  RDEF2 = N
  RDEF4 = N
  RDEF5 in {C, Y}
  ->  class DALIVE  [0.813]
```

Figure 2: Association Rules

There are several interesting association rules garnered. An example of interesting rules is

Listing 14: Sample Assocation Rule

```
Rule 122: (21/7, lift 3.4)
AGE > 71
RVISINF = N
RSBP > 145
RDEF3 = Y
STYPE in {PACS, POCS}
DDIAGISC = Y
-> class FDENNIS [0.652]
```

This means that a patient whose age is greater than 71, with Visible infarct in CT, blood pressure greater than 145, with a leg or foot deficit, Partial Anterior Circulation Stroke Syndrome (PACS) or Posterior Circulation Stroke Syndrome (POCS), and with Ischaemic Heart Disease is expected to be dependent (on a 6 month follow up).

4.3 Accuracy of Models

To get the accuracy of the model, we predicted the Testing Set as follows:

Listing 15: Predicting the Test Set

```
pred <- predict(treeModel, X1)
```

To get the ratio of the number of all correctly classified patients over the total number of patients, we simple use:

Listing 16: Tree Model Accuracy

```
pred <- predict(treeModel, X1)
```

We achieved a **60**% **accuracy** for this model.

Similarly, to get the accuracy of the boosted tree model, we invoke

Listing 17: Tree Model Accuracy

```
pred <- predict(boostTreeModel, X1)
```

We achieved a **62**% **accuracy** for the boosted model.

5 Conclusion

In this project, we used data mining techniques to build models to predict the status of the patient 6 months after the onset of stroke. We built decision tree models, generated association rules, and analyzed the accuracy of the models produced by the C5.0 algorithm and the CART model.

The researchers successfully generated a model with the 62% as the highest accuracy attained. We believe these results will be beneficial to medical professionals to make better decisions and realize trends in stroke prognosis.

5.1 Future Work

Since the researchers have a limited knowledge about the stroke and its complications, the features extracted are based through consulting with medical professionals. However, there might be interesting features not included in the dataset or there are certain features that will yield better accuracy for the decision tree. In line with this, using Factorial Analysis for Mixed Data may help in getting the best set of features.

In addition, there are other means of classification used by data mining. Usage of neural networks, bayesian classification, and other approaches may deem viable for this kind of dataset. The accuracy of the models should then be compared in order to produce the model that yields the best accuracy.

References

[1] http://www.strokecenter.org/patients/about-stroke/stroke-statistics/

[2] http://www.trialsjournal.com/

[3] http://cran.r-project.org/web/packages/C50/C50.pdf

[4] http://cran.r-project.org/web/packages/rpart/rpart.pdf

[5] http://www.strokeassociation.org/