Stat 6021 R Tutorial: ROC Curve and Multinomial Logistic Regression

There are two parts to this tutorial. In part 1, you will learn how to generate a Receiver Operating Characteristic (ROC) curve, as well as compute the Area Under the ROC Curve (AUC) to validate a logistic regression model. In part 2, you will learn how to fit a multinomial logistic regression model, which is used when we have a response variable that follows a multinomial distribution (categorical with more than 2 outcomes).

1. For the first part, we will use the data set "titanic.txt". The data set contains information regarding the survival of passengers along with a number of variables such as gender, age, and how much the passenger paid for the fare.

   (a) To validate a model, we need to split the data set into two: one part being the training data set, which we use to build a model, and the other part being the testing data set, which we use to validate the model. We will randomly perform this splitting. For example, type

   ```
   set.seed(111)
   sample<-sample.int(nrow(data), floor(.50*nrow(data)), replace = F)
   train<-data[sample, ]
   test<-data[-sample, ]
   ```

   In the example code above, the function `set.seed()` will ensure the same data points are sampled each time the code is run. This function should be removed if you want to assess the performance of the model over different randomly split data. The `sample.int()` function is used to randomly select the observations to be part of the training data set. In this example, I use half the data points to belong to the training data set, and the remaining to belong to the testing data set.

   (b) We will fit a logistic regression model using the amount paid and the gender as predictors,

   ```
   result<-glm(Survived ~ Fare + Sex, family=binomial, data=train)
   ```

In this example, notice that I did not use the `attach()` function. It is not recommended to use the `attach()` function if you are working with multiple data sets. In this example, I have two data sets, the training and testing data sets. Since there is no default data set attached, we need to specify which one we are using in the `glm()` function.

(c) To produce the ROC curve for this logistic regression model, we will be using a number of functions from the `ROCR` package. Type

```
library(ROCR)
preds<-predict(result,newdata=test, type="response")
rates<-prediction(preds, test$Survived)
roc_result<-performance(rates,measure="tpr", x.measure="fpr")
plot(roc_result)
lines(x = c(0,1), y = c(0,1), col="red")
```

The variable `preds` stores the predicted probabilities for our logistic regression model using the testing data set. The variable `rates` stores the numbers associated with the classification table. The variable `roc_result` stores the values of the true positive rate and false positive rate via the `performance()` function. The arguments `measure` and `x.measure` are used to plot the true positive rate on the y-axis and false positive rate on the x-axis respectively. The function `lines()` is used to overlay the diagonal line on the plot for ease of comparison between random guessing and our model's classification ability.

(d) To calculate the value of the AUC, type

```
auc<-performance(rates, measure = "auc")
auc
```

(e) To create a confusion matrix using 0.5 as the cutoff value, type

```
table(test$Survived, preds>0.5)
```

This places the actual class in the rows, and the predicted class in the columns. What is the false positive rate when the cutoff value is 0.5? How should the cutoff value be changed if we need to lower the false positive rate?

2. For the second part of this tutorial, we will use the data set "contraceptive.txt". The data come from a national survey conducted in Indonesia regarding the use of contraceptives. The subjects are married women who were either not pregnant or do not know if they were at the time of interview. The problem is to predict the current contraceptive method choice (no use, long-term methods, or short-term methods) of a woman based on her demographic and socio-economic characteristics. The variables in the data set are:

   - Wife's age

- Wife's education (coded on a 1 to 4 scale in increasing levels of education attainment)
- Husband's education (coded on a 1 to 4 scale in increasing levels of education attainment)
- Number of children born
- Wife's religion (coded 1 for Islam, 0 for other)
- Wife working? (coded 1 for no, 0 for yes)
- Husband's occupation (coded 1, 2, 3, or 4)
- Standard of Living Index (on a 1 to 4 scale)
- Media exposure (coded 1 for not good, 0 for good)
- Contraceptive method used (coded 1 for no use, 2 for long-term, 3 for short-term)

Suppose we want to investigate if the number of children a woman has and her religion influences the probability of which contraceptive method she uses.

(a) Before building our multinomial logistic regression models, notice how this data set, in the original .txt file, does not have headings for the columns. So, to read the data in, type

```
data<-read.table("contraceptive.txt", header=FALSE, sep="")
```

(b) In a data set with this many variables, it is advisable to give names for each column. To append names for each column, type

```
colnames(data)<-c("wife_age", "wife_edu", "hus_edu", "children",
"wife_rel", "wife_work", "hus_job", "sol", "media", "c_method")

attach(data)
```

I highly suggest using descriptive headings for the columns.

(c) Notice how the data set uses numbers to code categorical variables. R would treat these variables as quantitative even though they are categorical. To check, type `is.numeric(wife_rel)` and you will notice that R treats this column as a quantitative variable. To inform R that this column is categorical, type `wife_rel<-factor(wife_rel)` and then type `is.factor(wife_rel)` that this column is now treated as a categorical variable.

(d) Notice that the outcomes associated with this column are still numbers. Type `levels(wife_rel)` informs us that the classes are 0 and 1, in that order. To append descriptive names to these outcomes, type

```
levels(wife_rel)<-c("non_Islam","Islam")
```

This ensures 1 is coded for a woman who follows Islam and 0 is coded for a woman who does not follow Islam. Please perform the same operations for the column associated with contraceptive method.

(e) Now we are ready to fit the multinomial logistic regression models. The function `multinom()` belongs to the package `nnet`.

```
library(nnet)
result<-multinom(c_method ~ children + wife_rel)
summary(result)
```

(f) Notice this function does not provide the Wald test statistics and p-values. To do so, you have to calculate these on your own. You can type

```
z<-summary(result)$coefficients/summary(result)$standard.errors
p<-(1 - pnorm(abs(z)))*2
```

to generate the Wald test statistics and p-values for each of the coefficients.