Analysis Notebook

Jae Yong (Francisco) Lee

2018-11-19

Data Preparation

Data was sourced from Kaggle

```
setwd("~/repos/ckme136-capstone/")
hoq <- read.csv(file = "dataset/final_hoq.csv", header = TRUE,
    stringsAsFactors = TRUE)
str(hoq)
```

```
## 'data.frame':    3399 obs. of  12 variables:
##  $ h_type     : Factor w/ 2 levels "'Acute Care Hospitals'",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ h_ownership: Factor w/ 10 levels "'Government - Federal'",..: 7 5 2 7 7 8 7 7 6 6 ...
##  $ h_es       : Factor w/ 2 levels "No","Yes": 2 2 2 2 1 2 2 2 2 2 ...
##  $ h_ehr      : Factor w/ 2 levels "?","Y": 2 2 2 2 2 2 2 2 2 2 ...
##  $ h_rating   : int  5 5 5 5 5 5 5 5 5 5 ...
##  $ h_mortality: Factor w/ 4 levels "'Above the national average'",..: 4 1 4 1 1 3 4 1 4 3 ...
##  $ h_soc      : Factor w/ 4 levels "'Above the national average'",..: 3 3 1 1 1 1 4 1 1 1 ...
##  $ h_ra       : Factor w/ 4 levels "'Above the national average'",..: 4 1 1 1 1 3 1 1 1 1 ...
##  $ h_pex      : Factor w/ 4 levels "'Above the national average'",..: 1 4 1 1 1 1 1 4 1 1 ...
##  $ h_eoc      : Factor w/ 4 levels "'Above the national average'",..: 4 4 4 4 3 4 4 4 4 4 ...
##  $ h_toc      : Factor w/ 4 levels "'Above the national average'",..: 1 1 2 4 3 3 4 4 4 4 ...
##  $ h_imaging  : Factor w/ 4 levels "'Above the national average'",..: 3 4 4 4 3 3 1 1 4 3 ...
```

```
# Create ordered levels
hoq$h_mortality <- ordered(hoq$h_mortality, levels = c("'Not Available'",
    "'Below the national average'", "'Same as the national average'",
    "'Above the national average'"))
hoq$h_soc <- ordered(hoq$h_soc, levels = c("'Not Available'",
    "'Below the national average'", "'Same as the national average'",
    "'Above the national average'"))
hoq$h_ra <- ordered(hoq$h_ra, levels = c("'Not Available'", "'Below the national average'",
    "'Same as the national average'", "'Above the national average'"))
hoq$h_pex <- ordered(hoq$h_pex, levels = c("'Not Available'",
    "'Below the national average'", "'Same as the national average'",
    "'Above the national average'"))
hoq$h_eoc <- ordered(hoq$h_eoc, levels = c("'Not Available'",
    "'Below the national average'", "'Same as the national average'",
    "'Above the national average'"))
hoq$h_toc <- ordered(hoq$h_toc, levels = c("'Not Available'",
    "'Below the national average'", "'Same as the national average'",
    "'Above the national average'"))
hoq$h_imaging <- ordered(hoq$h_imaging, levels = c("'Not Available'",
    "'Below the national average'", "'Same as the national average'",
    "'Above the national average'"))
```

The hospital overall rating was categorized based on the literature review where ratings of 4 or above are considered to have high quality (Low Hospital Quality was abbreviated to LHQ and High Hospital Quality was abbreviated to HHQ).

```
hoq$h_rating <- cut(hoq$h_rating, breaks = c(0, 3, 5), labels = c("LHQ",
    "HHQ"))
write.csv(hoq, file = "nomRating.csv")
str(hoq)
```

```
## 'data.frame':    3399 obs. of  12 variables:
##  $ h_type     : Factor w/ 2 levels "'Acute Care Hospitals'",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ h_ownership: Factor w/ 10 levels "'Government - Federal'",..: 7 5 2 7 7 8 7 7 6 6 ...
##  $ h_es       : Factor w/ 2 levels "No","Yes": 2 2 2 2 1 2 2 2 2 2 ...
##  $ h_ehr      : Factor w/ 2 levels "?","Y": 2 2 2 2 2 2 2 2 2 2 ...
##  $ h_rating   : Factor w/ 2 levels "LHQ","HHQ": 2 2 2 2 2 2 2 2 2 2 ...
##  $ h_mortality: Ord.factor w/ 4 levels "'Not Available'"<..: 3 4 3 4 4 1 3 4 3 1 ...
##  $ h_soc      : Ord.factor w/ 4 levels "'Not Available'"<..: 1 1 4 4 4 4 3 4 4 4 ...
##  $ h_ra       : Ord.factor w/ 4 levels "'Not Available'"<..: 3 4 4 4 4 1 4 4 4 4 ...
##  $ h_pex      : Ord.factor w/ 4 levels "'Not Available'"<..: 4 3 4 4 4 4 4 3 4 4 ...
##  $ h_eoc      : Ord.factor w/ 4 levels "'Not Available'"<..: 3 3 3 3 1 3 3 3 3 3 ...
##  $ h_toc      : Ord.factor w/ 4 levels "'Not Available'"<..: 4 4 2 3 1 1 3 3 3 3 ...
##  $ h_imaging  : Ord.factor w/ 4 levels "'Not Available'"<..: 1 3 3 3 1 1 4 4 3 1 ...
```

Afterwards, *nomRating.csv* was balanced and renamed to *nomRating-balanced.csv*. False instances were down sampled to match the true instances (i.e., from 2435 to 964).

Methodology

Classification methods will be used to predict the overall hospital rating and identify the characteristics of hospitals. For predictive modelling, three classification algorithms will be explored: (1) decision tree, (2) Naïve Bayes, and (3) logistic regression.

Decision Tree

The first classification algorithm applied to the dataset was the decision tree model. The technique was realized through the use of Weka and the built-in J48 Decision Tree classifier. With the prepared dataset, initial assessment of the model displayed high accuracy rate. However, the primary metric of success of recall were significantly lower compared to the overall accuracy rate. Since 85.5% of the class attribute were FALSE, the model was biased toward the FALSE instances and resulted in high overall accuracy at the cost of low recall on TRUE instances. For the next iteration, F score will also be examined.

Logistic Regression

```
# install.packages('plyr') install.packages('corrplot')
# install.packages('gridExtra') install.packages('ggthemes')
# install.packages('caret') install.packages('MASS')

library(gridExtra)
library(ggthemes)
library(caret)
```

```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

```
library(MASS)
library(plyr)
library(corrplot)
```

```
## corrplot 0.84 loaded
```

```
library(ggplot2)
```

Data Preparation

Dealing with imbalanced class distribution Answers online: For logistic regression in particular, there was absolutely no benefit to creating a balanced sample. What was far more important was using all the data you had available.

For logistic regression models unbalanced training data affects only the estimate of the model intercept (although this of course skews all the predicted probabilities, which in turn compromises your predictions). Fortunately the intercept correction is straightforward: Provided you know, or can guess, the true proportion of 0s and 1s and know the proportions in the training set you can apply a rare events correction to the intercept.

How to remove outliers from dataset Unsolved

Naive Bayes

```
## install.packages('e1071')
library("e1071")
# hoq_df <- read.csv('~/repos/ckme136-capstone/dataset/')
# names(hoq_df)
```