# Homework 3

## DBA4711

**Han Shangru, Stanley (A0179442X)**

### Part a.

1. The proportion of unplanned readmissions in the data is 11.16%. This can bias the model towards the negative outcome, resulting in it predicting more false negatives and lowering its recall score. Hence, the majority class in the training data would be under-sampled before model fitting.

2. Out of the numerical variables, `numberInpatient` has the highest correlation with `readmission` at $0.1648$. This is corroborated by Figure 1 below, showing a generally increasing trend until about 10 inpatient visits, after which the variation blooms.

   Overall, we can expect `numberInpatient` to be significant in our modelling.

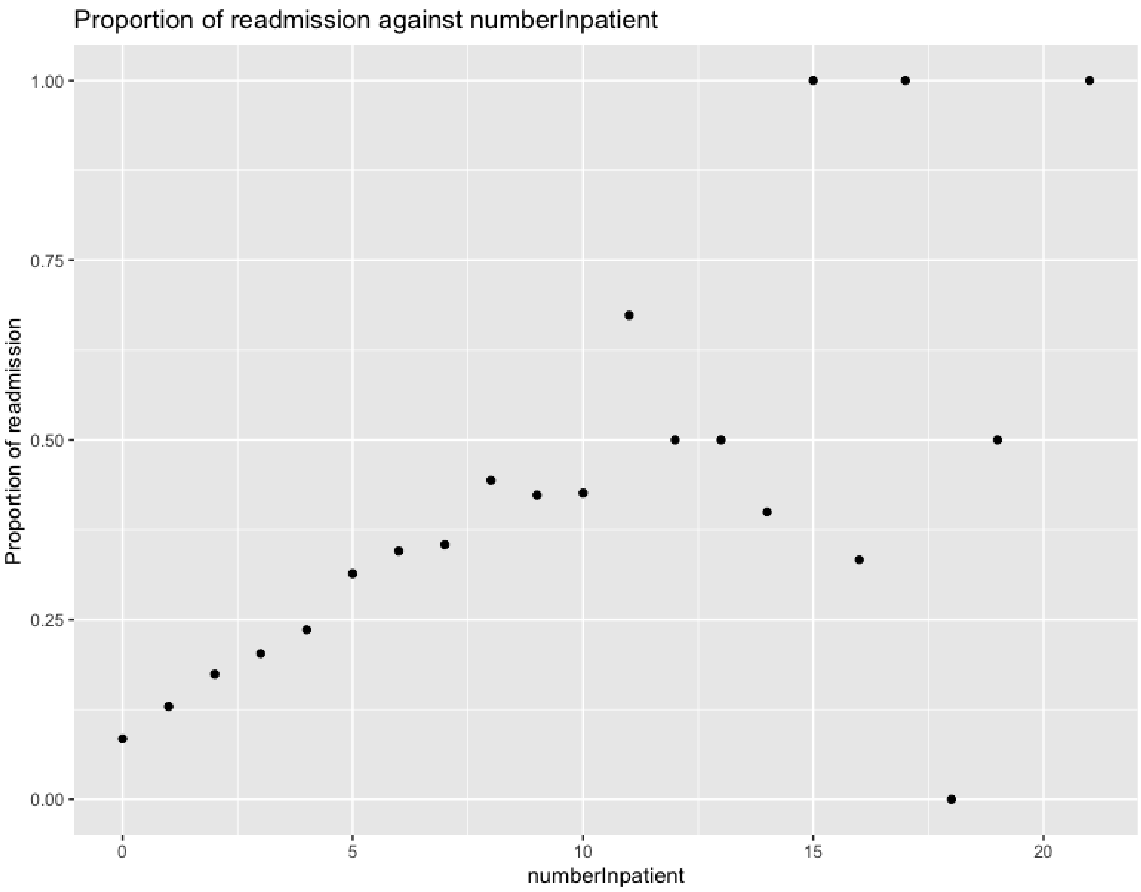

Figure 1. Plot of proportion of readmissions against number of inpatients visits.

## Part b.

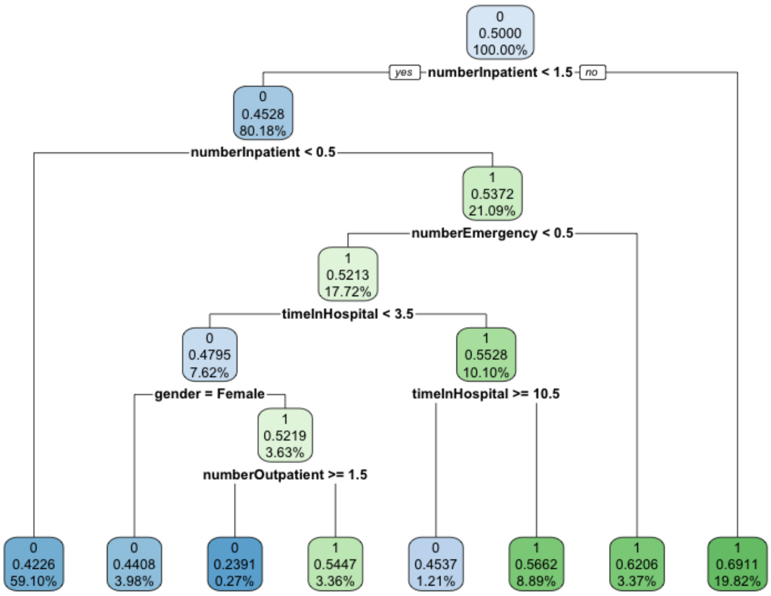

Figure 2a. Classification tree model as `model.class`.

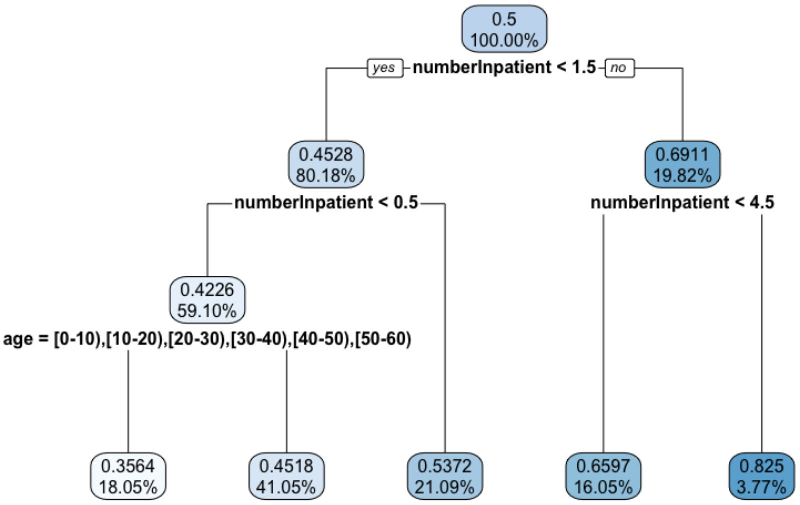

Figure 2b. Regression tree model as `model.anova`.

i. Both `class` and `anova` methods were experimented with to fit CART model named `model.class` and `model.anova` respectively, as shown in Figures 2a and 2b.

For `model.class`, the variables `numberInpatient`, `numberOutpatient`, `numberEmergency`, `timeInHospital`, and `gender` were used to make selections. Generally, patients with more inpatient or outpatient visits, a greater number of emergencies, or a moderate amount of time (about a week) spent in hospital are more likely to be selected for intervention. This is roughly in line with intuition as we can expect a positive correlation between the number of previous hospital visits and a patient's likelihood of being admitted again, unplanned or otherwise.

For `model.anova`, the variables `numberInpatient` and `age` were used to make selections. Generally, patients with more inpatient visits or over 60 years of age are more likely to be selected for intervention. The intuition for inpatient visits is similar to that in the classification model, while older patients also tend to have deteriorating health that can lead to a higher chance of readmission.

Using a threshold of 0.5 to separate positive and negative predictions, we observe the model properties as follow:

| | `model.class` | `model.anova` |
|---|---|---|
| $OSR^2$ | 0.0466 | 0.0585 |
| Accuracy | 0.70 | 0.66 |
| Sensitivity | 0.44 | 0.49 |
| Specificity | 0.73 | 0.68 |

`model.anova` has a higher out-of-sample $R^2$ value compared to `model.class`, suggesting the former is better able to generalise on new data. However, given that interventions are costly, the specificity of the model is more important. Hence, we would use `model.class` in the next few sections due to its higher accuracy and specificity.

ii. Confusion matrix of `model.class`:

```
          Reference
Prediction     0     1
         0 16571  1605
         1  6001  1264
```

With reference to the confusion matrix above, 7265 patients would be selected for intervention out of the 25441 observations in the test set.

The associated cost would be $\$1000 \times 7265 = \$7265000$.

iii. With reference to the same confusion matrix as Part ii., there are 1264 true positive cases. Therefore, the intervention is estimated to prevent $1264 \times 25\% = 316$ readmissions.

## Appendix

The R code used for this report is as follows:

```
library(ggplot2)
library(caret)
library(rpart)
library(rpart.plot)
```

```
osr2 <- function(pred, test, train) {
  SSE <- sum((test - pred)^2)
  SST <- sum((test - mean(train))^2)
  1 - SSE/SST
}

data <- read.csv("readmission.csv")

# Part a.

sum(is.na(data))
# -> there are 19777 NA values
names(which(colSums(is.na(data)) > 0))
# -> the NA values come from columns:
#    "row", "gender", "admissionType", "admissionSource"

summary(data)

n = nrow(data)
# proportion of unplanned re-admissions
length(which(data$readmission == 1)) / n * 100  # 11.16%

cor(data[, c("readmission",
             "numberOutpatient",
             "numberEmergency",
             "numberInpatient")])
cor(data[, c(
  "readmission",
  "acarbose",
  "chlorpropamide",
  "glimepiride",
  "glipizide",
  "glyburide",
  "glyburide.metformin",
  "insulin",
  "metformin",
  "nateglinide",
  "pioglitazone",
  "repaglinide",
  "rosiglitazone"
)])
cor(data[, c(
  "readmission",
  "timeInHospital",
  "numLabProcedures",
  "numNonLabProcedures",
  "numMedications",
  "numberDiagnoses"
)])
cor(data[, c(
  "readmission",
  "diagAcuteKidneyFailure",
  "diagAnemia",
  "diagAsthma",
  "diagAthlerosclerosis",
  "diagBronchitis",
  "diagCardiacDysrhythmia",
  "diagCardiomyopathy",
  "diagCellulitis",
  "diagCKD",
  "diagCOPD",
  "diagDyspnea",
  "diagHeartFailure",
  "diagHypertension",
  "diagHypertensiveCKD",
  "diagIschemicHeartDisease",
  "diagMyocardialInfarction",
  "diagOsteoarthritis",
  "diagPneumonia",
  "diagSkinUlcer"
)])

# plot the proportion of readmissions against numberInpatient
ggplot(
  aggregate(readmission ~ numberInpatient, data = data, FUN = mean),
  aes(x = numberInpatient, y = readmission)
) +
  geom_point() +
  labs(x = "numberInpatient", y = "Proportion of readmission") +
  ggtitle("Proportion of readmission against numberInpatient")

# Part b.

# split data
set.seed(998)
splitter <- createDataPartition(
  data$readmission, p = 0.75, list = F
)
data.train <- data[splitter,]
data.test <- data[-splitter,]

# undersample imbalanced training data
class_0 <- data.train[data.train$readmission == 0,]
class_1 <- data.train[data.train$readmission == 1,]
min_size <- min(nrow(class_0), nrow(class_1))
class_0.sampled <- class_0[sample(nrow(class_0), min_size),]
class_1.sampled <- class_1[sample(nrow(class_1), min_size),]
data.train <- rbind(class_0.sampled, class_1.sampled)
table(data.train$readmission)

# train CART models
model.class <- rpart(
  readmission ~ .,
  data = data.train,
  method = "class",
  control = rpart.control(cp = 0.002)
)
model.anova <- rpart(
  readmission ~ .,
  data = data.train,
  method = "anova",
  control = rpart.control(cp = 0.002)
)

# visualize models
rpart.plot(model.class, roundint = F, digits = 4)
rpart.plot(model.anova, roundint = F, digits = 4)

# predict on test data
pred.class <- predict(model.class, newdata = data.test)[, 2]
pred.anova <- predict(model.anova, newdata = data.test)

# out-of-sample R-squared values
osr2(pred.class, data.test$readmission, data.train$readmission)
osr2(pred.anova, data.test$readmission, data.train$readmission)

# threshold at 0.5
pred.class.thold <- as.integer(pred.class > 0.5)
pred.anova.thold <- as.integer(pred.anova > 0.5)
```

```
# confusion matrices
confusionMatrix(as.factor(pred.class.thold),
                as.factor(data.test$readmission),
                positive = "1")
confusionMatrix(as.factor(pred.anova.thold),
                as.factor(data.test$readmission),
                positive = "1")
```