ISYE 6501 - Week 5 Homework

Ujjawal Madan

11/06/2020

Contents

```
library(readr)
library(dplyr)
library(caret)
library(ggplot2)
library(knitr)
```

0.1 Question 14.1

Let's import the data and do the preprocessing.

```
set.seed(1, sample.kind = 'Rounding')

bcw <- read_csv("http://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/breast-c
                                    col_names = FALSE)

colnames(bcw) <- c('Sample_Code', 'Clump_Thickness', 'Uniformity_CellSize', 'Uniformity_CellShape', 'Margi

#Remove the first column since it's completely irrelavent to us
bcw<- bcw[,-1]

#Change class since it's a factor
bcw$Class <- as.factor(bcw$Class)

#The NAs are in the form of question marks
bcw$Bare_Nuclei[bcw$Bare_Nuclei == '?'] <- NA

#The indeces of the NA values
indeces <- which(!complete.cases(bcw))

#Change the form of the column with missing values to numeric
bcw$Bare_Nuclei <- as.numeric(bcw$Bare_Nuclei)
```

0.1.1 1. Imputing with mean and median

Let's replace the NA values with the mean of the column.

```
#Our dataset which will contain the mean value
mean_bcw <- bcw

#THis is our mean
mean_bare <- mean(bcw$Bare_Nuclei, na.rm = T)

#Change the values to the mean
```

```
mean_bcw$Bare_Nuclei[indeces] <- mean_bare

#These are the values that replaced the NAs.
mean_values <- mean_bcw[indeces,6]
```

Great! Let's do the same process but for the median values.

```
median_bcw <- bcw
median_bare <- median(bcw$Bare_Nuclei, na.rm = T)
median_bcw$Bare_Nuclei[indeces] <- median_bare
median_values <- median_bcw[indeces,6]
```

0.1.2 2. Imputing with regression

Let's impute the values that are NA with linear regression.

```
#Our regression version of the dataset
regression_bcw <- bcw

#THis is the data we will use for the regression
regression_data <- bcw[complete.cases(bcw),]

#THis is the linear model
model <- glm(Bare_Nuclei ~ ., data = regression_data)

#These are the predictions
predictions <- predict(model, bcw[!complete.cases(bcw), -6])

#Replace the NA values with the predictions and round them
regression_bcw$Bare_Nuclei[indeces] <- round(predictions)

#These are the values that replaced the NA Values.
regression_values <- regression_bcw[indeces,6]
```

0.1.3 3. Incorporating perturbation

Let's now incorporate perturbation. I used the rnorm function with the mean and standard deviation of the column.

```
perturb_bcw <- bcw
model <- glm(Bare_Nuclei ~ ., data = regression_data)
predictions <- predict(model, bcw[!complete.cases(bcw), -6])

#These are the errors we will purposefully incorporate into the values we will replace the NAs with
perturb_values <- rnorm(16, 0, 1)
predictions <- round(predictions + perturb_values)
for (i in seq(1, length(predictions))){
  if (predictions[i] <1) {predictions[i] <- 1}
  else if (predictions[i] > 10) {predictions[i] <- 10}
}
perturb_bcw$Bare_Nuclei[indeces] <- predictions
perturb_values <- perturb_bcw[indeces,6]

different_values <- cbind(mean_values, median_values, regression_values, perturb_values)
colnames(different_values) <- c('Mean', 'Median', "Regression", "Perturbation")
kable(different_values)
```

Mean	Median	Regression	Perturbation
3.544656	1	7	7
3.544656	1	3	4
3.544656	1	1	1
3.544656	1	2	3
3.544656	1	1	2
3.544656	1	1	1
3.544656	1	2	2
3.544656	1	1	2
3.544656	1	2	2
3.544656	1	6	6
3.544656	1	1	3
3.544656	1	1	1
3.544656	1	2	1
3.544656	1	1	1
3.544656	1	1	2
3.544656	1	1	1

Using this table, we can see what our NA values were replaced by in each version.

0.1.4 4. Compare the results with classificaiton models

Let's try running a KNN model on all of these different versions and see how they perform. Since there is relatively little data and since we are trying to find out which models perform better, let's use cross validation. The five versions are: mean replaced, median replaced, regression replaced, perturb based, na removed and binary variable introduced.

```
set.seed(1, sample.kind = 'Rounding')

control <- trainControl(method = "repeatedcv", number = 10, repeats = 5)

#Replaced by mean
mean_knn_fit <- train(Class ~., mean_bcw, method="knn", trControl=control, tuneGrid = data.frame(k = seq(1
mean_acc <- min(mean_knn_fit$results[,2])

#Replaced by median
median_knn_fit <- train(Class ~., median_bcw, method="knn", trControl=control, tuneGrid = data.frame(k = s
median_acc <- min(median_knn_fit$results[,2])

#Replaced by outputs of regression
regression_knn_fit <- train(Class ~., regression_bcw, method="knn", trControl=control, tuneGrid = data.fra
regression_acc <- min(regression_knn_fit$results[,2])

#Incorporating perturbation
perturb_knn_fit <- train(Class ~., perturb_bcw, method="knn", trControl=control, tuneGrid = data.frame(k =
perturb_acc <- min(perturb_knn_fit$results[,2])

#NAs removed
na_removed_bcw <- bcw[-indeces,]
removed_fit <- train(Class ~., na_removed_bcw, method="knn", trControl=control, tuneGrid = data.frame(k = 
removed_acc <- min(removed_fit$results[,2])

#Binary variable introduced
newvar_bcw <- bcw
newvar_bcw$missing <- as.factor(complete.cases(bcw))
newvar_fit <- train(Class ~., newvar_bcw, method="knn", trControl=control, tuneGrid = data.frame(k = seq(1
```

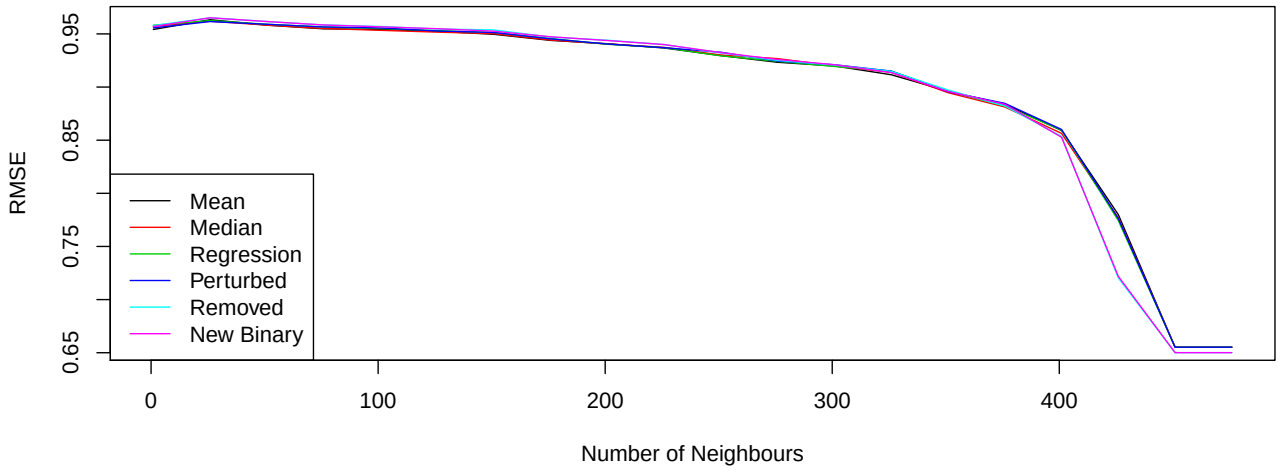
```r
newvar_acc <- min(newvar_fit$results[,2])


results <- cbind(mean_acc, median_acc, regression_acc, perturb_acc, removed_acc, newvar_acc)
colnames(results) <- c('Mean', 'Median', 'Regression', 'Perturb', 'Removed', 'New Binary')
kable(results)
```

Mean	Median	Regression	Perturb	Removed	New Binary
0.6552211	0.6552223	0.6552223	0.6552223	0.6500679	0.6500679

```r
plot(mean_knn_fit$results[,1], mean_knn_fit$results[,2],type="l",col='1', ylab = 'RMSE', xlab = 'Number of
lines(median_knn_fit$results[,1:2], col= '2')
lines(regression_knn_fit$results[,1:2], col= '3')
lines(perturb_knn_fit$results[,1:2], col= '4')
lines(removed_fit$results[,1:2], col= '5')
lines(newvar_fit$results[,1:2], col= '6')

legend('bottomleft',
       legend=c('Mean', 'Median', 'Regression', 'Perturbed', 'Removed', 'New Binary'),
       col = c(1,2,3,4,5,6),
       lty=1)
```



Interestingly, while introducing a binary variable does not ultimately reduce RMSE, RMSE does decrease faster as the number of neighbours increases. Ultimately it seems that you can't go wrong with any one of the models, although removing the values or introducing a new binary variable may work better.

0.2 Question 15.1

Describe a situation or problem from your job, everyday life, current events, etc., for which optimization would be appropriate. What data would you need?

An example could be where, if I am a condo designer, I might be trying to optimize: how many types of floor plans should I offer, how many units of each floor plan, and how many square feet for each floor plan - all to maximize overall intial sale price and ongoing revenue. For example, while a 1200 square feet condo may be more luxurious, a developer may realize that offering two 600 feet condos would be more cost effective. However, if units are made too small (like 300 square feet), demand for such units would be low and ultimately not maximize revenue. There

would be an inflection point for all three variables. Another example might be that if the entire building is made of 600 square feet units, price might decrease since there would be a surplus of 600 feet units in the market. The main constraint would be that overall square footage of the entire building would be fixed, and that floor plans have to have positive square footage.