hw10

2023-03-23

## Question 14.1

The breast cancer data set breast-cancer-wisconsin.data.txt from <http://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/> (description at <http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Original%29> ) has missing values. 1. Use the mean/mode imputation method to impute values for the missing data. 2. Use regression to impute values for the missing data. 3. Use regression with perturbation to impute values for the missing data. 4. (Optional) Compare the results and quality of classification models (e.g., SVM, KNN) build using (1) the data sets from questions 1,2,3; (2) the data that remains after data points with missing values are removed; and (3) the data set when a binary variable is introduced to indicate missing values.

data\_cancer <- read.csv('breast-cancer-wisconsin.data.txt', header = FALSE, na.strings = "?")  
#data\_cancer <- lapply(data\_cancer,as.numeric)  
  
# Adding column names to dataset - names comes from website in the homework prompt  
colnames(data\_cancer) <- c(  
'id',  
'Clump\_Thickness',  
'Uniformity\_of\_Cell\_Size',  
'Uniformity\_of\_Cell\_Shape',  
'Marginal\_Adhesion',  
'Single\_Epithelial\_Cell\_Size',  
'Bare\_Nuclei',  
'Bland\_Chromatin',  
'Normal\_Nucleoli',  
'Mitoses',  
'Class'  
)  
  
#Changing class (outcome data) from 2 and 4 to 0 and 1 with 0 being 'benign' and 4 being 'malignant'  
data\_cancer$Class <- as.factor(data\_cancer$Class)  
levels(data\_cancer$Class) = c(0,1)  
  
  
#There is missing data in this data set - lets find the rows with any missing data  
summary(data\_cancer)

## id Clump\_Thickness Uniformity\_of\_Cell\_Size  
## Min. : 61634 Min. : 1.000 Min. : 1.000   
## 1st Qu.: 870688 1st Qu.: 2.000 1st Qu.: 1.000   
## Median : 1171710 Median : 4.000 Median : 1.000   
## Mean : 1071704 Mean : 4.418 Mean : 3.134   
## 3rd Qu.: 1238298 3rd Qu.: 6.000 3rd Qu.: 5.000   
## Max. :13454352 Max. :10.000 Max. :10.000   
##   
## Uniformity\_of\_Cell\_Shape Marginal\_Adhesion Single\_Epithelial\_Cell\_Size  
## Min. : 1.000 Min. : 1.000 Min. : 1.000   
## 1st Qu.: 1.000 1st Qu.: 1.000 1st Qu.: 2.000   
## Median : 1.000 Median : 1.000 Median : 2.000   
## Mean : 3.207 Mean : 2.807 Mean : 3.216   
## 3rd Qu.: 5.000 3rd Qu.: 4.000 3rd Qu.: 4.000   
## Max. :10.000 Max. :10.000 Max. :10.000   
##   
## Bare\_Nuclei Bland\_Chromatin Normal\_Nucleoli Mitoses Class   
## Min. : 1.000 Min. : 1.000 Min. : 1.000 Min. : 1.000 0:458   
## 1st Qu.: 1.000 1st Qu.: 2.000 1st Qu.: 1.000 1st Qu.: 1.000 1:241   
## Median : 1.000 Median : 3.000 Median : 1.000 Median : 1.000   
## Mean : 3.545 Mean : 3.438 Mean : 2.867 Mean : 1.589   
## 3rd Qu.: 6.000 3rd Qu.: 5.000 3rd Qu.: 4.000 3rd Qu.: 1.000   
## Max. :10.000 Max. :10.000 Max. :10.000 Max. :10.000   
## NA's :16

cat('Percentage of NAs in dataset: ', nrow(data\_cancer[is.na(data\_cancer$Bare\_Nuclei),]) / nrow(data\_cancer) \* 100)

## Percentage of NAs in dataset: 2.288984

Looking the Summary of the dataset, we can see that the Bare\_Nuclei column has 16 NAs. The other columns do not seem to have any values missing nor do any of the ranges look off from what the data is described in the site (<http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Original%29>).

We can use mean imputation to enter the mean value for any NAs because the number of NAs to row values is under 5% (2.28%).

# We replaced NAs in Bare\_Nuclei with the remaining mean in Bare\_Nuclei <- we created a new data set for mean imputation  
data\_cancer.mean <- data\_cancer  
  
data\_cancer.mean$Bare\_Nuclei[is.na(data\_cancer.mean$Bare\_Nuclei)] <- mean(data\_cancer.mean$Bare\_Nuclei, na.rm = TRUE)   
  
# We replaced NAs in Bare\_Nuclei with the remaining mode in Bare\_Nuclei <- we created a new data set for mode imputation as well   
# Mode function is found online since R doesn't have it local? :/  
find\_mode <- function(x) {  
 u <- unique(x)  
 tab <- tabulate(match(x, u))  
 u[tab == max(tab)]  
}  
  
data\_cancer.mode <- data\_cancer  
  
data\_cancer.mode$Bare\_Nuclei[is.na(data\_cancer.mode$Bare\_Nuclei)] <- find\_mode(data\_cancer.mode$Bare\_Nuclei)

While both mean and mode imputation would work for imputing values, there would times where one method is better than another. In this situation, the mean is ~3.5 and the mode is 1 for imputed values for Bare Nuclei. The data scales from 1 to 10 with a value of 1 being no Bare Nuclei and a value of 10 being the max Bare Nuclei. Considering we are using this imputed data to help predict the type of cancer (benign or malignant), we would want to find the Bare Nuclei value that would better predict the type of cancer even as a False Positive. We would prefer to diagnose someone with malignant cancer early on and have the prediction be false than to not diagnose someone with malignant cancer and have the person actually have malignant cancer. In order to find the best method of imputation, we can use the data without missing values to see if there is a significance between low and high Nuclei compared to the type of cancer.

##Regression Imputation

set.seed(1)  
  
newdata<-data\_cancer  
# Takes rows without NAs and columns other than id and Class  
data\_removeNArows <- newdata[-which(is.na(newdata$Bare\_Nuclei), arr.ind=TRUE),2:10]  
  
model <- lm(Bare\_Nuclei~Clump\_Thickness+Uniformity\_of\_Cell\_Size+Uniformity\_of\_Cell\_Shape+Marginal\_Adhesion+Single\_Epithelial\_Cell\_Size+Bland\_Chromatin+Normal\_Nucleoli+Mitoses,data=data\_removeNArows)  
summary(model)

##   
## Call:  
## lm(formula = Bare\_Nuclei ~ Clump\_Thickness + Uniformity\_of\_Cell\_Size +   
## Uniformity\_of\_Cell\_Shape + Marginal\_Adhesion + Single\_Epithelial\_Cell\_Size +   
## Bland\_Chromatin + Normal\_Nucleoli + Mitoses, data = data\_removeNArows)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -9.7316 -0.9426 -0.3002 0.6725 8.6998   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -0.616652 0.194975 -3.163 0.00163 \*\*   
## Clump\_Thickness 0.230156 0.041691 5.521 4.83e-08 \*\*\*  
## Uniformity\_of\_Cell\_Size -0.067980 0.076170 -0.892 0.37246   
## Uniformity\_of\_Cell\_Shape 0.340442 0.073420 4.637 4.25e-06 \*\*\*  
## Marginal\_Adhesion 0.339705 0.045919 7.398 4.13e-13 \*\*\*  
## Single\_Epithelial\_Cell\_Size 0.090392 0.062541 1.445 0.14883   
## Bland\_Chromatin 0.320577 0.059047 5.429 7.91e-08 \*\*\*  
## Normal\_Nucleoli 0.007293 0.044486 0.164 0.86983   
## Mitoses -0.075230 0.059331 -1.268 0.20524   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 2.274 on 674 degrees of freedom  
## Multiple R-squared: 0.615, Adjusted R-squared: 0.6104   
## F-statistic: 134.6 on 8 and 674 DF, p-value: < 2.2e-16

#Using a linear regression model with predictors without NAs, we can fill in NAs of any predictors with NAs.   
set.seed(1)  
# Set up repeated k-fold cross-validation  
train.control <- trainControl(method = "cv", number = 10)  
# Train the model so we can predict the missing predictor values using other predictors.   
train.model <- train(Bare\_Nuclei ~., data = data\_removeNArows ,  
 method = "leapBackward",   
 tuneGrid = data.frame(nvmax = 1:8),  
 trControl = train.control  
 )  
train.model$results

## nvmax RMSE Rsquared MAE RMSESD RsquaredSD MAESD  
## 1 1 2.586251 0.5116429 1.787147 0.2803308 0.10903211 0.2020173  
## 2 2 2.406286 0.5690562 1.616582 0.2692246 0.10070987 0.1655963  
## 3 3 2.374041 0.5801646 1.595260 0.2560734 0.09567728 0.1924240  
## 4 4 2.272958 0.6134778 1.529782 0.2554610 0.09550541 0.1785070  
## 5 5 2.286116 0.6095671 1.540016 0.2436311 0.09189518 0.1688851  
## 6 6 2.282123 0.6112039 1.534321 0.2457943 0.09220504 0.1682379  
## 7 7 2.281253 0.6116476 1.535609 0.2477546 0.09302954 0.1718396  
## 8 8 2.284647 0.6107471 1.537842 0.2475400 0.09308246 0.1706002

train.model$bestTune

## nvmax  
## 4 4

data\_cancer.regression <- data\_cancer  
predictNAs <- predict(train.model, newdata = data\_cancer[which(is.na(newdata$Bare\_Nuclei), arr.ind=TRUE),])  
  
# Impute the NAs using predicted values  
data\_cancer.regression[which(is.na(newdata$Bare\_Nuclei)),]$Bare\_Nuclei <- predictNAs  
data\_cancer.regression[which(is.na(newdata$Bare\_Nuclei)),]

## id Clump\_Thickness Uniformity\_of\_Cell\_Size Uniformity\_of\_Cell\_Shape  
## 24 1057013 8 4 5  
## 41 1096800 6 6 6  
## 140 1183246 1 1 1  
## 146 1184840 1 1 3  
## 159 1193683 1 1 2  
## 165 1197510 5 1 1  
## 236 1241232 3 1 4  
## 250 169356 3 1 1  
## 276 432809 3 1 3  
## 293 563649 8 8 8  
## 295 606140 1 1 1  
## 298 61634 5 4 3  
## 316 704168 4 6 5  
## 322 733639 3 1 1  
## 412 1238464 1 1 1  
## 618 1057067 1 1 1  
## Marginal\_Adhesion Single\_Epithelial\_Cell\_Size Bare\_Nuclei Bland\_Chromatin  
## 24 1 2 5.4585352 7  
## 41 9 6 7.9816106 7  
## 140 1 1 0.9872832 2  
## 146 1 2 1.6218560 2  
## 159 1 3 0.9807851 1  
## 165 1 2 2.2157441 3  
## 236 1 2 2.7152652 3  
## 250 1 2 1.7634059 3  
## 276 1 2 2.0741942 2  
## 293 1 2 6.0866099 6  
## 295 1 2 0.9872832 2  
## 298 1 2 2.5265324 2  
## 316 6 7 5.2438347 4  
## 322 1 2 1.7634059 3  
## 412 1 1 0.9872832 2  
## 618 1 1 0.6634986 1  
## Normal\_Nucleoli Mitoses Class  
## 24 3 1 1  
## 41 8 1 0  
## 140 1 1 0  
## 146 1 1 0  
## 159 1 1 0  
## 165 1 1 0  
## 236 1 1 0  
## 250 1 1 0  
## 276 1 1 0  
## 293 10 1 1  
## 295 1 1 0  
## 298 3 1 0  
## 316 9 1 0  
## 322 1 1 0  
## 412 1 1 0  
## 618 1 1 0

##Regression with Perturbation

#using MICE to impute missing values using perturbation  
imp\_perturbation <- mice(data\_cancer, method = 'norm.nob', m=1)

##   
## iter imp variable  
## 1 1 Bare\_Nuclei  
## 2 1 Bare\_Nuclei  
## 3 1 Bare\_Nuclei  
## 4 1 Bare\_Nuclei  
## 5 1 Bare\_Nuclei

imp\_perturbation

## Class: mids  
## Number of multiple imputations: 1   
## Imputation methods:  
## id Clump\_Thickness   
## "" ""   
## Uniformity\_of\_Cell\_Size Uniformity\_of\_Cell\_Shape   
## "" ""   
## Marginal\_Adhesion Single\_Epithelial\_Cell\_Size   
## "" ""   
## Bare\_Nuclei Bland\_Chromatin   
## "norm.nob" ""   
## Normal\_Nucleoli Mitoses   
## "" ""   
## Class   
## ""   
## PredictorMatrix:  
## id Clump\_Thickness Uniformity\_of\_Cell\_Size  
## id 0 1 1  
## Clump\_Thickness 1 0 1  
## Uniformity\_of\_Cell\_Size 1 1 0  
## Uniformity\_of\_Cell\_Shape 1 1 1  
## Marginal\_Adhesion 1 1 1  
## Single\_Epithelial\_Cell\_Size 1 1 1  
## Uniformity\_of\_Cell\_Shape Marginal\_Adhesion  
## id 1 1  
## Clump\_Thickness 1 1  
## Uniformity\_of\_Cell\_Size 1 1  
## Uniformity\_of\_Cell\_Shape 0 1  
## Marginal\_Adhesion 1 0  
## Single\_Epithelial\_Cell\_Size 1 1  
## Single\_Epithelial\_Cell\_Size Bare\_Nuclei  
## id 1 1  
## Clump\_Thickness 1 1  
## Uniformity\_of\_Cell\_Size 1 1  
## Uniformity\_of\_Cell\_Shape 1 1  
## Marginal\_Adhesion 1 1  
## Single\_Epithelial\_Cell\_Size 0 1  
## Bland\_Chromatin Normal\_Nucleoli Mitoses Class  
## id 1 1 1 1  
## Clump\_Thickness 1 1 1 1  
## Uniformity\_of\_Cell\_Size 1 1 1 1  
## Uniformity\_of\_Cell\_Shape 1 1 1 1  
## Marginal\_Adhesion 1 1 1 1  
## Single\_Epithelial\_Cell\_Size 1 1 1 1

#imputing missing values using perturbation   
data\_cancer.perturbation <- complete(imp\_perturbation)  
#We need to apply absolute value on imputed values since range is from 0 to 10  
data\_cancer.perturbation$Bare\_Nuclei <- abs(data\_cancer.perturbation$Bare\_Nuclei)  
data\_cancer.perturbation[which(is.na(newdata$Bare\_Nuclei)),]

## id Clump\_Thickness Uniformity\_of\_Cell\_Size Uniformity\_of\_Cell\_Shape  
## 24 1057013 8 4 5  
## 41 1096800 6 6 6  
## 140 1183246 1 1 1  
## 146 1184840 1 1 3  
## 159 1193683 1 1 2  
## 165 1197510 5 1 1  
## 236 1241232 3 1 4  
## 250 169356 3 1 1  
## 276 432809 3 1 3  
## 293 563649 8 8 8  
## 295 606140 1 1 1  
## 298 61634 5 4 3  
## 316 704168 4 6 5  
## 322 733639 3 1 1  
## 412 1238464 1 1 1  
## 618 1057067 1 1 1  
## Marginal\_Adhesion Single\_Epithelial\_Cell\_Size Bare\_Nuclei Bland\_Chromatin  
## 24 1 2 8.0453734 7  
## 41 9 6 7.2909289 7  
## 140 1 1 1.0648244 2  
## 146 1 2 1.2055555 2  
## 159 1 3 3.0016967 1  
## 165 1 2 0.3845370 3  
## 236 1 2 4.4008439 3  
## 250 1 2 1.7263537 3  
## 276 1 2 0.7294873 2  
## 293 1 2 4.3360134 6  
## 295 1 2 3.8740989 2  
## 298 1 2 2.4250905 2  
## 316 6 7 1.4098944 4  
## 322 1 2 0.2504595 3  
## 412 1 1 3.1125003 2  
## 618 1 1 4.6100580 1  
## Normal\_Nucleoli Mitoses Class  
## 24 3 1 1  
## 41 8 1 0  
## 140 1 1 0  
## 146 1 1 0  
## 159 1 1 0  
## 165 1 1 0  
## 236 1 1 0  
## 250 1 1 0  
## 276 1 1 0  
## 293 10 1 1  
## 295 1 1 0  
## 298 3 1 0  
## 316 9 1 0  
## 322 1 1 0  
## 412 1 1 0  
## 618 1 1 0

Comparing the regression imputation data to the perturbation imputation data for Bare Nuclei, both have imputations ranging from 0 to 10 (for perturbation, some of the values appear to be negative and I took the absolute value in order to fit the imputation in the range). These imputations are different from the mean and mode imputations as they have variety, and are depended on the other predictors for the values. For this data set, I would choose to use the linear regression imputation over the other methods used as the variety of imputation values and reasonableness of imputing values (no values outside of 0 to 10) make the linear regression imputation most fitting with the rest of the Bare Nuclei data points. The regression and perturbation imputed values came out with decimals, while the other values had integer values; while the data description mentioned the range was between 0 to 10, there wasn’t clarity on if values could have decimals. These imputed values stick out like sore thumbs in the dataset, but should work for prediction purposes.

##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?

During this semester, I started to go back to the gym, and I’ve been taking my health more seriously than the last couple of years of my life. While going to the gym is great, the real improvement of one’s body comes from having ample rest, and eating properly. My goal is to gain muscle, and lose fat at the same time. While this is fairly difficult, usually beginners (me) can follow a process called ‘body re-composition’. In order to do so, one has to be an calorie deficit and maintain at least 1g protein per lb of body weight. This is fairly difficult as this limits many dietary choices, however they are still many choices to choice from. One way I can use optimization is to find the cheapest methods of weekly meals while achieving my diet constraints. Other than the number of calories, and amount of protein I need to eat a day, constraints such as no processed food, diversified meal choices, and a minimum amount fats and carbs would be key factors in optimizing cost. I would need data on food statistics (amount of protein, fats, carbs, $ etc), which can be found online or through calorie tracking apps.