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.

##Finding Missing Data and using Mean/Mode Imputation

```
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_Sha
pe+Marginal_Adhesion+Single_Epithelial_Cell_Size+Bland_Chromatin+Normal_Nucle
oli+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)),]
```

```
##     Marginal_Adhesion Single_Epithelial_Cell_Size Bare_Nuclei
## 24                  1                           2   5.4585352
## 41                  9                           6   7.9816106
## 140                 1                           1   0.9872832
## 146                 1                           2   1.6218560
## 159                 1                           3   0.9807851
## 165                 1                           2   2.2157441
## 236                 1                           2   2.7152652
## 250                 1                           2   1.7634059
## 276                 1                           2   2.0741942
## 293                 1                           2   6.0866099
## 295                 1                           2   0.9872832
## 298                 1                           2   2.5265324
## 316                 6                           7   5.2438347
## 322                 1                           2   1.7634059
## 412                 1                           1   0.9872832
## 618                 1                           1   0.6634986
```

##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)),]
```

```
##     Marginal_Adhesion Single_Epithelial_Cell_Size Bare_Nuclei
Bland_Chromatin
## 24                  1                           2   8.0453734
## 41                  9                           6   7.2909289
```

```
## 140                1                    1   1.0648244
## 146                1                    2   1.2055555
## 159                1                    3   3.0016967
## 165                1                    2   0.3845370
## 236                1                    2   4.4008439
## 250                1                    2   1.7263537
## 276                1                    2   0.7294873
## 293                1                    2   4.3360134
## 295                1                    2   3.8740989
## 298                1                    2   2.4250905
## 316                6                    7   1.4098944
## 322                1                    2   0.2504595
## 412                1                    1   3.1125003
## 618                1                    1   4.6100580
```

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.