ISYE 6501 Homework 10

2024-10-30

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.

Load Data Attribute Information[1]:

1. Sample code number: id number
2. Clump Thickness: 1 - 10
3. Uniformity of Cell Size: 1 - 10
4. Uniformity of Cell Shape: 1 - 10
5. Marginal Adhesion: 1 - 10
6. Single Epithelial Cell Size: 1 - 10
7. Bare Nuclei: 1 - 10
8. Bland Chromatin: 1 - 10
9. Normal Nucleoli: 1 - 10
10. Mitoses: 1 - 10
11. Class: (2 for benign, 4 for malignant)

Reference 1:
http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Original%29

```
# Read a txt file
df<-read.table("breast-cancer-wisconsin.data.txt", stringsAsFactor = FALSE,
header = F, sep = ",", na.strings="?")
head(df,2)
```

```
##        V1 V2 V3 V4 V5 V6 V7 V8 V9 V10 V11
## 1 1000025  5  1  1  1  2  1  3  1   1   2
## 2 1002945  5  4  4  5  7 10  3  2   1   2
```

Checking for missing data

```
#assigning colnames to data
colnames(df) <- c("ID", "Clump_Thickness", "Uniform_Cell_Size",
"Uniform_Cell_Shape",
                  "Marg_Adhesion", "Single_Epith_Cell_Size", "Bare_Nuclei",
"Bland_Chromatin",
                  "Normal_Nucleoli", "Mitoses", "Class")
df$Class <- as.factor(df$Class)
levels(df$Class) <- c(0, 1)
summary(df)
```

```
##        ID           Clump_Thickness  Uniform_Cell_Size Uniform_Cell_Shape
##  Min.   :   61634   Min.   : 1.000   Min.   : 1.000    Min.   : 1.000
##  1st Qu.:  870688   1st Qu.: 2.000   1st Qu.: 1.000    1st Qu.: 1.000
##  Median : 1171710   Median : 4.000   Median : 1.000    Median : 1.000
##  Mean   : 1071704   Mean   : 4.418   Mean   : 3.134    Mean   : 3.207
##  3rd Qu.: 1238298   3rd Qu.: 6.000   3rd Qu.: 5.000    3rd Qu.: 5.000
##  Max.   :13454352   Max.   :10.000   Max.   :10.000    Max.   :10.000
##
##  Marg_Adhesion    Single_Epith_Cell_Size  Bare_Nuclei     Bland_Chromatin
##  Min.   : 1.000   Min.   : 1.000         Min.   : 1.000   Min.   : 1.000
##  1st Qu.: 1.000   1st Qu.: 2.000         1st Qu.: 1.000   1st Qu.: 2.000
##  Median : 1.000   Median : 2.000         Median : 1.000   Median : 3.000
##  Mean   : 2.807   Mean   : 3.216         Mean   : 3.545   Mean   : 3.438
##  3rd Qu.: 4.000   3rd Qu.: 4.000         3rd Qu.: 6.000   3rd Qu.: 5.000
##  Max.   :10.000   Max.   :10.000         Max.   :10.000   Max.   :10.000
##                                          NA's   :16
##  Normal_Nucleoli     Mitoses       Class
##  Min.   : 1.000   Min.   : 1.000   0:458
##  1st Qu.: 1.000   1st Qu.: 1.000   1:241
##  Median : 1.000   Median : 1.000
##  Mean   : 2.867   Mean   : 1.589
##  3rd Qu.: 4.000   3rd Qu.: 1.000
##  Max.   :10.000   Max.   :10.000
##
```

```
#which column contains the missing data
df[is.na(df$Bare_Nuclei),]
```

```
##          ID Clump_Thickness Uniform_Cell_Size Uniform_Cell_Shape
Marg_Adhesion
## 24  1057013               8                 4                  5
1
## 41  1096800               6                 6                  6
9
## 140 1183246               1                 1                  1
1
## 146 1184840               1                 1                  3
1
## 159 1193683               1                 1                  2
```

```
1
## 165 1197510                5                1                1
1
## 236 1241232                3                1                4
1
## 250  169356                3                1                1
1
## 276  432809                3                1                3
1
## 293  563649                8                8                8
1
## 295  606140                1                1                1
1
## 298   61634                5                4                3
1
## 316  704168                4                6                5
6
## 322  733639                3                1                1
1
## 412 1238464                1                1                1
1
## 618 1057067                1                1                1
1
##     Single_Epith_Cell_Size Bare_Nuclei Bland_Chromatin Normal_Nucleoli
Mitoses
## 24                       2          NA               7               3
1
## 41                       6          NA               7               8
1
## 140                      1          NA               2               1
1
## 146                      2          NA               2               1
1
## 159                      3          NA               1               1
1
## 165                      2          NA               3               1
1
## 236                      2          NA               3               1
1
## 250                      2          NA               3               1
1
## 276                      2          NA               2               1
1
## 293                      2          NA               6              10
1
## 295                      2          NA               2               1
1
## 298                      2          NA               2               3
1
## 316                      7          NA               4               9
```

```
1
## 322                     2          NA                3                1
1
## 412                     1          NA                2                1
1
## 618                     1          NA                1                1
1
##     Class
## 24      1
## 41      0
## 140     0
## 146     0
## 159     0
## 165     0
## 236     0
## 250     0
## 276     0
## 293     1
## 295     0
## 298     0
## 316     0
## 322     0
## 412     0
## 618     0
```

```
#check for % of missing observation (threshold < 5%)
print(sprintf("Percent of missing observation = %0.3f", 16/nrow(df)*100))
```

```
## [1] "Percent of missing observation = 2.289"
```

There were 16 NAs found in the "Bare Nuclei" column, which is below the 5% threshold, making imputation an acceptable approach.

Mean imputation

```
#mean imputation
df.mean<-df
df.mean<-df.mean %>% mutate_at(vars(Bare_Nuclei),~ifelse(is.na(.x), mean(.x,
na.rm = TRUE), .x))
#check it was imputed correctly
head(df.mean,24)
```

```
##         ID Clump_Thickness Uniform_Cell_Size Uniform_Cell_Shape
Marg_Adhesion
## 1  1000025               5                 1                  1
1
## 2  1002945               5                 4                  4
5
## 3  1015425               3                 1                  1
1
## 4  1016277               6                 8                  8
```

```
1
## 5  1017023                4                  1                   1
3
## 6  1017122                8                 10                  10
8
## 7  1018099                1                  1                   1
1
## 8  1018561                2                  1                   2
1
## 9  1033078                2                  1                   1
1
## 10 1033078                4                  2                   1
1
## 11 1035283                1                  1                   1
1
## 12 1036172                2                  1                   1
1
## 13 1041801                5                  3                   3
3
## 14 1043999                1                  1                   1
1
## 15 1044572                8                  7                   5
10
## 16 1047630                7                  4                   6
4
## 17 1048672                4                  1                   1
1
## 18 1049815                4                  1                   1
1
## 19 1050670               10                  7                   7
6
## 20 1050718                6                  1                   1
1
## 21 1054590                7                  3                   2
10
## 22 1054593               10                  5                   5
3
## 23 1056784                3                  1                   1
1
## 24 1057013                8                  4                   5
1
##    Single_Epith_Cell_Size Bare_Nuclei Bland_Chromatin Normal_Nucleoli
Mitoses
## 1                       2    1.000000               3                1
1
## 2                       7   10.000000               3                2
1
## 3                       2    2.000000               3                1
1
## 4                       3    4.000000               3                7
```

```
1
## 5                        2    1.000000                3                1
1
## 6                        7   10.000000                9                7
1
## 7                        2   10.000000                3                1
1
## 8                        2    1.000000                3                1
1
## 9                        2    1.000000                1                1
5
## 10                       2    1.000000                2                1
1
## 11                       1    1.000000                3                1
1
## 12                       2    1.000000                2                1
1
## 13                       2    3.000000                4                4
1
## 14                       2    3.000000                3                1
1
## 15                       7    9.000000                5                5
4
## 16                       6    1.000000                4                3
1
## 17                       2    1.000000                2                1
1
## 18                       2    1.000000                3                1
1
## 19                       4   10.000000                4                1
2
## 20                       2    1.000000                3                1
1
## 21                       5   10.000000                5                4
4
## 22                       6    7.000000                7               10
1
## 23                       2    1.000000                2                1
1
## 24                       2    3.544656                7                3
1
##    Class
## 1      0
## 2      0
## 3      0
## 4      0
## 5      0
## 6      1
## 7      0
## 8      0
```

```
## 9      0
## 10     0
## 11     0
## 12     0
## 13     1
## 14     0
## 15     1
## 16     1
## 17     0
## 18     0
## 19     1
## 20     0
## 21     1
## 22     1
## 23     0
## 24     1
```

```
#double check if mean was calculated correctly
mean(df.mean$Bare_Nuclei)
```

```
## [1] 3.544656
```

Mode imputation

```
#found this mode function in the internet
getmode <- function(v) {
   uniqv <- unique(v)
   uniqv[which.max(tabulate(match(v, uniqv)))]
}
df.mode<-df
mode.result <- getmode(df.mode$Bare_Nuclei)
print(mode.result)
```

```
## [1] 1
```

```
#fill NA with Mode of 1s
df.mode$Bare_Nuclei[is.na(df.mode$Bare_Nuclei)] <- mode.result
#check it was imputed correctly
head(df.mode,24)
```

```
##         ID Clump_Thickness Uniform_Cell_Size Uniform_Cell_Shape
Marg_Adhesion
## 1  1000025               5                 1                  1
1
## 2  1002945               5                 4                  4
5
## 3  1015425               3                 1                  1
1
## 4  1016277               6                 8                  8
1
## 5  1017023               4                 1                  1
```

```
3
## 6  1017122                8                10                10
8
## 7  1018099                1                 1                 1
1
## 8  1018561                2                 1                 2
1
## 9  1033078                2                 1                 1
1
## 10 1033078                4                 2                 1
1
## 11 1035283                1                 1                 1
1
## 12 1036172                2                 1                 1
1
## 13 1041801                5                 3                 3
3
## 14 1043999                1                 1                 1
1
## 15 1044572                8                 7                 5
10
## 16 1047630                7                 4                 6
4
## 17 1048672                4                 1                 1
1
## 18 1049815                4                 1                 1
1
## 19 1050670               10                 7                 7
6
## 20 1050718                6                 1                 1
1
## 21 1054590                7                 3                 2
10
## 22 1054593               10                 5                 5
3
## 23 1056784                3                 1                 1
1
## 24 1057013                8                 4                 5
1
##    Single_Epith_Cell_Size Bare_Nuclei Bland_Chromatin Normal_Nucleoli
Mitoses
## 1                       2           1               3               1
1
## 2                       7          10               3               2
1
## 3                       2           2               3               1
1
## 4                       3           4               3               7
1
## 5                       2           1               3               1
```

```
1
## 6                      7          10                9                7
1
## 7                      2          10                3                1
1
## 8                      2           1                3                1
1
## 9                      2           1                1                1
5
## 10                     2           1                2                1
1
## 11                     1           1                3                1
1
## 12                     2           1                2                1
1
## 13                     2           3                4                4
1
## 14                     2           3                3                1
1
## 15                     7           9                5                5
4
## 16                     6           1                4                3
1
## 17                     2           1                2                1
1
## 18                     2           1                3                1
1
## 19                     4          10                4                1
2
## 20                     2           1                3                1
1
## 21                     5          10                5                4
4
## 22                     6           7                7               10
1
## 23                     2           1                2                1
1
## 24                     2           1                7                3
1
##    Class
## 1      0
## 2      0
## 3      0
## 4      0
## 5      0
## 6      1
## 7      0
## 8      0
## 9      0
## 10     0
```

```
## 11     0
## 12     0
## 13     1
## 14     0
## 15     1
## 16     1
## 17     0
## 18     0
## 19     1
## 20     0
## 21     1
## 22     1
## 23     0
## 24     1
```

Regression Imputation

```
set.seed(123)

newdata<-df
missing.index<-which(is.na(newdata$Bare_Nuclei), arr.ind=TRUE)
newdata.1 <- newdata[-missing.index,2:10]# all other predictors data points
except for the missing value and response variable

#Linear Model
model <-
lm(Bare_Nuclei~Clump_Thickness+Uniform_Cell_Size+Uniform_Cell_Shape+Marg_Adhe
sion+Single_Epith_Cell_Size+Bland_Chromatin+Normal_Nucleoli+Mitoses,data=newd
ata.1 )
summary(model)

##
## Call:
## lm(formula = Bare_Nuclei ~ Clump_Thickness + Uniform_Cell_Size +
##     Uniform_Cell_Shape + Marg_Adhesion + Single_Epith_Cell_Size +
##     Bland_Chromatin + Normal_Nucleoli + Mitoses, data = newdata.1)
##
## 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 ***
## Uniform_Cell_Size      -0.067980   0.076170  -0.892  0.37246
## Uniform_Cell_Shape      0.340442   0.073420   4.637 4.25e-06 ***
## Marg_Adhesion           0.339705   0.045919   7.398 4.13e-13 ***
## Single_Epith_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
```

Variable Selection using stepwise

```
# Fit the full model
full.model <- model
# Stepwise regression model
step.model <- stepAIC(full.model, direction = "both",
                      trace = FALSE)
summary(step.model)

##
## Call:
## lm(formula = Bare_Nuclei ~ Clump_Thickness + Uniform_Cell_Shape +
##     Marg_Adhesion + Bland_Chromatin, data = newdata.1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.8115 -0.9531 -0.3111  0.6678  8.6889
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)        -0.53601    0.17514  -3.060   0.0023 **
## Clump_Thickness     0.22617    0.04121   5.488 5.75e-08 ***
## Uniform_Cell_Shape  0.31729    0.05086   6.239 7.76e-10 ***
## Marg_Adhesion       0.33227    0.04431   7.499 2.03e-13 ***
## Bland_Chromatin     0.32378    0.05606   5.775 1.17e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.274 on 678 degrees of freedom
## Multiple R-squared:  0.6129, Adjusted R-squared:  0.6107
## F-statistic: 268.4 on 4 and 678 DF,  p-value: < 2.2e-16
```

Since regular linear regression included several insignificant predictors, stepwise regression in both directions was applied to remove these predictors and create a more streamlined model.

```
# Set seed for reproducibility
set.seed(123)
# Set up repeated k-fold cross-validation
train.control <- trainControl(method = "cv", number = 10)
# Train the model
step.model <- train(Bare_Nuclei ~., data = newdata.1 ,
```

```
                    method = "leapBackward",
                    tuneGrid = data.frame(nvmax = 1:4),
                    trControl = train.control
                    )
step.model$results
```

```
##   nvmax     RMSE  Rsquared      MAE    RMSESD RsquaredSD     MAESD
## 1     1 2.548924 0.5228085 1.765516 0.2208983 0.08742394 0.1577356
## 2     2 2.432898 0.5624336 1.642851 0.2839852 0.10115619 0.1857652
## 3     3 2.362419 0.5900746 1.565549 0.2804584 0.09463731 0.1827845
## 4     4 2.278984 0.6185545 1.534310 0.2734543 0.08969728 0.1880005
```

Find the best model

```
step.model$bestTune
```

```
##   nvmax
## 4     4
```

Cross-validation identified a smaller model with the lowest RMSE of 2.278 and the highest R-Squared of 0.618. The optimal model included four predictors: Clump_Thickness, Uniform_Cell_Shape, Marg_Adhesion, and Bland_Chromatin.

Predicting the missing values

```
predicted.missing<-predict(step.model,newdata=df[missing.index,])
```

Impute with regressed data points

```
df.final.regression<-df

df.final.regression[missing.index,]$Bare_Nuclei<-
as.integer(predicted.missing)#make predicted values integers
#final data with imputed regressed values
head(df.final.regression,24)
```

```
##         ID Clump_Thickness Uniform_Cell_Size Uniform_Cell_Shape
Marg_Adhesion
## 1  1000025               5                 1                  1
1
## 2  1002945               5                 4                  4
5
## 3  1015425               3                 1                  1
1
## 4  1016277               6                 8                  8
1
## 5  1017023               4                 1                  1
3
## 6  1017122               8                10                 10
8
## 7  1018099               1                 1                  1
1
```

```
## 8  1018561                2                1                2
1
## 9  1033078                2                1                1
1
## 10 1033078                4                2                1
1
## 11 1035283                1                1                1
1
## 12 1036172                2                1                1
1
## 13 1041801                5                3                3
3
## 14 1043999                1                1                1
1
## 15 1044572                8                7                5
10
## 16 1047630                7                4                6
4
## 17 1048672                4                1                1
1
## 18 1049815                4                1                1
1
## 19 1050670               10                7                7
6
## 20 1050718                6                1                1
1
## 21 1054590                7                3                2
10
## 22 1054593               10                5                5
3
## 23 1056784                3                1                1
1
## 24 1057013                8                4                5
1
##    Single_Epith_Cell_Size Bare_Nuclei Bland_Chromatin Normal_Nucleoli
Mitoses
## 1                       2           1               3               1
1
## 2                       7          10               3               2
1
## 3                       2           2               3               1
1
## 4                       3           4               3               7
1
## 5                       2           1               3               1
1
## 6                       7          10               9               7
1
## 7                       2          10               3               1
1
```

```
## 8                      2           1                3                1
1
## 9                      2           1                1                1
5
## 10                     2           1                2                1
1
## 11                     1           1                3                1
1
## 12                     2           1                2                1
1
## 13                     2           3                4                4
1
## 14                     2           3                3                1
1
## 15                     7           9                5                5
4
## 16                     6           1                4                3
1
## 17                     2           1                2                1
1
## 18                     2           1                3                1
1
## 19                     4          10                4                1
2
## 20                     2           1                3                1
1
## 21                     5          10                5                4
4
## 22                     6           7                7               10
1
## 23                     2           1                2                1
1
## 24                     2           5                7                3
1
##    Class
## 1      0
## 2      0
## 3      0
## 4      0
## 5      0
## 6      1
## 7      0
## 8      0
## 9      0
## 10     0
## 11     0
## 12     0
## 13     1
## 14     0
## 15     1
```

```
## 16     1
## 17     0
## 18     0
## 19     1
## 20     0
## 21     1
## 22     1
## 23     0
## 24     1
```

Imputation with Regression + Perturbation

```
set.seed(123)
n <- rnorm(16, mean = predicted.missing, sd = sd(predicted.missing))#generate
16 random numbers based off missing predicted values
n

##  [1]  4.2231494  7.4742594  4.4229474  1.7772688  1.2657577  5.9960455
##  [7]  3.7312046 -1.0250090  0.5602511  5.1042934  3.6853723  3.3196246
## [13]  6.1272047  2.0073698 -0.2378873  4.6021659

#bounding the negative numbers to positive only
abs(n)

##  [1] 4.2231494 7.4742594 4.4229474 1.7772688 1.2657577 5.9960455 3.7312046
##  [8] 1.0250090 0.5602511 5.1042934 3.6853723 3.3196246 6.1272047 2.0073698
## [15] 0.2378873 4.6021659
```

Finally, Impute with pertubed regressed data points

```
df.final.regression.pertubed<-df
df.final.regression.pertubed[missing.index,]$Bare_Nuclei<-
as.integer(abs(n))#make predicted values integers
#final data with imputed  perturbed regressed values
head(df.final.regression.pertubed,24)

##         ID Clump_Thickness Uniform_Cell_Size Uniform_Cell_Shape
Marg_Adhesion
## 1  1000025               5                 1                  1
1
## 2  1002945               5                 4                  4
5
## 3  1015425               3                 1                  1
1
## 4  1016277               6                 8                  8
1
## 5  1017023               4                 1                  1
3
## 6  1017122               8                10                 10
8
## 7  1018099               1                 1                  1
1
```

```
## 8  1018561                2                1                2
1
## 9  1033078                2                1                1
1
## 10 1033078                4                2                1
1
## 11 1035283                1                1                1
1
## 12 1036172                2                1                1
1
## 13 1041801                5                3                3
3
## 14 1043999                1                1                1
1
## 15 1044572                8                7                5
10
## 16 1047630                7                4                6
4
## 17 1048672                4                1                1
1
## 18 1049815                4                1                1
1
## 19 1050670               10                7                7
6
## 20 1050718                6                1                1
1
## 21 1054590                7                3                2
10
## 22 1054593               10                5                5
3
## 23 1056784                3                1                1
1
## 24 1057013                8                4                5
1
##    Single_Epith_Cell_Size Bare_Nuclei Bland_Chromatin Normal_Nucleoli
Mitoses
## 1                       2           1               3               1
1
## 2                       7          10               3               2
1
## 3                       2           2               3               1
1
## 4                       3           4               3               7
1
## 5                       2           1               3               1
1
## 6                       7          10               9               7
1
## 7                       2          10               3               1
1
```

```
## 8                      2           1                3                1
1
## 9                      2           1                1                1
5
## 10                     2           1                2                1
1
## 11                     1           1                3                1
1
## 12                     2           1                2                1
1
## 13                     2           3                4                4
1
## 14                     2           3                3                1
1
## 15                     7           9                5                5
4
## 16                     6           1                4                3
1
## 17                     2           1                2                1
1
## 18                     2           1                3                1
1
## 19                     4          10                4                1
2
## 20                     2           1                3                1
1
## 21                     5          10                5                4
4
## 22                     6           7                7               10
1
## 23                     2           1                2                1
1
## 24                     2           4                7                3
1
##    Class
## 1      0
## 2      0
## 3      0
## 4      0
## 5      0
## 6      1
## 7      0
## 8      0
## 9      0
## 10     0
## 11     0
## 12     0
## 13     1
## 14     0
## 15     1
```

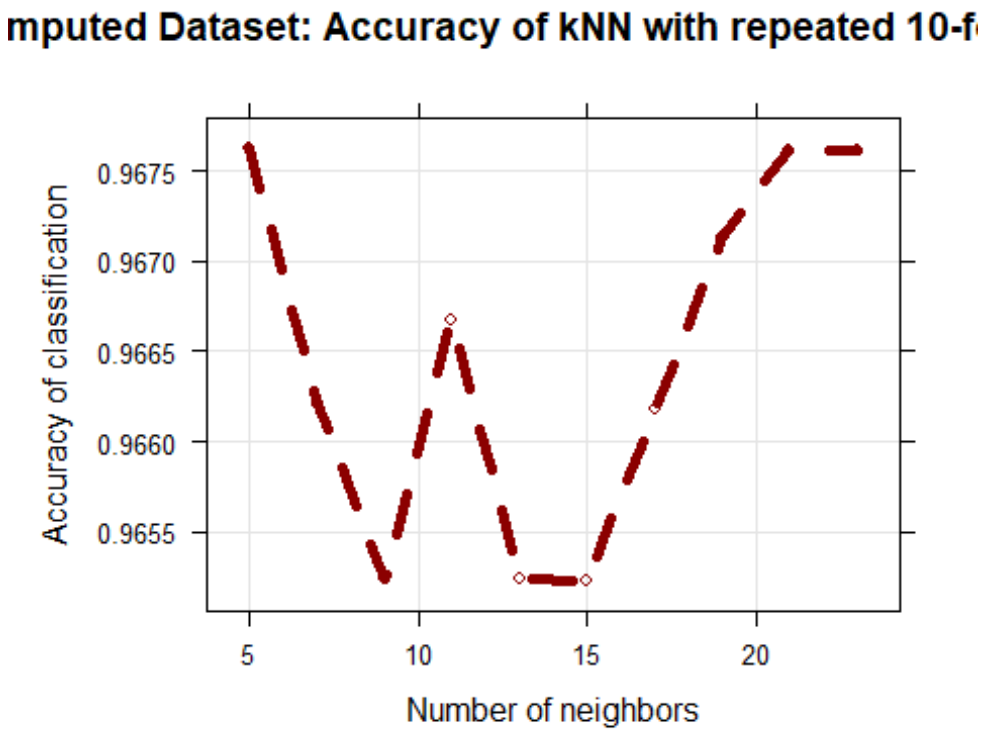
```
## 16     1
## 17     0
## 18     0
## 19     1
## 20     0
## 21     1
## 22     1
## 23     0
## 24     1
```

KNN Comparisons

```
df1<-read.table("breast-cancer-wisconsin.data.txt", stringsAsFactor = FALSE,
header = F, sep = ",", na.strings="?")
head(df1,2)

##        V1 V2 V3 V4 V5 V6 V7 V8 V9 V10 V11
## 1 1000025  5  1  1  1  2  1  3  1   1   2
## 2 1002945  5  4  4  5  7 10  3  2   1   2
```

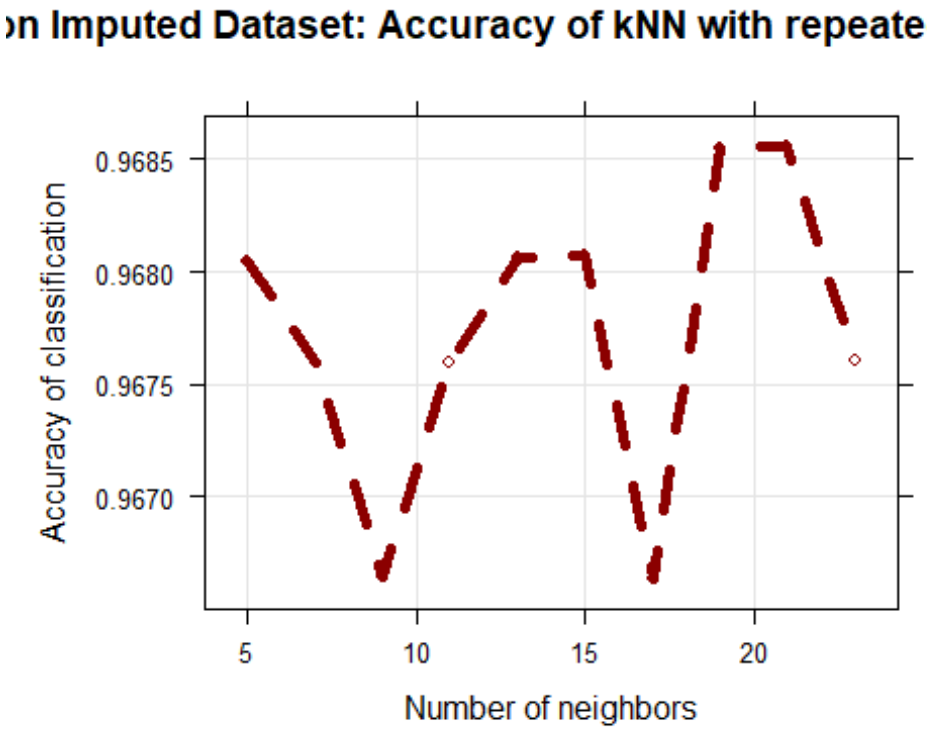
KNN vs Mean imputation

```
ctrl <- trainControl(method="repeatedcv",number=10,repeats = 3)
knn.mean <- train(x=df.mean[,1:10],y=as.factor(df.mean[,11]), method = "knn",
trControl = ctrl,
                preProcess = c("center","scale"), tuneLength = 10)

plot(knn.mean,col = "dark red",lwd=5,lty=2,cex.lab=1.25,cex.main=1.5,
     main="Mean Imputed Dataset: Accuracy of kNN with repeated 10-fold CV",
     xlab="Number of neighbors",
     ylab="Accuracy of classification")
```



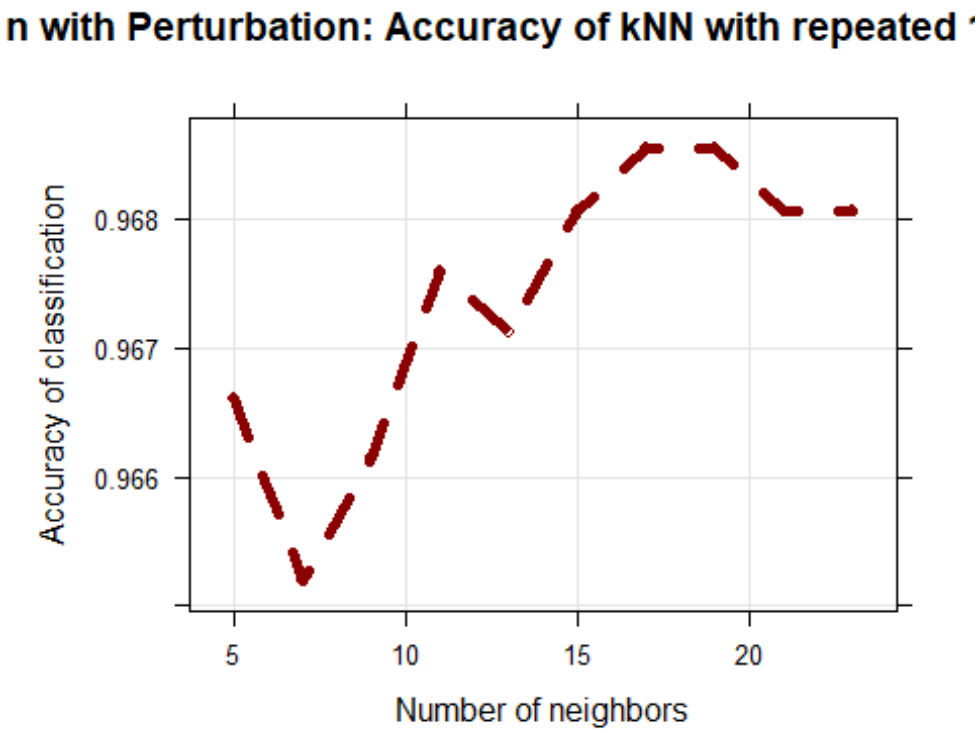
KNN vs Regular Regression

```
knn.regressed <-
train(x=df.final.regression[,1:10],y=as.factor(df.final.regression[,11]),
method = "knn", trControl = ctrl,
                    preProcess = c("center","scale"), tuneLength = 10)

plot(knn.regressed,col = "dark red",lwd=5,lty=2,cex.lab=1.25,cex.main=1.5,
     main="Regression Imputed Dataset: Accuracy of kNN with repeated 10-fold
CV",
     xlab="Number of neighbors",
     ylab="Accuracy of classification")
```



KNN vs Perturbed Regression

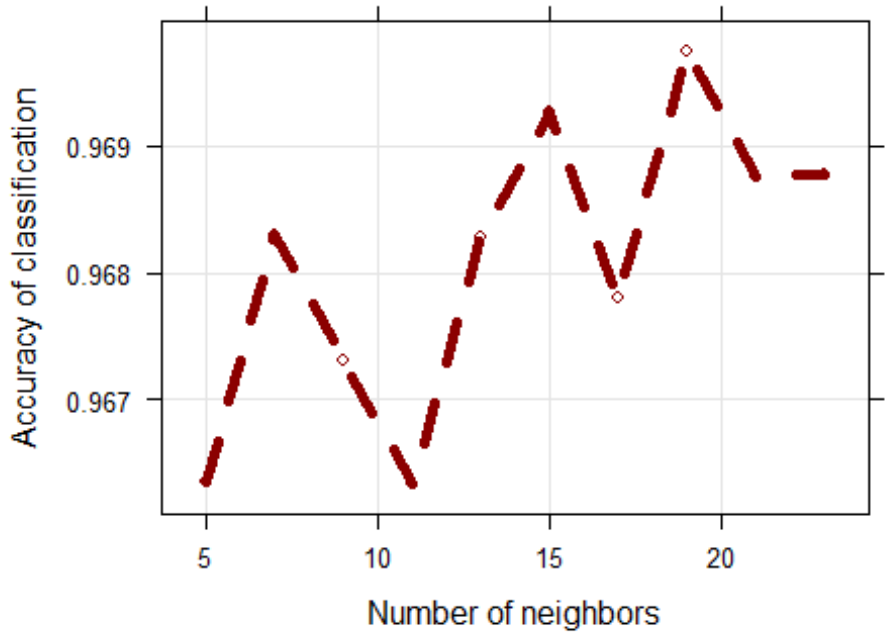
```
knn.perturbreg <-
train(x=df.final.regression.pertubed[,1:10],y=as.factor(df.final.regression.p
ertubed[,11]), method = "knn", trControl = ctrl,
                      preProcess = c("center","scale"), tuneLength = 10)

plot(x=knn.perturbreg,col = "dark red",lwd=5,lty=2,cex.lab=1.25,cex.main=1.5,
     main="Regression with Perturbation: Accuracy of kNN with repeated 10-
fold CV",
     xlab="Number of neighbors",
     ylab="Accuracy of classification")
```



KNN vs Removed missing data points

```
df1.removed<-na.omit(df1)
knn.removed <- train(x=df1.removed[,1:10],y=as.factor(df1.removed[,11]),
method = "knn", trControl = ctrl,
                        preProcess = c("center","scale"), tuneLength = 10)

plot(knn.removed,col = "dark red",lwd=5,lty=2,cex.lab=1.25,cex.main=1.5,
     main="Missing values removed: Accuracy of kNN with repeated 10-fold CV",
     xlab="Number of neighbors",
     ylab="Accuracy of classification")
```

values removed: Accuracy of kNN with repeated 10-



Using 10-fold cross-validation, the accuracies were as follows:

On average, KNN accuracies were consistently high across different methods, reaching slightly over 90% overall.

However, the optimal number of K neighbors varied by method:

Mean approach: 5 K neighbors Regular regression approach: 20 K neighbors Perturbed regression approach: 19 K neighbors Missing approach: 19 K neighbors

Summary

The “Bare_Nuclei” column contained 16 missing values, accounting for just 2.3% of the entire dataset, which is below the 5% threshold, making imputation acceptable.

The mean value for Bare Nuclei was calculated as 3.54.

An initial linear regression with all predictors showed multiple insignificant predictors.

Stepwise regression in both directions reduced the number of predictors, achieving an RMSE of 2.278 and an R-Squared of 0.618.

Imputation methods—including mean, missing, and perturbed regression—were compared using KNN. Results indicated high accuracy across methods, mostly in the 90% range. Regular regression and perturbed regression showed optimal K values around 20, while mean imputation achieved optimal results with a lower K of 5 neighbors.

Based on these findings, mean imputation appears favorable for simplicity and effectiveness, particularly given the low percentage of missing values. The key takeaway is that with a small proportion of missing data, straightforward methods like mean imputation are often best.

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?

One situation where optimization would be beneficial is in planning an efficient daily workout routine. The goal would be to maximize both fitness gains and time management by deciding the optimal duration and type of exercises each day.

To approach this, I'd need data on factors like workout times, types of exercises, and their impact on muscle groups, along with information about recovery times and energy levels throughout the week. With these data points, I could set constraints (like available time each day and necessary rest periods) and use an optimization model to create a balanced workout schedule that fits well with my other daily activities.