Applied Predictive Modeling 1/7/2018

Chapter 3: Data Pre-processing

Name: Kamaru-Deen Lawal

Email: NA

1

(a)

```
library(e1071)
library(mlbench)
data(Glass)
str(Glass)
summary(Glass)

# Correlation Matrix
sapply(Glass, class) # gives the class of each feature in Glass
sapply(Glass, is.factor) # tells uls whether the features in Glass are factors or not
cor(Glass[sapply(Glass, function(x) !is.factor(x))]) # remove factor col and make cor
    matrix

# Summary
summary(Glass)

# Type (categorical Counts)
w = table(Glass$Type)
t = as.data.frame(w)

# Visualization for Predictors

# Refractive Index (RI)

# Summary
summary(Glass$RI)

# Boxplot
bp <- boxplot(Glass$RI)
length(bp$out) # 17 outliers out of 214 points

# Distribution
hist(Glass$RI) # histogram for RI feature
skewness(Glass$RI) # 1.602
RITrans <- BoxCoxTrans(Glass$RI) # est lam = -2
RITrans
hist(predict(RITrans, Glass$RI)) # histogram of RI feature after boxcox transformation
skewness(predict(RITrans, Glass$RI)) # 1.565

# Sodium (Na)
hist(Glass$Na)
skewness(Glass$Na) # 0.4478343
NaTrans <- BoxCoxTrans(Glass$Na) # est lam = -.1
hist(predict(NaTrans, Glass$Na))
skewness(predict(NaTrans, Glass$Na)) # .033

# Magnesium (Mg)
```

```r
hist(Glass$Mg)
skewness(Glass$Mg) # -1.136
MgTrans <- BoxCoxTrans(Glass$Mg) # no transformation applied
MgTrans
hist(predict(MgTrans, Glass$Mg))
skewness(predict(MgTrans, Glass$Mg)) # -1.136

# Aluminum (Al)
hist(Glass$Al)
skewness(Glass$Al) # .8946
AlTrans <- BoxCoxTrans(Glass$Al) # est lam = .5
hist(predict(AlTrans, Glass$Al))
skewness(predict(AlTrans, Glass$Al)) # .091

# Silicon (Si)
hist(Glass$Si)
skewness(Glass$Si) # -.720
SiTrans <- BoxCoxTrans(Glass$Si) # est lam = 2
SiTrans
hist(predict(SiTrans, Glass$Si))
skewness(predict(SiTrans, Glass$Si)) # -.650

# Potassium (K)
hist(Glass$K)
skewness(Glass$K) # 6.46
KTrans <- BoxCoxTrans(Glass$K) # no transformation applied
KTrans

# Calcium (Ca)
hist(Glass$Ca)
skewness(Glass$Ca) # 2.018
CaTrans <- BoxCoxTrans(Glass$Ca) # est lam = -1.1
CaTrans
hist(predict(CaTrans, Glass$Ca))
skewness(predict(CaTrans, Glass$Ca)) # -.193

# Barium (Ba)
hist(Glass$Ba)
skewness(Glass$Ba) # 3.368
BaTrans <- BoxCoxTrans(Glass$Ba)
BaTrans # no transformation applied (lambda couldn't be estimated)

# Iron (Fe)
hist(Glass$Fe)
skewness(Glass$Fe) # 1.729
FeTrans <- BoxCoxTrans(Glass$Fe)
FeTrans # no est for lambda
```

(b) Out of the 214 data points in the set. 17 of them have refractive indices either greater than

$$Q3 + 1.5 \cdot IQR$$

or less than

$$Q1 - 1.5 \cdot IQR$$

Mg, Si, K, Ca, Ba, and Fe are all skewed predictors.

(c) Transforming all of skewed predictors seems like it would help the model. Predictors Ba, Fe, and K could not be transformed because they had 0 values in their respective columns.

2

(a)

```
library(mlbench)
data(Soybean) # 683 datapoints total

# date
w = table(Soybean$date) # 682 values => 1 missing
barplot(w, xlab = "Month", ylab = "Frequency") # .1% missing

# plant.stand
w = table(Soybean$plant.stand) # 647 values => 36 missing
barplot(w, xlab = "plant.stand", ylab = "Frequency") # 5.2% missing

# precip
w = table(Soybean$precip) # 645 values => 38 missing
barplot(w, xlab = "precip", ylab = "Frequency") # 5.6% missing

# temp
w = table(Soybean$temp) # 653 values => 30 missing
barplot(w, xlab = "temp", ylab = "Frequency") # 4.4% missing

# hail
w = table(Soybean$hail) # 562 values => 121 missing
barplot(w, xlab = "hail", ylab = "Frequency") # 17.7% missing

# crop.hist
w = table(Soybean$crop.hist) # 667 values => 16 missing
barplot(w, xlab = "crop.hist", ylab = "Frequency") # 2.3% missing

# area.dam
w = table(Soybean$area.dam) # 682 values => 1 missing
barplot(w, xlab = "area.dam", ylab = "Frequency") # .1% missing

# sever
w = table(Soybean$sever) # 562 values => 121 missing
barplot(w, xlab = "sever", ylab = "Frequency") # 17.7% missing

# seed.tmt
w = table(Soybean$seed.tmt) # 562 values => 121 missing
barplot(w, xlab = "seed.tmt", ylab = "Frequency") # 17.7%

# germ
w = table(Soybean$germ) # 571 values => 112 missing
barplot(w, xlab = "germ", ylab = "Frequency") # 16.4% missing

# plant.growth
w = table(Soybean$plant.growth) # 667 values => 16 missing
barplot(w, xlab = "plant.growth", ylab = "Frequency") # 2.3% missing

# leaves
w = table(Soybean$leaves) # 683 vals => 0 miss
barplot(w, xlab = "leaves", ylab = "Frequency") # 0% miss
```

```r
# leaf.halo
w = table(Soybean$leaf.halo) # 599 vals => 84 miss
barplot(w, xlab = "leaf.halo", ylab = "Frequency") # 12.3% miss

# leaf.marg
w = table(Soybean$leaf.marg) # 599 vals => 84 miss
barplot(w, xlab = "leaf.marg", ylab = "Frequency") # 12.3% miss

# leaf.size
w = table(Soybean$leaf.size) # 599 vals => 84 miss
barplot(w, xlab = "leaf.size", ylab = "Frequency") # 12.3% miss

# leaf.shread
w = table(Soybean$leaf.shread) # 583 vals => 100 miss
barplot(w, xlab = "leaf.shread", ylab = "Frequency") # 14.6% miss

# leaf.malf
w = table(Soybean$leaf.malf) # 599 vals => 84 miss
barplot(w, xlab = "leaf.malf", ylab = "Frequency") # 12.3% miss

# leaf.mild
w = table(Soybean$leaf.mild) # 575 vals => 108 miss
barplot(w, xlab = "leaf.mild", ylab = "Frequency") # 15.8% miss

# stem
w = table(Soybean$stem) # 667 vals => 16 miss
barplot(w, xlab = "stem", ylab = "Frequency") # 2.3% miss

# lodging
w = table(Soybean$lodging) # 562 vals => 121 miss
barplot(w, xlab = "lodging", ylab = "Frequency") # 17.7% miss

# stem.cankers
w = table(Soybean$stem.cankers) # 645 vals => 38 miss
barplot(w, xlab = "stem.cankers", ylab = "Frequency") # 5.6% miss

# canker.lesion
w = table(Soybean$canker.lesion) # 575 vals => 108 miss
barplot(w, xlab = "canker.lesion", ylab = "Frequency") # 15.8% miss

# fruiting.bodies
w = table(Soybean$fruiting.bodies) # 577 vals => 106 miss
barplot(w, xlab = "fruiting.bodies", ylab = "Frequency") # 15.5% miss

# ext.decay
w = table(Soybean$ext.decay) # 645 vals => 38 miss
barplot(w, xlab = "ext.decay", ylab = "Frequency") # 5.6% miss

# mycelium
w = table(Soybean$mycelium) # 645 vals => 38 miss
barplot(w, xlab = "mycelium", ylab = "Frequency") # 15.8% miss

# int.discolor
w = table(Soybean$int.discolor) # 645 vals => 38 miss
barplot(w, xlab = "int.discolor", ylab = "Frequency") # 15.8% miss
```

```r
# sclerotia
w = table(Soybean$sclerotia) # 645 vals => 38 miss
barplot(w, xlab = "sclerotia", ylab = "Frequency") # 15.8% miss

# fruit.pods
w = table(Soybean$fruit.pods) # 599 vals => 84 miss
barplot(w, xlab = "fruit.pods", ylab = "Frequency") # 12.3% miss

# fruit.spots
w = table(Soybean$fruit.spots) # 577 vals => 106 miss
barplot(w, xlab = "fruit.spots", ylab = "Frequency") # 15.5% miss

# seed
w = table(Soybean$seed) # 591 vals => 92 miss
barplot(w, xlab = "seed", ylab = "Frequency") # 13.5% miss

# mold.growth
w = table(Soybean$mold.growth) # 591 vals => 92 miss
barplot(w, xlab = "mold.growth", ylab = "Frequency") # 13.5% miss

# seed.discolor
w = table(Soybean$seed.discolor) # 577 vals => 106 miss
barplot(w, xlab = "seed.discolor", ylab = "Frequency") # 15.5% miss

# seed.size
w = table(Soybean$seed.size) # 591 vals => 92 miss
barplot(w, xlab = "seed.size", ylab = "Frequency") # 13.5% miss

# shriveling
w = table(Soybean$shriveling) # 577 vals => 106 miss
barplot(w, xlab = "shriveling", ylab = "Frequency") # 15.5% miss

# roots
w = table(Soybean$roots) # 652 vals => 31 miss
barplot(w, xlab = "roots", ylab = "Frequency") # 4.5% miss

# Nero Zero Features
?Soybean
nearZeroVar(Soybean)
```

leaf.mild, mycelium, a nd sclerotia all had ratios greater than 20 for the ratio of the most prevalent value to the frequency of the second most prevalent category. In addition all of these categories had less than 3 categories (unique values over sample size was significantly less than 10%). We conclude it would be advantageous to remove the predictors.

(b) hail, server, seed.tmt, germ, and lodging are all missing greater than 16%. canker.lesion, fruiting.bodies, mycelium, and int.discolor are all missing greater than 15% of the data. leaf.halo, leaf.marg, leaf.size, leaf.shread, leaf.malf, and leaf.mild were all missing greater than 12% of the data. All of the leaf. features are missing a similar percentage of data.

(c) I would start off by dropping all features with missing values (0% missing values), and then build different models based on the remaining features. Assess the predictive power of these models. If the model performs well on the datasets, then stop. Otherwise repeat the process above by gradually increasing the cutoff for what percentage of data can be missing. With regard to data imputation of the categorical variables I would start by simply adding the mode for each predictor. If the model

still lacked predictive power I would consider imputing using kNN, for $k = 5$, and the metric is simply the majority vote for the corresponding feature.

3

(a)

```
library(caret)
data(BloodBrain)
?BloodBrain
dim(bbbDescr)
length(logBBB)
```

(b)

```
# Degenerate Predictor Analysis
attrs <- attributes(bbbDescr)$names
numpreds <- length(attributes(bbbDescr)$names)
numsampls <- 208
plot(bbbDescr$negative, logBBB)
nearZeroVar(bbbDescr) # 3, 16, 17, 22, 25, 50, 60
plot(bbbDescr$negative, logBBB)
plot(bbbDescr$peoe_vsa.2.1, logBBB)
plot(bbbDescr$peoe_vsa.3.1, logBBB)
plot(bbbDescr$a_acid, logBBB)
plot(bbbDescr$vsa_acid, logBBB)
plot(bbbDescr$frac.anion7., logBBB)
plot(bbbDescr$alert, logBBB)

# Skew
skewVals <- abs(apply(bbbDescr, 2, skewness)) > 2
lessThanTwo <- 184 - sum(skewVals) # 161 predictors

# Example Predictors w/ abs(skew) < 2
plot(bbbDescr$tpsa, logBBB)
plot(bbbDescr$peoe_vsa.6.1, logBBB)
plot(bbbDescr$vsa_other, logBBB)

# Correlation Heat Map
install.packages("reshape2")
library(corrplot)
library(reshape2)
corr <- cor(bbbDescr[sapply(bbbDescr, function(x) !is.factor(x))])
corrplot(corr, method = "circle")

# Correlations Sorted
corr[upper.tri(corr, diag = TRUE)] <- NA
m <- melt(corr)
m <- m[order(- abs(m$value)), ] # sort by descending absolute correlation
df <- na.omit(m) # omit NA values
list <- split(df, 1:nrow(df)) # converts dataframe to list
```

(c)

```
pcaobject <- prcomp(bbbDescr, center = TRUE, scale. = TRUE)
percentvariance <- pcaobject$sd^2 / sum(pcaobject$sd^2) * 100
plot(percentvariance, type = "s", col = "green", ylab = "% Variance", xlab = "Component")
```

There are a a lot of strong relationships between the predictor data. Correlations in the predictor set could be reduced by applying some type of dimension reduction technique such as PCA. Some of the correlations show that some variables are measuring the same exact quantity. The percent variance accounted for with each component leveled off around 40, and it seems like only a few of the predictors ended up contributing significantly to the individual components.