Assignment 2 - Random Forest

Code

Gordon Wall (gwall2)

Load Relevant Libraries

Load Relevant Data

Hide

```
data("BreastCancer")
```

Examine and Tame Data

Levels

Hide

```
levels(BreastCancer$Class)
```

```
[1] "benign"    "malignant"
```

Skim for variable statistics by type

Hide

```
skim(BreastCancer)
```

```
-- Data Summary ------------------------
                           Values
Name                       BreastCancer
Number of rows             699
Number of columns          11
_____________________
Column type frequency:
  character                1
  factor                   10
_____________________
Group variables            None

-- Variable type: character ------------------------------------------------------------------
-----------------------------
# A tibble: 1 x 8
  skim_variable n_missing complete_rate   min   max empty n_unique whitespace
* <chr>              <int>         <dbl> <int> <int> <int>    <int>      <int>
1 Id                     0             1     5     8     0      645          0

-- Variable type: factor ---------------------------------------------------------------------
-----------------------------
# A tibble: 10 x 6
   skim_variable   n_missing complete_rate ordered n_unique top_counts
 * <chr>               <int>         <dbl> <lgl>      <int> <chr>
 1 Cl.thickness            0         1     TRUE          10 1: 145, 5: 130, 3: 108, 4: 80
 2 Cell.size               0         1     TRUE          10 1: 384, 10: 67, 3: 52, 2: 45
 3 Cell.shape              0         1     TRUE          10 1: 353, 2: 59, 10: 58, 3: 56
 4 Marg.adhesion           0         1     TRUE          10 1: 407, 2: 58, 3: 58, 10: 55
 5 Epith.c.size            0         1     TRUE          10 2: 386, 3: 72, 4: 48, 1: 47
 6 Bare.nuclei            16         0.977 FALSE         10 1: 402, 10: 132, 2: 30, 5: 30
 7 Bl.cromatin             0         1     FALSE         10 2: 166, 3: 165, 1: 152, 7: 73
 8 Normal.nucleoli         0         1     FALSE         10 1: 443, 10: 61, 3: 44, 2: 36
 9 Mitoses                 0         1     FALSE          9 1: 579, 2: 35, 3: 33, 10: 14
10 Class                   0         1     FALSE          2 ben: 458, mal: 241
```

all the numeric data is classified as a factor

by default in this dataset

We will change variables 2:10 to numeric before

use in the model

Tidy Data and Re-check

Hide

```
BreastCancer[,2:10] <- sapply(BreastCancer[,2:10], as.numeric)
skim(BreastCancer)
```

```
-- Data Summary ------------------------
                           Values
Name                       BreastCancer
Number of rows             699
Number of columns          11
_____________________
Column type frequency:
  factor                   2
  numeric                  9
_____________________
Group variables            None

-- Variable type: factor ---------------------------------------------------------------------
-----------------------------
# A tibble: 2 x 6
  skim_variable n_missing complete_rate ordered n_unique top_counts
* <chr>              <int>         <dbl> <lgl>      <int> <chr>
1 Id                     0             1 FALSE        645 118: 6, 127: 5, 119: 3, 101: 2
2 Class                  0             1 FALSE          2 ben: 458, mal: 241

-- Variable type: numeric --------------------------------------------------------------------
-----------------------------
# A tibble: 9 x 11
  skim_variable   n_missing complete_rate  mean    sd    p0   p25   p50   p75  p100 hist
* <chr>               <int>         <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <chr>
1 Cl.thickness            0             1  4.42  2.82     1     2     4     6    10 ▇▇▇▅▃
2 Cell.size               0             1  3.13  3.05     1     1     1     5    10 ▇▂▁▁▂
3 Cell.shape              0             1  3.21  2.97     1     1     1     5    10 ▇▂▁▁▁
4 Marg.adhesion           0             1  2.81  2.86     1     1     1     4    10 ▇▂▁▁▁
5 Epith.c.size            0             1  3.22  2.21     1     2     2     4    10 ▇▃▁▁▁
6 Bare.nuclei            16             1  3.54  3.64     1     1     1     6    10 ▇▁▁▁▂
7 Bl.cromatin             0             1  3.44  2.44     1     2     3     5    10 ▇▅▂▃▁
8 Normal.nucleoli         0             1  2.87  3.05     1     1     1     4    10 ▇▁▁▁▁
9 Mitoses                 0             1  1.57  1.62     1     1     1     1     9 ▇▁▁▁▁
```

All relevant numeric variables have been changed to numeric

There are 699 observations but only 645 unique values from the Id variable per the n_unique output of our skim statistics

Upon further inspection, the duplicated Id values are not redundant rows; the Id values may be the same, but the other row information is unique thus, they will be kept and Id variable will be dropped as it is negligible in this scenario

(had they been completely duplicated rows, we would consider removing them as long as they could be determined not to be unique data points that coincidentally had the same values in each variable; not likely for 16 rows)

Removal of ID column

Hide

```
bc <- BreastCancer[,-1]
skim(bc)
```

```
-- Data Summary ------------------------
                           Values
Name                       bc
Number of rows             699
Number of columns          10
_____________________
Column type frequency:
  factor                   1
  numeric                  9
_____________________
Group variables            None

-- Variable type: factor ---------------------------------------------------------------------
-----------------------------
# A tibble: 1 x 6
  skim_variable n_missing complete_rate ordered n_unique top_counts
* <chr>             <int>         <dbl> <lgl>      <int> <chr>
1 Class                 0             1 FALSE          2 ben: 458, mal: 241

-- Variable type: numeric --------------------------------------------------------------------
-----------------------------
# A tibble: 9 x 11
  skim_variable   n_missing complete_rate  mean    sd    p0   p25   p50   p75  p100 hist
* <chr>               <int>         <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <chr>
1 Cl.thickness            0         1      4.42  2.82     1     2     4     6    10 ▇▇▇▅▃
2 Cell.size               0         1      3.13  3.05     1     1     1     5    10 ▇▂▁▁▂
3 Cell.shape              0         1      3.21  2.97     1     1     1     5    10 ▇▂▁▁▂
4 Marg.adhesion           0         1      2.81  2.86     1     1     1     4    10 ▇▂▁▁▂
5 Epith.c.size            0         1      3.22  2.21     1     2     2     4    10 ▇▃▁▁▂
6 Bare.nuclei            16         0.977  3.54  3.64     1     1     1     6    10 ▇▁▁▁▂
7 Bl.cromatin             0         1      3.44  2.44     1     2     3     5    10 ▇▅▁▃▁
8 Normal.nucleoli         0         1      2.87  3.05     1     1     1     4    10 ▇▁▁▁▂
9 Mitoses                 0         1      1.57  1.62     1     1     1     1     9 ▇▁▁▁▁
```

Our data looks much nicer now

Bare.nuclei variable still needs to be addressed

as it’s the only one with 16 missing entries

Examine distinct values of Bare.nuclei

Hide

```
bc %>% distinct(Bare.nuclei)
```

Bare.nuclei <dbl>
1
10
2
4
3
9
7
NA
5
8

1-10 of 11 rows Previous **1** 2 Next

per skimming the data, 16 values are missing from the Bare.nuclei variable based on judgement, there are not enough data points in this set to justify removing rows (deleting 16/699 obsv is 2.3% of our data), so these NA values will be imputed with KNN instead

Imputing missing values with K-Nearest-Neighbors (KNN)

Hide

```
library(VIM)

bc.complete <- kNN(bc, variable = "Bare.nuclei", k = 5)
skim(bc.complete)
bc.complete <- bc.complete[,-11]

bc.complete %>% distinct(Bare.nuclei)
```

Build Training Model

Partition Dataset

Hide

```
installed.packages("randomForest")
```

```
     Package LibPath Version Priority Depends Imports LinkingTo Suggests Enhances License Licens
e_is_FOSS
     License_restricts_use OS_type Archs MD5sum NeedsCompilation Built
```

Model training data

Hide

```
library(randomForest)

rf.model <- train(Class~., data = train, method = "rf")
rf.model
```

```
Random Forest

490 samples
  9 predictor
  2 classes: 'benign', 'malignant'

No pre-processing
Resampling: Bootstrapped (25 reps)
Summary of sample sizes: 490, 490, 490, 490, 490, 490, ...
Resampling results across tuning parameters:

  mtry  Accuracy   Kappa
  2     0.9622850  0.9158223
  5     0.9563368  0.9023158
  9     0.9508223  0.8899742

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was mtry = 2.
```

Re-model training data with required mtry values (c(2,6,8))

Hide

```
rf.model2
```

```
Random Forest

490 samples
  9 predictor
  2 classes: 'benign', 'malignant'

No pre-processing
Resampling: Bootstrapped (25 reps)
Summary of sample sizes: 490, 490, 490, 490, 490, 490, ...
Resampling results across tuning parameters:

  mtry  Accuracy   Kappa
  2     0.9605014  0.9130698
  6     0.9524235  0.8952448
  8     0.9495591  0.8886232

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was mtry = 2.
```

Accuracy is highest at mtry = 2 (96%)

Build Test Model

Predict Test data with second random forest model

Hide

```
probs <- predict(rf.model2, test, type = "prob")
head(probs)
```

	benign <dbl>	malignant <dbl>
5	0.994	0.006
6	0.008	0.992
8	1.000	0.000
14	0.992	0.008
15	0.016	0.984
17	1.000	0.000

6 rows

Hide

```
pred.class <- predict(rf.model2, test, type = "raw")
head(pred.class)
```

```
[1] benign    malignant benign    benign    malignant benign
Levels: benign malignant
```

Compare predicted results with actual classifications

Hide

```
comparison <- table(test$Class, pred.class)
comparison
```

```
           pred.class
            benign malignant
  benign       133         4
  malignant      1        71
```

Final Thoughts:

I’m happy with the results of this model
Only 1 tumor in 699 was predicted as benign but
was actually malignant; this boasts great accuracy
In the real world, a medical research team would strive
to predict malignancy with 100% accuracy, but this random
forest model has done well to predict the test data
The model also predicted 4 incorrect malignant tumors,
which ended up being benign in reality
However, this inaccuracy is not as big of a deal in
application because incorrectly removing a benign tumor
has far less reprecussions (mainly cost-based) than does
incorrectly leaving a malignant tumor inside a patient