ECON4330 Final Project

Jerry Fang, Marcus Murphy, Changhong Liang

2022-12-09

1. Introduction and Questions

Since October is Breast Cancer Awareness Month, we have decided then to explore Breast Cancer Prediction and diagnosis. We would like to know "What machine-learning methods can we use to predict and diagnose breast cancer, and which of the methods we selected are the most effective in correctly diagnosing?" Furthermore, what observable factors have the strongest contribution to cancer malignancy and benignity prediction?

2. Background

Excluding skin cancers, breast cancer is the most common cancer diagnosed among women in the United States, accounting for nearly 1 in 3 cancers. It is also the second leading cause of cancer death among women after lung cancer. The treatment options and mortality risk from breast cancer crucially depends on whether the tumor is detected early, and whether the tumor is malignant or benign. Early detection of breast cancer could also help slow down the disease's progression and potentially reduce the mortality rate through appropriate therapeutic interventions at the right time.

Machine learning algorithms applied in healthcare setting are uniquely suited to play a significant role because they are precisely designed to make predictions about the nature (benign vs. belignant) and the progression of the breast cancer based on large number of observable features. The machine learning algorithms' high performance in predicting and diagnosis of the diseases means that well-trained algorithimics can potentially reduce costs of medicine, help doctors and patients make real time decisions, and to save people's lives. The most common data mining modeling goals are classification and prediction.

3. Literature Review

The most related paper to our study is a recent paper titled *"Machine Learning Algorithms For Breast Cancer Prediction And Diagnosis" (Naji, Filali, et al., 2021)* which predicts and diagnoses breast cancer. It explores which machine learning algorithms can more accurately predict the status (benign vs. malignant) of breast cancer among 10 covariates. The methods used in the Naji, Filali et al (2021) study include Support Vector Machine (SVM) (a supervised model that uses classification and regression analysis), Random Forest, Logistic Regression, Decision tree (C4.5) and K-Nearest Neighbours (KNN).They found that Support Vector Machine outperformed all other algorithms and achieved a predictive accuracy of 97.2%. Although there is some overlap with our analysis, we contribute to the existing analysis by also incorporating boosting, classification trees, and classification forests.

4. Dataset

The entire dataset consisted of breast cancer data scraped from the "Breast Cancer Wisconsin (Diagnostic) Data Set. The raw data focuses on fine-needle aspirate images of cell nuclei and features radius (the mean of distances from center of the nucleus to the perimeter), texture (standard deviation of the gray-scale values of the image), perimeter, area, smoothness (local variation in radius lengths), compactness ($\frac{(\text{Perimeter})^2}{(\text{Area}-1)}$), concavity (severity of concave portions of the contour), concave points (number of concave points on the contour), symmetry, and fractal dimension ("Coastline Approximation" − 1). Then, the mean, standard deviation, and worst/largest values of each feature were computed for each value, resulting in 30 real-valued features. The table below provides a snapshot of the raw data. However, we use R to read the csv file and process the data/variables involved in the dataset.

5. Methodology

5.1 text preprocessing

We have used text preprocessing to read the data and remove any variables that we don't need for the Machine Learning methods. Commands we used: dplyr::select(), mutate().

Variable list of dataset:

```
## 'data.frame':    569 obs. of  31 variables:
##  $ diagnosis              : Factor w/ 2 levels "M","B": 1 1 1 1 1 1 1 1 1 1 ...
##  $ radius_mean            : num  18 20.6 19.7 11.4 20.3 ...
##  $ texture_mean           : num  10.4 17.8 21.2 20.4 14.3 ...
##  $ perimeter_mean         : num  122.8 132.9 130 77.6 135.1 ...
##  $ area_mean              : num  1001 1326 1203 386 1297 ...
##  $ smoothness_mean        : num  0.1184 0.0847 0.1096 0.1425 0.1003 ...
##  $ compactness_mean       : num  0.2776 0.0786 0.1599 0.2839 0.1328 ...
##  $ concavity_mean         : num  0.3001 0.0869 0.1974 0.2414 0.198 ...
##  $ concave.points_mean    : num  0.1471 0.0702 0.1279 0.1052 0.1043 ...
##  $ symmetry_mean          : num  0.242 0.181 0.207 0.26 0.181 ...
##  $ fractal_dimension_mean : num  0.0787 0.0567 0.06 0.0974 0.0588 ...
##  $ radius_se              : num  1.095 0.543 0.746 0.496 0.757 ...
##  $ texture_se             : num  0.905 0.734 0.787 1.156 0.781 ...
##  $ perimeter_se           : num  8.59 3.4 4.58 3.44 5.44 ...
##  $ area_se                : num  153.4 74.1 94 27.2 94.4 ...
##  $ smoothness_se          : num  0.0064 0.00522 0.00615 0.00911 0.01149 ...
##  $ compactness_se         : num  0.049 0.0131 0.0401 0.0746 0.0246 ...
##  $ concavity_se           : num  0.0537 0.0186 0.0383 0.0566 0.0569 ...
##  $ concave.points_se      : num  0.0159 0.0134 0.0206 0.0187 0.0188 ...
##  $ symmetry_se            : num  0.03 0.0139 0.0225 0.0596 0.0176 ...
##  $ fractal_dimension_se   : num  0.00619 0.00353 0.00457 0.00921 0.00511 ...
##  $ radius_worst           : num  25.4 25 23.6 14.9 22.5 ...
##  $ texture_worst          : num  17.3 23.4 25.5 26.5 16.7 ...
##  $ perimeter_worst        : num  184.6 158.8 152.5 98.9 152.2 ...
##  $ area_worst             : num  2019 1956 1709 568 1575 ...
##  $ smoothness_worst       : num  0.162 0.124 0.144 0.21 0.137 ...
##  $ compactness_worst      : num  0.666 0.187 0.424 0.866 0.205 ...
##  $ concavity_worst        : num  0.712 0.242 0.45 0.687 0.4 ...
##  $ concave.points_worst   : num  0.265 0.186 0.243 0.258 0.163 ...
##  $ symmetry_worst         : num  0.46 0.275 0.361 0.664 0.236 ...
##  $ fractal_dimension_worst: num  0.1189 0.089 0.0876 0.173 0.0768 ...
```

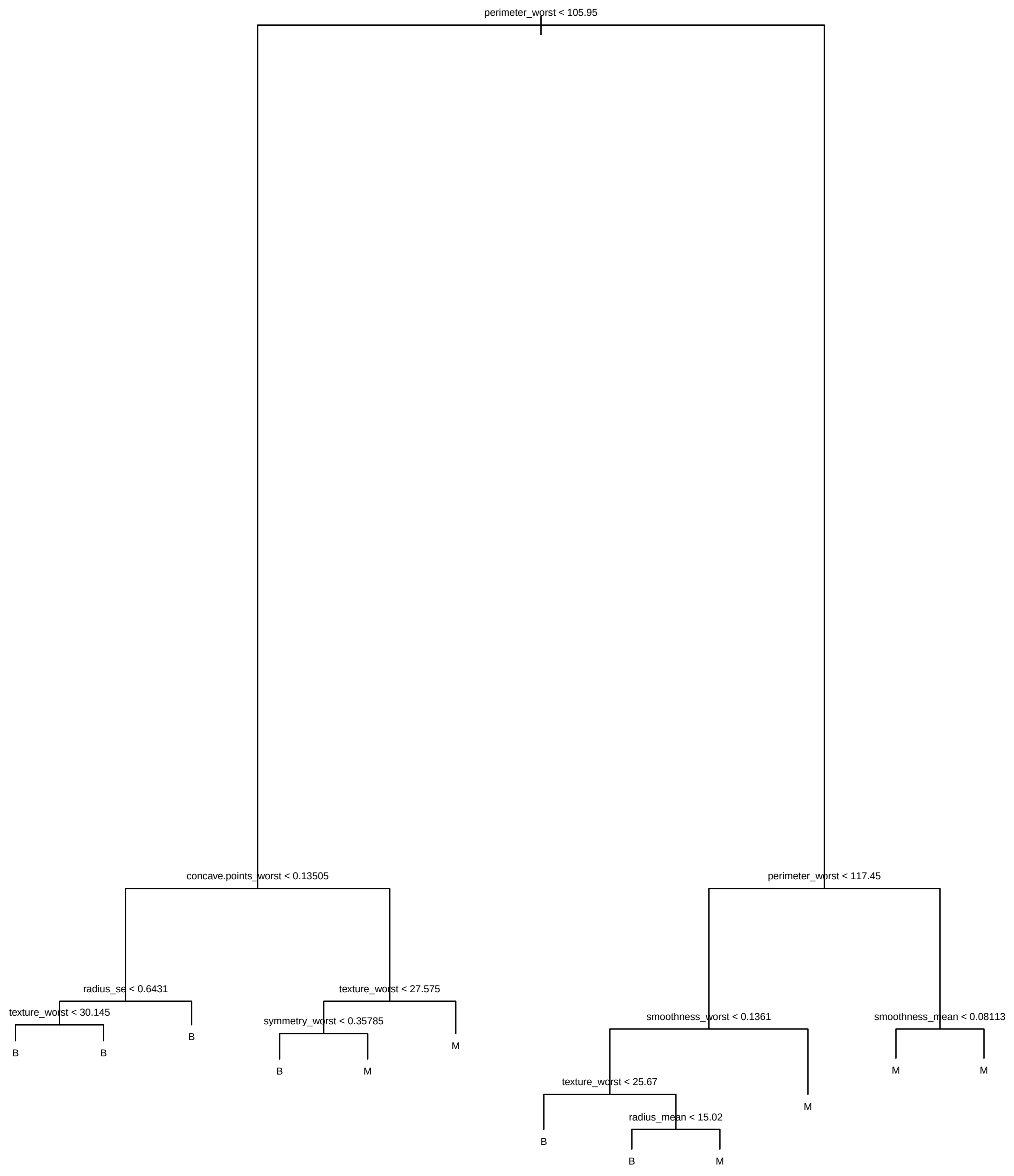
Summary Statistics of dataset:

```
##  diagnosis  radius_mean      texture_mean   perimeter_mean     area_mean
##  M:212      Min.   : 6.981   Min.   : 9.71   Min.   : 43.79   Min.   : 143.5
##  B:357      1st Qu.:11.700   1st Qu.:16.17   1st Qu.: 75.17   1st Qu.: 420.3
##             Median :13.370   Median :18.84   Median : 86.24   Median : 551.1
##             Mean   :14.127   Mean   :19.29   Mean   : 91.97   Mean   : 654.9
##             3rd Qu.:15.780   3rd Qu.:21.80   3rd Qu.:104.10   3rd Qu.: 782.7
##             Max.   :28.110   Max.   :39.28   Max.   :188.50   Max.   :2501.0
##  smoothness_mean   compactness_mean  concavity_mean    concave.points_mean
##  Min.   :0.05263   Min.   :0.01938   Min.   :0.00000   Min.   :0.00000
##  1st Qu.:0.08637   1st Qu.:0.06492   1st Qu.:0.02956   1st Qu.:0.02031
##  Median :0.09587   Median :0.09263   Median :0.06154   Median :0.03350
##  Mean   :0.09636   Mean   :0.10434   Mean   :0.08880   Mean   :0.04892
##  3rd Qu.:0.10530   3rd Qu.:0.13040   3rd Qu.:0.13070   3rd Qu.:0.07400
##  Max.   :0.16340   Max.   :0.34540   Max.   :0.42680   Max.   :0.20120
##  symmetry_mean    fractal_dimension_mean   radius_se        texture_se
##  Min.   :0.1060   Min.   :0.04996        Min.   :0.1115   Min.   :0.3602
##  1st Qu.:0.1619   1st Qu.:0.05770        1st Qu.:0.2324   1st Qu.:0.8339
##  Median :0.1792   Median :0.06154        Median :0.3242   Median :1.1080
##  Mean   :0.1812   Mean   :0.06280        Mean   :0.4052   Mean   :1.2169
##  3rd Qu.:0.1957   3rd Qu.:0.06612        3rd Qu.:0.4789   3rd Qu.:1.4740
##  Max.   :0.3040   Max.   :0.09744        Max.   :2.8730   Max.   :4.8850
##   perimeter_se       area_se        smoothness_se      compactness_se
##  Min.   : 0.757   Min.   :  6.802   Min.   :0.001713   Min.   :0.002252
##  1st Qu.: 1.606   1st Qu.: 17.850   1st Qu.:0.005169   1st Qu.:0.013080
##  Median : 2.287   Median : 24.530   Median :0.006380   Median :0.020450
##  Mean   : 2.866   Mean   : 40.337   Mean   :0.007041   Mean   :0.025478
##  3rd Qu.: 3.357   3rd Qu.: 45.190   3rd Qu.:0.008146   3rd Qu.:0.032450
##  Max.   :21.980   Max.   :542.200   Max.   :0.031130   Max.   :0.135400
##   concavity_se     concave.points_se   symmetry_se       fractal_dimension_se
##  Min.   :0.00000   Min.   :0.000000   Min.   :0.007882   Min.   :0.0008948
##  1st Qu.:0.01509   1st Qu.:0.007638   1st Qu.:0.015160   1st Qu.:0.0022480
##  Median :0.02589   Median :0.010930   Median :0.018730   Median :0.0031870
##  Mean   :0.03189   Mean   :0.011796   Mean   :0.020542   Mean   :0.0037949
##  3rd Qu.:0.04205   3rd Qu.:0.014710   3rd Qu.:0.023480   3rd Qu.:0.0045580
##  Max.   :0.39600   Max.   :0.052790   Max.   :0.078950   Max.   :0.0298400
##   radius_worst   texture_worst   perimeter_worst    area_worst
##  Min.   : 7.93   Min.   :12.02   Min.   : 50.41   Min.   : 185.2
##  1st Qu.:13.01   1st Qu.:21.08   1st Qu.: 84.11   1st Qu.: 515.3
##  Median :14.97   Median :25.41   Median : 97.66   Median : 686.5
##  Mean   :16.27   Mean   :25.68   Mean   :107.26   Mean   : 880.6
##  3rd Qu.:18.79   3rd Qu.:29.72   3rd Qu.:125.40   3rd Qu.:1084.0
##  Max.   :36.04   Max.   :49.54   Max.   :251.20   Max.   :4254.0
##  smoothness_worst  compactness_worst concavity_worst  concave.points_worst
##  Min.   :0.07117   Min.   :0.02729   Min.   :0.0000   Min.   :0.00000
##  1st Qu.:0.11660   1st Qu.:0.14720   1st Qu.:0.1145   1st Qu.:0.06493
##  Median :0.13130   Median :0.21190   Median :0.2267   Median :0.09993
##  Mean   :0.13237   Mean   :0.25427   Mean   :0.2722   Mean   :0.11461
##  3rd Qu.:0.14600   3rd Qu.:0.33910   3rd Qu.:0.3829   3rd Qu.:0.16140
##  Max.   :0.22260   Max.   :1.05800   Max.   :1.2520   Max.   :0.29100
##  symmetry_worst   fractal_dimension_worst
##  Min.   :0.1565   Min.   :0.05504
##  1st Qu.:0.2504   1st Qu.:0.07146
##  Median :0.2822   Median :0.08004
##  Mean   :0.2901   Mean   :0.08395
##  3rd Qu.:0.3179   3rd Qu.:0.09208
##  Max.   :0.6638   Max.   :0.20750
```

5.2 Classification Trees

We have used this method to create a classification tree to categorize the variables of the dataset and determine whether it is Benign or Malignant. We also used this method to check the test CE using the LOOCV and created the confusion matrix. In addition, we pruned the tree to simplify the number of branches on the tree but still classify whether the diagnosis is Benign or Malignant.

Plots the tree and provides a summary via the CART algorithm.



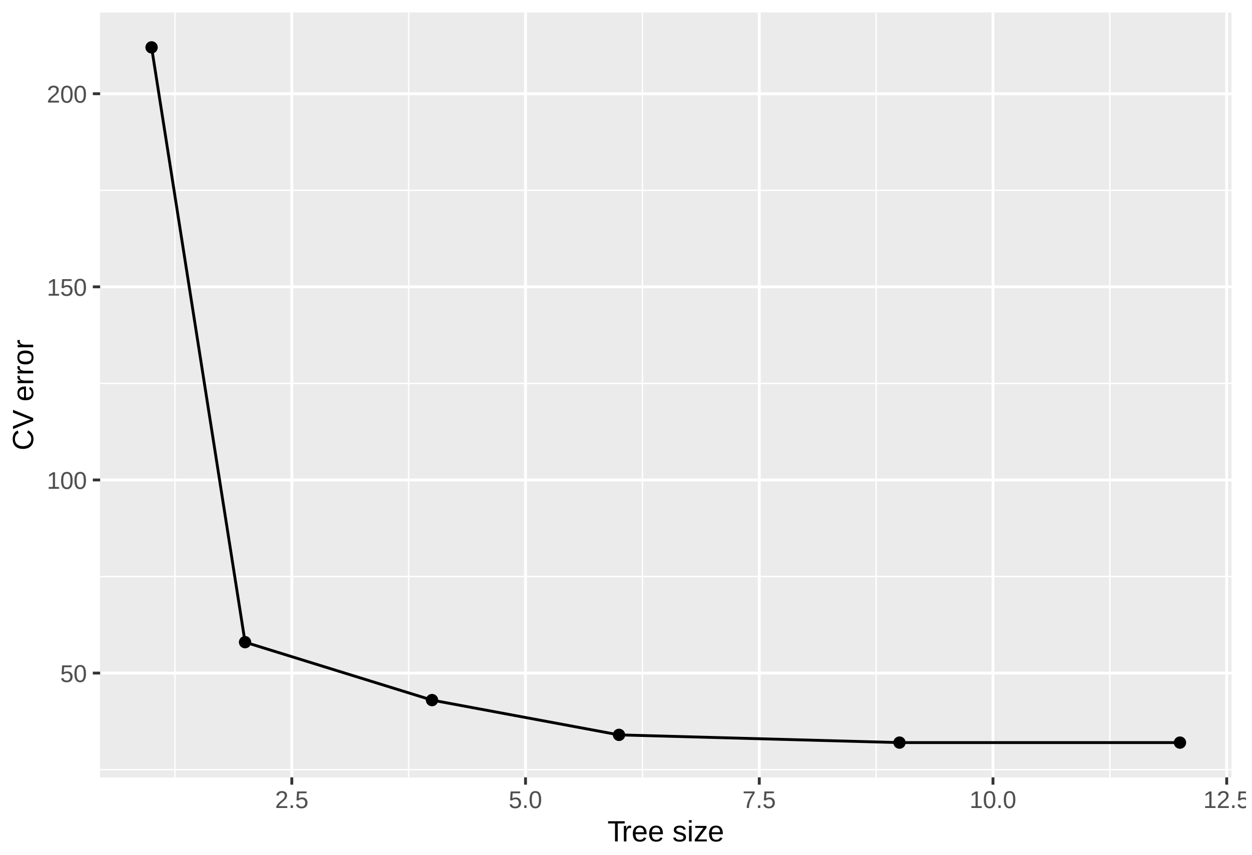
```
##
## Classification tree:
## tree(formula = diagnosis ~ radius_mean + texture_mean + perimeter_mean +
##     area_mean + smoothness_mean + compactness_mean + concavity_mean +
##     concave.points_mean + symmetry_mean + fractal_dimension_mean +
##     radius_se + texture_se + perimeter_se + area_se + smoothness_se +
##     compactness_se + concavity_se + concave.points_se + symmetry_se +
##     symmetry_se + fractal_dimension_se + radius_worst + texture_worst +
##     perimeter_worst + area_worst + smoothness_worst + compactness_worst +
##     concavity_worst + concave.points_worst + symmetry_worst +
##     fractal_dimension_worst, data = Diagnosis)
## Variables actually used in tree construction:
## [1] "perimeter_worst"      "concave.points_worst" "radius_se"
## [4] "texture_worst"        "symmetry_worst"       "smoothness_worst"
## [7] "radius_mean"          "smoothness_mean"
## Number of terminal nodes:  12
## Residual mean deviance:  0.09294 = 51.77 / 557
## Misclassification error rate: 0.01757 = 10 / 569
```

Use the LOOCV to check the test CE and creates the confusion matrix.

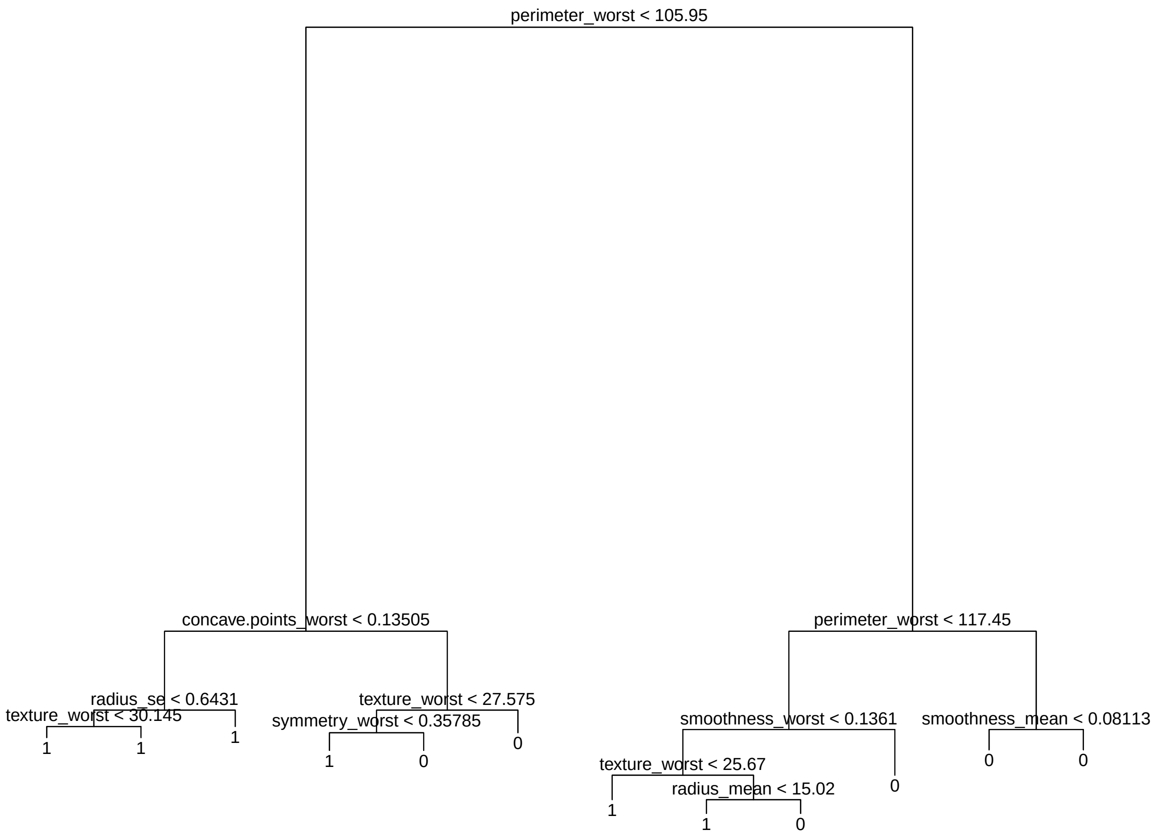
```
##
## predict.Diagnosis_classification   0   1
##                                 0 204   5
##                                 1   7 352
```

Pruning the classification trees

```
## $size
## [1] 12  9  6  4  2  1
##
## $dev
## [1]  32  32  34  43  58 212
##
## $k
## [1]  -Inf   0.0   3.0   4.5   9.0 166.0
##
## $method
## [1] "misclass"
##
## attr(,"class")
## [1] "prune"         "tree.sequence"
```



Plots the pruned tree



Double checks the CE via LOOCV and again, this creates the confusion matrix

```
##
##       0   1
##   0 206   4
##   1   6 353
```

5.3 Classification Forest

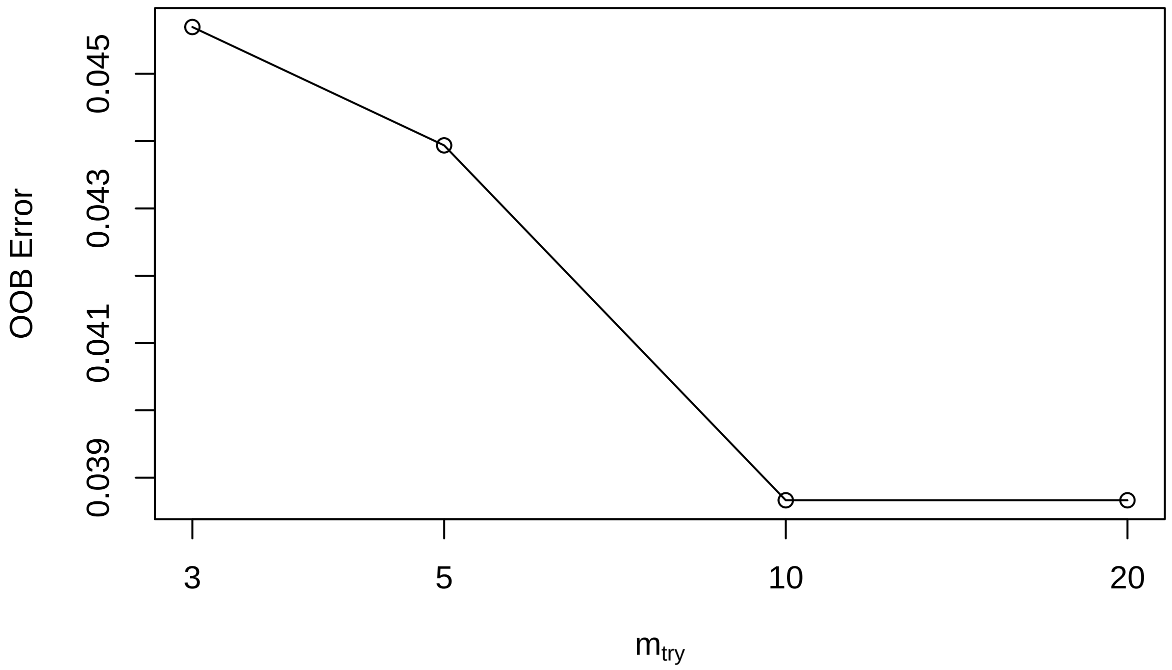
We have used classification forests to obtain the OOB as well as the error rate for whether the Diagnosis is Benign or Malignant. The chart provides different error rates. We also tuned m with a plot and chose m that corresponds with the lowest OOB Error always changes. In this case, we decided to draw the m presenting most in recurrent running of the codes and ended up choosing $m = 5$. Finally, we create importance plots to determine the accuracy as well as the Gini coefficient.

Trains the model and finds the error rate (OOB, Malignant, and Benign)

```
##        OOB          M          B
## 0.03690685 0.06603774 0.01960784
```

Find the choice of m using tuneRF

```
## mtry = 5  OOB error = 4.39%
## Searching left ...
## mtry = 3     OOB error = 4.57%
## -0.04 0.05
## Searching right ...
## mtry = 10    OOB error = 3.87%
## 0.12 0.05
## mtry = 20    OOB error = 3.87%
## 0 0.05
```



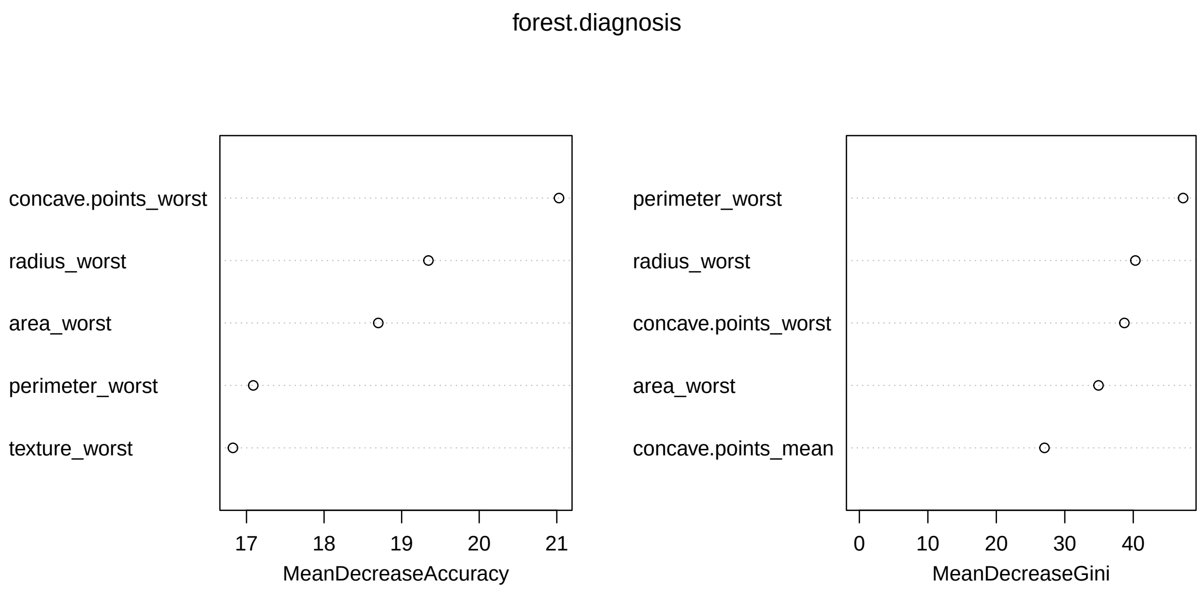
```
##
## Call:
##  randomForest(x = x, y = y, mtry = res[which.min(res[, 2]), 1])
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 10
##
##         OOB estimate of  error rate: 3.69%
## Confusion matrix:
##     M   B class.error
## M 199  13  0.06132075
## B   8 349  0.02240896
```

As shown in the plot, the m corresponding with the lowest OOB Error always changes. So we decided to draw the m that occurs the most in recurrent codes. In this case, we choose 8.

Now, we will use $m = 8$ (or $mtry = 8$) to find the error rate (OOB, Malignant, and Benign).

```
##        OOB          M          B
## 0.03339192 0.05188679 0.02240896
```

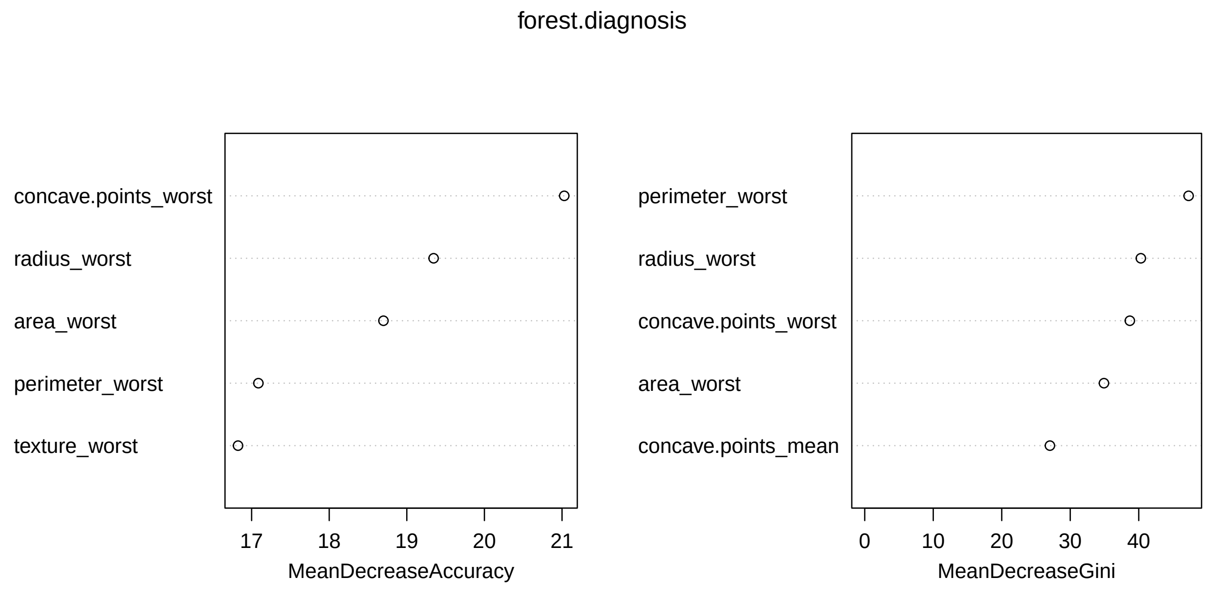
Creates the first Importance Plot. Note: the plot below provides a measure of the mean decrease in prediction accuracy anda measure of the total decrease in training MSE (or RSS) resulting from plots over that variable averaged over all trees.



```
## quartz_off_screen
##                 3
```

```
## pdf
##   2
```

Creates the Second Importance Plot. Note: the plot below provides a measure of the mean decrease in prediction accuracy and a measure of the total decrease in training MSE (or RSS) resulting from plots over that variable averaged over all trees.



```
## quartz_off_screen
##                 3

## pdf
##   2
```

5.4 Logistics regression (multi)

When running the Logistics regression Machine Learning method, we found the coefficients as well as the standard errors of each of the variables.

Runs a Logistic regression using the glm function, and diagnosis `results = binomial` and provides the corresponding summary statistics

```
## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##
## Call:
## glm(formula = diagnosis ~ radius_mean + texture_mean + perimeter_mean +
##     area_mean + smoothness_mean + compactness_mean + concavity_mean +
##     concave.points_mean + symmetry_mean + fractal_dimension_mean +
##     radius_se + texture_se + perimeter_se + area_se + smoothness_se +
##     compactness_se + concavity_se + concave.points_se + symmetry_se +
##     symmetry_se + fractal_dimension_se + radius_worst + texture_worst +
##     perimeter_worst + area_worst + smoothness_worst + compactness_worst +
##     concavity_worst + concave.points_worst + symmetry_worst +
##     fractal_dimension_worst, family = binomial, data = Diagnosis)
##
## Deviance Residuals:
```

```
##    Min      1Q  Median      3Q     Max
##  -8.49   -8.49    8.49    8.49    8.49
##
## Coefficients:
##                           Estimate Std. Error z value Pr(>|z|)
## (Intercept)              3.100e+06  2.816e+05  11.011  < 2e-16 ***
## radius_mean             -2.612e+06  2.693e+05  -9.700  < 2e-16 ***
## texture_mean            -2.107e+05  1.471e+04 -14.325  < 2e-16 ***
## perimeter_mean          -1.585e+06  2.464e+04 -64.334  < 2e-16 ***
## area_mean                1.400e+05  3.907e+03  35.833  < 2e-16 ***
## smoothness_mean          1.640e+08  8.361e+06  19.621  < 2e-16 ***
## compactness_mean         6.915e+06  3.213e+06   2.152 0.031361 *
## concavity_mean          -1.120e+06  1.408e+06  -0.795 0.426625
## concave.points_mean      1.846e+07  5.382e+06   3.430 0.000603 ***
## symmetry_mean           -4.356e+07  7.772e+05 -56.052  < 2e-16 ***
## fractal_dimension_mean   4.555e+07  2.169e+06  21.003  < 2e-16 ***
## radius_se               -3.581e+07  1.169e+06 -30.642  < 2e-16 ***
## texture_se              -6.852e+06  2.005e+05 -34.177  < 2e-16 ***
## perimeter_se            -1.830e+06  4.720e+04 -38.771  < 2e-16 ***
## area_se                  6.879e+05  1.835e+04  37.488  < 2e-16 ***
## smoothness_se           -8.061e+08  1.224e+07 -65.864  < 2e-16 ***
## compactness_se           1.908e+08  5.732e+06  33.282  < 2e-16 ***
## concavity_se            -1.645e+08  5.341e+06 -30.800  < 2e-16 ***
## concave.points_se        1.356e+09  4.013e+07  33.785  < 2e-16 ***
## symmetry_se             -3.110e+08  4.126e+06 -75.376  < 2e-16 ***
## fractal_dimension_se    -1.627e+09  6.597e+07 -24.664  < 2e-16 ***
## radius_worst             6.596e+06  2.143e+05  30.781  < 2e-16 ***
## texture_worst            6.276e+05  2.437e+04  25.754  < 2e-16 ***
## perimeter_worst          3.807e+05  1.219e+04  31.229  < 2e-16 ***
## area_worst              -9.631e+04  2.741e+03 -35.140  < 2e-16 ***
## smoothness_worst         2.325e+07  3.298e+06   7.051 1.78e-12 ***
## compactness_worst       -9.670e+06  3.999e+05 -24.179  < 2e-16 ***
## concavity_worst          3.258e+07  1.523e+06  21.386  < 2e-16 ***
## concave.points_worst    -1.540e+08  5.471e+06 -28.151  < 2e-16 ***
## symmetry_worst           2.662e+07  3.392e+05  78.464  < 2e-16 ***
## fractal_dimension_worst  3.980e+07  5.340e+06   7.453 9.13e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance:   751.44  on 568  degrees of freedom
## Residual deviance: 32006.76  on 538  degrees of freedom
## AIC: 32069
##
## Number of Fisher Scoring iterations: 25
```

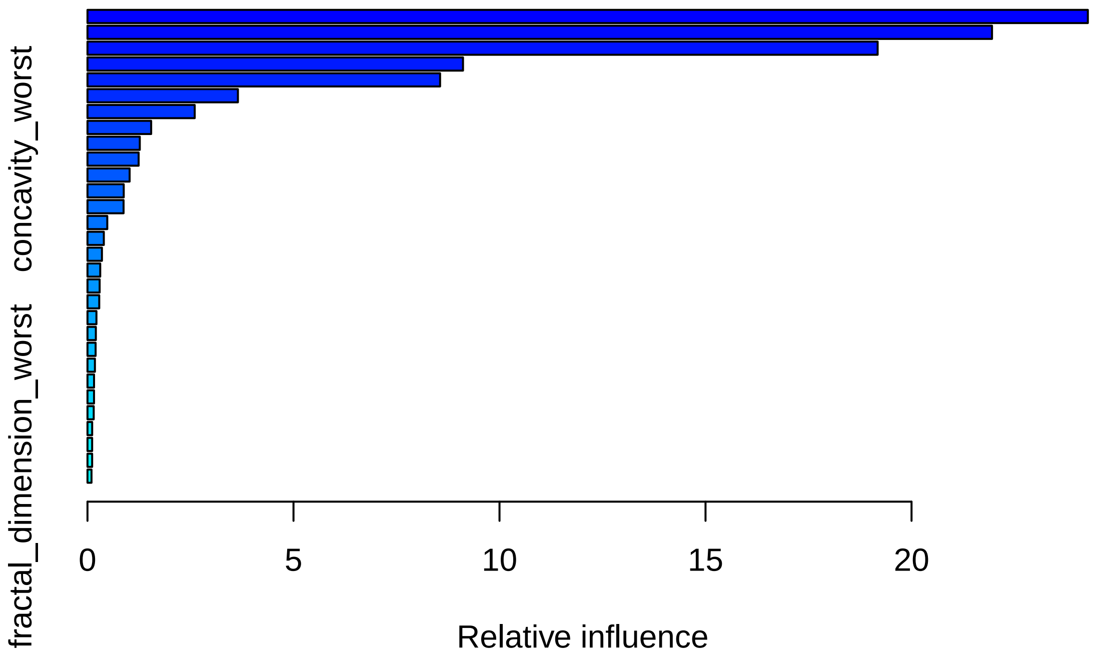
Provides the coefficients for each of the variables from the Logistics Regression.

```
##             (Intercept)             radius_mean            texture_mean
##            3.100286e+06           -2.611694e+06           -2.106608e+05
##          perimeter_mean               area_mean         smoothness_mean
##           -1.585158e+06            1.400100e+05            1.640417e+08
##        compactness_mean          concavity_mean     concave.points_mean
##             6.915038e+06         -1.119655e+06          1.846203e+07
##            symmetry_mean fractal_dimension_mean             radius_se
##            -4.356399e+07          4.554902e+07         -3.581468e+07
##               texture_se          perimeter_se               area_se
##            -6.852471e+06         -1.830026e+06          6.879468e+05
##            smoothness_se        compactness_se          concavity_se
##            -8.061208e+08          1.907881e+08         -1.644867e+08
##        concave.points_se           symmetry_se  fractal_dimension_se
##             1.355626e+09         -3.109821e+08         -1.627064e+09
##             radius_worst         texture_worst       perimeter_worst
##             6.596245e+06          6.275811e+05          3.807160e+05
##               area_worst      smoothness_worst     compactness_worst
##            -9.630807e+04          2.325375e+07         -9.669639e+06
##          concavity_worst  concave.points_worst        symmetry_worst
##             3.258101e+07         -1.540096e+08          2.661681e+07
## fractal_dimension_worst
##             3.979628e+07
```

5.5 Boosting

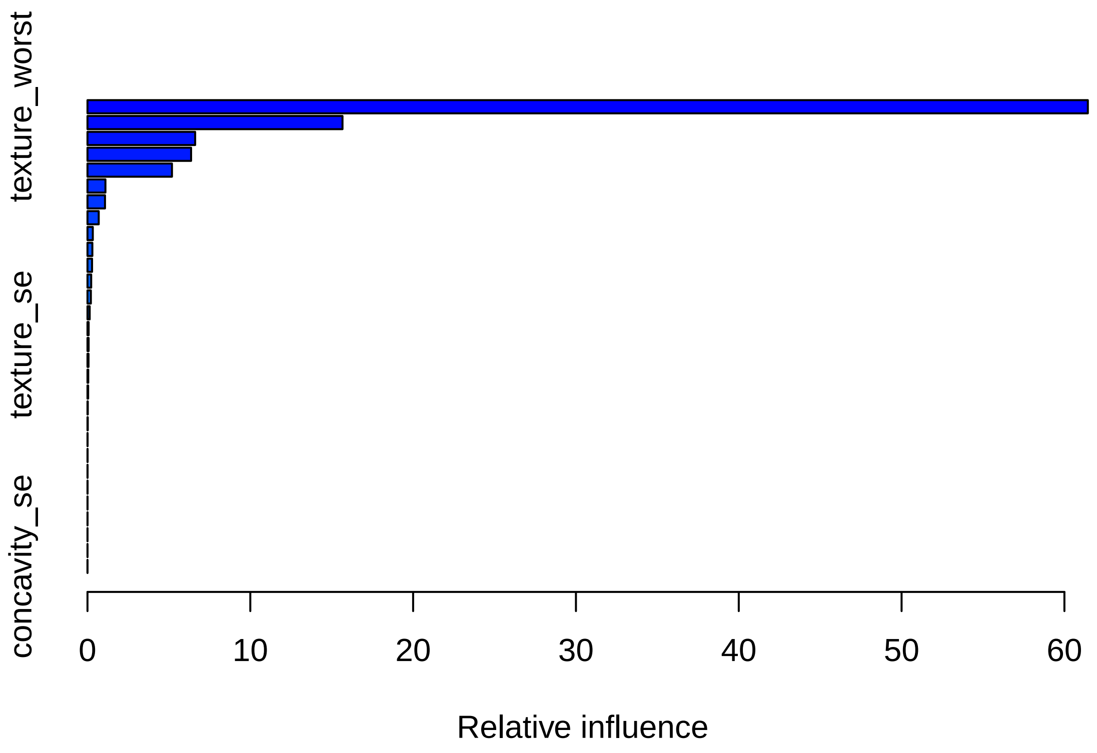
Boosting is thus far more accurate for predictions. The coefficients seem to imply that as many of the cell nuclei become larger and more dense, the diagnosis would become more serious. Hence, it is important to check regularly for suspicious lumps.

Boosted classification model is computed using the `gbm()`, and the argument, `distribution = bernoulli`, implies we are using classification trees. This provides information about the relative influence via dataset as well as in a graph.



```
##                                             var    rel.inf
## concave.points_worst       concave.points_worst 24.2793981
## area_worst                           area_worst 21.9532781
## perimeter_worst                 perimeter_worst 19.1774330
## radius_worst                       radius_worst  9.1140568
## concave.points_mean         concave.points_mean  8.5578646
## texture_worst                     texture_worst  3.6515332
## area_se                                 area_se  2.6032073
## texture_mean                       texture_mean  1.5456940
## concavity_mean                   concavity_mean  1.2696843
## concavity_worst                 concavity_worst  1.2421963
## compactness_worst             compactness_worst  1.0228737
## perimeter_mean                   perimeter_mean  0.8796910
## area_mean                             area_mean  0.8760130
## symmetry_worst                   symmetry_worst  0.4807475
## compactness_se                   compactness_se  0.3968284
## compactness_mean               compactness_mean  0.3511683
## smoothness_worst               smoothness_worst  0.3093689
## concave.points_se             concave.points_se  0.2950885
## radius_mean                         radius_mean  0.2852502
## fractal_dimension_mean   fractal_dimension_mean  0.2176672
## concavity_se                       concavity_se  0.2035083
## fractal_dimension_se       fractal_dimension_se  0.1978075
## smoothness_mean                 smoothness_mean  0.1809173
## perimeter_se                       perimeter_se  0.1595039
## smoothness_se                     smoothness_se  0.1589315
## symmetry_se                             symmetry_se  0.1517046
## symmetry_mean                         symmetry_mean  0.1140421
## texture_se                               texture_se  0.1136315
## radius_se                                 radius_se  0.1134380
## fractal_dimension_worst fractal_dimension_worst  0.0974729
```

Provides the influence of each variable, sort of similar to importance plots



```
##                        var      rel.inf
## 1            texture_worst 61.439790875
## 2      concave.points_mean 15.661843669
## 3            symmetry_worst  6.613850802
## 4     concave.points_worst  6.364095909
## 5               area_worst  5.189635682
## 6             texture_mean  1.102862638
## 7                  area_se  1.076064965
## 8          perimeter_worst  0.692140848
## 9         smoothness_worst  0.328198133
## 10         concavity_worst  0.300134559
## 11        compactness_mean  0.282025127
## 12          compactness_se  0.227357133
## 13 fractal_dimension_worst  0.207803563
## 14           smoothness_se  0.135481779
## 15             symmetry_se  0.071360511
## 16              texture_se  0.070665395
## 17           symmetry_mean  0.064717786
## 18          concavity_mean  0.046772092
## 19  fractal_dimension_mean  0.041305031
## 20            radius_worst  0.018955080
## 21    fractal_dimension_se  0.017329045
## 22       compactness_worst  0.008618446
## 23         smoothness_mean  0.007967842
## 24       concave.points_se  0.006962445
## 25          perimeter_mean  0.006666503
## 26               radius_se  0.005148085
## 27            perimeter_se  0.004403249
## 28               area_mean  0.003610940
## 29             radius_mean  0.002676206
## 30            concavity_se  0.001555664
```

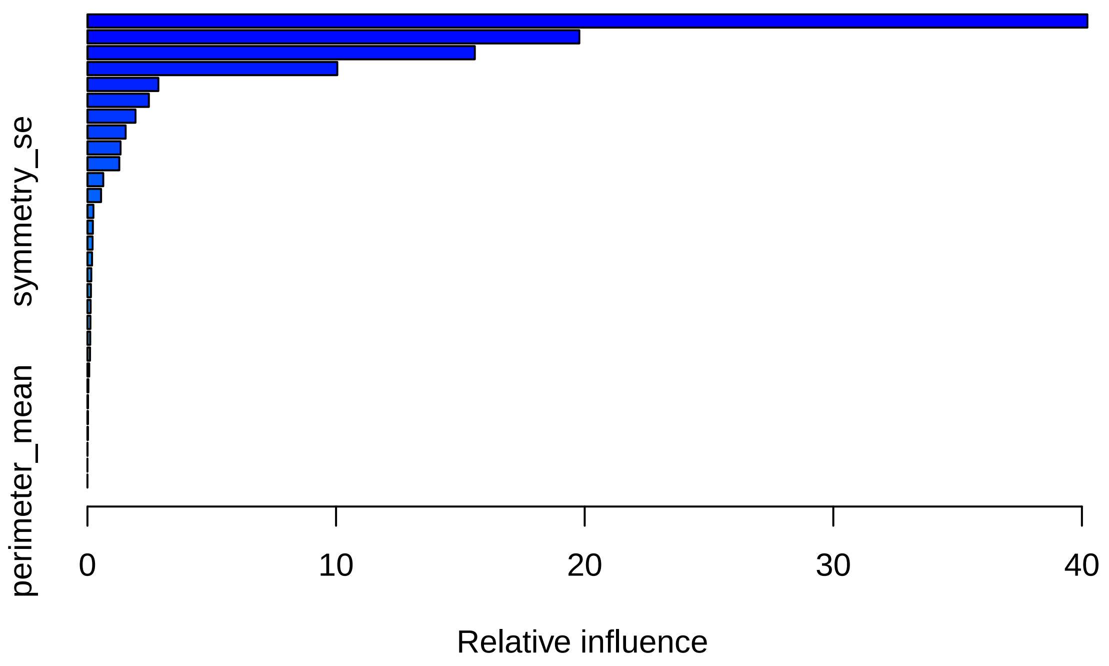
Tuning parameter choice, selecting all tuning parameters optimally through hyper-parameter search.

```
## [1] 81
```

Loop through each possible shrinkage and tree depth combination, determining the optimal number of trees for each, as well as the corresponding minimal deviance (where minimum is over number of trees holding shrinkage and tree depth fixed)

```
##    shrinkage interaction.depth optimal_trees min_deviance
## 1      0.100                 1           346    0.1783031
## 2      0.025                 1           933    0.1873544
## 3      0.050                 1           890    0.1890660
## 4      0.025                 2           591    0.1951681
## 5      0.050                 2           284    0.1965360
## 6      0.250                 1            53    0.1971467
## 7      0.010                 2           998    0.2015700
## 8      0.100                 2           143    0.2026046
## 9      0.025                 3           407    0.2027760
## 10     0.010                 3           991    0.2032197
```

The optimal tuning parameters are shrinkage = 0.1, interaction.depth = 1, and number of trees = 346.



```
##                                                var     rel.inf
## perimeter_worst                    perimeter_worst 40.21837597
## concave.points_worst          concave.points_worst 19.78322941
## radius_worst                          radius_worst 15.58026943
## concave.points_mean            concave.points_mean 10.04893701
## area_se                                    area_se  2.85193015
## concavity_worst                    concavity_worst  2.47045849
## texture_worst                        texture_worst  1.93178451
## texture_mean                          texture_mean  1.53632664
## symmetry_worst                      symmetry_worst  1.33116416
## smoothness_worst                  smoothness_worst  1.28057044
## area_worst                              area_worst  0.63620559
## smoothness_se                        smoothness_se  0.54793587
## symmetry_se                            symmetry_se  0.23889690
## symmetry_mean                        symmetry_mean  0.21728977
## compactness_se                      compactness_se  0.20537734
## area_mean                                area_mean  0.18575113
## fractal_dimension_mean   fractal_dimension_mean  0.15476839
## fractal_dimension_se       fractal_dimension_se  0.13989391
## smoothness_mean                    smoothness_mean  0.12713273
## compactness_mean                  compactness_mean  0.11945355
## fractal_dimension_worst fractal_dimension_worst  0.11130950
## texture_se                              texture_se  0.10434790
## radius_se                                radius_se  0.07079057
## compactness_worst                compactness_worst  0.03771637
## concave.points_se                concave.points_se  0.02400415
## concavity_mean                      concavity_mean  0.02105633
## perimeter_se                          perimeter_se  0.02074112
## concavity_se                          concavity_se  0.00428265
## radius_mean                            radius_mean  0.00000000
## perimeter_mean                      perimeter_mean  0.00000000
```

6. Findings

For the Classification Tree, we found that the Misclassification error rate to be approximately $0.1757 = \frac{10}{569}$. We also double checked the CE via LOOCV and created the confusion matrix.

We also found the error rates of OOB, Benign and Malignant using the Classification Forest method. We know that the m corresponding with the lowest OOB Error always changes, so we decided to draw the m that occurs the most in recurrent codes. In this case, we chose $m = 8$ (or $mtry=8) to find the corresponding error rates of OOB, Benign, and Malignant.

For boosting, we found that concave.points_worst, perimeter_worst, concave.points_mean, area_worst, and radius_worst are the most important variables for this study.

We found that there is a strong relationship between breast cancer malignancy and cell parameters. That is, the more malignant the breast cancer is, the more abnormal and aggressive the cancer cells are in terms of their size, shape, and other characteristics. For example, malignant breast cancer cells are larger, irregular in shape, and have more abnormal nuclei compared to normal breast cells. Also, malignant cells may experience increased proliferation and migration, as well as lack of adhesion to other cells, which contributes to the spread of cancer. Overall, the cell parameters of breast cancer cells provide important clues about malignancy of cancer helping us to guide treatment decisions/options.

7. Conclusion and Final Thoughts

On the project as a whole, it should be noted that all the results are obtained using the WBCD database; therefore it is possible that our results are only applicable to the population of patients in the sample. It would have ideal if we could valicate our predictions in other samples that are not used in the training the algorithms. This could be considered as a limitation of our work. In future work, we would like to validate the findings using samples drawn from other settings. Although this study shows accuracy of machine learning methods on data of cell parameters, it should be noted that misclassifications can sometimes be detected from demographic variables. For example, if a patient with a lump has a history of breast cancer in their family, doctors will be more likely to further investigate a benign lump than say, a seemingly healthy person with no history. The reverse side of the argument is not as clean: a misclassified malignancy will be less likely to be disproven, and physicians will be more likely to treat it as soon as possible. This is not necessarily a good thing, though, because many cancer prevention and care methods have less-than-ideal or even very strong negative side effects (e.g., Chemotherapy and radiation can both cause infertility, and radiation therapy can also cause cancer). Hence, prediction accuracy is extremely important.

In summary, this study mainly shows the accuracy of different machine learning methods in using breast cell parameters in predicting the type of breast cancers. These parameters prove to be effective in predicting, but prediction would likely be enhanced through the inclusion of demographic variables such as family medical history, diet, whether the subject regularly exercises, etc. The study emphasizes the need to regularly inspect oneself for irregularities, as it is often these lesions with dense and large cells that prove to be malignant.

Role of Assignment

Jerry Fang: Jerry scraped, cleaned, and preprocessed the Data. Came up with formula for logistics regression as well as classification forests/trees and boosting. Debugged code when necessary. Browsed for information from articles to include in the Conclusion and Final Thoughts section. Formatted Bibliography.

Marcus Murphy: Idea for project, focusing on using ML methods for diagnosing breast cancer. Write-ups, part of presentation.

Changhong Liang: scraping, cleaning and preprocessing the Data, building classification tree, classification forest and boosting tree. Debugging and tuning all the models. Finding relevant essays to supplement the background, findings and conclusion.

References:

Naji, M.A. et. al.(2021). Procedia Computer Science, Volume 191, 2021, Pages 487-492. Machine Learning Algorithms For Breast Cancer Prediction And Diagnosis. https://doi.org/10.1016/j.procs.2021.07.062

"How is Breast Cancer Diagnosed?" Centers for Disease Control and Prevention, 26 Sept. 2022, https://www.cdc.gov/cancer/breast/basic_info/diagnosis.htm

"Breast Cancer Statistics: How Common Is Breast Cancer?" American Cancer Society, https://www.cancer.org/cancer/breast-cancer/about/how-common-is-breast-cancer.html

Sauter, Edward R. "Breast Cancer Prevention: Current Approaches and Future Directions." European Journal of Breast Health, US National Library of Medicine, 1 Apr. 2018, https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5939980/