Breast Cancer

Melih Gündüz/41476216168

2024-05-04

```
install.packages("mlbench")
install.packages("tidymodels")
install.packages("DALEX")
install.packages("rpart.plot")
install.packages("caret")
install.packages("yardstick")
install.packages("AER")
install.packages("randomForest")
library(randomForest)
library(AER)
library(yardstick)
library(caret)
library(tidymodels)
library(DALEX)
library(rpart.plot)
library(mlbench)
```

1. Problem, Features, and Target

The dataset is related to breast cancer diagnosis. It includes features such as Cl.thickness, Cell.shape, Marg.adhesion, Epith.c.size, Bare.nuclei, Bl.cromatin, Normal.nucleoli, Mitoses, and Class. The target variable is Class, which indicates whether the breast cancer diagnosis is benign or malignant. The aim of the analysis is to create models based on the given features to predict the outcome of breast cancer and evaluate the models' performance.

2.Dataset

```
data("BreastCancer")
BreastCancer <- BreastCancer[,-1]
BreastCancer <- na.omit(BreastCancer)
str(BreastCancer)
```

```
'data.frame':   683 obs. of  10 variables:
 $ Cl.thickness   : Ord.factor w/ 10 levels "1"<"2"<"3"<"4"<..: 5 5 3 6 4 8 1 2 2 4 ...
 $ Cell.size      : Ord.factor w/ 10 levels "1"<"2"<"3"<"4"<..: 1 4 1 8 1 10 1 1 1 2 ...
 $ Cell.shape     : Ord.factor w/ 10 levels "1"<"2"<"3"<"4"<..: 1 4 1 8 1 10 1 2 1 1 ...
 $ Marg.adhesion  : Ord.factor w/ 10 levels "1"<"2"<"3"<"4"<..: 1 5 1 1 3 8 1 1 1 1 ...
 $ Epith.c.size   : Ord.factor w/ 10 levels "1"<"2"<"3"<"4"<..: 2 7 2 3 2 7 2 2 2 2 ...
 $ Bare.nuclei    : Factor w/ 10 levels "1","2","3","4",..: 1 10 2 4 1 10 10 1 1 1 ...
 $ Bl.cromatin    : Factor w/ 10 levels "1","2","3","4",..: 3 3 3 3 3 9 3 3 1 2 ...
 $ Normal.nucleoli: Factor w/ 10 levels "1","2","3","4",..: 1 2 1 7 1 7 1 1 1 1 ...
 $ Mitoses        : Factor w/ 9 levels "1","2","3","4",..: 1 1 1 1 1 1 1 1 5 1 ...
 $ Class          : Factor w/ 2 levels "benign","malignant": 1 1 1 1 1 2 1 1 1 1 ...
 - attr(*, "na.action")= 'omit' Named int [1:16] 24 41 140 146 159 165 236 250 276 293 ...
  ..- attr(*, "names")= chr [1:16] "24" "41" "140" "146" ...
```

The dataset has a total of 683 observations and 10 variables. Among these, 5 are ord.factor variables and 5 are factor variables.

3.Splitting The Dataset

```
set.seed(123)
breast_split <- initial_split(data = BreastCancer, prop = 0.80)
breast_train <- breast_split |> training()
breast_test <- breast_split |> testing()
```

It involves understanding the relationships of the parameters of the model in the dataset. It aims to predict the target variable and divides this dataset into two subsets, 'train' and 'test', according to 80% and 20%, respectively.

4.Train a Logistic Regression and A Decision Tree Model

4.1Train a Logistic Regression Model

This code creates a logistic regression model using the 'glm' function. The target variable is used as predictors, and all other variables from the 'train' dataset are used as predictors. We use the binomial distribution as the model's distribution.

```
lr_model <- glm(breast_train$Class~., data=breast_train,
                family = "binomial")
summary(lr_model)
```

```
Call:
glm(formula = breast_train$Class ~ ., family = "binomial", data = breast_train)

Coefficients:
                    Estimate Std. Error z value Pr(>|z|)
(Intercept)          -14.245 235078.226   0.000    1.000
Cl.thickness.L        49.853 207326.237   0.000    1.000
Cl.thickness.Q        12.965 141461.542   0.000    1.000
Cl.thickness.C         4.181 150804.584   0.000    1.000
Cl.thickness^4         4.355 126781.510   0.000    1.000
Cl.thickness^5        24.522 208802.221   0.000    1.000
Cl.thickness^6        16.915 101541.660   0.000    1.000
Cl.thickness^7        -9.489  79315.553   0.000    1.000
Cl.thickness^8       -33.883 117185.241   0.000    1.000
Cl.thickness^9       -25.532 109681.214   0.000    1.000
Cell.size.L           70.190 334555.901   0.000    1.000
Cell.size.Q           -8.089 216146.615   0.000    1.000
Cell.size.C           -7.860 217508.161   0.000    1.000
Cell.size^4           -4.039 398663.104   0.000    1.000
Cell.size^5          -37.165 252845.449   0.000    1.000
Cell.size^6          -39.587 225027.654   0.000    1.000
Cell.size^7          -22.854 248183.806   0.000    1.000
Cell.size^8           29.290 167182.827   0.000    1.000
Cell.size^9           11.390 129376.012   0.000    1.000
Cell.shape.L          18.232 369045.334   0.000    1.000
Cell.shape.Q           5.383 147885.151   0.000    1.000
Cell.shape.C         -12.526 184474.669   0.000    1.000
Cell.shape^4         -30.832 251387.800   0.000    1.000
```

Cell.shape^5	-40.866	227153.767	0.000	1.000
Cell.shape^6	27.056	295677.775	0.000	1.000
Cell.shape^7	7.250	264423.085	0.000	1.000
Cell.shape^8	4.338	175212.721	0.000	1.000
Cell.shape^9	-11.771	183510.617	0.000	1.000
Marg.adhesion.L	37.784	163332.398	0.000	1.000
Marg.adhesion.Q	-7.680	223650.991	0.000	1.000
Marg.adhesion.C	-25.901	237147.166	0.000	1.000
Marg.adhesion^4	-23.192	213838.394	0.000	1.000
Marg.adhesion^5	10.438	360441.563	0.000	1.000
Marg.adhesion^6	45.372	253378.810	0.000	1.000
Marg.adhesion^7	1.309	262610.018	0.000	1.000
Marg.adhesion^8	-50.955	298490.926	0.000	1.000
Marg.adhesion^9	12.331	217857.001	0.000	1.000
Epith.c.size.L	-41.080	405390.602	0.000	1.000
Epith.c.size.Q	-3.561	264824.905	0.000	1.000
Epith.c.size.C	8.409	174954.857	0.000	1.000
Epith.c.size^4	59.312	242846.542	0.000	1.000
Epith.c.size^5	4.495	307423.749	0.000	1.000
Epith.c.size^6	9.832	300709.917	0.000	1.000
Epith.c.size^7	30.889	223324.718	0.000	1.000
Epith.c.size^8	44.962	139984.613	0.000	1.000
Epith.c.size^9	14.394	114027.765	0.000	1.000
Bare.nuclei2	-2.876	206244.059	0.000	1.000
Bare.nuclei3	26.705	110433.032	0.000	1.000
Bare.nuclei4	53.603	137712.792	0.000	1.000
Bare.nuclei5	42.476	34421.028	0.001	0.999
Bare.nuclei6	168.435	419379.930	0.000	1.000
Bare.nuclei7	47.175	185294.339	0.000	1.000
Bare.nuclei8	11.667	165639.576	0.000	1.000
Bare.nuclei9	56.414	378708.564	0.000	1.000
Bare.nuclei10	51.746	99466.691	0.001	1.000
Bl.cromatin2	16.134	139334.613	0.000	1.000
Bl.cromatin3	16.860	133268.230	0.000	1.000
Bl.cromatin4	62.011	132349.546	0.000	1.000
Bl.cromatin5	17.774	162609.494	0.000	1.000
Bl.cromatin6	36.412	264899.799	0.000	1.000
Bl.cromatin7	34.428	166292.130	0.000	1.000
Bl.cromatin8	44.837	288138.235	0.000	1.000
Bl.cromatin9	76.061	413307.228	0.000	1.000
Bl.cromatin10	46.423	184120.049	0.000	1.000
Normal.nucleoli2	3.761	170335.342	0.000	1.000
Normal.nucleoli3	11.737	121603.582	0.000	1.000

```
Normal.nucleoli4     -13.530 121188.678   0.000    1.000
Normal.nucleoli5     -19.565 220850.737   0.000    1.000
Normal.nucleoli6      -2.381 122473.095   0.000    1.000
Normal.nucleoli7     -80.531 250835.375   0.000    1.000
Normal.nucleoli8     -29.811 168511.515   0.000    1.000
Normal.nucleoli9      19.233 308634.611   0.000    1.000
Normal.nucleoli10     30.824 148436.072   0.000    1.000
Mitoses2               2.079 182637.734   0.000    1.000
Mitoses3              14.596 132558.225   0.000    1.000
Mitoses4              49.765 473182.475   0.000    1.000
Mitoses5             -30.285 365358.156   0.000    1.000
Mitoses6            -122.913 355685.123   0.000    1.000
Mitoses7             -47.058 520977.593   0.000    1.000
Mitoses8             -28.539 419355.313   0.000    1.000
Mitoses10             -5.926 408589.429   0.000    1.000

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 7.0812e+02  on 545  degrees of freedom
Residual deviance: 2.2325e-08  on 465  degrees of freedom
AIC: 162

Number of Fisher Scoring iterations: 25
```

The model's fit yielded a null deviance 7.0812e+02 with 545 degrees of freedom, and a residual deviance of 2.2325e-08 with 465 degrees of freedom. The AIC value is 162. AIC allows us to compare the quality of different models.

4.1.1 Logistic Model Performance

4.1.1.a

This predicts the probability of breast cancer occurrence using a logistic regression model (lr_model) with features from the test dataset.

```r
breast_probs <- predict(lr_model, breast_test[,-10],
                        type = "response")
head(breast_probs)
```

```
           1            3            9           18           22           33
2.220446e-16 2.220446e-16 2.220446e-16 2.220446e-16 1.000000e+00 1.000000e+00
```

This involves predicting the probability of breast cancer occurrence for specific observations in the test dataset.

4.1.1.b

This code is used to divide values between 0 and 1 into two categories. Specifically, if a value falls between 0 and 0.5, it is categorized as the probability of having benign breast cancer; if it falls between 0.5 and 1, it is categorized as the probability of having malignant breast cancer.

```
breast_classes <- ifelse(breast_probs>0.5, 1 ,0)
head(breast_classes)
```

```
 1  3  9 18 22 33
 0  0  0  0  1  1
```

The output shows that in some rows, breast cancer is classified as benign, while in other rows, it is classified as malignant.

4.1.1.c

This code is used to evaluate the model's classification performance.

```
confusionMatrix(table(ifelse(breast_test$Class == "malignant", "1", "0"),
                      breast_classes), positive= "1")
```

```
Confusion Matrix and Statistics

   breast_classes
     0  1
  0 88  2
  1  5 42

               Accuracy : 0.9489
                 95% CI : (0.8976, 0.9792)
```

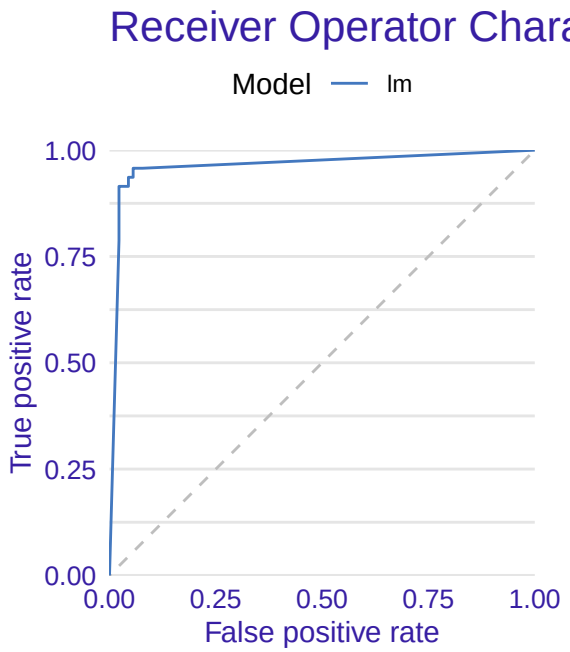
```
    No Information Rate : 0.6788
    P-Value [Acc > NIR] : 8.208e-15

                  Kappa : 0.8849

 Mcnemar's Test P-Value : 0.4497

            Sensitivity : 0.9545
            Specificity : 0.9462
         Pos Pred Value : 0.8936
         Neg Pred Value : 0.9778
             Prevalence : 0.3212
         Detection Rate : 0.3066
   Detection Prevalence : 0.3431
      Balanced Accuracy : 0.9504

       'Positive' Class : 1
```

The results of model evaluation on the test dataset are as follows: Accuracy (0.9489), Kappa (0.8849), McNemar's Test P-Value (0.4497), Sensitivity (0.9545), Specificity (0.9462), Positive Predictive Value (0.8936), Negative Predictive Value (0.9778), Prevalence (0.3212), Detection Rate (0.3066), Detection Prevalence (0.3431), Balanced Accuracy (0.9504). Overall, it performs well in terms of performance.

4.1.2 ROC Curve

This code is used to visualize the ROC Curve graph and to visualize the performance of the logistic regression model.

```r
explain_lr <- explain(model = lr_model,
                      data = breast_test[, -10],
                      y = breast_test$Class == "malignant",
                      type = "classification",
                      verbose = FALSE)
performance_lr <- model_performance(explain_lr)
plot(performance_lr, geom = "roc")
```




```
performance_lr
```

```
Measures for:  classification
recall     : 0.893617
precision  : 0.9545455
f1         : 0.9230769
accuracy   : 0.9489051
auc        : 0.963357

Residuals:
           0%           10%           20%           30%           40%
-1.000000e+00 -2.220446e-16 -2.220446e-16 -2.220446e-16 -2.220446e-16
          50%           60%           70%           80%           90%
-2.220446e-16 -2.220446e-16  2.220446e-16  2.220446e-16  2.220446e-16
         100%
 1.000000e+00
```

Recall (0.893617) is the rate of correctly predicting positives. Precision (0.9545455) is the rate of true positives among predicted positives. F1 (0.9230769) is the harmonic mean of Recall and Precision, summarizing the model's classification performance in a single metric. Accuracy (0.9489051) is the rate of correctly classifying all observations. AUC (0.963357) represents the area under the ROC curve and is used to measure the model's prediction performance. The AUC value is approaching 1, indicating that the model's performance is improving. The

Residuals section shows the residuals of the model's predictions, and generally, the residuals appear to have low values.

4.2.Training Decison Tree

4.2.a

The purpose of this code is to be used for classifying decision trees.

```
dt_model <- decision_tree() |>
set_engine("rpart") |>
set_mode("classification")
```

4.2.b

This code is used to classify using the "breast_train" dataset.

```
dt_breast <- dt_model |>
  fit(Class~., data = breast_train)
dt_breast
```

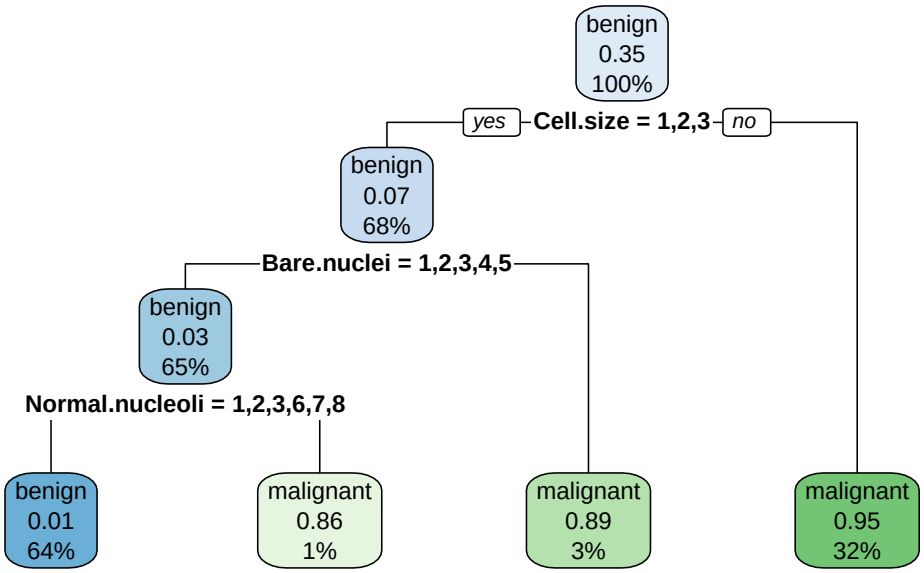
```
parsnip model object

n= 546

node), split, n, loss, yval, (yprob)
      * denotes terminal node

1) root 546 192 benign (0.64835165 0.35164835)
  2) Cell.size=1,2,3 373  27 benign (0.92761394 0.07238606)
    4) Bare.nuclei=1,2,3,4,5 355  11 benign (0.96901408 0.03098592)
      8) Normal.nucleoli=1,2,3,6,7,8 348   5 benign (0.98563218 0.01436782) *
      9) Normal.nucleoli=4,5,10 7   1 malignant (0.14285714 0.85714286) *
    5) Bare.nuclei=7,8,9,10 18   2 malignant (0.11111111 0.88888889) *
  3) Cell.size=4,5,6,7,8,9,10 173   8 malignant (0.04624277 0.95375723) *
```

In this context, column values are classified according to whether they represent benign or malignant breast cancer. This classification has been applied on the Breast_train dataset, which consists of 546 observations. The distinction between benign and malignant classes is observed more frequently, with a higher occurrence of benign classification.

4.2.c

This code has been used to plot the decision tree.

```
rpart.plot(dt_breast$fit)
```



The root node is split based on cell size, with 68% being benign (yes) and 32% being malignant (no). The probability of the root node is 0.35. The sub-node is then split based on Bare.nuclei, with 65% being benign (yes) and 3% being malignant (no). The probability of the sub-node is 0.07. The second sub-node is split based on Normal.nucleoli, with 64% being benign (yes) and 1% being malignant (no). The probability of the second sub-node is 0.03. Leaf nodes are observed as follows: one leaf node has 64% benign cases with a probability of 0.01. The second leaf node is 1% malignant with a probability of 0.86. The third leaf node is 3% malignant with a probability of 0.89. The fourth leaf node is 32% malignant with a probability of 0.95.

4.2.d

In this code, we are reclassifying the decision tree dataset.

```
breast_predictions <- dt_breast |>
  predict(new_data = breast_test)
breast_predictions
```

```
# A tibble: 137 x 1
   .pred_class
   <fct>
 1 benign
 2 benign
 3 benign
 4 benign
 5 malignant
 6 malignant
 7 benign
 8 malignant
 9 malignant
10 benign
# i 127 more rows
```

The new dataset created from Breast_predictions sorts benign and malignant cases in the pred.class column.

4.2.e

This code is used to predict the probabilities of new data in the "breast_test" dataset using the decision tree model.

```
dt_breast |>
predict(new_data = breast_test,
        type = "prob")
```

```
# A tibble: 137 x 2
   .pred_benign .pred_malignant
          <dbl>           <dbl>
 1       0.986          0.0144
 2       0.986          0.0144
 3       0.986          0.0144
 4       0.986          0.0144
 5       0.0462         0.954
```

```
 6       0.0462          0.954
 7       0.986           0.0144
 8       0.0462          0.954
 9       0.0462          0.954
10       0.986           0.0144
# i 127 more rows
```

The probability values of being benign and malignant are provided for each observation.

4.2.1 Decision Model Performance

This code plots a confusion matrix table, showing the values for benign and malignant cases.

```
breast_results <- tibble(predicted=breast_predictions$.pred_class,
                         actual=breast_test$Class)
breast_results|> conf_mat(truth = actual, estimate = predicted)
```

```
           Truth
Prediction  benign malignant
  benign        87         3
  malignant      3        44
```

This model shows the performance of correctly and incorrectly predicting benign and malignant cases. It generally predicts benign cases correctly as benign but sometimes predicts them as malignant. Similarly, it generally predicts malignant cases correctly as malignant but sometimes predicts them as benign.

4.2.1.a

This code is used to calculate the accuracy value. It is the ratio of the cases that we predicted in the model to all cases.

```
breast_results |> accuracy(truth = actual, estimate = predicted)
```

```
# A tibble: 1 x 3
  .metric  .estimator .estimate
  <chr>    <chr>          <dbl>
1 accuracy binary         0.956
```

The accuracy value (0.9562044) is quite good.

4.2.1.b

This code is used to calculate the sensitivity value.

```
breast_results |> sens(truth = actual, estimate = predicted)
```

```
# A tibble: 1 x 3
  .metric .estimator .estimate
  <chr>   <chr>          <dbl>
1 sens    binary         0.967
```

The sensitivity value (0.9666667) is quite good.

4.2.1.c

This code is used to calculate the F-measure value.

```
breast_results |> f_meas(truth = actual, estimate = predicted)
```

```
# A tibble: 1 x 3
  .metric .estimator .estimate
  <chr>   <chr>          <dbl>
1 f_meas  binary         0.967
```

The F-measure value (0.9666667) is quite good.

5.The Overfitting Problem

This code collects the necessary information to measure the performance of the decision tree model in the dataset.

```
breastfit <- dt_model |>
  last_fit(Class ~., split = breast_split)
breastfit |> collect_metrics()
```

```
# A tibble: 3 x 4
  .metric     .estimator .estimate .config
  <chr>       <chr>          <dbl> <chr>
1 accuracy    binary        0.956  Preprocessor1_Model1
2 roc_auc     binary        0.949  Preprocessor1_Model1
3 brier_class binary        0.0426 Preprocessor1_Model1
```

The accuracy value (0.95620438) has turned out to be quite good. The ROC AUC value (0.94893617) has also turned out to be quite good. The Brier classification value (0.04264961) has been obtained. Overall, the classification performance of this model has turned out to be quite good.

5.a

This code collects predictions on the dataset from the model and evaluates the performance of the model.

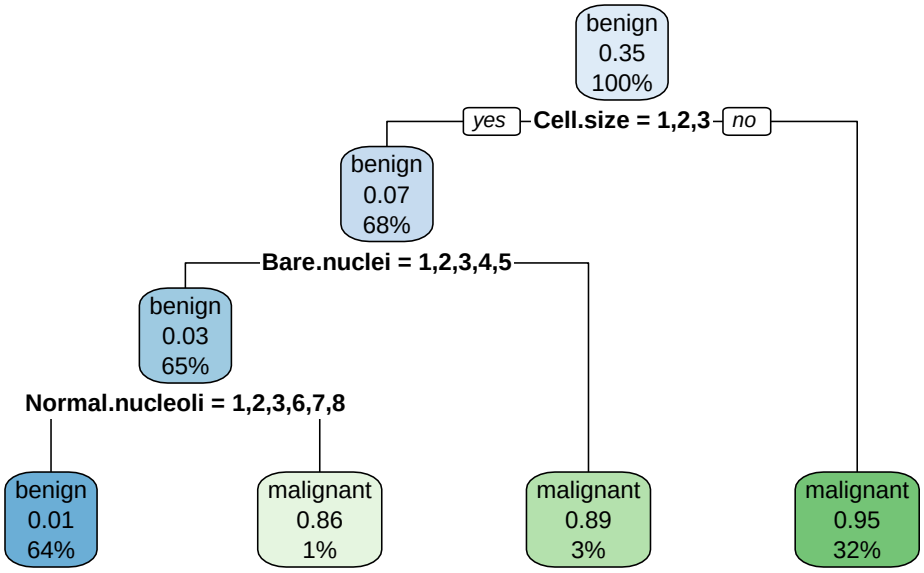
```
breastfit |> collect_predictions()
```

```
# A tibble: 137 x 7
   .pred_class .pred_benign .pred_malignant id               .row Class .config
   <fct>              <dbl>           <dbl> <chr>           <int> <fct> <chr>
 1 benign            0.986          0.0144 train/test split     1 beni~ Prepro~
 2 benign            0.986          0.0144 train/test split     3 beni~ Prepro~
 3 benign            0.986          0.0144 train/test split     9 beni~ Prepro~
 4 benign            0.986          0.0144 train/test split    18 beni~ Prepro~
 5 malignant         0.0462         0.954  train/test split    22 mali~ Prepro~
 6 malignant         0.0462         0.954  train/test split    32 mali~ Prepro~
 7 benign            0.986          0.0144 train/test split    35 beni~ Prepro~
 8 malignant         0.0462         0.954  train/test split    42 mali~ Prepro~
 9 malignant         0.0462         0.954  train/test split    43 mali~ Prepro~
10 benign            0.986          0.0144 train/test split    44 beni~ Prepro~
# i 127 more rows
```

The probabilities of the classes, namely benign and malignant, are provided.

5.b

```
breast_dt <- rpart(Class ~ ., data = breast_train,
                   method = "class")
rpart.plot(breast_dt)
```

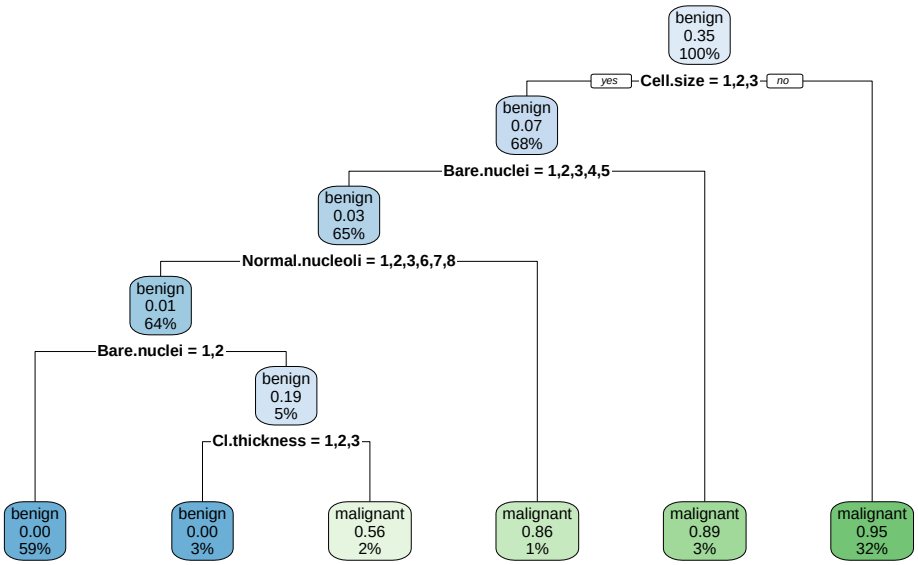


The root node is split based on cell size, with 68% being benign (yes) and 32% being malignant (no). The probability of the root node is 0.35. The sub-node is then split based on Bare.nuclei, with 65% being benign (yes) and 3% being malignant (no). The probability of the sub-node is 0.07. The second sub-node is split based on Normal.nucleoli, with 64% being benign (yes) and 1% being malignant (no). The probability of the second sub-node is 0.03. Leaf nodes are observed as follows: one leaf node has 64% benign cases with a probability of 0.01. The second leaf node is 1% malignant with a probability of 0.86. The third leaf node is 3% malignant with a probability of 0.89. The fourth leaf node is 32% malignant with a probability of 0.95.

5.c

This code demonstrates the effects of parameters in the decision tree.

```
less_dt <- rpart(Class ~ ., data = breast_train,
                 method = "class",
                 maxdepth =30,
                 cp = 0.00000001)
rpart.plot(less_dt)
```



When we look at the 5th decision tree, it appears different. The reason is that there is more branching, and as a result, the probabilities and details are more visible. As we decrease the value of the cp parameter and increase the value of the maxdepth parameter, the branching of the decision tree increases.

6. The Imbalancedness Problems

This code checks for imbalance.

```
table(breast_train$Class)/dim(breast_train) [1]
```

```
   benign malignant
0.6483516 0.3516484
```

The probability of being benign is 0.6483516, and the probability of being malignant is 0.3516484. The main goal here is to address the imbalance by balancing these values, aiming for them to be close. Therefore, a randomForest is performed below to address the imbalance.

6.a

This code is used to balance the imbalance.

```r
imbalanced <- randomForest(Class ~ ., data = breast_train,
                           type="classification")
imb_predic <- predict(imbalanced, breast_train)
imbalanced
```

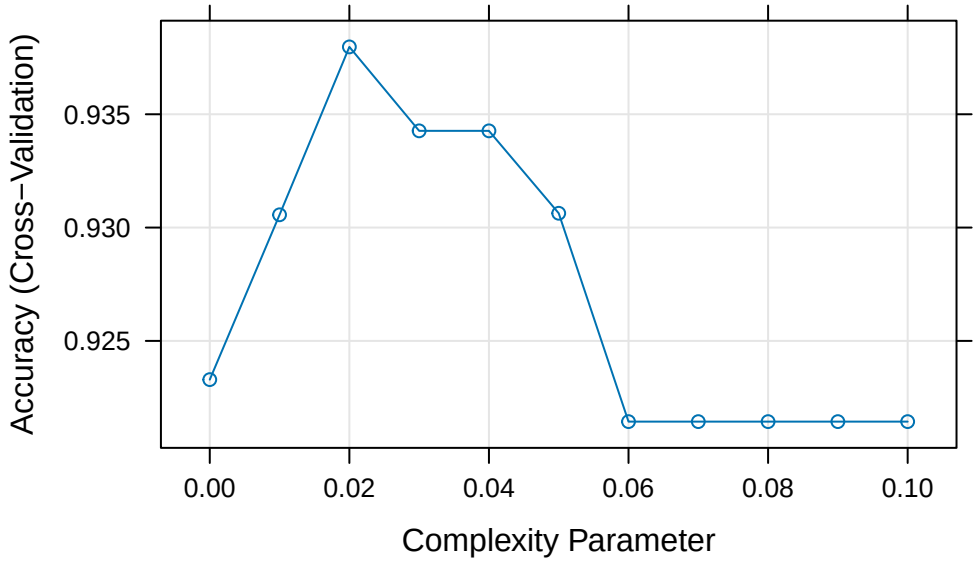
```
Call:
 randomForest(formula = Class ~ ., data = breast_train, type = "classification")
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 3

        OOB estimate of  error rate: 2.2%
Confusion matrix:
          benign malignant class.error
benign       345         9  0.02542373
malignant      3       189  0.01562500
```

The error rate of 2.75% has been estimated in the Random Forest model. The goal here is to classify benign and malignant cases correctly. The error rate for benign cases is 0.03107345, and for malignant cases, it is 0.02083333.

7. Improve The Prediction Performance Of The Decision Tree Model Tuning Hyperparameters (Grid Search in Caret)

This code is used to visually select the best hyperparameter value in grid search.

```r
fit_control <- trainControl(method = "cv", number = 10)
hyp_dt_model <- train(Class ~ .,
                  data = breast_train,
                  method = "rpart",
                  trControl = fit_control,
                  tuneGrid = expand.grid(cp = seq(0, 0.1, 0.01)))
plot(hyp_dt_model)
```

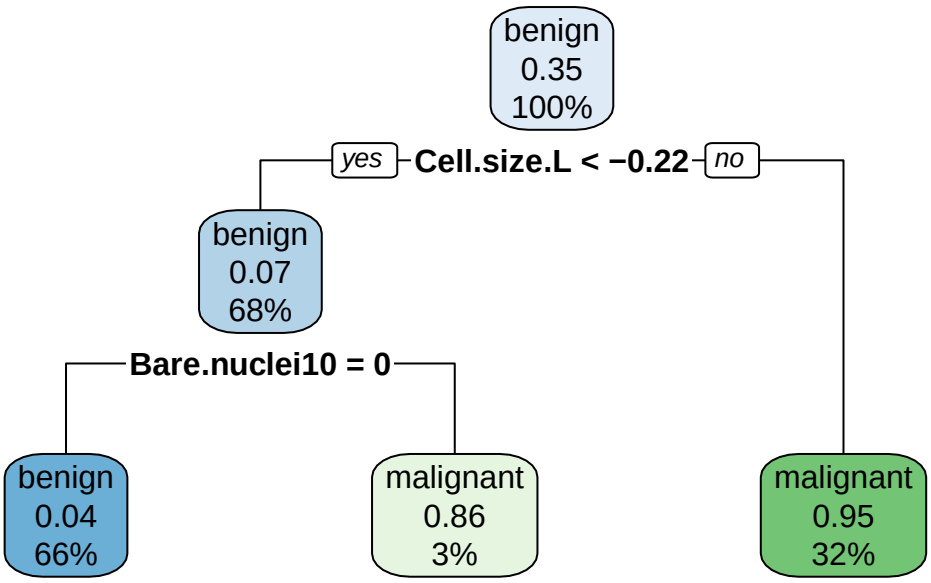


The goal in the graph is to select the highest accuracy value, which corresponds to the elbow point. Therefore, we can choose the observation point at 0.935.

7.a

The code is used to plot a decision tree.

```
rpart.plot(hyp_dt_model$finalModel)
```



The decision tree model built based on the graph above has resulted in a moderate level of branching, navigating between benign and malignant conditions based on specific criteria.