Breast Cancer Prediction Analysis

Michael Adekola

Executive Summary

In this report, my aim is to analyse breast cancer which is a common disease that usually occurs in women over the age of 50. Young women can also be diagnosed with breast cancer, and rarely males are too. By using R-studio, I explored the breast dataset and designed a decision tree model and random forest model which both predict the existence of breast cancer in a patient. This is done by preparing the dataset which involved loading the relevant packages into RStudio and cleaning the data. In the exploration phase, I explored the missing values and decided the appropriate method to impute them. The dataset was randomly

sampled the data by splitting 70% into a training set and 30% into a test set. This allowed for the creation of our models using the training set. An evaluation of the models is described in terms of a confusion matrix, which outlines the metrics of accuracy, recall and precision. In terms of these metrics, it can be concluded that:

- Our decision tree is more accurate than the random forest
- Both models predicted the positive results well based on recall
- The decision tree was more precise than the random forest

Table of Contents

Preparing the Data

The **breast-cancer.csv** spreadsheet contains historical data which will be used for conducting analysis and developing our model. This dataset includes a variety of attributes related to patients and ultrasonic imaging; a total of 286 observations and 10 variables.

Step 1: Load the Data File

At first, the following packages were installed to assist with R coding later on in analysis; 'PerformanceAnalytics', 'dplyr', 'ggplot2', 'gridExtra', 'arm', 'car', 'psych', 'caTools', 'caret', 'mice', 'VIM', 'rpart', 'rattle', and 'randomForest' *(Refer to appendix 1 for a brief description of each)*. With the column names unidentified, it was important to add a header label after loading the dataset into R Studio. This will determine the output and give certainty to the data being analysed. Using the code below, the dataset was imported, with the variable list titled. Each variable is detailed in *Table 1*. A description of each column is depicted in *Table 2*.

canc <- read.csv(file.choose(), header = F ,col.names = c("class","age", "menopause", "tumor.size", "inv.nodes", "node.caps", "deg.malig", "breast", "breast.quad","irradiat"),colClasses = "factor", na.strings = c("?"))

	class	age	menopause	tumor.size	inv.nodes	node.caps	deg.malig	breast	breast.quad	irradiat
1	no-recurrence-events	30-39	premeno	30-34	0-2	no	3	left	left_low	no
2	no-recurrence-events	40-49	premeno	20-24	0-2	no	2	right	right_up	no
3	no-recurrence-events	40-49	premeno	20-24	0-2	no	2	left	left_low	no
4	no-recurrence-events	60-69	ge40	15-19	0-2	no	2	right	left_up	no

Table 1: Brief header of dataset

Column	Attribute header	Description
1	class	The 'class' column shows that there are 2 types of events presented, recurrence events and no-recurrence events. There are more no-recurrence cases than recurrence cases in the dataset presented.
2	age	The 'age' column shows the age of the patient at the time of diagnosis.
3	menopause	The 'menopause' column indicates whether the patient is premenopausal or postmenopausal at the time of diagnosis. There are 3 types of ge40, itg40 and premeno.
4	tumor.size	The size of the tumor is usually measured in millimetres (mm).
5	inv.nodes	The lymph nodes that contain metastatic breast cancer are visible when examined. The number range is (0 - 39).

6	node.caps	This column details whether the cancer contains metastasise to a lymph node. It means it has spread to the tumor to the lymph node. The column shows 2 characters, yes or no.
7	deg.malig	The 'Degree of malignancy' has a range of 1-3 for the tumor. Grade 1 tumors are normal cells. Grade 2 tumors don't look like normal tumor cells and they grow faster. Grade 3 tumors look abnormal; they grow abnormally fast.
8	breast	The 'breast' column contains 2 characters, left or right. This is due to breast cancer usually occurring on the left or right side of the breast.
9	breast.quad	The 'breast.quad' column details the quadrants of how a breast is divided. These four quadrants are left-up, left-low, right-up, right-low, central.
10	irradiat	Radiation therapy treatment uses high doses of radiation to destroy cancer cells and shrink tumors. Thus, the 'irradiat' column contains 2 characters, yes or no.

Table 2: Description of each column

Next, the following codes were used to summarise the results and define the regular expressions of each variable presented. The return of these codes can be viewed in *Appendix 2*.

```
names(canc)
str(canc)
head(canc)
summary(canc)
```

Exploring the Data

Before cleaning the data, it was also important to determine any irregularities within. Quickly it was observed that the dataset had no headers or names, it also had the missing values (NA's) coded as question mark "?". Lastly, as this is a classification problem, all the variables including the target class have to be a factor. Without writing multiple lines of code, we decided to resolve these issues while loading the dataset into Rstudio with the code below.

```
canc <- read.csv(file.choose(), header = F ,col.names = c("class","age", "menopause",
"tumor.size", "inv.nodes", "node.caps", "deg.malig", "breast",
"breast.quad","irradiat"),colClasses = "factor", na.strings = c("?"))
```

Basic analysis was then conducted using the ggplot2 library to make comparisons between the independent variables and the target variable. From the below graphs, seen in *Figure 1*, it can be observed that the 'age' group and 'breast' quadrant seem to follow normal distributions. Moreover, when we see the histogram of 'inv.nodes', 'class' variables occur the most with fewer axillary lymph nodes (0-2). Looking at the 'Degree of Malignancy' histogram, we can see that the higher degree of malignancy is, the more recurrence-events

increases but there is a fluctuation of no-recurrence-events, it is most frequent at 2, then 1, and then 3.

RStudio Plot Display

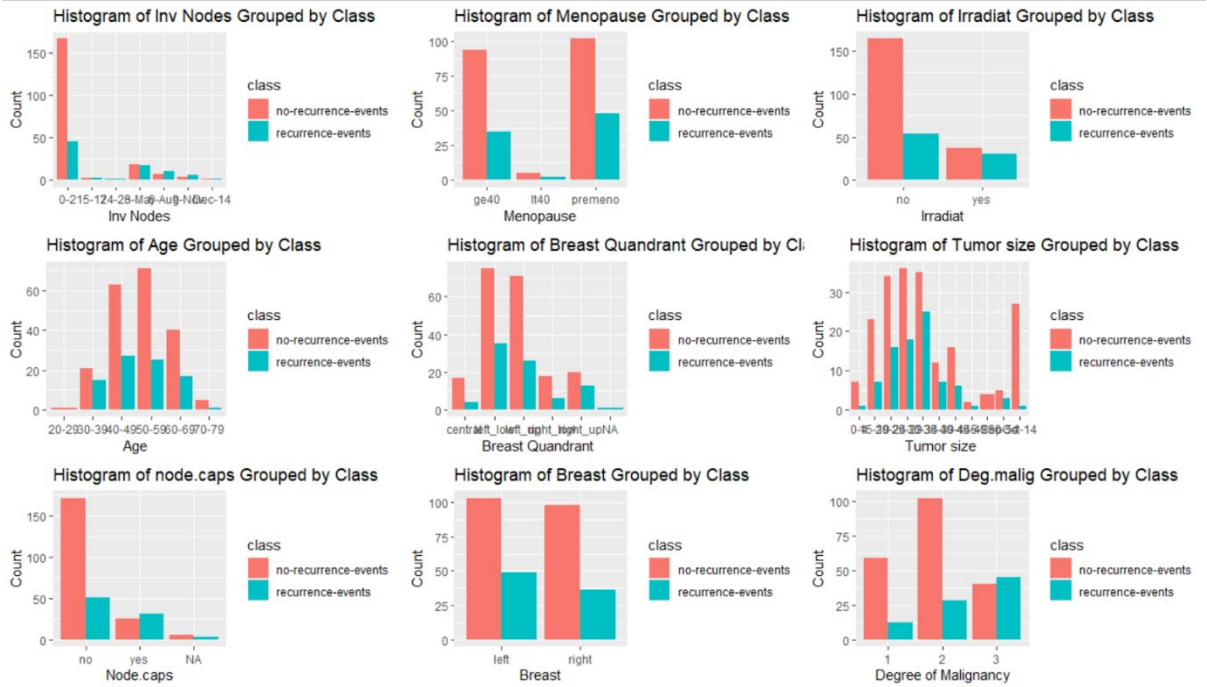

Figure 1: RStudio Grid Graph

Step 2: Identify Missing Values

With the raw data presented being only 286 observations, it was decided not to omit any values and impute values that can have more predictive power for the objective. If the missing values were omitted, it can introduce bias in our model. We viewed the number of NA values in each column using the *is.na* function, as seen in *Figure 2*.

```
> colSums(is.na(canc))
      class         age   menopause  tumor.size   inv.nodes   node.caps   deg.malig      breast
          0           0           0           0           0           8           0           0
breast.quad    irradiat
          1           0
> canc[!complete.cases(canc),]
                    class   age menopause tumor.size inv.nodes node.caps deg.malig breast breast.quad
146 no-recurrence-events 40-49   premeno      25-29       0-2       <NA>         2   left   right_low
164 no-recurrence-events 60-69      ge40      25-29      3-May       <NA>         1  right    left_up
165 no-recurrence-events 60-69      ge40      25-29      3-May       <NA>         1  right   left_low
184 no-recurrence-events 50-59      ge40      30-34      9-Nov       <NA>         3   left    left_up
185 no-recurrence-events 50-59      ge40      30-34      9-Nov       <NA>         3   left   left_low
207    recurrence-events 50-59      ge40      30-34       0-2         no          3   left       <NA>
234    recurrence-events 70-79      ge40      15-19      9-Nov       <NA>         1   left   left_low
264    recurrence-events 50-59      lt40      20-24       0-2       <NA>         1   left    left_up
265    recurrence-events 50-59      lt40      20-24       0-2       <NA>         1   left   left_low
    irradiat
146      yes
164      yes
165      yes
184      yes
185      yes
207       no
234      yes
264       no
265       no
> md.pattern(canc)
    class age menopause tumor.size inv.nodes deg.malig breast irradiat breast.quad node.caps
277     1   1         1          1         1         1      1        1           1         1 0
8       1   1         1          1         1         1      1        1           1         0 1
1       1   1         1          1         1         1      1        1           0         1 1
        0   0         0          0         0         0      0        0           1         8 9
```

Figure 2: is.na function to view NA values

From above, we can see that node caps have 8 missing values and breast quadrant has 1 missing value. The missing data proportion table, as seen in *Table 1*, shows exactly where the missing values are located and their corresponding row numbers.

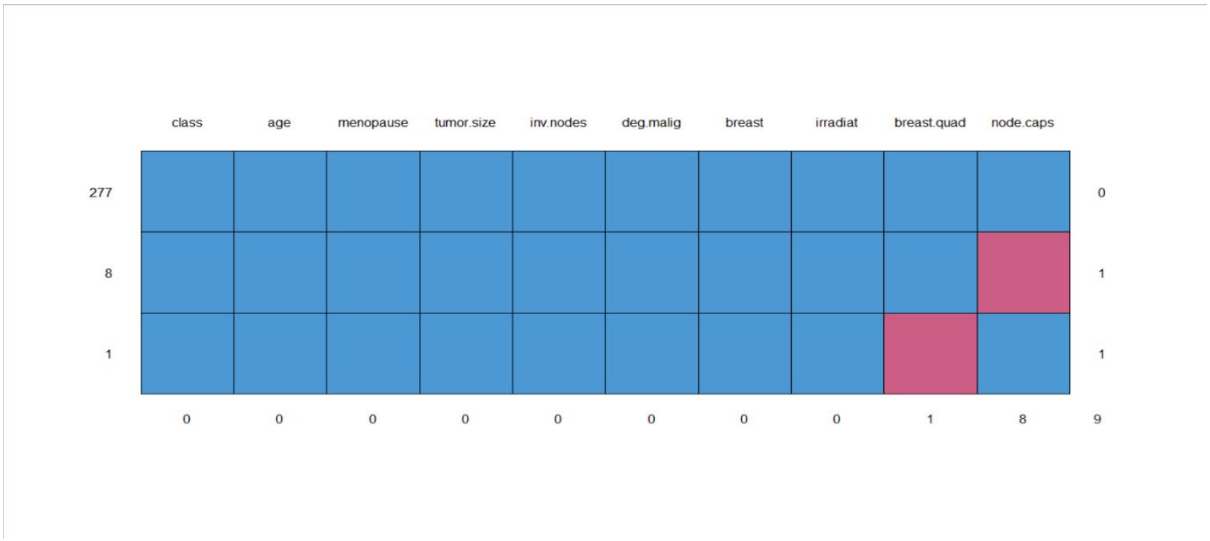

Table 1: Missing Data Proportion Table

From this, it can be seen that there are 277 rows with no missing numbers, 8 rows with missing data points and they are from node caps and 1 row with missing data points from the breast quadrant. The proportion of missing data was calculated for each variable below. The node cap has the highest proportion of missing data at 2.8 percent while the breast quadrant is 0.35 percent.

```
> prop <- function(x) {sum(is.na(x))/ length(x)*100}
> apply(canc,2,prop)
      class         age   menopause  tumor.size   inv.nodes   node.caps   deg.malig      breast
  0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   2.7972028   0.0000000   0.0000000
breast.quad    irradiat
  0.3496503   0.0000000
```

We also decided to not impute the missing values with mean and median values as it is not a very smart way, and variables like menopause cannot be averaged. The ‘mice’ package used logistic regression and polynomial regression algorithms to estimate missing values and resolve uncertainty for the node.caps and breast.quad respectively (KDNugget, 2020).

```
> print(impute)
Class: mids
Number of multiple imputations:  3
Imputation methods:
      class         age   menopause  tumor.size   inv.nodes   node.caps   deg.malig      breast breast.quad
         ""          ""          ""          ""          ""    "logreg"          ""          ""    "polyreg"
   irradiat
         ""
```

The Mice algorithm suggested three estimates for the missing values and after carefully observing the estimates against other variables in the missing rows, the third estimate was best fitted so we imputed the missing values with our new estimates. As seen below, there are no more missing values, making sure the data is prepared so there are no bias estimates and invalid conclusions before analysis.

```
> cancer_clean <- complete(impute, 3)
> colSums(is.na(cancer_clean))
      class         age   menopause  tumor.size   inv.nodes   node.caps   deg.malig      breast breast.quad
          0           0           0           0           0           0           0           0           0
  irradiat
          0
```

Sampling Data

Data sampling is part of the data mining process, where a representative sample of the data is created. Once the data has been prepared, it is sampled by splitting the data. 70% is split into a training set and 30% into a test set. The training set is used to create models, whilst the test set is used to check the model is correct. The training set is much larger than the test to allow for the model to be trained on a larger amount of data. This allows the model to be tested on a greater variety of data constellations (Rohrich 2020).

Step 1: Randomly Select 70% of the Dataset as Training Data

```
> set.seed(777)
> train.index <- sample(1:nrow(cancer_clean), 0.7* nrow(cancer_clean))
> print(sort(train.index))
  [1]   1   2   4   5   6   7   9  10  11  12  13  14  15  16  18  19  20  21  22  23  24  26  27  30  31  33  34  35  36
 [30]  37  38  39  41  42  45  46  47  50  51  53  54  56  57  58  59  60  61  63  64  65  66  68  72  73  74  78  79  80
 [59]  81  82  83  85  88  89  90  91  93  96  97  98  99 100 102 104 107 109 110 111 113 114 115 116 117 118 119 120 121
 [88] 124 125 126 127 128 129 131 133 134 136 137 139 140 142 143 144 147 148 149 150 151 153 154 155 156 157 160 161 162
[117] 163 165 167 169 170 171 173 174 175 176 177 178 181 182 183 185 188 189 190 191 192 194 195 196 198 200 201 202 205
[146] 206 208 210 212 214 215 216 219 220 222 223 224 225 226 227 229 233 234 235 236 237 239 242 243 244 245 246 248 249
[175] 250 251 252 253 254 257 258 259 260 262 263 264 265 266 268 269 270 271 273 277 278 279 280 282 283 284
> cancer.train <- cancer_clean[train.index,]
> dim(cancer.train)
[1] 200  10
```

The dim() function demonstrates that the train data contains 200 data points from 10 variables, which is 70% of total observations.

Step 2: Select the 30% Left as the Testing Data

```
> cancer.test <- cancer_clean[-train.index,]
> dim(cancer.test)
[1] 86 10
```

The dim() function demonstrates that the train data contains 86 data points from 10 variables, which is 30% of total observations.

Step 3: Check Proportion of Train & Test Data

The prop.table() function is used to check for the proportion of labels in both the training and test split. This ensures the data is not imbalanced.

```
> prop.table(table(cancer.test$class))

no-recurrence-events    recurrence-events
           0.6744186            0.3255814
```

Building the Model

The data model will develop a clear visual and summary of the data set. The data model explores data-orientated structures by organising elements of data and standardising how they relate to each other and real-world entity properties (Cole, 2020). A decision tree model and random forest model establish the relationship between these variables.

Step 1: Set Seed for Randomisation

```
> set.seed(777)
```

The set.seed function is executed to ensure that every time the code is run, the same result is consistently produced.

Step 2: Build the Decision Tree Model

```
cancer.tree <- rpart(class ~., data = cancer.train, method = "class")
print(cancer.tree)
summary(cancer.tree)
fancyRpartPlot(cancer.tree, caption = NULL)
```

2.1) To specify the decision tree model, we began by executing the following code:

```
> cancer.tree <- rpart(class ~., data = cancer.train, method = "class")
```

This line specifies the model to use the cancer.train dataset and the method to use the variable *class*, which is either a recurrent or non-recurrent event.

2.2) We then used the print() function to view a description of the cancer.tree dataset. This function provides details of the breast cancer decision tree.

```
> print(cancer.tree)
n= 200

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

 1) root 200 57 no-recurrence-events (0.7150000 0.2850000)
   2) deg.malig=1,2 139 25 no-recurrence-events (0.8201439 0.1798561)
     4) tumor.size=10-14,45-49,5-9 20  0 no-recurrence-events (1.0000000 0.0000000) *
     5) tumor.size=0-4,15-19,20-24,25-29,30-34,35-39,40-44,50-54 119 25 no-recurrence-events (0.7899160 0.2100840)
      10) age=20-29,40-49,50-59,60-69 98 17 no-recurrence-events (0.8265306 0.1734694) *
      11) age=30-39,70-79 21  8 no-recurrence-events (0.6190476 0.3809524)
        22) tumor.size=0-4,20-24,25-29,40-44 13  3 no-recurrence-events (0.7692308 0.2307692) *
        23) tumor.size=15-19,30-34 8  3 recurrence-events (0.3750000 0.6250000) *
   3) deg.malig=3 61 29 recurrence-events (0.4754098 0.5245902)
     6) inv.nodes=0-2 34 11 no-recurrence-events (0.6764706 0.3235294)
      12) age=40-49,50-59,70-79 19  2 no-recurrence-events (0.8947368 0.1052632) *
      13) age=30-39,60-69 15  6 recurrence-events (0.4000000 0.6000000) *
     7) inv.nodes=12-14,15-17,24-26,3-5,6-8,9-11 27  6 recurrence-events (0.2222222 0.7777778) *
```

2.3) We used the summary() function to summarise each variable in the breast cancer dataset.

```
> summary(cancer.tree)
```

The summary is displayed in *Appendix 3*.

2.4) Lastly, we visualised the decision tree model using the fancyRpartPlot() function. This function visualises the cancer.tree object, depicted in *Figure 3*.

```
> fancyRpartPlot(cancer.tree, caption = NULL)
```

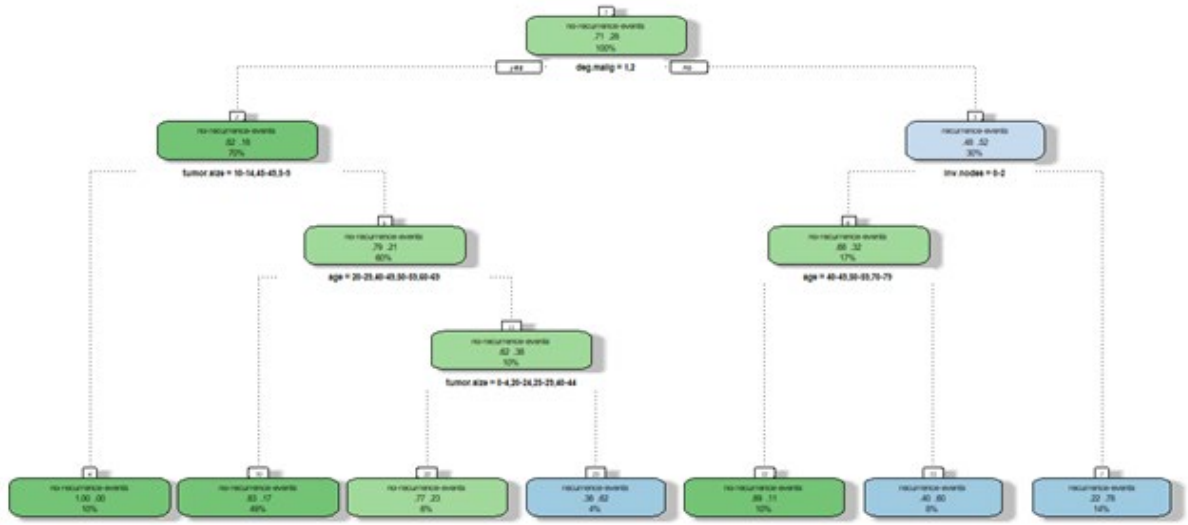

Figure 3: Decision tree model

The root node has a majority class of no-recurrence-events and the probability for no-recurrence-event in the root node is 0.71. The node asks if the degree of malignancy is 1 or 2? If yes then go down to the root's left child, 70 percent are no-recurrence-event and the probability is 0.82 and keep going down the nodes. Our model shows that the most important questions to ask are degree of malignancy, inv-nodes and tumor size.

```
> summary(cancer.tree)
Call:
rpart(formula = class ~ ., data = cancer.train, method = "class")
  n= 200

          CP nsplit rel error    xerror      xstd
1 0.13157895      0 1.0000000 1.0000000 0.1119994
2 0.05263158      2 0.7368421 0.8947368 0.1081405
3 0.01169591      3 0.6842105 0.9473684 0.1101497
4 0.01000000      6 0.6491228 0.8421053 0.1059626

Variable importance
  deg.malig   inv.nodes         age   node.caps  tumor.size breast.quad    irradiat   menopause
         26          21          16          12          11           7           5           1
```

Step 3: Interpreting the Decision Tree Model

3.1) *Predicting the model*

I made predictions using the decision model through the predict() function. This function was used to predict the cancer.tree object. By using "class" as the type, the class with the highest probability will be returned.

```
> cancer.predictions <- predict(cancer.tree, cancer.test, type = "class")
> head(cancer.predictions)
                   3                    8                   17                   25                   28
no-recurrence-events no-recurrence-events no-recurrence-events no-recurrence-events    recurrence-events
                  29
no-recurrence-events
Levels: no-recurrence-events recurrence-events
```

3.2) Comparison table

We used a comparison table to generate a comparison between the predicted class and actual class.

```
> cancer.comparison <- cancer.test
> cancer.comparison$predictions <- cancer.predictions
> cancer.comparison[,c("class", "predictions")]
                    class          predictions
3   no-recurrence-events no-recurrence-events
8   no-recurrence-events no-recurrence-events
17  no-recurrence-events no-recurrence-events
25  no-recurrence-events no-recurrence-events
28  no-recurrence-events    recurrence-events
29  no-recurrence-events no-recurrence-events
32  no-recurrence-events no-recurrence-events
40  no-recurrence-events no-recurrence-events
43  no-recurrence-events no-recurrence-events
44  no-recurrence-events no-recurrence-events
48  no-recurrence-events no-recurrence-events
49  no-recurrence-events no-recurrence-events
52  no-recurrence-events no-recurrence-events
55  no-recurrence-events no-recurrence-events
62  no-recurrence-events no-recurrence-events
67  no-recurrence-events no-recurrence-events
69  no-recurrence-events no-recurrence-events
70  no-recurrence-events no-recurrence-events
71  no-recurrence-events no-recurrence-events
75  no-recurrence-events no-recurrence-events
76  no-recurrence-events no-recurrence-events
77  no-recurrence-events no-recurrence-events
84  no-recurrence-events no-recurrence-events
86  no-recurrence-events no-recurrence-events
87  no-recurrence-events no-recurrence-events
92  no-recurrence-events no-recurrence-events
94  no-recurrence-events no-recurrence-events
95  no-recurrence-events no-recurrence-events
101 no-recurrence-events no-recurrence-events
103 no-recurrence-events no-recurrence-events
105 no-recurrence-events no-recurrence-events
106 no-recurrence-events no-recurrence-events
108 no-recurrence-events no-recurrence-events
112 no-recurrence-events no-recurrence-events
122 no-recurrence-events no-recurrence-events
123 no-recurrence-events no-recurrence-events
130 no-recurrence-events no-recurrence-events
132 no-recurrence-events    recurrence-events
135 no-recurrence-events no-recurrence-events
138 no-recurrence-events no-recurrence-events
141 no-recurrence-events no-recurrence-events
145 no-recurrence-events    recurrence-events
146 no-recurrence-events no-recurrence-events
152 no-recurrence-events    recurrence-events
158 no-recurrence-events no-recurrence-events
159 no-recurrence-events no-recurrence-events
164 no-recurrence-events no-recurrence-events
166 no-recurrence-events no-recurrence-events
168 no-recurrence-events no-recurrence-events
172 no-recurrence-events    recurrence-events
179 no-recurrence-events no-recurrence-events
180 no-recurrence-events no-recurrence-events
184 no-recurrence-events    recurrence-events
186 no-recurrence-events no-recurrence-events
187 no-recurrence-events no-recurrence-events
193 no-recurrence-events no-recurrence-events
197 no-recurrence-events no-recurrence-events
199 no-recurrence-events no-recurrence-events
203    recurrence-events no-recurrence-events
204    recurrence-events no-recurrence-events
207    recurrence-events no-recurrence-events
209    recurrence-events no-recurrence-events
211    recurrence-events no-recurrence-events
213    recurrence-events no-recurrence-events
217    recurrence-events no-recurrence-events
218    recurrence-events no-recurrence-events
221    recurrence-events no-recurrence-events
228    recurrence-events no-recurrence-events
230    recurrence-events no-recurrence-events
231    recurrence-events no-recurrence-events
232    recurrence-events no-recurrence-events
238    recurrence-events no-recurrence-events
240    recurrence-events no-recurrence-events
241    recurrence-events no-recurrence-events
247    recurrence-events    recurrence-events
255    recurrence-events no-recurrence-events
256    recurrence-events    recurrence-events
261    recurrence-events no-recurrence-events
267    recurrence-events no-recurrence-events
272    recurrence-events no-recurrence-events
274    recurrence-events no-recurrence-events
275    recurrence-events no-recurrence-events
276    recurrence-events    recurrence-events
281    recurrence-events    recurrence-events
285    recurrence-events    recurrence-events
286    recurrence-events    recurrence-events
```

We can see here that there are 28 observations which were incorrectly predicted.

3.3) View misclassified rows

I viewed the details of the 28 incorrectly predicted observations. I did this by creating an object called *disagreement.index* which stores observations where the actual class is not the same as the predicted class. This was executed using the following statement:

disagreement.index <- cancer.comparison$class != cancer.comparison$predictions

I then generated a comparison table of the object *disagreement.index* to view the attributes of the incorrectly predicted observations using the following statement:

cancer.comparison[disagreement.index,]

```
> disagreement.index <- cancer.comparison$class != cancer.comparison$predictions
> cancer.comparison[disagreement.index,]
                class   age menopause tumor.size inv.nodes node.caps deg.malig breast breast.quad irradiat          predictions
28  no-recurrence-events 60-69      ge40      25-29      0-2        no         3  right      left_up       no    recurrence-events
132 no-recurrence-events 40-49   premeno      40-44      3-5       yes         3  right      left_up      yes    recurrence-events
145 no-recurrence-events 60-69      ge40      45-49      6-8       yes         3   left      central       no    recurrence-events
152 no-recurrence-events 60-69      ge40      30-34      3-5       yes         3   left     left_low       no    recurrence-events
172 no-recurrence-events 30-39   premeno      15-19      0-2        no         1   left     left_low       no    recurrence-events
184 no-recurrence-events 50-59      ge40      30-34     9-11       yes         3   left      left_up      yes    recurrence-events
203    recurrence-events 40-49   premeno      40-44      0-2        no         1   left     left_low       no no-recurrence-events
204    recurrence-events 50-59      ge40      35-39      0-2        no         2   left     left_low       no no-recurrence-events
207    recurrence-events 50-59      ge40      30-34      0-2        no         3   left     right_up       no no-recurrence-events
209    recurrence-events 50-59   premeno      30-34      0-2        no         3   left     right_up       no no-recurrence-events
211    recurrence-events 40-49   premeno      20-24      0-2        no         2   left     left_low       no no-recurrence-events
213    recurrence-events 40-49   premeno      30-34      0-2        no         3  right     right_up       no no-recurrence-events
217    recurrence-events 50-59      ge40      20-24      0-2        no         2   left      left_up       no no-recurrence-events
218    recurrence-events 40-49   premeno      15-19      0-2        no         2   left      left_up       no no-recurrence-events
221    recurrence-events 40-49   premeno      25-29      0-2        no         3   left     right_up       no no-recurrence-events
228    recurrence-events 50-59   premeno      30-34      0-2        no         3  right      left_up      yes no-recurrence-events
230    recurrence-events 60-69      ge40      45-49      0-2        no         1  right     right_up      yes no-recurrence-events
231    recurrence-events 50-59   premeno      50-54     9-11       yes         2  right      left_up       no no-recurrence-events
232    recurrence-events 40-49   premeno      30-34      3-5        no         2  right      left_up       no no-recurrence-events
238    recurrence-events 40-49   premeno      25-29      0-2        no         2  right     left_low       no no-recurrence-events
240    recurrence-events 40-49   premeno      20-24      3-5       yes         2  right     right_up      yes no-recurrence-events
241    recurrence-events 60-69      ge40      20-24      3-5        no         2   left     left_low      yes no-recurrence-events
255    recurrence-events 40-49   premeno      30-34      0-2       yes         3  right     right_up       no no-recurrence-events
261    recurrence-events 60-69      ge40      25-29      3-5        no         2  right     right_up       no no-recurrence-events
267    recurrence-events 40-49   premeno      30-34      3-5       yes         2   left     right_up       no no-recurrence-events
272    recurrence-events 50-59   premeno      25-29      0-2        no         3  right     left_low      yes no-recurrence-events
274    recurrence-events 60-69      ge40      30-34      0-2       yes         2  right     right_up      yes no-recurrence-events
275    recurrence-events 60-69      ge40      30-34      3-5       yes         2   left      central      yes no-recurrence-events
```

This comparison table demonstrates that of the misclassified rows, it was more common for classes to be incorrectly predicted as no-recurrence events, when they, in fact, were recurrence events. Only 6 of the 28 misclassified observations were predicted to be recurrence events but were actually no-recurrence.

Step 4: Random Forest Model

A Random forest model is used to predict the class of cancer against all other variables. Below is the summary of the result. From the diagram, it can be seen that the number of trees and number of variables at each split is 100 and 2 respectively. We can also see that the out-of-bag error is 24.5%

```
Call:
 randomForest(formula = class ~ ., data = cancer.train, importance = TRUE,      ntree = 100, mtry = 2)
               Type of random forest: classification
                     Number of trees: 100
No. of variables tried at each split: 2

        OOB estimate of  error rate: 24.5%
Confusion matrix:
                     no-recurrence-events recurrence-events class.error
no-recurrence-events                  130                13  0.09090909
recurrence-events                      36                21  0.63157895
```

Step 5: Plot the Random Forest Model

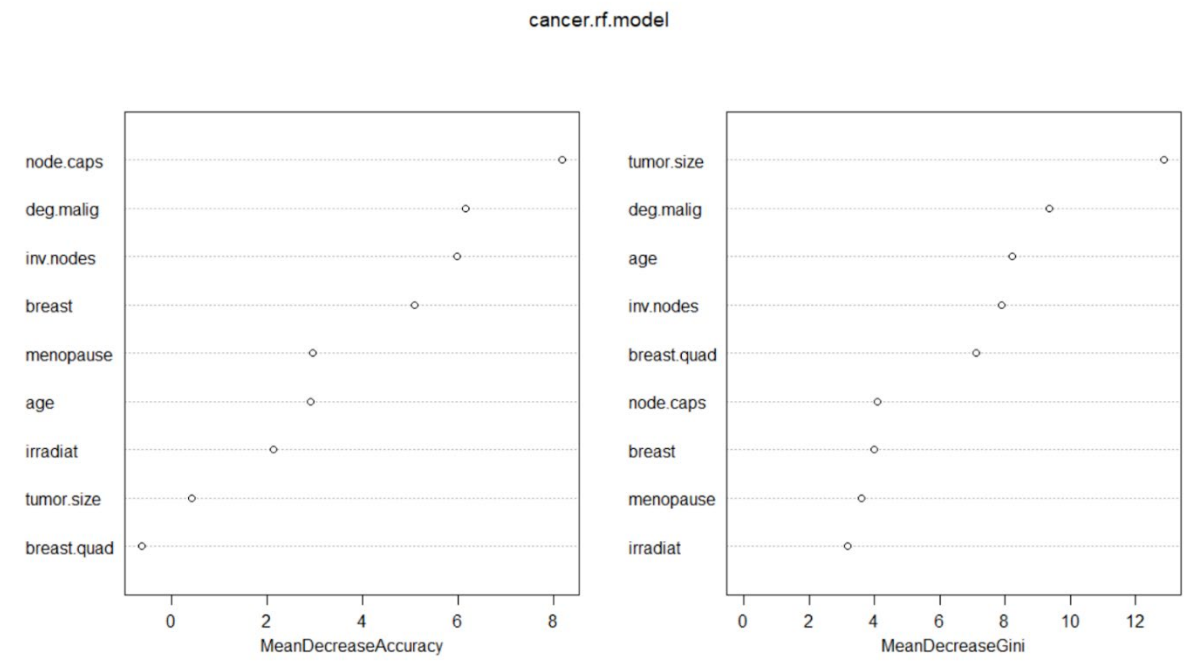

Figure 4: Plot of variable importance

Random forest has an inbuilt function to plot the variable importance according to two metrics. This can be seen in *Figure 2*.
MeanDecreaseAccuracy: Provides a rough estimate of the loss in output of prediction when the relevant variable is omitted from the training set. Node caps, degree of malignancy and inv nodes are the most important variables according to this metric.

MeanDecreaseGini: The Gini is a measure of node impurity and it has tumor size, degree of malignancy and age as its most important variables.

Evaluating the Models

Confusion Matrix

A confusion matrix is a table which focuses on how the classifier performs on a set of test data. The four different outcomes of the matrix are:

- True positive: predicted positive, outcome is positive
- True negative: predicted negative, outcome is negative
- False positive: predicted positive, outcome is negative
- False negative: predicted negative, outcome is positive

These combinations allow us to measure accuracy, recall and precision.

A confusion Matrix was created for both the decision tree and the random forest model.

```
> tree.confusion <- table(cancer.predictions, cancer.test$class)
> print(tree.confusion)

cancer.predictions     no-recurrence-events recurrence-events
  no-recurrence-events                   52                 22
  recurrence-events                       6                  6
```

```
preds.rf.cancer         no-recurrence-events recurrence-events
  no-recurrence-events                   54                 25
  recurrence-events                       4                  3
```

In the tree model, six observations were both incorrectly classified while in the forest model, 4 non-recurrence events were incorrectly classified and 3 recurrence events misclassified. The confusion matrix function was expanded to generate further statistics, including the metrics of accuracy, precision and recall.

```
> confusionMatrix(tree.confusion, mode = "prec_recall")
Confusion Matrix and Statistics


cancer.predictions     no-recurrence-events recurrence-events
  no-recurrence-events                   52                 22
  recurrence-events                       6                  6

               Accuracy : 0.6744
                 95% CI : (0.5648, 0.7716)
    No Information Rate : 0.6744
    P-Value [Acc > NIR] : 0.551008

                  Kappa : 0.1301

 Mcnemar's Test P-Value : 0.004586

              Precision : 0.7027
                 Recall : 0.8966
                     F1 : 0.7879
             Prevalence : 0.6744
         Detection Rate : 0.6047
   Detection Prevalence : 0.8605
      Balanced Accuracy : 0.5554

       'Positive' Class : no-recurrence-events
```

Figure 5: Confusion matrix statistics for decision tree model

```
preds.rf.cancer         no-recurrence-events recurrence-events
  no-recurrence-events                   54                 25
  recurrence-events                       4                  3

               Accuracy : 0.6628
                 95% CI : (0.5528, 0.7612)
    No Information Rate : 0.6744
    P-Value [Acc > NIR] : 0.6393430

                  Kappa : 0.0474

 Mcnemar's Test P-Value : 0.0002041

              Precision : 0.6835
                 Recall : 0.9310
                     F1 : 0.7883
             Prevalence : 0.6744
         Detection Rate : 0.6279
   Detection Prevalence : 0.9186
      Balanced Accuracy : 0.5191

       'Positive' Class : no-recurrence-events
```

Figure 6: Confusion matrix statistics for random forest model

Accuracy

Accuracy, also known as the recognition rate, is a metric that measures the percentage of test observations that are correctly classified. To ensure that the usage of data does not mislead decisions, the accuracy of data is estimated prior. Models that are not at least 80% accurate cannot be used in a clinical setting. We used the following code to calculate accuracy:

```
> cancer.rf.accuracy <- sum(diag(cancer.rf.confusion)) / sum(cancer.rf.confusion)
> print(cancer.rf.accuracy)
[1] 0.6627907
```

Our models are at 67.44% and 66.28 accuracy. This means 67.44% of our test observations were correctly classified as either a recurrence or non-recurrence event. However, this also means over 30% of our observations were inaccurate and incorrectly classified. This means they cannot be used in a clinical setting. The decision tree seems to perform better than the random forest.

Recall

Recall, also known as the "true positive rate" or sensitivity of data, looks at the proportion of relevant results out of the number of actual relevant samples (e.g. of the observations that are actually positive, how many were predicted positive). This metric helps understand how the classification models work when applied to the breast cancer dataset. With no relevant results, recall is not defined, and the value of NA is returned (Alex, 2019). The recall from our models are 0.8966 and 0.9310. This means that our models predicted the positive results well, as approximately 90% of observations that are actually positive were predicted to be positive.

Precision

The precision metric focuses on the performance of the classifier. It determines how many of the classes which were predicted to be positive, are actually positive. The precision metric obtained from the confusion matrix is 0.7027 for the decision tree and 0.6835 for the forest mode. This means that of the classes predicted to be positive, 70% were actually positive. This means the decision tree model was quite precise and the classifier performed well. However, the random forest did not classify well. Perhaps might need some parameter tuning.

Parameter Tuning

Often known as hyperparameter optimization is choosing the optimal values for our models in order to increase the model architecture (Wikipedia 2020). We have tuned our models and have come up with the best values and increased accuracy. For the decision tree model, we used the two-fold method to determine the number of splits and complexity parameter (cp). We used the one-standard error rule to determine the minimum cp and we pruned the tree as shown below.

```
set.seed(777)
tree.params <- rpart.control(minsplit=3, minbucket=round(5 / 3), maxdepth=15, cp=0.01169591)

## Fit decision model to training set
## Use parameters from above and Gini index for splitting
plotcp(cancer.tree)
cancer.tree <- rpart(class ~ ., data = cancer.train,
                     control=tree.params, parms=list(split="gini"))
```

As expected the result of the pruning led to a slight increase in the model accuracy but it is stil not enough to make it a good model. The accuracy of the model increased from 0.67% to 0.69%.

```
> confusionMatrix(tree.confusion, mode = "prec_recall")
Confusion Matrix and Statistics


cancer.predictions     no-recurrence-events recurrence-events
  no-recurrence-events                   54                22
  recurrence-events                       4                 6

               Accuracy : 0.6977
                 95% CI : (0.5892, 0.7921)
    No Information Rate : 0.6744
    P-Value [Acc > NIR] : 0.3695804

                  Kappa : 0.1743

 Mcnemar's Test P-Value : 0.0008561

              Precision : 0.7105
                 Recall : 0.9310
                     F1 : 0.8060
             Prevalence : 0.6744
         Detection Rate : 0.6279
   Detection Prevalence : 0.8837
      Balanced Accuracy : 0.5727

       'Positive' Class : no-recurrence-events
```

Figure 7: Confusion matrix statistics for tuned tree model

For the random forest, we have used the boosting method to tune the model. We tried to boost by performing a grid tuning, we then performed a gradient boosting using the “XGB” package. These boosting were computationally heavy tasks so we decided to run the process in parallel creating 8 copies of R to run our models simultaneously. After trying both methods they produced the exact same result. The boosting for the forest had a similar accuracy as the tree model but different precision, recall.

```
Confusion Matrix and Statistics

                     Reference
Prediction            no-recurrence-events recurrence-events
  no-recurrence-events                   53                 21
  recurrence-events                       5                  7

               Accuracy : 0.6977
                 95% CI : (0.5892, 0.7921)
    No Information Rate : 0.6744
    P-Value [Acc > NIR] : 0.369580

                  Kappa : 0.1922

 Mcnemar's Test P-Value : 0.003264

              Precision : 0.7162
                 Recall : 0.9138
                     F1 : 0.8030
             Prevalence : 0.6744
         Detection Rate : 0.6163
   Detection Prevalence : 0.8605
      Balanced Accuracy : 0.5819

       'Positive' Class : no-recurrence-events
```

Figure 8: Confusion matrix statistics for boosted random forest

Conclusion

This report presented an analysis of breast cancer production using RStudio and constructed a decision tree and random forest model to establish an in-depth data analysis. It can be concluded that the decision tree model is most suited to meet our business objective of predicting breast cancer likelihood than the random forest model. This is because the decision tree has a higher accuracy and precision. This means a higher amount of observations were correctly classified, and more classes which were predicted to be positive were actually positive. The decision tree also has a high recall rate meaning that a high amount of the observations that were actually positive were also predicted to be positive. From the models, it has been found that patients with a tumor size 10/18 are always more likely to have a cancer recurrence. Despite our models returning detailed information, they cannot be used in a clinical setting as they do not reach 80% accuracy. It would be interesting to test the analysis with other classification algorithms such as K-Neighbour classifiers, Logistic regression and SVC.

References

Cole, Z 2020, *Types of Data Models: Conceptual, Logical and Physical*, Erwin, viewed 1 November 2020, <https://erwin.com/blog/types-of-data-models-conceptual-logical-physical/?fbclid=IwAR3K017fNjEWlZA6DAeFpmFpAKIAZo949y2g-Ck5cqQT4eHt7gpdYCWS00I>.

Moayedikia, A 2020, 'Lecture 6', INF30030 *Business Analytics*, Learning Material on Canvas, Swinburne University of Technology, [Accessed 11 September 2020].

Moayedikia, A 2020, 'Lecture 7', INF30030 *Business Analytics*, Learning Material on Canvas, Swinburne University of Technology, [Accessed 11 September 2020].

Moayedikia, A 2020, 'Lecture 9', INF30030 *Business Analytics*, Learning Material on Canvas, Swinburne University of Technology, [Accessed 11 September 2020].

Rohrich, G 2020, *Training, Validating and Testing – Why Proper Model Selection is Essential*, Towards Data Science, viewed 31 October 2020, <https://towardsdatascience.com/train-test-split-c3eed34f763b>.

Rstudio.com. 2020. *About Rstudio*. [online] Available at: <https://rstudio.com/about/> [Accessed 9 September 2020].

Wikipedia 2020, *Hyperparamter Optimization*, Wikipedia, viewed 1 November 2020, <https://en.wikipedia.org/wiki/Hyperparameter_optimization>.

Appendices

Appendix 1: Brief Description of the Installed Packages Used

Installed Package	Description
Performance Analytics	Provides econometric functions for performance and risk analysis of financial instruments
dplyr	Focuses on data frames and provides a set of tools for efficiently manipulate datasets
ggplot2	Enables the creation of custom plots to create multi-layered visuals with ease
gridExtra	Works with "grid" graphics and to arranges multiple grid-based plots on a page, and draw tables
arm	Plots the balance statistics before and after matching (used for regression and multilevel models)
car	Companion package to an applied regression model
psych	Basic data analysis and psychometric analysis
caTools	Contains several basic utility functions including: moving (rolling, running) window statistic functions, read/write for GIF and ENVI binary files etc.
caret	Streamline the model training process for complex regression and classification problems
keras	Focus on enabling fast experimentation
mice	Implements a method to deal with missing data by creating multiple imputations
VIM	Introduces new tools for the visualization of missing or imputed values
rpart	Splitting the dataset recursively, which means that the subsets that arise from a split are further split until a predetermined termination criterion is reached
rattle	Providing a graphical user interface and functionality to data mining
randomForest	Generating a large number of decision trees

Appendix 2: Summary of *canc* dataset

```
> names(canc)
 [1] "class"       "age"         "menopause"   "tumor.size"  "inv.nodes"   "node.caps"   "deg.malig"   "breast"
 [9] "breast.quad" "irradiat"
> str(canc)
'data.frame':   286 obs. of  10 variables:
 $ class      : Factor w/ 2 levels "no-recurrence-events",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ age        : Factor w/ 6 levels "20-29","30-39",..: 2 3 3 5 3 5 4 5 3 3 ...
 $ menopause  : Factor w/ 3 levels "ge40","lt40",..: 3 3 3 1 3 1 3 1 3 3 ...
 $ tumor.size : Factor w/ 11 levels "0-4","14-Oct",..: 6 4 4 3 1 3 5 4 10 4 ...
 $ inv.nodes  : Factor w/ 7 levels "0-2","11-Sep",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ node.caps  : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...
 $ deg.malig  : Factor w/ 3 levels "1","2","3": 3 2 2 2 2 2 2 1 2 2 ...
 $ breast     : Factor w/ 2 levels "left","right": 1 2 1 2 2 1 1 1 1 2 ...
 $ breast.quad: Factor w/ 5 levels "central","left_low",..: 2 5 2 3 4 2 2 2 2 3 ...
 $ irradiat   : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...
> head(canc)
                  class   age menopause tumor.size inv.nodes node.caps deg.malig breast breast.quad irradiat
1 no-recurrence-events 30-39   premeno      30-34       0-2        no         3   left    left_low       no
2 no-recurrence-events 40-49   premeno      20-24       0-2        no         2  right   right_up       no
3 no-recurrence-events 40-49   premeno      20-24       0-2        no         2   left    left_low       no
4 no-recurrence-events 60-69      ge40      15-19       0-2        no         2  right    left_up       no
5 no-recurrence-events 40-49   premeno        0-4       0-2        no         2  right   right_low       no
6 no-recurrence-events 60-69      ge40      15-19       0-2        no         2   left    left_low       no
> summary(canc)
                  class        age        menopause     tumor.size  inv.nodes   node.caps  deg.malig   breast
 no-recurrence-events:201   20-29: 1   ge40   :129   30-34  :60   0-2   :213   no  :222   1: 71     left :152
 recurrence-events   : 85   30-39:36   lt40   :  7   25-29  :54   11-Sep: 10   yes : 56   2:130     right:134
                            40-49:90   premeno:150   20-24  :50   14-Dec:  3   NA's:  8   3: 85
                            50-59:96                 15-19  :30   15-17 :  6
                            60-69:57                 14-Oct :28   24-26 :  1
                            70-79: 6                 40-44  :22   5-Mar : 36
                                                     (Other):42   8-Jun : 17

   breast.quad  irradiat
 central  : 21   no :218
 left_low :110   yes: 68
 left_up  : 97
 right_low: 24
 right_up : 33
 NA's     :  1
```

Appendix 3: Summary of *cancer.tree* dataset

```
> summary(cancer.tree)
Call:
rpart(formula = class ~ ., data = cancer.train, method = "class")
  n= 200

          CP nsplit rel error    xerror      xstd
1 0.13157895      0 1.0000000 1.0000000 0.1119994
2 0.05263158      2 0.7368421 0.8947368 0.1081405
3 0.01169591      3 0.6842105 0.9473684 0.1101497
4 0.01000000      6 0.6491228 0.8421053 0.1059626

Variable importance
  deg.malig   inv.nodes         age  tumor.size   node.caps breast.quad    irradiat   menopause
         26          21          16          11          11           7           5           1

Node number 1: 200 observations,    complexity param=0.1315789
  predicted class=no-recurrence-events  expected loss=0.285  P(node) =1
    class counts:   143    57
   probabilities: 0.715 0.285
  left son=2 (139 obs) right son=3 (61 obs)
  Primary splits:
      deg.malig  splits as  LLR,         improve=10.076580, (0 missing)
      inv.nodes  splits as  LRRRRRR,     improve= 8.838049, (0 missing)
      node.caps  splits as  LR,          improve= 7.022500, (0 missing)
      irradiat   splits as  LR,          improve= 4.118817, (0 missing)
      tumor.size splits as  LLRRRRRRLLR, improve= 3.364902, (0 missing)
  Surrogate splits:
      inv.nodes  splits as  LRRRLRL,     agree=0.745, adj=0.164, (0 split)
      node.caps  splits as  LR,          agree=0.715, adj=0.066, (0 split)
      tumor.size splits as  LLLLLLRLLLL, agree=0.705, adj=0.033, (0 split)

Node number 2: 139 observations,    complexity param=0.01169591
  predicted class=no-recurrence-events  expected loss=0.1798561  P(node) =0.695
    class counts:   114    25
   probabilities: 0.820 0.180
  left son=4 (20 obs) right son=5 (119 obs)
  Primary splits:
      tumor.size splits as  RLRRRRRRLLR, improve=1.5113960, (0 missing)
      age        splits as  LRLLLR,      improve=1.3666150, (0 missing)
      menopause  splits as  LRR,         improve=0.9917037, (0 missing)
      inv.nodes  splits as  L---LRR,     improve=0.6465072, (0 missing)
      irradiat   splits as  LR,          improve=0.6114048, (0 missing)

Node number 3: 61 observations,    complexity param=0.1315789
  predicted class=recurrence-events     expected loss=0.4754098  P(node) =0.305
    class counts:    29    32
   probabilities: 0.475 0.525
```

```
  left son=6 (34 obs) right son=7 (27 obs)
  Primary splits:
      inv.nodes   splits as  LRRRRRR, improve=6.210543, (0 missing)
      node.caps   splits as  LR,      improve=5.932057, (0 missing)
      age         splits as  -RLLRL,  improve=2.671725, (0 missing)
      breast.quad splits as  LRLRL,   improve=1.885281, (0 missing)
      irradiat    splits as  LR,      improve=1.701288, (0 missing)
  Surrogate splits:
      node.caps   splits as  LR,          agree=0.820, adj=0.593, (0 split)
      irradiat    splits as  LR,          agree=0.689, adj=0.296, (0 split)
      breast.quad splits as  LRLRL,       agree=0.672, adj=0.259, (0 split)
      age         splits as  -LLRLL,      agree=0.607, adj=0.111, (0 split)
      tumor.size  splits as  LLLLRLLR--L, agree=0.590, adj=0.074, (0 split)

Node number 4: 20 observations
  predicted class=no-recurrence-events  expected loss=0  P(node) =0.1
    class counts:    20     0
   probabilities: 1.000 0.000

Node number 5: 119 observations,    complexity param=0.01169591
  predicted class=no-recurrence-events  expected loss=0.210084  P(node) =0.595
    class counts:    94    25
   probabilities: 0.790 0.210
  left son=10 (98 obs) right son=11 (21 obs)
  Primary splits:
      age         splits as  LRLLLR,      improve=1.4889960, (0 missing)
      menopause   splits as  LRR,         improve=0.9514267, (0 missing)
      irradiat    splits as  LR,          improve=0.6307561, (0 missing)
      breast.quad splits as  RRRLL,       improve=0.3617403, (0 missing)
      tumor.size  splits as  L-LRLLLR--L, improve=0.2840336, (0 missing)
  Surrogate splits:
      inv.nodes splits as  L---LLR, agree=0.832, adj=0.048, (0 split)

Node number 6: 34 observations,    complexity param=0.05263158
  predicted class=no-recurrence-events  expected loss=0.3235294  P(node) =0.17
    class counts:    23    11
   probabilities: 0.676 0.324
  left son=12 (19 obs) right son=13 (15 obs)
  Primary splits:
      age         splits as  -RLLRL,      improve=4.10340600, (0 missing)
      tumor.size  splits as  LLLLRLRL--R, improve=1.60172100, (0 missing)
      breast.quad splits as  RLRRL,       improve=0.36220640, (0 missing)
      irradiat    splits as  LR,          improve=0.19452230, (0 missing)
      menopause   splits as  RLL,         improve=0.05882353, (0 missing)
  Surrogate splits:
      breast.quad splits as  RRLRL,       agree=0.676, adj=0.267, (0 split)
      tumor.size  splits as  LLRLLLLL--R, agree=0.618, adj=0.133, (0 split)
```

```
      menopause   splits as  RLL,         agree=0.588, adj=0.067, (0 split)
      irradiat    splits as  LR,          agree=0.588, adj=0.067, (0 split)

Node number 7: 27 observations
  predicted class=recurrence-events     expected loss=0.2222222  P(node) =0.135
    class counts:     6    21
   probabilities: 0.222 0.778

Node number 10: 98 observations
  predicted class=no-recurrence-events  expected loss=0.1734694  P(node) =0.49
    class counts:    81    17
   probabilities: 0.827 0.173

Node number 11: 21 observations,    complexity param=0.01169591
  predicted class=no-recurrence-events  expected loss=0.3809524  P(node) =0.105
    class counts:    13     8
   probabilities: 0.619 0.381
  left son=22 (13 obs) right son=23 (8 obs)
  Primary splits:
      tumor.size  splits as  L-RLLR-L---, improve=1.53937700, (0 missing)
      breast.quad splits as  LRRLL,       improve=1.19047600, (0 missing)
      deg.malig   splits as  RL-,         improve=0.36630040, (0 missing)
      breast      splits as  RL,          improve=0.01385281, (0 missing)
  Surrogate splits:
      inv.nodes   splits as  L---LLR, agree=0.714, adj=0.250, (0 split)
      breast.quad splits as  LLRLL,   agree=0.667, adj=0.125, (0 split)

Node number 12: 19 observations
  predicted class=no-recurrence-events  expected loss=0.1052632  P(node) =0.095
    class counts:    17     2
   probabilities: 0.895 0.105

Node number 13: 15 observations
  predicted class=recurrence-events     expected loss=0.4  P(node) =0.075
    class counts:     6     9
   probabilities: 0.400 0.600

Node number 22: 13 observations
  predicted class=no-recurrence-events  expected loss=0.2307692  P(node) =0.065
    class counts:    10     3
   probabilities: 0.769 0.231

Node number 23: 8 observations
  predicted class=recurrence-events     expected loss=0.375  P(node) =0.04
    class counts:     3     5
   probabilities: 0.375 0.625
```