Neural Networks in R

Breast cancer is "the most common cause of cancer death among women worldwide". In the United States, breast cancer is second to lung cancer related deaths, a national health critical issue. Statistical facts show breast cancer as the most frequently diagnosed cancer in women in 140 out of 184 countries (Henderson, 2015). Key to survival and reduction of breast cancer-related mortality is closely linked with early detection and intervention.

Early signs of irregular cells growth are detected by sampling and analyzing nuclear changes and parameter using diagnostic tools. The results of these nuclear morphometry tests are evaluated for structural deviations, which are representative of cancer diagnosis (Narasimha, Vasavi, & Kumar, 2013). Now, considering the significance in accuracy of these evaluations, one may question how the medical industry uses machine learning models like neural networks to augment diagnosis and judgement of such vital medical assessments.

The following is post-study report of numerous breast cancer preventive screenings, scrutinizing cell nuclei parameters in order to classify the specimens either malignant or benign. The data have been previously categorized, thus, the intent here is to employ a neural network methodology to replicate this categorization and measure the algorithm's effectiveness supporting medical professionals in the identification and early detection of breast carcinoma.

Method Rationale. The particular machine and deep learning algorithm of neural network contains exceptional calculations which are optimal for the prediction and classification of responses like the one here evaluated. By employing a multilayer perceptron all the data will be evaluated as it passes through the layers, and by means of the retro-feeding tactic of the process (backpropagation), the weight and bias of each neuron is adjusted repeatedly until the model converges the labeled classification. Furthermore, the model allows users to define the number

of layers to be used, providing a sense of control of how the model performs (Burkov, 2019).

Data. The data at hand [Breast Cancer Wisconsin (Diagnostic) Data Set], comes from the UCI Machine Learning Repository, where it has been maintained after been donated. This multivariate file contains 569 instances and 32 attributes including the response variable named, *diagnosis* (Dua, and Graff, 2019). Per appendix A1, the first variable of the files refers to unique identifier (*ID*), which for the purpose of this assessment has been excluded as shown below.

```
> wdbc$ID <- NULL     # Exclusion of ID variable
```

As previously stated, the dataset features quantitative data representative of the images obtained by means of a fine needle aspirate (FNA) process. These digitized samples were studied, measure and recorded, ultimately enabling the classification and diagnosing of every instance as malignant or benign. Essentially, there is a total of 10 real-value features per cell, however, given the 3-dimensional fragmentation of each cell sampling, it produces a total of 30 observations across 3 planes (Dua, and Graff, 2019). Figure 1.1 provides a preview of the dataframe content and additional statistical insight.

Variables description:

1. ID number
2. Diagnosis (M = malignant, B = benign
3 – 30. Cell measurements:
   a) radius (mean of distances from center to points on the perimeter)
   b) texture (standard deviation of gray-scale values)
   c) perimeter
   d) area
   e) smoothness (local variation in radius lengths)
   f) compactness (perimeter^2 / area - 1.0)
   g) concavity (severity of concave portions of the contour)
   h) concave points (number of concave portions of the contour)
   i) symmetry
   j) fractal dimension ("coastline approximation" - 1)

```
head(wdbc, 4)
diagnosis radius texture perimeter   area smoothness compactness concavity concave symmetry fractal
        M  17.99   10.38    122.80 1001.0    0.11840     0.27760    0.3001 0.14710   0.2419 0.07871
        M  20.57   17.77    132.90 1326.0    0.08474     0.07864    0.0869 0.07017   0.1812 0.05667
        M  19.69   21.25    130.00 1203.0    0.10960     0.15990    0.1974 0.12790   0.2069 0.05999
        M  11.42   20.38     77.58  386.1    0.14250     0.28390    0.2414 0.10520   0.2597 0.09744
radius2 texture2 perimeter2  area2 smoothness2 compactness2 concavity2 concave2 symmetry2 fractal2
 1.0950   0.9053      8.589 153.40    0.006399      0.04904    0.05373  0.01587   0.03003 0.006193
 0.5435   0.7339      3.398  74.08    0.005225      0.01308    0.01860  0.01340   0.01389 0.003532
 0.7456   0.7869      4.585  94.03    0.006150      0.04006    0.03832  0.02058   0.02250 0.004571
 0.4956   1.1560      3.445  27.23    0.009110      0.07458    0.05661  0.01867   0.05963 0.009208
radius3 texture3 perimeter3  area3 smoothness3 compactness3 concavity3 concave3 symmetry3 fractal3
  25.38    17.33     184.60 2019.0      0.1622       0.6656     0.7119   0.2654    0.4601  0.11890
  24.99    23.41     158.80 1956.0      0.1238       0.1866     0.2416   0.1860    0.2750  0.08902
  23.57    25.53     152.50 1709.0      0.1444       0.4245     0.4504   0.2430    0.3613  0.08758
  14.91    26.50      98.87  567.7      0.2098       0.8663     0.6869   0.2575    0.6638  0.17300
```

Figure 1.1 – Data preview, including all 30 measurements and the response variable.

Exploratory Analysis. Throughout the exploratory data analysis (EDA) step, basic commands such as str and summary were used. As shown in appendixes A1 and A2, the set originally had 32 variables, and the *diagnosis* value was a "character" type. Looking the summary ( ) command results, one can anticipate a likely correlation between some variables due to their nature (i.e., radius-area-compactness). Moreover, scaling and centering the distributions should be considered.

Before conducting any in-depth preprocessing, the first approach was to transform the *diagnosis* values as factors. Per the bellow code, the original values of "M", and "B" (malignant, and benign) were transformed and renamed as "1", and "0".

```
> # Transform diagnosis values to factor
> wdbc$diagnosis <- factor(wdbc$diagnosis, levels = c("M", "B"),
labels = c("1", "0"))
```

Subsequently, the describe {psych} command was applied to review statistical traits and identify any potential errors. Comparing each plane summary, figure 1.2 and appendix A3, it is clear that plane II emphasis was in cells with smaller dimensions than plane I, and plane III.

```
> describe(wdbc[1:11])
            vars   n   mean     sd median trimmed    mad    min     max   range  skew kurtosis   se
diagnosis*     1 569   1.63   0.48   2.00    1.66   0.00   1.00    2.00    1.00 -0.53    -1.73 0.02
radius         2 569  14.13   3.52  13.37   13.82   2.82   6.98   28.11   21.13  0.94     0.81 0.15
texture        3 569  19.29   4.30  18.84   19.04   4.17   9.71   39.28   29.57  0.65     0.73 0.18
perimeter      4 569  91.97  24.30  86.24   89.74  18.84  43.79  188.50  144.71  0.99     0.94 1.02
area           5 569 654.89 351.91 551.10  606.13 227.28 143.50 2501.00 2357.50  1.64     3.59 14.75
smoothness     6 569   0.10   0.01   0.10    0.10   0.01   0.05    0.16    0.11  0.45     0.82 0.00
compactness    7 569   0.10   0.05   0.09    0.10   0.05   0.02    0.35    0.33  1.18     1.61 0.00
concavity      8 569   0.09   0.08   0.06    0.08   0.06   0.00    0.43    0.43  1.39     1.95 0.00
concave        9 569   0.05   0.04   0.03    0.04   0.03   0.00    0.20    0.20  1.17     1.03 0.00
symmetry      10 569   0.18   0.03   0.18    0.18   0.03   0.11    0.30    0.20  0.72     1.25 0.00
fractal       11 569   0.06   0.01   0.06    0.06   0.01   0.05    0.10    0.05  1.30     2.95 0.00
```

Figure 1.2 – Sample of statistical summary, Plane I.

After some extensive time of data familiarization, four variables (radius, area, texture, concave) were selected to compare their values and distributions in relation to their respective classifications. Per appendix A4, the developed code was to display these named variables and their density proportions. Figure 1.3, portrays a density visualization of these variable's measures in relation with the ultimate taxonomy of the cells of been malignant or benign.

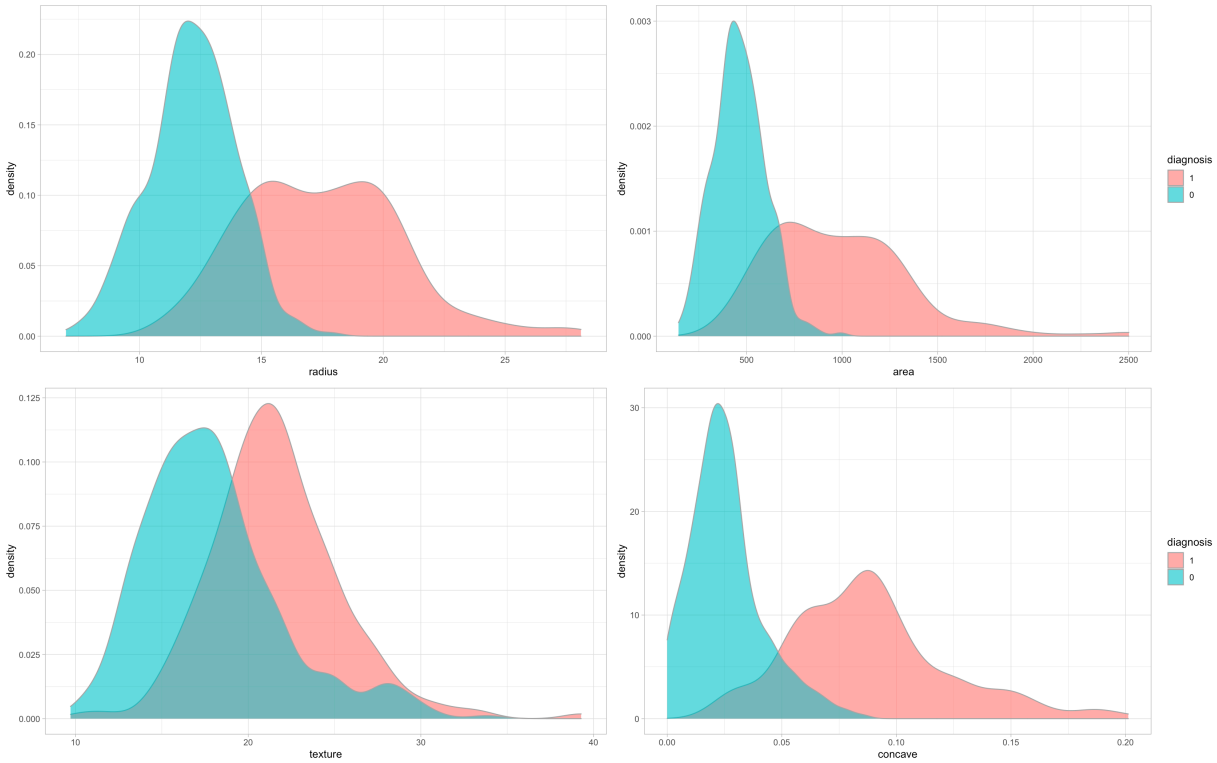


Figure 1.3 – Density representation, four variables from Plane I.

Next, to further the exploration a graphical representation of the original *diagnosis* values

was generated. Appendix A5, illustrates a decision tree graphic which highlights variables like concave, area, and concavity as key splitting nodes part of the classification process. From a comparative standpoint, the presupposition here is that these named variables will be identify as attributes with higher weights throughout this neural network model as well.

Finally, the variable distributions were analyze, looking for potential correlations among variables, and/or anomalies. Appendix A6 contains the code and additional graphics representing these distributions while segregated by planes (Plane I, Plane II, and Plane III).

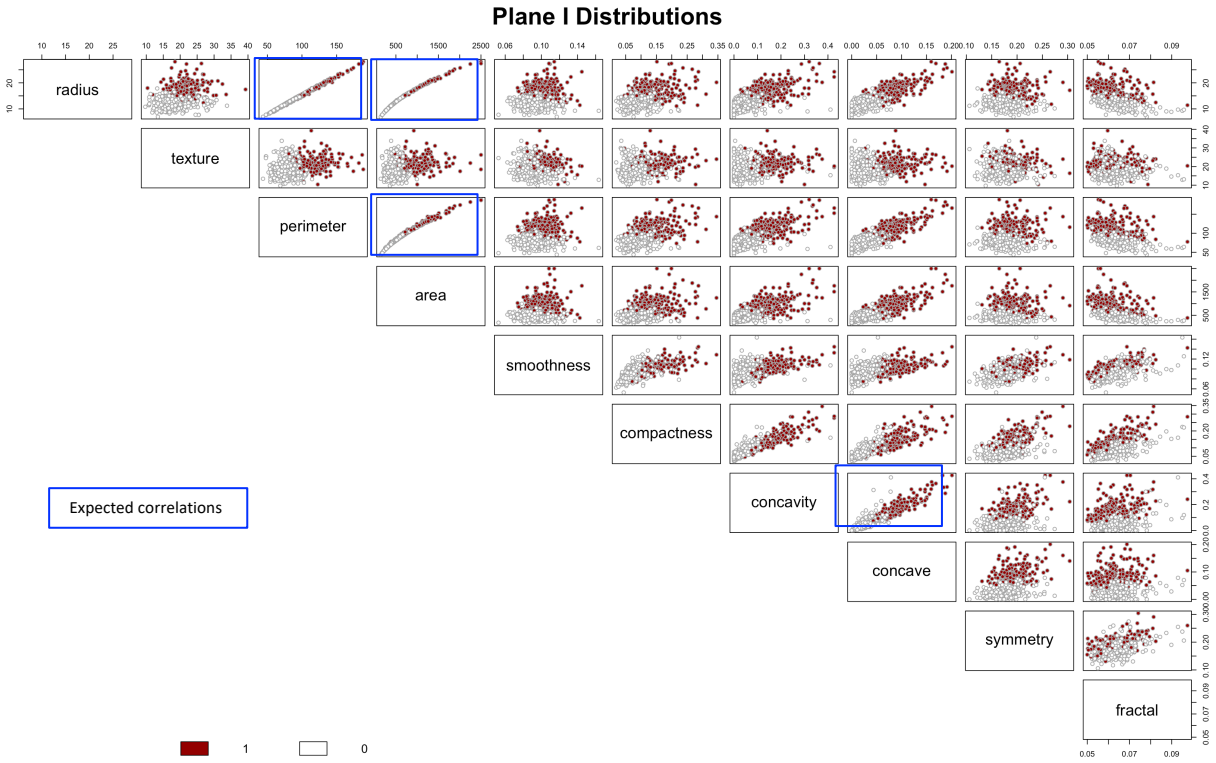


Figure 1.4 – Variables distributions and comparison, for plane I.

Preprocessing. Upon completion of the EDA phase, one additional step was taken in preparation for the algorithm. This was converting the values of the response variable into a numeric type. This step prove to be essential for the classification of the targeted variable within

the model. The following code shows the command used to transform these values.

```
> # Diagnosis values converted to numeric for accurate representation
> wdbc$diagnosis <- as.character(wdbc$diagnosis)
> wdbc$diagnosis <- as.numeric(wdbc$diagnosis)
```

Algorithm Intuition. The neural network model rely on the proper preparation of the data to be analyzed. For this assessment the dataset was divided into three different subsets. The ratios of these were 0.70, 0.15, and 0.15. The reason for creating three sets was to not only test the trained model, but to cross-validate once again the model with the third set. As displayed on the bellow code, a seed was established, and three samples were taken from the vector.

```
> set.seed(1234)
> #split the data into a training and test set
> ind <- sample(3,nrow(wdbc),replace=TRUE,prob= c(0.70,0.15,0.15))
> train <- wdbc[ind == 1, ]
> test <- wdbc[ind == 2, ]
> QC <- wdbc[ind == 3, ]
```

The designated proportions were assigned to three new variables, *train, test*, and *QC* followed by running the model against the subsets. Figure 1.5 shows the train data results.

```
> # Run the neuralnet command to build the model.
> nn <- neuralnet(diagnosis ~.,
+                 data = train, hidden = c(4,4),
+                 lifesign = "minimal", linear.output = FALSE, err.fct = "sse", likelihood=TRUE)
hidden: 4, 4    thresh: 0.01    rep: 1/1    steps:     432      error: 11.00197 aic: 320.00394  bic: 911.72195  time: 0.25 secs
```

Figure 1.5 – Neuralnet command results against the train data with 2 layers of 4 neurons each.

Modeling Fitting. After numerous repetitions, and testing of different proportions and layers, the best results were linked to a (4, 4) approach. As projected, *diagnosis* was the designated response variable and all others were used as independent attributes. The response time for the 569 iterations using the (4,4) hidden layers-set was 0.25 seconds, an error around 11.00 and an AIC value of 320.00.

The following code was used to generated the model illustrated on figure 1.6.

```
> # plot the NN
> plot(nn,radius = 0.05, arrow.length = 0.16, intercept = TRUE,
intercept.factor = 0.025, information = TRUE, information.pos = 8,
col.entry.synapse = "black", col.entry = "maroon4", line_stag= 0.03,
col.hidden = "blue", col.hidden.synapse = "dimgrey",
col.out = "maroon4", col.out.synapse = "darkblue",
col.intercept = "red", fontsize = 10, dimension = 2, show.weights = T)
```

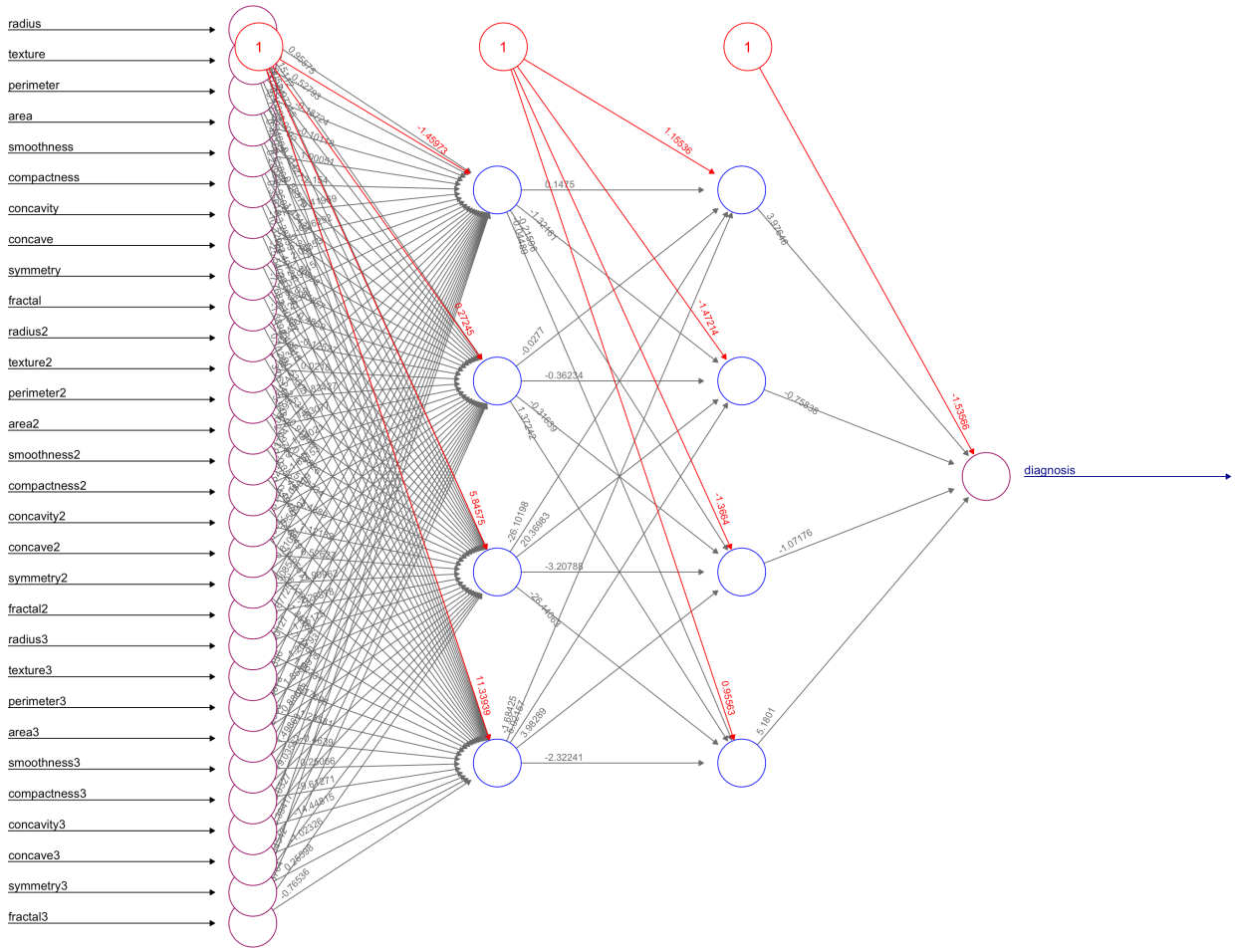


Figure 1.6 – Neural network depiction of the train data.

The displayed diagram illustrates all the independent variables, across the three named planes, as the inputs, the 2 sets of layers with 4 neurons each [with respective weight values], and the 3 bias with arrows in red. Finally the output, is a single response as projected.

An alternate display was generated for a different perspective, using the plotnet ( ) command from the NeuralNetTools package. This depiction (see figure 1.7) is unique given the connectors colors and thickness indicating the weight and values (i.e., positive vs. negative) of each neuron.

```
> # Alternate display
> plotnet(nn, y_names="Diagnosis", max_sp = TRUE, pad_x=1,
cex_val=.85, circle_col="grey95", circle_cex=7,
bord_col="darkblue",pos_col="blue", neg_col="red", alpha_val=0.6,
line_stag= 0.03)
```

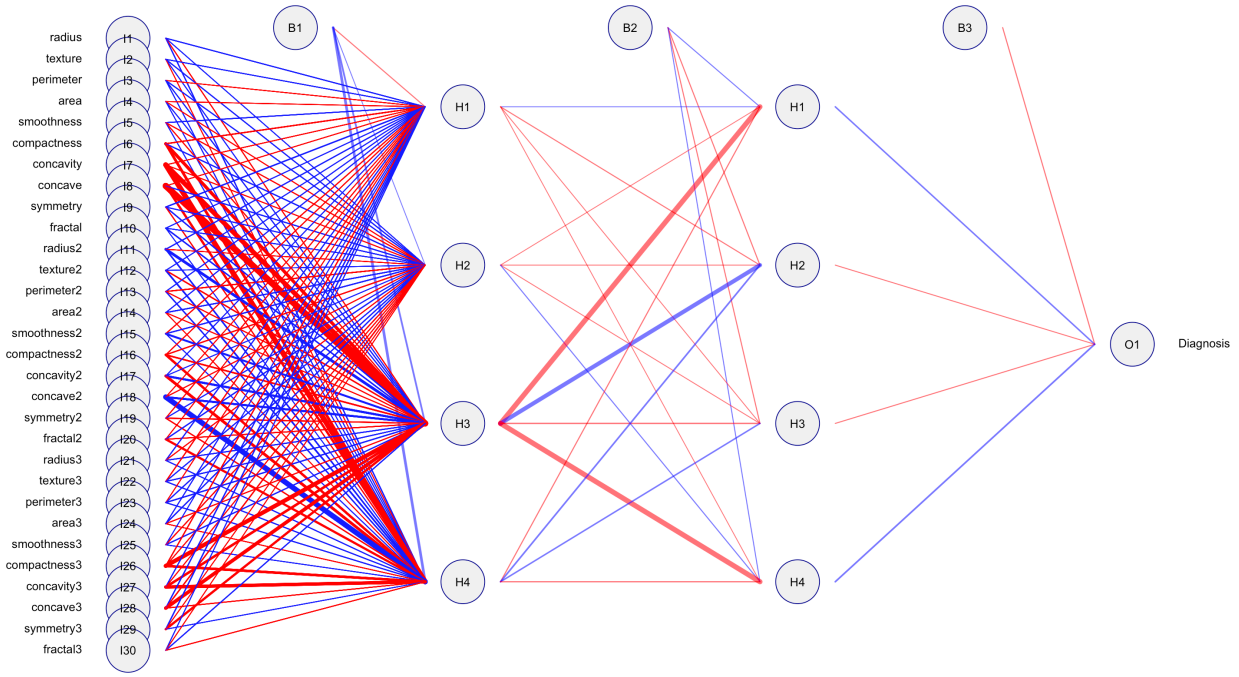


Figure 1.7 – Alternate view of the neural network, accentuating with colors (red = negative, blue=positive) the apportioned weights

Model Properties and Evaluation. To further look into the model's characteristics the nn$result.matrix was executed. In the appendix A7 illustration, one can appreciate a preview of the first layer interactions with the inputs and the assigned weight of each of these. Followed this observation, the train model was evaluated for probability and accuracy. Figure 1.8 shows a

confusion matrix comparing the predicted against actual values, revealing 248 accurate benign classifications and precisely 119 malignant responses, scoring the model with a cumulative accuracy rate of 93.6%.

```
> # Train dataset assessment
> library("Metrics")
> #Model evaluation; Round the predicted probabilities
> mypredict<-compute(nn, nn$covariate)$net.result
> mypredict<-apply(mypredict, c(1), round)
> length(mypredict)
[1] 392
> length(train$diagnosis)
[1] 392
> # confusion matrix for the training set
> n2<-(mypredict==train$diagnosis)
> table(mypredict[1:length(train$diagnosis)], train$diagnosis, dnn =c("Actual","Predicted"))
      Predicted
Actual   0   1
     0 248  19
     1   6 119
> accuracy(train$diagnosis, mypredict[1:length(train$diagnosis)]) #cross-entropy
[1] 0.9362245
```

Figure 1.8 – Train dataset confusion matrix and accuracy rate.

The above mentioned accuracy was once again verified by comparing the true positive rate (TPR), and the false positive rate [ROC curve], as displayed bellow.

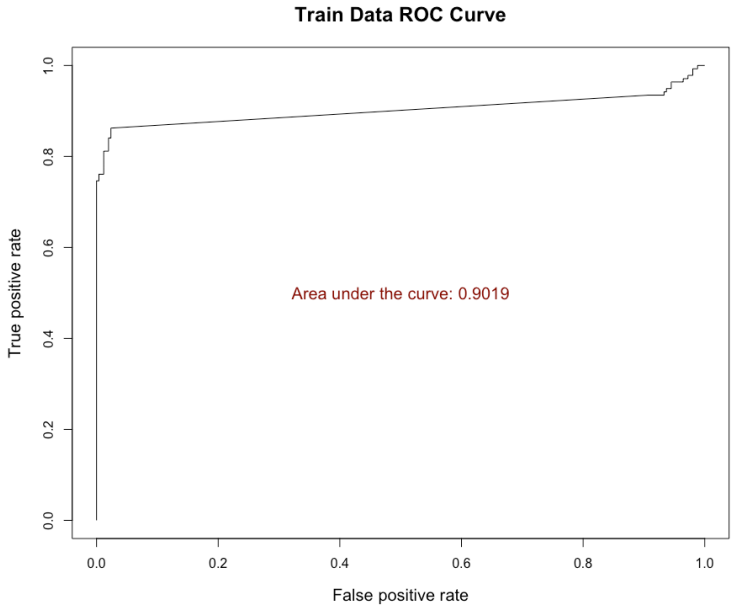


Figure 1.9 – Train model ROC Curve, covering over ~90%

Completed the training portion of the model, the same model was ran against the test data subset, followed by creating a confusion matrix to observe the cross-entropy error result and ultimate accuracy of the model while assessing the test data. Lastly, the same method was applied to the QC cross-validation dataset, resulting in a sustained accuracy level above 90%.

```
> #Model evaluation; Round the predicted probabilities
> testPred <- compute(nn, test[,1:31])$net.result
> testPred <- apply(testPred, c(1), round)
> # confusion matrix for the test set
> n3 <- (testPred==test$diagnosis)
> table(testPred[1:length(test$diagnosis)], test$diagnosis, dnn =c("Predicted", "Actual"))
         Actual
Predicted  0  1
        0 47  6
        1  2 30
> accuracy(testPred[1:length(test$diagnosis)], test$diagnosis) #cross-entropy
[1] 0.9058824
```

Figure 2.1 – Test dataset accuracy calculated above 90%.

```
> #Model evaluation; Round the predicted probabilities
> QC_chk <- compute(nn, QC[, 1:31])$net.result
> QC_chk <- apply(QC_chk, c(1), round)
> # confusion matrix for the test set
> n4<-(QC_chk==QC$diagnosis)
> table(QC_chk[1:length(QC$diagnosis)], QC$diagnosis, dnn =c("Predicted", "Actual"))
         Actual
Predicted  0  1
        0 53  8
        1  1 30
> accuracy(QC_chk[1:length(QC$diagnosis)], QC$diagnosis) #cross-entropy
[1] 0.9021739
```

Figure 2.2 – QC dataset accuracy calculated above 90%.

In summary, and per the illustrations the model responded quite well in minimum time. Knowing that the data set here was small, it is understandable that larger dataset will increase the computational cost. Nevertheless, the assessment demonstrates how much practicality the neural network algorithm offers, enhancing the early detection and classification of breast cancer patients-at-risk, by thoroughly analyzing all the particular nuclear parameters. These examples underlined an average of 90% accuracy of prediction, across the three different tested samples, but with extra labeled data the model can get improved even further.

References

Burkov, A. (2019). The hundred-page machine learning book.

Dietrich, D., Heller, B., & B. Yang. (2015). Data science & big data analytics: Discovering, analyzing, visualizing and presenting data. John Wiley & Sons, Inc.

Dua, D. and Graff, C. (2019). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

Henderson, I.G., MD. (2015). *Breast Cancer : Fundamentals of Evidence-Based Disease Management*. Oxford University Press.

Narasimha, A., Vasavi, B., & Kumar, H. M. (2013). Significance of nuclear morphometry in benign and malignant breast aspirates. *International journal of applied & basic medical research*, *3*(1), 22–26. https://doi.org/10.4103/2229-516X.112237

Appendix

```
> str(wdbc)          # Review structure of the dataset
'data.frame':   569 obs. of  32 variables:
 $ ID          : int  842302 842517 84300903 84348301 84358402 843786 844359
 $ diagnosis   : chr  "M" "M" "M" "M" ...
 $ radius      : num  18 20.6 19.7 11.4 20.3 ...
 $ texture     : num  10.4 17.8 21.2 20.4 14.3 ...
 $ perimeter   : num  122.8 132.9 130 77.6 135.1 ...
 $ area        : num  1001 1326 1203 386 1297 ...
 $ smoothness  : num  0.1184 0.0847 0.1096 0.1425 0.1003 ...
 $ compactness : num  0.2776 0.0786 0.1599 0.2839 0.1328 ...
 $ concavity   : num  0.3001 0.0869 0.1974 0.2414 0.198 ...
 $ concave     : num  0.1471 0.0702 0.1279 0.1052 0.1043 ...
 $ symmetry    : num  0.242 0.181 0.207 0.26 0.181 ...
 $ fractal     : num  0.0787 0.0567 0.06 0.0974 0.0588 ...
 $ radius2     : num  1.095 0.543 0.746 0.496 0.757 ...
 $ texture2    : num  0.905 0.734 0.787 1.156 0.781 ...
 $ perimeter2  : num  8.59 3.4 4.58 3.44 5.44 ...
 $ area2       : num  153.4 74.1 94 27.2 94.4 ...
 $ smoothness2 : num  0.0064 0.00522 0.00615 0.00911 0.01149 ...
 $ compactness2: num  0.049 0.0131 0.0401 0.0746 0.0246 ...
 $ concavity2  : num  0.0537 0.0186 0.0383 0.0566 0.0569 ...
 $ concave2    : num  0.0159 0.0134 0.0206 0.0187 0.0188 ...
 $ symmetry2   : num  0.03 0.0139 0.0225 0.0596 0.0176 ...
 $ fractal2    : num  0.00619 0.00353 0.00457 0.00921 0.00511 ...
 $ radius3     : num  25.4 25 23.6 14.9 22.5 ...
 $ texture3    : num  17.3 23.4 25.5 26.5 16.7 ...
 $ perimeter3  : num  184.6 158.8 152.5 98.9 152.2 ...
 $ area3       : num  2019 1956 1709 568 1575 ...
 $ smoothness3 : num  0.162 0.124 0.144 0.21 0.137 ...
 $ compactness3: num  0.666 0.187 0.424 0.866 0.205 ...
 $ concavity3  : num  0.712 0.242 0.45 0.687 0.4 ...
 $ concave3    : num  0.265 0.186 0.243 0.258 0.163 ...
 $ symmetry3   : num  0.46 0.275 0.361 0.664 0.236 ...
 $ fractal3    : num  0.1189 0.089 0.0876 0.173 0.0768 ...
```

A1 – Dataset structure shows 32 variables including the response and 569 observations.

```
> summary(wdbc)      # Run the summary command
       ID            diagnosis            radius          texture         perimeter          area
 Min.   :     8670   Length:569         Min.   : 6.981   Min.   : 9.71   Min.   : 43.79   Min.   : 143.5
 1st Qu.:   869218   Class :character   1st Qu.:11.700   1st Qu.:16.17   1st Qu.: 75.17   1st Qu.: 420.3
 Median :   906024   Mode  :character   Median :13.370   Median :18.84   Median : 86.24   Median : 551.1
 Mean   : 30371831                      Mean   :14.127   Mean   :19.29   Mean   : 91.97   Mean   : 654.9
 3rd Qu.:  8813129                      3rd Qu.:15.780   3rd Qu.:21.80   3rd Qu.:104.10   3rd Qu.: 782.7
 Max.   :911320502                      Max.   :28.110   Max.   :39.28   Max.   :188.50   Max.   :2501.0
   smoothness       compactness       concavity          concave           symmetry         fractal
 Min.   :0.05263   Min.   :0.01938   Min.   :0.00000   Min.   :0.00000   Min.   :0.1060   Min.   :0.04996
 1st Qu.:0.08637   1st Qu.:0.06492   1st Qu.:0.02956   1st Qu.:0.02031   1st Qu.:0.1619   1st Qu.:0.05770
 Median :0.09587   Median :0.09263   Median :0.06154   Median :0.03350   Median :0.1792   Median :0.06154
 Mean   :0.09636   Mean   :0.10434   Mean   :0.08880   Mean   :0.04892   Mean   :0.1812   Mean   :0.06280
 3rd Qu.:0.10530   3rd Qu.:0.13040   3rd Qu.:0.13070   3rd Qu.:0.07400   3rd Qu.:0.1957   3rd Qu.:0.06612
 Max.   :0.16340   Max.   :0.34540   Max.   :0.42680   Max.   :0.20120   Max.   :0.3040   Max.   :0.09744
    radius2          texture2        perimeter2         area2          smoothness2       compactness2
 Min.   :0.1115   Min.   :0.3602   Min.   : 0.757   Min.   :  6.802   Min.   :0.001713   Min.   :0.002252
 1st Qu.:0.2324   1st Qu.:0.8339   1st Qu.: 1.606   1st Qu.: 17.850   1st Qu.:0.005169   1st Qu.:0.013080
 Median :0.3242   Median :1.1080   Median : 2.287   Median : 24.530   Median :0.006380   Median :0.020450
 Mean   :0.4052   Mean   :1.2169   Mean   : 2.866   Mean   : 40.337   Mean   :0.007041   Mean   :0.025478
 3rd Qu.:0.4789   3rd Qu.:1.4740   3rd Qu.: 3.357   3rd Qu.: 45.190   3rd Qu.:0.008146   3rd Qu.:0.032450
 Max.   :2.8730   Max.   :4.8850   Max.   :21.980   Max.   :542.200   Max.   :0.031130   Max.   :0.135400
  concavity2        concave2         symmetry2          fractal2          radius3         texture3
 Min.   :0.00000   Min.   :0.000000   Min.   :0.007882   Min.   :0.0008948   Min.   : 7.93   Min.   :12.02
 1st Qu.:0.01509   1st Qu.:0.007638   1st Qu.:0.015160   1st Qu.:0.0022480   1st Qu.:13.01   1st Qu.:21.08
 Median :0.02589   Median :0.010930   Median :0.018730   Median :0.0031870   Median :14.97   Median :25.41
 Mean   :0.03189   Mean   :0.011796   Mean   :0.020542   Mean   :0.0037949   Mean   :16.27   Mean   :25.68
 3rd Qu.:0.04205   3rd Qu.:0.014710   3rd Qu.:0.023480   3rd Qu.:0.0045580   3rd Qu.:18.79   3rd Qu.:29.72
 Max.   :0.39600   Max.   :0.052790   Max.   :0.078950   Max.   :0.0298400   Max.   :36.04   Max.   :49.54
  perimeter3        area3         smoothness3       compactness3       concavity3        concave3
 Min.   : 50.41   Min.   : 185.2   Min.   :0.07117   Min.   :0.02729   Min.   :0.0000   Min.   :0.00000
 1st Qu.: 84.11   1st Qu.: 515.3   1st Qu.:0.11660   1st Qu.:0.14720   1st Qu.:0.1145   1st Qu.:0.06493
 Median : 97.66   Median : 686.5   Median :0.13130   Median :0.21190   Median :0.2267   Median :0.09993
 Mean   :107.26   Mean   : 880.6   Mean   :0.13237   Mean   :0.25427   Mean   :0.2722   Mean   :0.11461
 3rd Qu.:125.40   3rd Qu.:1084.0   3rd Qu.:0.14600   3rd Qu.:0.33910   3rd Qu.:0.3829   3rd Qu.:0.16140
 Max.   :251.20   Max.   :4254.0   Max.   :0.22260   Max.   :1.05800   Max.   :1.2520   Max.   :0.29100
  symmetry3       fractal3
 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
```

A2 – Descriptive statistics.

```
> describe(wdbc[1:11])
            vars   n   mean     sd median trimmed    mad    min     max   range  skew kurtosis    se
diagnosis*     1 569   1.63   0.48   2.00    1.66   0.00   1.00    2.00    1.00 -0.53    -1.73  0.02
radius         2 569  14.13   3.52  13.37   13.82   2.82   6.98   28.11   21.13  0.94     0.81  0.15
texture        3 569  19.29   4.30  18.84   19.04   4.17   9.71   39.28   29.57  0.65     0.73  0.18
perimeter      4 569  91.97  24.30  86.24   89.74  18.84  43.79  188.50  144.71  0.99     0.94  1.02
area           5 569 654.89 351.91 551.10  606.13 227.28 143.50 2501.00 2357.50  1.64     3.59 14.75
smoothness     6 569   0.10   0.01   0.10    0.10   0.01   0.05    0.16    0.11  0.45     0.82  0.00
compactness    7 569   0.10   0.05   0.09    0.10   0.05   0.02    0.35    0.33  1.18     1.61  0.00
concavity      8 569   0.09   0.08   0.06    0.08   0.06   0.00    0.43    0.43  1.39     1.95  0.00
concave        9 569   0.05   0.04   0.03    0.04   0.03   0.00    0.20    0.20  1.17     1.03  0.00
symmetry      10 569   0.18   0.03   0.18    0.18   0.03   0.11    0.30    0.20  0.72     1.25  0.00
fractal       11 569   0.06   0.01   0.06    0.06   0.01   0.05    0.10    0.05  1.30     2.95  0.00
> describe(wdbc[12:21])
             vars   n  mean    sd median trimmed   mad  min    max  range skew kurtosis   se
radius2         1 569  0.41  0.28   0.32    0.36  0.16 0.11   2.87   2.76 3.07    17.45 0.01
texture2        2 569  1.22  0.55   1.11    1.16  0.47 0.36   4.88   4.52 1.64     5.26 0.02
perimeter2      3 569  2.87  2.02   2.29    2.51  1.14 0.76  21.98  21.22 3.43    21.12 0.08
area2           4 569 40.34 45.49  24.53   31.69 13.63 6.80 542.20 535.40 5.42    48.59 1.91
smoothness2     5 569  0.01  0.00   0.01    0.01  0.00 0.00   0.03   0.03 2.30    10.32 0.00
compactness2    6 569  0.03  0.02   0.02    0.02  0.01 0.00   0.14   0.13 1.89     5.02 0.00
concavity2      7 569  0.03  0.03   0.03    0.03  0.02 0.00   0.40   0.40 5.08    48.24 0.00
concave2        8 569  0.01  0.01   0.01    0.01  0.01 0.00   0.05   0.05 1.44     5.04 0.00
symmetry2       9 569  0.02  0.01   0.02    0.02  0.01 0.01   0.08   0.07 2.18     7.78 0.00
fractal2       10 569  0.00  0.00   0.00    0.00  0.00 0.00   0.03   0.03 3.90    25.94 0.00
> describe(wdbc[22:31])
             vars   n   mean     sd median trimmed    mad    min     max   range skew kurtosis    se
radius3         1 569  16.27   4.83  14.97   15.73   3.65   7.93   36.04   28.11 1.10     0.91  0.20
texture3        2 569  25.68   6.15  25.41   25.39   6.42  12.02   49.54   37.52 0.50     0.20  0.26
perimeter3      3 569 107.26  33.60  97.66  103.42  25.01  50.41  251.20  200.79 1.12     1.04  1.41
area3           4 569 880.58 569.36 686.50  788.02 319.65 185.20 4254.00 4068.80 1.85     4.32 23.87
smoothness3     5 569   0.13   0.02   0.13    0.13   0.02   0.07    0.22    0.15 0.41     0.49  0.00
compactness3    6 569   0.25   0.16   0.21    0.23   0.13   0.03    1.06    1.03 1.47     2.98  0.01
concavity3      7 569   0.27   0.21   0.23    0.25   0.20   0.00    1.25    1.25 1.14     1.57  0.01
concave3        8 569   0.11   0.07   0.10    0.11   0.07   0.00    0.29    0.29 0.49    -0.55  0.00
symmetry3       9 569   0.29   0.06   0.28    0.28   0.05   0.16    0.66    0.51 1.43     4.37  0.00
fractal3       10 569   0.08   0.02   0.08    0.08   0.01   0.06    0.21    0.15 1.65     5.16  0.00
```

A3 – Statistical summary of dataset, separated by planes.

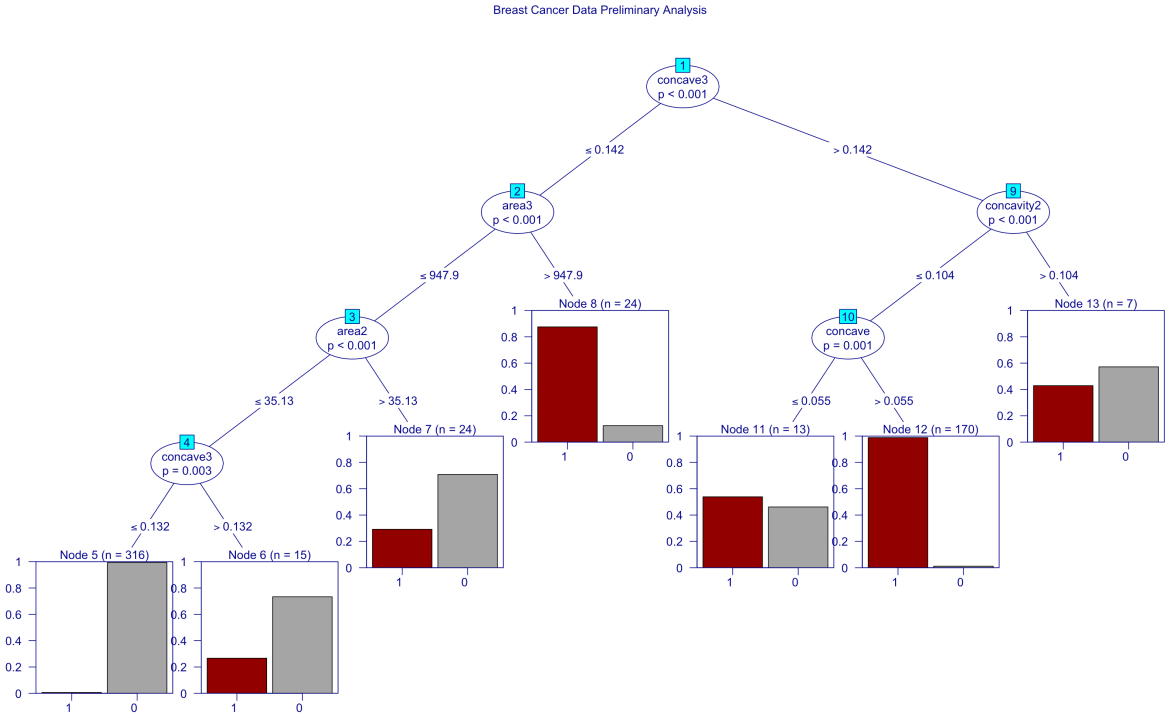
```r
# Explore density of diagnosis across four variables
library(gridExtra)
g1 <- ggplot(data = wdbc)+
  theme_light()+
  geom_density(mapping = aes(radius,  fill = diagnosis), col="darkgrey", show.legend = FALSE, alpha=0.6)
g2 <- ggplot(data = wdbc)+
  theme_light()+
  geom_density(mapping = aes(area, fill = diagnosis), col="darkgrey", show.legend = TRUE, alpha=0.6)
g3 <- ggplot(data = wdbc)+
  theme_light()+
  geom_density(mapping = aes(texture, fill = diagnosis), col="darkgrey", show.legend = FALSE, alpha=0.6)
g4 <- ggplot(data = wdbc)+
  theme_light()+
  geom_density(mapping = aes(concave, fill = diagnosis), col="darkgrey", show.legend = TRUE, alpha=0.6)
grid.arrange(arrangeGrob(g1,  g2,  g3, g4),  nrow=1)
```

A4 – Density code to assess variables: “radius”, “area”, “texture”, and “concave”.

```r
# Test how the actual sample looks like
set.seed(1234)
ind <- sample(2, nrow(wdbc), replace = T, prob = c(1.0, 0.0))
train.data <- wdbc[ind == 1, ]
test.data <- wdbc[ind == 2, ]

# Run the method on a training data
library("party")
library("partykit")
myFormula<-diagnosis~.
model <- ctree(myFormula, data = train.data)

plot.new()
plot(model, type="extended", ep_args = list(justmin =8),
     main ="Breast Cancer Data Preliminary Analysis",
     drop_terminal=F, tnex=1.5,
     gp=gpar(fontsize = 12, col="dark blue"),
     inner_panel = node_inner(model, fill=c("white","cyan"), pval=T),
     terminal_panel=node_barplot(model, fill=c("darkred","darkgray"), beside=T, ymax=1,
                                 just = c(.95,.5), ylines=T, widths = 1.5, gap=0.05, reverse=F, id=T),
     margins = c(2, 4, 4, 2))
```

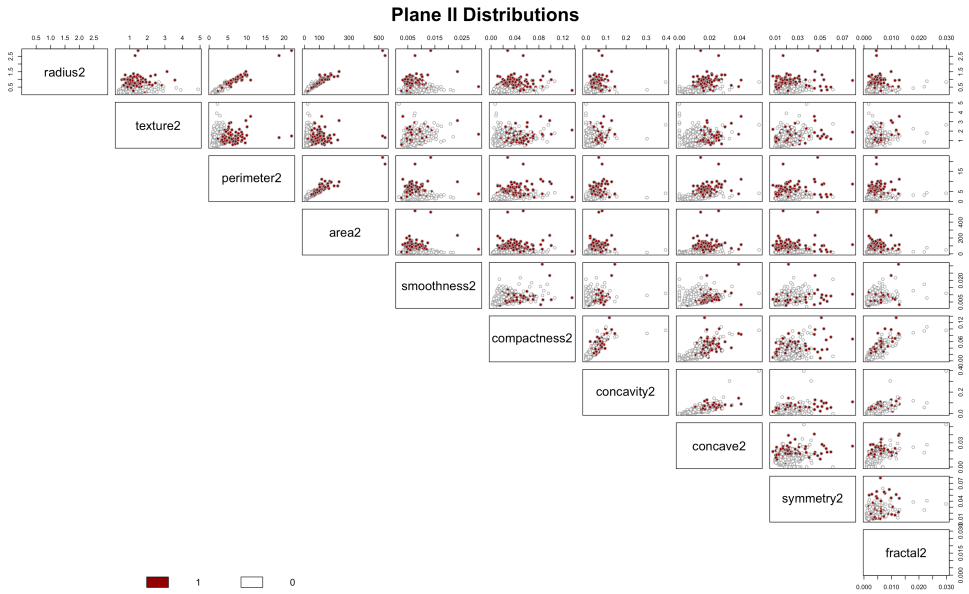


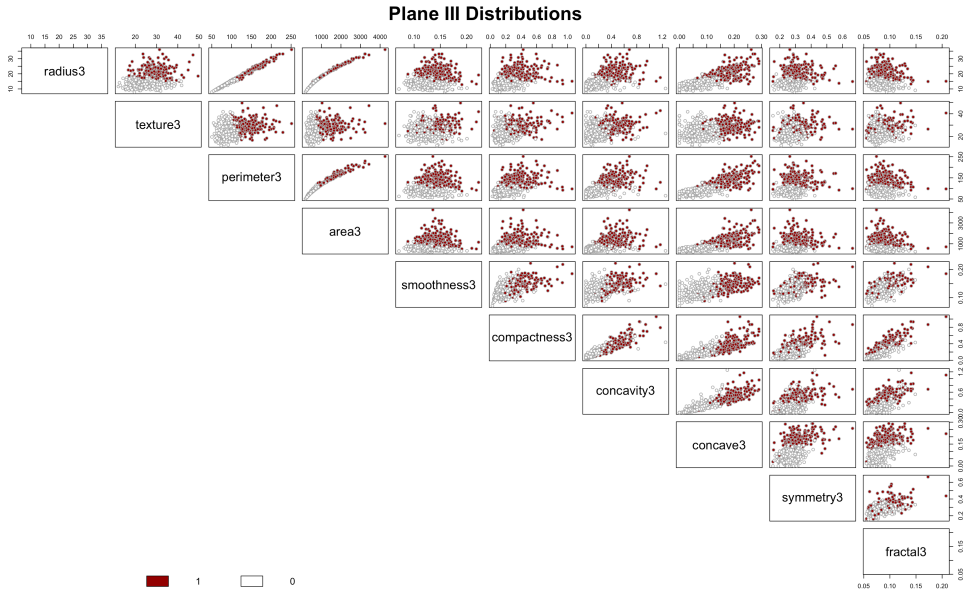
A5 – Code and decision tree representation of original medical diagnosis.

```r
#*********************************************************************
library("graphics")
# Plot Plan I
clrs <- c("darkred", "white")
pairs(wdbc[2:11], fill=clrs, main = "Plane I Distributions", cex.main= 2, cex.labels = 2,
      lower.panel = NULL, pch = 21, col="darkgrey", bg = clrs [unclass(wdbc$diagnosis)])
par (xpd = T)
legend (0.10, 0.01, horiz = TRUE, as.vector(unique(wdbc$diagnosis)),
        fill=clrs, bty = "n")

#*********************************************************************
# Plane II
clrs <- c("darkred", "white")
pairs(wdbc[12:21], fill=clrs, main = "Plane II Distributions", cex.main= 2, cex.labels = 2,
      lower.panel = NULL, pch = 21, col="darkgrey", bg = clrs [unclass(wdbc$diagnosis)])
par (xpd = T)
legend (0.10, 0.01, horiz = TRUE, as.vector(unique(wdbc$diagnosis)),
        fill=clrs, bty = "n")
#*********************************************************************

# Plane III
clrs <- c("darkred", "white")
pairs(wdbc[22:31], fill=clrs, main = "Plane III Distributions", cex.main= 2, cex.labels = 2,
      lower.panel = NULL, pch = 21, col="darkgrey", bg = clrs [unclass(wdbc$diagnosis)])
par (xpd = T)
legend (0.10, 0.01, horiz = TRUE, as.vector(unique(wdbc$diagnosis)),
        fill=clrs, bty = "n")
```





A6 – Code and distribution of Plane II, and Plane III.

```
> result.matrix         # number of training steps,
                                  [,1]
error                      11.001968150
reached.threshold           0.007884844
steps                     432.000000000
aic                       320.003936299
bic                       911.721950428
Intercept.to.1layhid1      -1.459727660
radius.to.1layhid1          0.955747842
texture.to.1layhid1         0.527933206
perimeter.to.1layhid1      -0.187244174
area.to.1layhid1           -0.101176160
smoothness.to.1layhid1      1.000510646
compactness.to.1layhid1    -2.154000657
concavity.to.1layhid1      -0.419687039
concave.to.1layhid1        -0.262918424
symmetry.to.1layhid1        2.043527038
fractal.to.1layhid1         0.395150686
radius2.to.1layhid1         1.265217463
texture2.to.1layhid1        0.343672751
perimeter2.to.1layhid1      0.553446756
area2.to.1layhid1           0.615410849
smoothness2.to.1layhid1    -1.063950233
compactness2.to.1layhid1   -0.837743377
concavity2.to.1layhid1      0.252347655
concave2.to.1layhid1       -0.181439224
symmetry2.to.1layhid1      -0.710624371
fractal2.to.1layhid1       -0.166400656
radius3.to.1layhid1         0.952246789
texture3.to.1layhid1       -1.317760557
perimeter3.to.1layhid1     -0.208979706
area3.to.1layhid1           0.525672809
smoothness3.to.1layhid1     0.060188309
compactness3.to.1layhid1   -1.545942019
concavity3.to.1layhid1      1.036577082
concave3.to.1layhid1        1.587212047
symmetry3.to.1layhid1       1.707676677
fractal3.to.1layhid1        0.350893745
```

A7 – Preview of inputs and first layer iteration using the nn$result.matrix command.