*Create a title page here*

**Neural Network for Breast Cancer Detection, Diagnosis and Classification**

**1. Introduction**

Breast cancer is a significant health concern, particularly affecting women worldwide. It is a complex disease with a high mortality rate, making early and accurate diagnosis of paramount importance. According to the World Health Organization (WHO), breast cancer is the most common cancer among women, with over 2.2 million new cases diagnosed globally in 2020. The high prevalence and impact of breast cancer underscore the need for advanced and efficient diagnostic methods. Early detection and accurate diagnosis are essential to improving survival rates. Mammography and other imaging techniques have been vital tools in this fight, but manual examination and interpretation by medical professionals can be time-consuming and potentially subject to human error. In response to these challenges, we turn to innovative technology and machine learning to develop automated systems that can aid in the detection and diagnosis of breast cancer, thereby saving lives and reducing the burden on healthcare systems.

The primary objective of this study is to assess the Neural Network (NN) Algorithm in the detection and diagnosis of breast cancer. Specifically, we aim to investigate the effectiveness of Neural Network for predicting breast cancer and classifying it into benign and malignant categories. This identification is crucial for early and accurate diagnosis of breast cancer.

The chosen analysis approach involves the application of artificial intelligence and machine learning techniques to the field of breast cancer detection and diagnosis. This approach is highly relevant due to the potential to automate and enhance the accuracy of breast cancer diagnosis, as traditional methods can be time-consuming and subject to human error. Machine learning models, such as artificial neural networks (ANN), have shown promise in analyzing mammography images and improving the efficiency of diagnosis. By assessing these methods, we aim to determine which models offer the highest accuracy and potential for clinical implementation. This research seeks to address a critical need in the healthcare domain by exploring advanced technology for early breast cancer detection, ultimately contributing to improved patient outcomes and survival rates.

By leveraging these state-of-the-art technologies, we aim to provide a more efficient and accurate means of detecting breast cancer at an early stage, which is critical for effective treatment and improved patient outcomes. The automation of breast cancer detection and diagnosis can significantly reduce the time and effort required, enabling healthcare professionals to prioritize high-risk patients and streamline the diagnostic process. Furthermore, our analysis will compare different approaches to determine which model or architecture is the most effective, ultimately contributing to the development of a robust computer-aided diagnosis system. This research aims to advance the field of breast cancer detection and diagnosis, with a focus on harnessing the potential of artificial intelligence to save lives and enhance healthcare outcomes.

**2.1 Understanding the Data**

The dataset for this was selected from the module’s recommended datasets list. Its features were computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. They describe characteristics of the cell nuclei present in the image. There are 569 observations and 31 study variables in total.

**Table 1.0: Description of Data Variables**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Variable Name** | **Type** | **Description** |
| 1 | Diagnosis | Categorical (M = malignant, B = benign) | Benign breast cancer patients  Malignant breast cancer patients |
| 2 | Radius | Continuous | Mean of distances from center to points on the perimeter |
| 3 | Texture | Continuous | Standard deviation of gray-scale values |
| 4 | Perimeter | Continuous | Perimeter |
| 5 | Area | Continuous | Area |
| 6 | Smoothness | Continuous | Local variation in radius lengths |
| 7 | Compactness | Continuous | Perimeter2 / area - 1.0 |
| 8 | Concavity | Continuous | severity of concave portions of the contour) |
| 9 | Concave | Continuous | Number of concave portions of the contour |
| 10 | Symmetry | Continuous | Symmetry |
| 11 | Fractal | Continuous | It is measured using “coastline approximation” -1 |

**2.2. Exploratory Analysis**

Using the ‘str’ function, it was observed that the variables in the dataset consist of numeric majorly, except for the target variable (diagnosis) which is in character form. Now, to ensure that all the target variable is in factor. Encoding was done as shown in Table 1 above.

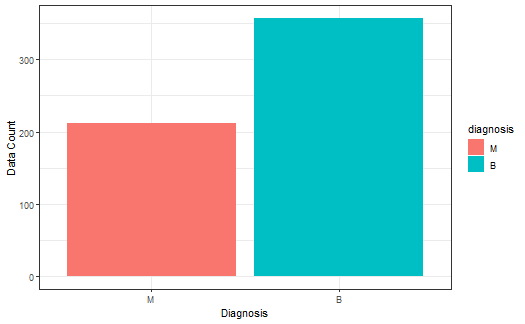
The ‘summary’ of the data shows that there are 212 malignant cancer data and 357 benign cancer data, in the ratio 0.37:0.63 indicating a slight unbalance in the dataset. Further, the minimum and maximum, as well as measures of central tendency (mean, median) and spread (1st and 3rd quartiles) for each of these variables were obtained

**Table 1.0: Frequency table for Diagnosis**

|  |  |  |
| --- | --- | --- |
|  | **M** | **B** |
| **Frequency** | 212 | 357 |
| **Proportion** | 0.3726 | 0.6274 |

**Variable Analysis**

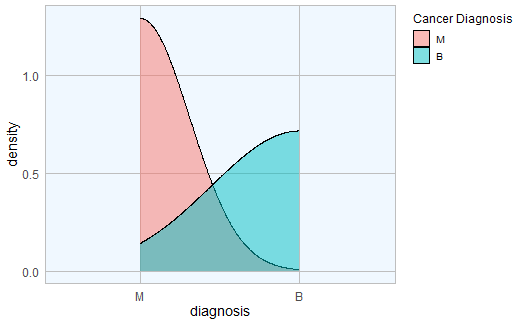
The number of people diagnosed with benign cancer is more than with a number of people that is diagnosed with malignant. We will perform further analysis to find out more about the relevant parameter that contributes to the diagnosis.



**Figure 2.1: Diagnosis as Malignant vs Benign**

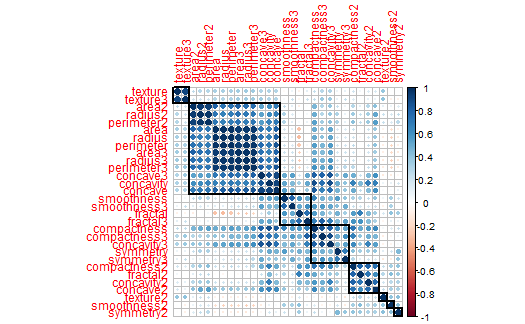
**Density Plot**

We can gather that the majority of the dataset lies with the malignant and the least with the benign.



**Figure 2.2: Density graph for breast cancer diagnosis**

**Correlation**

Correlations is used to determine whether and to what extent two or more variables are related linearly. Pearson's Correlation Coefficient, one of the most commonly used correlation measures, provides a numerical value that represents the strength and direction of this linear relationship. The correlation matrix in the figure below shows that there is a great correlation between some variables:

**Figure 2.3: Correlation Matrix of study variables**

**2.3. Preprocessing**

**Check for missing data**

Next, we check how many NA records we have, per column. Fortunately, there are no records of missing data in the dataset.

**Data Splitting** *- Splitting the data set into Train and Test*

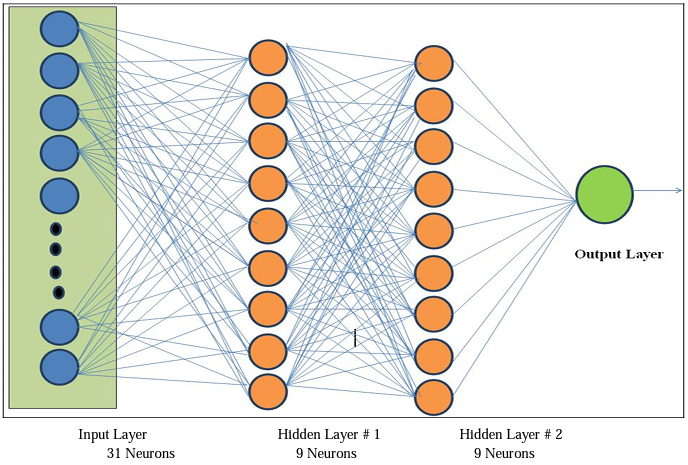
Data Splitting Functions A series of test/training partitions are created using createDataPartition while createResample creates one or more bootstrap samples.

**2.4. Neural Network**

Neural networks (NN), a branch of machine learning also known as artificial neural networks (ANN), are computational models—essentially algorithms. These networks possess a remarkable capacity to derive meaning from imprecise or complex data, uncovering intricate patterns and identifying trends that may elude human comprehension or other conventional computer techniques. Neural networks have revolutionized our daily lives in numerous ways, exemplified by their integration into ridesharing apps, Gmail's intelligent email sorting, and product recommendations on platforms like Amazon.

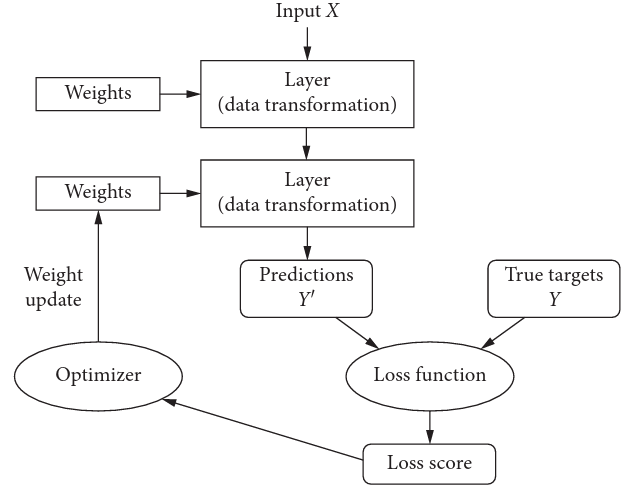
One of the most groundbreaking features of neural networks is their ability to learn autonomously. These characteristic parallels the human brain, which comprises neurons—the fundamental units for transmitting information in both biological brains and neural networks. Alex Cardinell, Founder and CEO of Cortx, an artificial intelligence company specializing in natural language processing solutions, including an automated grammar correction application called Perfect Tense, points out, Human brains and artificial neural networks share similarities in their learning processes. In both cases, neurons continuously adjust their responses based on stimuli. When a task is executed correctly, neurons receive positive feedback and become more likely to trigger in similar future instances. Conversely, if neurons receive negative feedback, they learn to be less likely to trigger in subsequent instances.

The whole of neurons in the input layer of the NN is equal to the number of characteristics in the dataset in its architecture. The hidden layer is another network component, with the number of hidden layers being counted as one layer. The NN architecture is illustrated below.



**Figure 2.4: Neural Network algorithm**

A detailed visual explanation of neural networks (NNs) is shown in Figure 2.5.



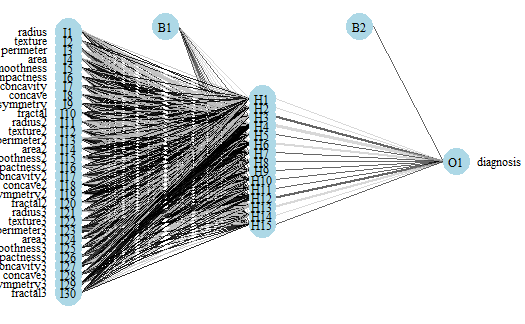
**Figure 2.5: Detailed process of a neural network (NN)**

**2.5. Model Fitting**

The dataset was divided into training and test set using R code, after which we perform the neural network model and evaluate the confusion matrix with model methods such as sensitivity, specificity, positive prediction value, negative prediction value, and prevalence of the data.

**3. Results**

**3.1 Output**

****

**Figure 3.0: Neural Network Algorithm for Breast Cancer Diagnosis**

The NN architecture has been employed in breast cancer diagnosis and classification because of its ability of feature extraction that can be used to enhance and easily see malignancy in breast masses. Thus, aiding in the process of early detection of breast cancer, so that it can be treated at a lower stage, before it spreads more.

| **Predicted diagnosis**

**Actual diagnosis** | B | M | Row Total |

-----------------|-----------|-----------|-----------|

M | 10 | 70 | 80 |

| 0.047 | 0.332 | |

-----------------|-----------|-----------|-----------|

B | 128 | 3 | 131 |

| 0.607 | 0.014 | |

-----------------|-----------|-----------|-----------|

Column Total | 138 | 73 | 211 |

-----------------|-----------|-----------|-----------|

**3.2. Model Properties**

After experimenting with the selected parameters in the training process and system testing, the accuracy level is 93.84%, and the error is 6.16% i.e 94% of the test values are correctly classified, and misclassification rate is around 6%. The rate at which there was no information, i.e., "No Information Rate" is 62%.

This p-value is associated with a hypothesis test comparing the model's accuracy to the accuracy achieved by the "No Information Rate." In this case, the p-value is 0.00, which suggests that there is statistically significant difference between the model's accuracy and the "No Information Rate."

**3.3. Evaluation of the model**

The confusion matrix, which is often used to evaluate the performance of a classification model is provided below. In a confusion matrix, the rows represent the predicted classes, while the columns represent the actual classes or categories. Each cell in the matrix represents the number of observations that fall into a particular combination of actual and predicted diagnosis.

In this case, the model correctly classified 70 instances as Malignant "M" (True Positives), and it correctly classified 128 instances as Benign "B" (True Negatives). However, it also made 3 false positive errors (classified "B" as "M") and 10 false negative errors (classified "M" as "B")..

**Table 3.0: Confusion Matrix**

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | *Actual* | |
|  |  | M | B |
| *Prediction* | M | 70 | 3 |
| B | 10 | 128 |

The model is observed to be 93.8% accurate, with 87.5% and 97.7% Sensitivity and Specificity respectively. The prevalence of the data is recorded to be 37.9%.

**4. Conclusion**

**4.1 Summary**

Automating the detection of breast cancer to enhance the care of patients is a challenging task. The current study proposes a NN approach that analyzes the diagnosis and the automatic detection of breast cancer. The proposed model was found to successfully obtain correct results that might decrease human mistakes in the diagnosis process and reduce the cost of cancer diagnosis. The approach presented in this study achieved an accuracy of 94%. The use of multiple NN models in a meta-learning framework allowed for better generalization and improved accuracy, particularly in detecting malignant tumors. The approach in medical imaging datasets could be extended to other types of cancer or medical conditions. The model produced by NN is more and it has the potential to make essential advancements in breast cancer prediction. Based on these findings, we can infer that machine learning techniques can automatically detect the disease with high accuracy.

**4.2 Limitations of the study**

* The main limitation of this study is to use the secondary database, future study should be done based on primary data for more accuracy of the results related to breast cancer identification.
* It is incredibly costly and time-consuming to conduct a medical diagnostic process in the area of medicine.
* Furthermore, the dataset used in the study is limited in terms of the number of samples and the diversity of the cases, this may result in a loss of valuable information that could have contributed to a more comprehensive understanding of the study.
* It is hard to explain the detail of each attribute because there is no reference about this dissertation. Because of that, we maybe make some misunderstanding to attributes.
* The outcomes may change by the period and environment when the dataset is collected.

**4.3 Recommendation/Improvement Areas:**

* In terms of future work, we suggest several avenues for further research. One potential direction is to explore the use of other meta-learning algorithms and compare their performance with the approach presented in this study.
* Another direction is to investigate the impact of incorporating clinical data, such as patient history or biopsy results, into the classification model.
* In the future, designing novel architecture can help reduce the number of trainable parameters while maintaining or improving performance. The idea is to find simpler, more efficient structures that can capture the underlying patterns in the data with fewer parameters.
* Machine learning techniques may be used as a clinical assistant to detect breast cancer, which will be very beneficial for new doctors of a physician in the event of a misdiagnosis.
* There should be increased efforts to ensure adequate keeping of information or updating of database of breast cancer for future analysis.
* Efforts should be made to increase awareness and proper documentation of other factors that may contribute to the diagnosis of breast cancer.

library(ggplot2)  
library(corrplot)

## corrplot 0.92 loaded

library(tidyverse)

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.3 ✔ readr 2.1.4  
## ✔ forcats 1.0.0 ✔ stringr 1.5.0  
## ✔ lubridate 1.9.3 ✔ tibble 3.2.1  
## ✔ purrr 1.0.2 ✔ tidyr 1.3.0

## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

library(dplyr)  
library(caret)

## Loading required package: lattice  
##   
## Attaching package: 'caret'  
##   
## The following object is masked from 'package:purrr':  
##   
## lift

library(nnet)  
library(NeuralNetTools)  
library(gmodels)

nn <- read.csv("wdbc.csv")

str(nn)

## 'data.frame': 569 obs. of 32 variables:  
## $ ID : int 842302 842517 84300903 84348301 84358402 843786 844359 84458202 844981 84501001 ...  
## $ 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 ...

# Transforming the target variable that I will use to submit to the categorical algorithm into factors and removing the ID and odd target variable  
diagnosis = factor(nn$diagnosis, levels = c("M", "B"), labels = c("M", "B"))  
data = nn[,-1]  
data = data[,-1]  
  
# Adding the newly re-coded variable to the data  
nndata <- cbind(diagnosis, data)

summary(nndata)

## diagnosis radius texture perimeter area   
## M:212 Min. : 6.981 Min. : 9.71 Min. : 43.79 Min. : 143.5   
## B:357 1st Qu.:11.700 1st Qu.:16.17 1st Qu.: 75.17 1st Qu.: 420.3   
## Median :13.370 Median :18.84 Median : 86.24 Median : 551.1   
## Mean :14.127 Mean :19.29 Mean : 91.97 Mean : 654.9   
## 3rd Qu.:15.780 3rd Qu.:21.80 3rd Qu.:104.10 3rd Qu.: 782.7   
## Max. :28.110 Max. :39.28 Max. :188.50 Max. :2501.0   
## smoothness compactness concavity concave   
## Min. :0.05263 Min. :0.01938 Min. :0.00000 Min. :0.00000   
## 1st Qu.:0.08637 1st Qu.:0.06492 1st Qu.:0.02956 1st Qu.:0.02031   
## Median :0.09587 Median :0.09263 Median :0.06154 Median :0.03350   
## Mean :0.09636 Mean :0.10434 Mean :0.08880 Mean :0.04892   
## 3rd Qu.:0.10530 3rd Qu.:0.13040 3rd Qu.:0.13070 3rd Qu.:0.07400   
## Max. :0.16340 Max. :0.34540 Max. :0.42680 Max. :0.20120   
## symmetry fractal radius2 texture2   
## Min. :0.1060 Min. :0.04996 Min. :0.1115 Min. :0.3602   
## 1st Qu.:0.1619 1st Qu.:0.05770 1st Qu.:0.2324 1st Qu.:0.8339   
## Median :0.1792 Median :0.06154 Median :0.3242 Median :1.1080   
## Mean :0.1812 Mean :0.06280 Mean :0.4052 Mean :1.2169   
## 3rd Qu.:0.1957 3rd Qu.:0.06612 3rd Qu.:0.4789 3rd Qu.:1.4740   
## Max. :0.3040 Max. :0.09744 Max. :2.8730 Max. :4.8850   
## perimeter2 area2 smoothness2 compactness2   
## Min. : 0.757 Min. : 6.802 Min. :0.001713 Min. :0.002252   
## 1st Qu.: 1.606 1st Qu.: 17.850 1st Qu.:0.005169 1st Qu.:0.013080   
## Median : 2.287 Median : 24.530 Median :0.006380 Median :0.020450   
## Mean : 2.866 Mean : 40.337 Mean :0.007041 Mean :0.025478   
## 3rd Qu.: 3.357 3rd Qu.: 45.190 3rd Qu.:0.008146 3rd Qu.:0.032450   
## Max. :21.980 Max. :542.200 Max. :0.031130 Max. :0.135400   
## concavity2 concave2 symmetry2 fractal2   
## Min. :0.00000 Min. :0.000000 Min. :0.007882 Min. :0.0008948   
## 1st Qu.:0.01509 1st Qu.:0.007638 1st Qu.:0.015160 1st Qu.:0.0022480   
## Median :0.02589 Median :0.010930 Median :0.018730 Median :0.0031870   
## Mean :0.03189 Mean :0.011796 Mean :0.020542 Mean :0.0037949   
## 3rd Qu.:0.04205 3rd Qu.:0.014710 3rd Qu.:0.023480 3rd Qu.:0.0045580   
## Max. :0.39600 Max. :0.052790 Max. :0.078950 Max. :0.0298400   
## radius3 texture3 perimeter3 area3   
## Min. : 7.93 Min. :12.02 Min. : 50.41 Min. : 185.2   
## 1st Qu.:13.01 1st Qu.:21.08 1st Qu.: 84.11 1st Qu.: 515.3   
## Median :14.97 Median :25.41 Median : 97.66 Median : 686.5   
## Mean :16.27 Mean :25.68 Mean :107.26 Mean : 880.6   
## 3rd Qu.:18.79 3rd Qu.:29.72 3rd Qu.:125.40 3rd Qu.:1084.0   
## Max. :36.04 Max. :49.54 Max. :251.20 Max. :4254.0   
## smoothness3 compactness3 concavity3 concave3   
## Min. :0.07117 Min. :0.02729 Min. :0.0000 Min. :0.00000   
## 1st Qu.:0.11660 1st Qu.:0.14720 1st Qu.:0.1145 1st Qu.:0.06493   
## Median :0.13130 Median :0.21190 Median :0.2267 Median :0.09993   
## Mean :0.13237 Mean :0.25427 Mean :0.2722 Mean :0.11461   
## 3rd Qu.:0.14600 3rd Qu.:0.33910 3rd Qu.:0.3829 3rd Qu.:0.16140   
## Max. :0.22260 Max. :1.05800 Max. :1.2520 Max. :0.29100   
## 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

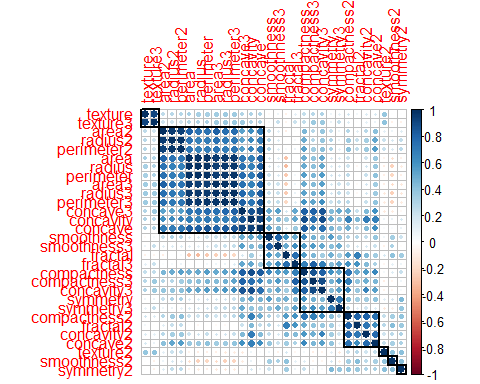
prop.table(table(nndata$diagnosis))

##   
## M B   
## 0.3725835 0.6274165

ggplot(nndata, aes(x=diagnosis, fill = diagnosis))+   
 theme\_bw()+  
 geom\_bar()+  
 labs(x = "Diagnosis", y = "Data Count")

# Density graph for TBSA (Total burn surface area)  
ggplot(nndata, aes(x = diagnosis, fill = diagnosis)) +  
 geom\_density(alpha=0.5) +  
 scale\_fill\_discrete(name = "Cancer Diagnosis", labels = c("M", "B"))

corr\_mat <- cor(nndata[,2:ncol(nndata)])  
corrplot(corr\_mat, order = "hclust", tl.cex = 1, addrect = 8)



nndata %>%  
 summarise\_all(  
 funs(sum(is.na(.)))  
 )

## Warning: `funs()` was deprecated in dplyr 0.8.0.  
## ℹ Please use a list of either functions or lambdas:  
##   
## # Simple named list: list(mean = mean, median = median)  
##   
## # Auto named with `tibble::lst()`: tibble::lst(mean, median)  
##   
## # Using lambdas list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))  
## Call `lifecycle::last\_lifecycle\_warnings()` to see where this warning was  
## generated.

## diagnosis radius texture perimeter area smoothness compactness concavity  
## 1 0 0 0 0 0 0 0 0  
## concave symmetry fractal radius2 texture2 perimeter2 area2 smoothness2  
## 1 0 0 0 0 0 0 0 0  
## compactness2 concavity2 concave2 symmetry2 fractal2 radius3 texture3  
## 1 0 0 0 0 0 0 0  
## perimeter3 area3 smoothness3 compactness3 concavity3 concave3 symmetry3  
## 1 0 0 0 0 0 0 0  
## fractal3  
## 1 0

## Splitting the data set into Train and Test with 60 and 40 percent respectively

index <- sample(2, nrow(nndata), replace=TRUE, prob = c(0.60, 0.40))  
traindata <- nndata[index==1, ]  
testdata <- nndata[index==2, ]

## Create Neural network model

model\_nnet <- nnet(diagnosis ~ ., data = traindata, size=15, rang = 1, decay = 8e-4, maxit = 200)

## # weights: 481  
## initial value 458.555789   
## iter 10 value 150.238608  
## iter 20 value 93.884341  
## iter 30 value 93.522835  
## iter 40 value 68.166960  
## iter 50 value 55.336683  
## iter 60 value 51.127621  
## iter 70 value 50.397121  
## iter 80 value 50.276321  
## iter 90 value 49.717312  
## iter 100 value 48.082009  
## iter 110 value 44.721274  
## iter 120 value 39.492581  
## iter 130 value 38.725659  
## iter 140 value 37.938664  
## iter 150 value 37.777069  
## iter 160 value 37.277396  
## iter 170 value 36.095843  
## iter 180 value 30.714087  
## iter 190 value 24.999692  
## iter 200 value 24.116981  
## final value 24.116981   
## stopped after 200 iterations

## Plot a neural interpretation diagram for a neural network object

par(mar = numeric(4), family = 'serif')  
plotnet(model\_nnet, alpha = 0.6)

## Predict

pred\_nnet <- predict(model\_nnet, testdata,type = c("class"))

## Accuracy

accuracy <- sum(pred\_nnet == testdata$diagnosis)/nrow(testdata)  
accuracy

## [1] 0.9237668

## Create cross table to summarize the result

CrossTable(testdata$diagnosis, pred\_nnet, prop.chisq = FALSE,  
 prop.c = FALSE, prop.r = FALSE, dnn = c("Actual diagnosis",  
 "Predicted diagnosis"))

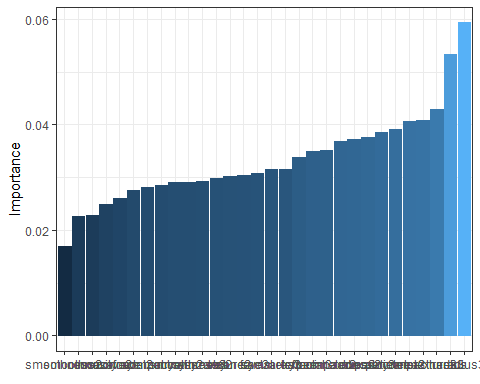
##   
##   
## Cell Contents  
## |-------------------------|  
## | N |  
## | N / Table Total |  
## |-------------------------|  
##   
##   
## Total Observations in Table: 223   
##   
##   
## | Predicted diagnosis   
## Actual diagnosis | B | M | Row Total |   
## -----------------|-----------|-----------|-----------|  
## M | 10 | 74 | 84 |   
## | 0.045 | 0.332 | |   
## -----------------|-----------|-----------|-----------|  
## B | 132 | 7 | 139 |   
## | 0.592 | 0.031 | |   
## -----------------|-----------|-----------|-----------|  
## Column Total | 142 | 81 | 223 |   
## -----------------|-----------|-----------|-----------|  
##   
##

pred\_nnet = factor(pred\_nnet, levels = c("M", "B"), labels = c("M", "B"))  
cm\_nnet <- confusionMatrix(pred\_nnet, testdata$diagnosis, positive = "M")  
cm\_nnet

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction M B  
## M 74 7  
## B 10 132  
##   
## Accuracy : 0.9238   
## 95% CI : (0.8807, 0.955)  
## No Information Rate : 0.6233   
## P-Value [Acc > NIR] : <2e-16   
##   
## Kappa : 0.8365   
##   
## Mcnemar's Test P-Value : 0.6276   
##   
## Sensitivity : 0.8810   
## Specificity : 0.9496   
## Pos Pred Value : 0.9136   
## Neg Pred Value : 0.9296   
## Prevalence : 0.3767   
## Detection Rate : 0.3318   
## Detection Prevalence : 0.3632   
## Balanced Accuracy : 0.9153   
##   
## 'Positive' Class : M   
##

## Relative importance of input variables in neural networks using Garson’s algorithm

garson(model\_nnet)



lekprofile(model\_nnet)

