Breast Cancer Prediction

Submitted in Partial Fulfillment of the requirements

for the Award of the Degree of

Masters of Science

in

Big Data Analytics

by

Gayatri Krishna : 21BDA16

Under the Supervision of

Jayati Kaushik

Associate Professor

Department of Advanced Computing

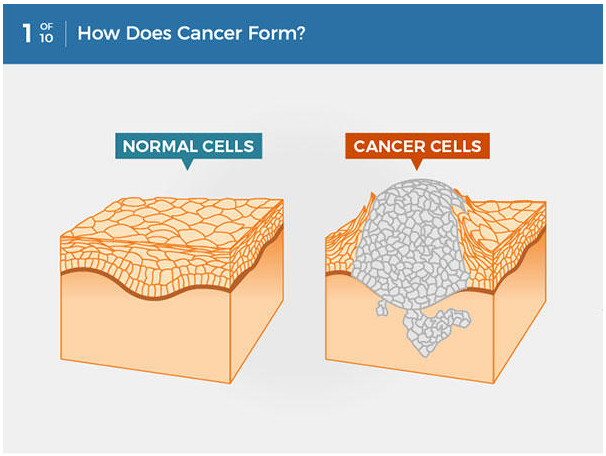
June 2022

Contents

Chapter 1

INTRODUCTION

Cancer is a disease that seriously threatens human health. Cancer is a condition in which some cells in the body develop uncontrollably and spread to other parts of the body. Cancer can develop practically anyplace in the human body, which contains trillions of cells. Human cells normally develop and multiply (a process known as cell division) to generate new cells as the body requires them. When cells age or get damaged, they die, and new cells replace them.

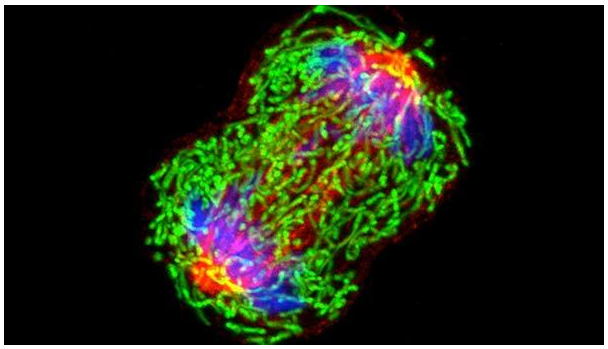


Normal Cell v/s Cancer Cell

When this ordered mechanism fails,damaged cells grow and reproduce when they should not. These cells can combine to produce tumours, which are tissue masses. Tumors can be malignant (cancerous) or benign (not cancerous).Cancerous tumours infiltrate neighbouring tissues and can move to distant locations in the body to produce new tumours. This process is called metastasis.

1.1 Breast Cancer

Breast cancer is the leading cancer among females. Breast cancer is the most prevalent malignant tumour in women. It is the second leading cause of mortality. Breast cancer is cancer of the breast tissue, and symptoms include breast lumps, epidermal tissue dimples, form changes, and red plaques on the epidermis.



Division of Breast Tissues

Cancer can produce osteocope, enlarged lymph nodes, and dyspnea if it spreads. Early detection and diagnosis of breast cancer can assist medical workers in providing suitable treatments or post-surgery relapse monitoring.

1.2 Diagnosing Breast Cancer

Breast cancer is typically diagnosed using a fine-needle aspiration cell technique. The degree of canceration can be determined by observing the abnormal cell morphology of the collected tissue sections under the light microscope. The following tests and methods are used to diagnose breast cancer:

- Breast exam: The doctor will examine both of your breasts as well as the lymph nodes in your armpit, feeling for lumps or other abnormalities.
- Mammogram: Breast cancer screening is a frequent practise. If an abnormality is discovered on a screening mammography, your doctor may advise you to get a diagnostic mammogram to further assess the abnormality.

- Biopsy: Biopsy samples are sent to a laboratory for testing to determine whether the cells are malignant. A biopsy sample is also evaluated to establish the type of cells involved in breast cancer, the tumour's grade, and if the cancer cells have hormone receptors or other receptors that may influence your treatment options.
- Ultrasound: Ultrasound image create images of structures deep within the body by using sound waves. A new breast lump can be ultra-sounded to establish whether it is a solid mass or a fluid-filled cyst.
- MRI: An MRI machine creates images of the interior of your breast using a magnet and radio waves.

1.3 Problem Statement

Our goal here is to forecast whether or not a woman has breast cancer. There are several classification methods employed, and the accuracy of each approach is determined. This will assist us in determining which model will provide the most accurate results.

Keywords:-Breast cancer,malign type,benign type cancer, cancer,supervised learning technique,Logistic regression, Decision Tree, Random Forest, Accuracy,Prediction

Chapter 2

DATASET

2.1 About the Data

A fine needle aspirate (FNA) of a breast lump is used to generate the features in a digital image. They characterize the traits of the visible cell nuclei in the picture.

Creators:-

- Dr. William H. Wolberg, General Surgery Dept. University of Wisconsin
- W. Nick Street, Computer Sciences Dept. University of Wisconsin
- Olvi L. Mangasarian, Computer Sciences Dept. University of Wisconsin

Data Set Characteristics	Associated Tasks	Observations	Attributes	Missing Values
Multivariate	Classification	569	32	No

2.2 Attribute Information

There are 32 attributes and 569 observations in the dataset. The diagnosis, which comes in two types, is the dependent variable. The dataset's attributes are its characteristics. The qualities are as follows:

- ID number
- Diagnosis (M = malignant, B = benign)
- Ten real-valued features are computed for each cell nucleus:
  - radius (mean of distances from center to points on the perimeter)
  - texture (standard deviation of gray-scale values)
  - perimeter
  - area
  - smoothness (local variation in radius lengths)
  - compactness (perimeter^2 / area - 1.0)
  - concavity (severity of concave portions of the contour)
  - concave points (number of concave portions of the contour)
  - symmetry
  - fractal dimension ("coastline approximation" - 1)

Note:- All feature values are recorded with four significant digits.

id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	symmetry_mean	fractal_dimension_mean	radius_se	texture_se	perimeter_se	area_se	smoothness_se	compactness_se	concavity_se	concave points_se	symmetry_se	fractal_dimension_se	radius_worst	texture_worst	perimeter_worst	area_worst	smoothness_worst	compactness_worst	concavity_worst	concave points_worst	symmetry_worst	fractal_dimension_worst
842302	M	17.99	10.38	122.8	1001	0.1184	0.2776	0.3001	0.1471	0.2419	0.07871	1.095	0.9053	8.589	153.4	0.0064	0.049	0.0537	0.0587	0.03	0.006193	25.38	17.33	184.6	2019	0.1622	0.6656	0.7119	0.2654	0.4601	0.1189
842517	M	20.57	17.77	132.9	1326	0.08474	0.07864	0.0869	0.07017	0.1812	0.05667	0.5435	0.7339	3.398	74.08	0.0052	0.0131	0.0186	0.0134	0.0139	0.003532	24.99	23.41	158.8	1956	0.1238	0.1866	0.2416	0.186	0.275	0.08902
8E+07	M	19.69	21.25	130	1203	0.1096	0.1599	0.1974	0.1279	0.2069	0.05999	0.7456	0.7869	4.585	94.03	0.0062	0.0401	0.0383	0.02058	0.0225	0.004571	23.57	25.53	152.5	1709	0.1444	0.4245	0.4504	0.243	0.3613	0.08758
8E+07	M	11.42	20.38	77.58	386.1	0.1425	0.2839	0.2414	0.1052	0.2597	0.09744	0.4956	1.156	3.445	27.23	0.0091	0.0746	0.0566	0.01867	0.0596	0.009208	14.91	26.5	98.87	567.7	0.2098	0.8663	0.6869	0.2575	0.6638	0.173
8E+07	M	20.29	14.34	135.1	1297	0.1003	0.1328	0.198	0.1043	0.1809	0.05883	0.7572	0.7813	5.438	94.44	0.0115	0.0246	0.0569	0.01885	0.0176	0.005115	22.54	16.67	152.2	1575	0.1374	0.205	0.4	0.1625	0.2364	0.07678
843786	M	12.45	15.7	82.57	477.1	0.1278	0.17	0.1578	0.08089	0.2087	0.07613	0.3345	0.8902	2.217	27.19	0.0075	0.0335	0.0367	0.01137	0.0217	0.005082	15.47	23.75	103.4	741.6	0.1791	0.5249	0.5355	0.1741	0.3985	0.1244
844359	M	18.25	19.98	119.6	1040	0.09463	0.109	0.1127	0.074	0.1794	0.05742	0.4467	0.7732	3.18	53.91	0.0043	0.0138	0.0225	0.01039	0.0137	0.002179	22.88	27.66	153.2	1606	0.1442	0.2576	0.3784	0.1932	0.3063	0.08368

Glimpse of the Dataset

2.3 Source of Data

The UCI Machine Learning Repository was used to get the data, `https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+\%28Diagnostic\%29`.

Chapter 3

ANALYSIS

The software used here is R.

Libraries Used:

```
library(tidyverse)
library(ggcorrplot)
library(lattice)
library(psych)
library(DataExplorer)
library(car)
library(caret)
library(scales)
library(modelr)
library(broom)
library(cowplot)
library(corrplot)
library(pROC)
library(caTools)
library(superml)
library(ggplot2)
library(GGally)
library(rmarkdown)
library(lattice)
library(gclus)
library(dplyr)
library(plotly)
library(MLmetrics)
library(plotROC)
library(caret)
options(warn=-1)
```

3.1 Data Pre-processing

Data preparation includes data prepossessing, which is any type of processing done on raw data to get it ready for another data processing technique. The first task was to check for missing

	id	diagnosis	texture_mean	area_mean	symmetry_mean	texture_se	smoothness_se	symmetry_se	fractal_dimension_se	smoothness_worst	symmetry_worst	fractal_dimension_worst
1	842302	1	10.38	1001.0	0.2419	0.9053	0.006399	0.030030	0.006193	0.16220	0.4601	0.11890
2	842517	1	17.77	1326.0	0.1812	0.7339	0.005225	0.013890	0.003532	0.12380	0.2750	0.08902
3	84300903	1	21.25	1203.0	0.2069	0.7869	0.006150	0.022500	0.004571	0.14440	0.3613	0.08758
4	84348301	1	20.38	386.1	0.2597	1.1560	0.009110	0.059630	0.009208	0.20980	0.6638	0.17300
5	84358402	1	14.34	1297.0	0.1809	0.7813	0.011490	0.017560	0.005115	0.13740	0.2364	0.07678

values and it was observed that there were no missing values in the dataset.

Data type is an attribute of a piece of data that instructs a computer system how to interpret its value. Knowing the different sorts of data helps to ensure that each property's value is as expected and that data is collected in the correct format.

```
 $ id                    : int  842302 842517 84300903 84348301 84358402 843786 844359 84458202 844981
84501001 ...
 $ diagnosis             : chr  "M" "M" "M" "M" ...
 $ radius_mean           : num  18 20.6 19.7 11.4 20.3 ...
 $ texture_mean          : num  10.4 17.8 21.2 20.4 14.3 ...
 $ perimeter_mean        : num  122.8 132.9 130 77.6 135.1 ...
 $ area_mean             : num  1001 1326 1203 386 1297 ...
 $ smoothness_mean       : num  0.1184 0.0847 0.1096 0.1425 0.1003 ...
 $ compactness_mean      : num  0.2776 0.0786 0.1599 0.2839 0.1328 ...
 $ concavity_mean        : num  0.3001 0.0869 0.1974 0.2414 0.198 ...
 $ concave.points_mean   : num  0.1471 0.0702 0.1279 0.1052 0.1043 ...
 $ symmetry_mean         : num  0.242 0.181 0.207 0.26 0.181 ...
 $ fractal_dimension_mean : num  0.0787 0.0567 0.06 0.0974 0.0588 ...
 $ radius_se             : num  1.095 0.543 0.746 0.496 0.757 ...
 $ texture_se            : num  0.905 0.734 0.787 1.156 0.781 ...
 $ perimeter_se          : num  8.59 3.4 4.58 3.44 5.44 ...
 $ area_se               : num  153.4 74.1 94 27.2 94.4 ...
 $ smoothness_se         : num  0.0064 0.00522 0.00615 0.00911 0.01149 ...
 $ compactness_se        : num  0.049 0.0131 0.0401 0.0746 0.0246 ...
 $ concavity_se          : num  0.0537 0.0186 0.0383 0.0566 0.0569 ...
 $ concave.points_se     : num  0.0159 0.0134 0.0206 0.0187 0.0188 ...
 $ symmetry_se           : num  0.03 0.0139 0.0225 0.0596 0.0176 ...
 $ fractal_dimension_se  : num  0.00619 0.00353 0.00457 0.00921 0.00511 ...
 $ radius_worst          : num  25.4 25 23.6 14.9 22.5 ...
 $ texture_worst         : num  17.3 23.4 25.5 26.5 16.7 ...
 $ perimeter_worst       : num  184.6 158.8 152.5 98.9 152.2 ...
 $ area_worst            : num  2019 1956 1709 568 1575 ...
 $ smoothness_worst      : num  0.162 0.124 0.144 0.21 0.137 ...
 $ compactness_worst     : num  0.666 0.187 0.424 0.866 0.205 ...
 $ concavity_worst       : num  0.712 0.242 0.45 0.687 0.4 ...
 $ concave.points_worst  : num  0.265 0.186 0.243 0.258 0.163 ...
 $ symmetry_worst        : num  0.46 0.275 0.361 0.664 0.236 ...
 $ fractal_dimension_worst: num  0.1189 0.089 0.0876 0.173 0.0768 ...
```

Structure of the Dataset

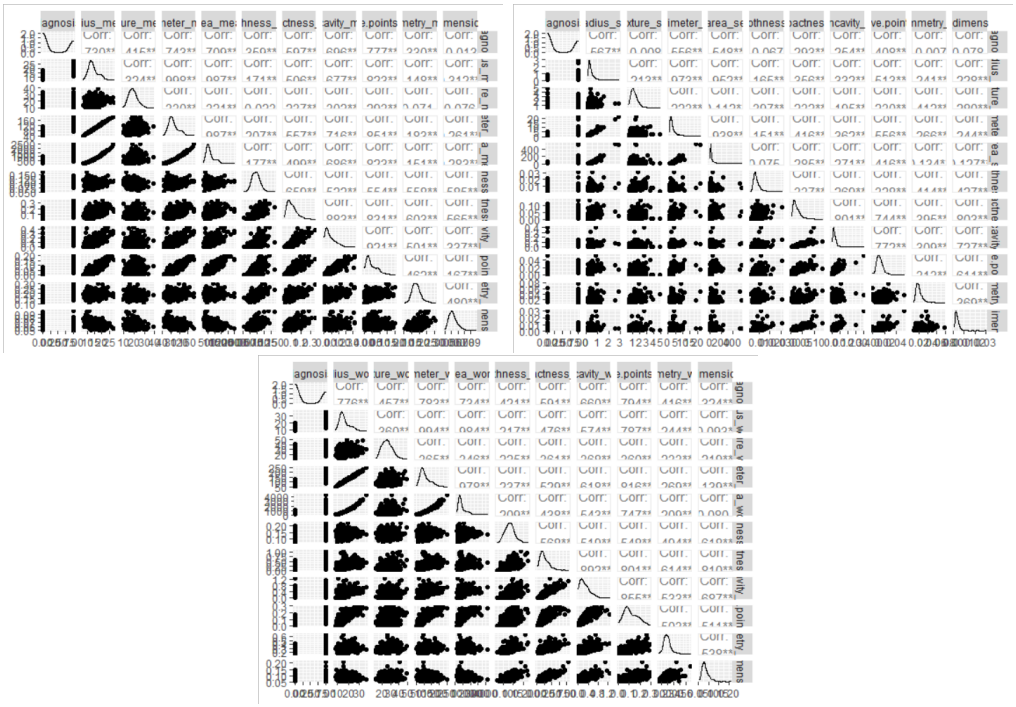
A machine learning algorithm works well with numerical data. It is crucial to label encode the categorical attribute in order to transform it into a numerical one. Most attributes, it has been found, have numerical data types, but the dependent variable, diagnosis, has character data types. The diagnosis attribute is classified as a binary class. The **"superml"** library provided by R has numerous functions to apply Label Encoder to the data. Here the label Encoder converted into numeric format (ie) 0 and 1, where 1 denotes Malignant and 0 denotes Benign.

3.2 Exploratory Data Analysis

Exploratory data analysis is a way of evaluating data sets to highlight their key features, frequently utilising statistical graphics and other techniques for data visualisation. There are 32 attributes in the dataset, which are broken out into mean, square, and worst categories. Not all of these characteristics will be able to determine which form of breast cancer will develop.

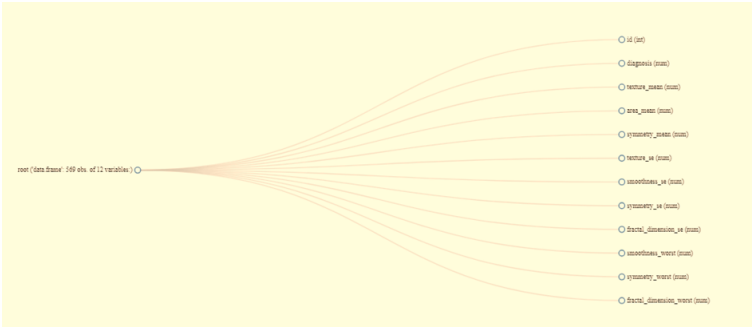
Multicollinearity:

We attempt to analyse the relationship between the variables and exclude those that are highly associated with the aid of the pair plot from the ggplot package. The presence of multicollinearity, which will go against the assumptions of building a regression model, is indicated by highly linked variables.



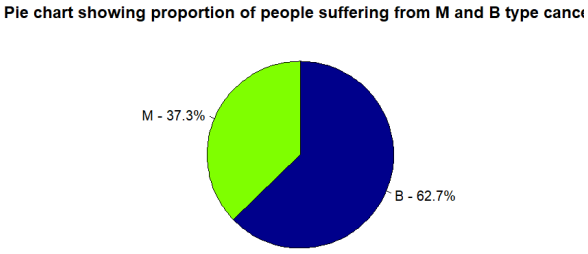
Pairplots

It is evident from the pair-plots above that attributes with correlation values higher than 0.7 lead to multicollinearity. We also used the **variance inflation factor** to support this, excluding variables with VIF values larger than 10.Thus, the dimension of the data is 569 observations and 12 rows. The reduced data is diplayed in the below flow chart.



Proportion of people suffering from M and B type cancer:

The target variable, diagnosis, is divided into the B type and M type categories. To depict the percentage of patients with B and M type diagnoses, a pie chart is used.



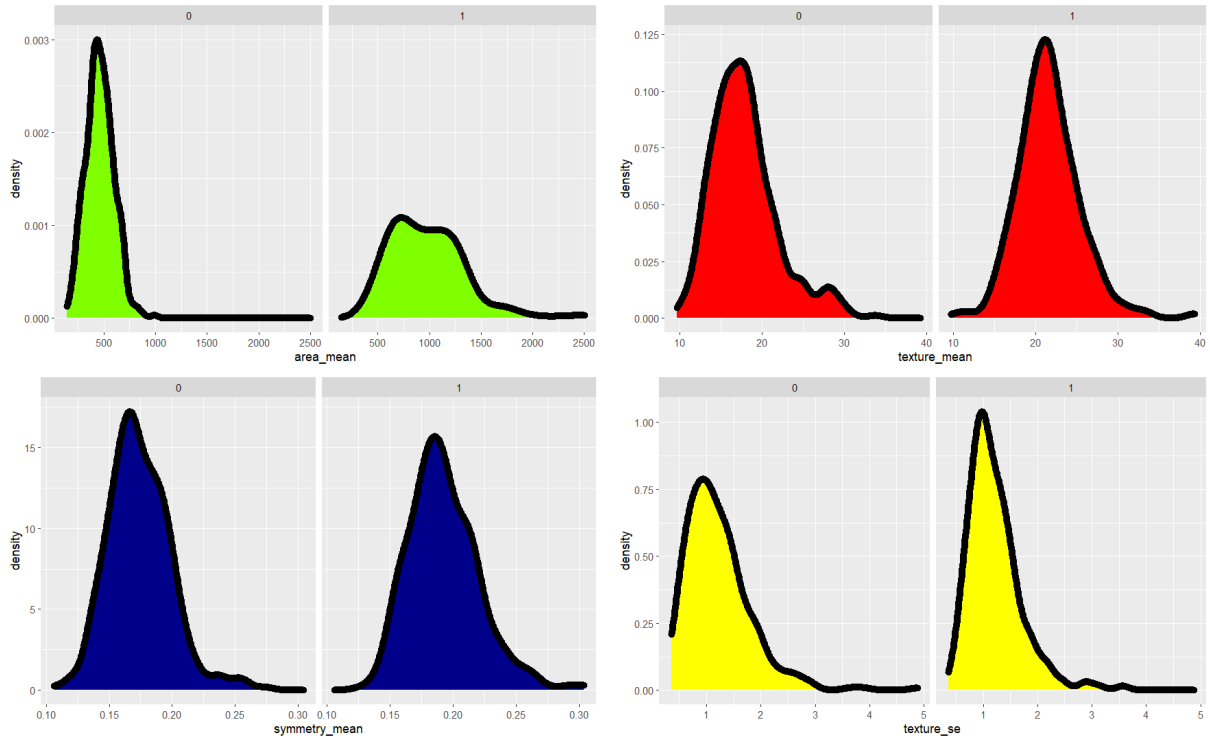

```{r}
tab1 <- table(data$diagnosis)
tab1
```

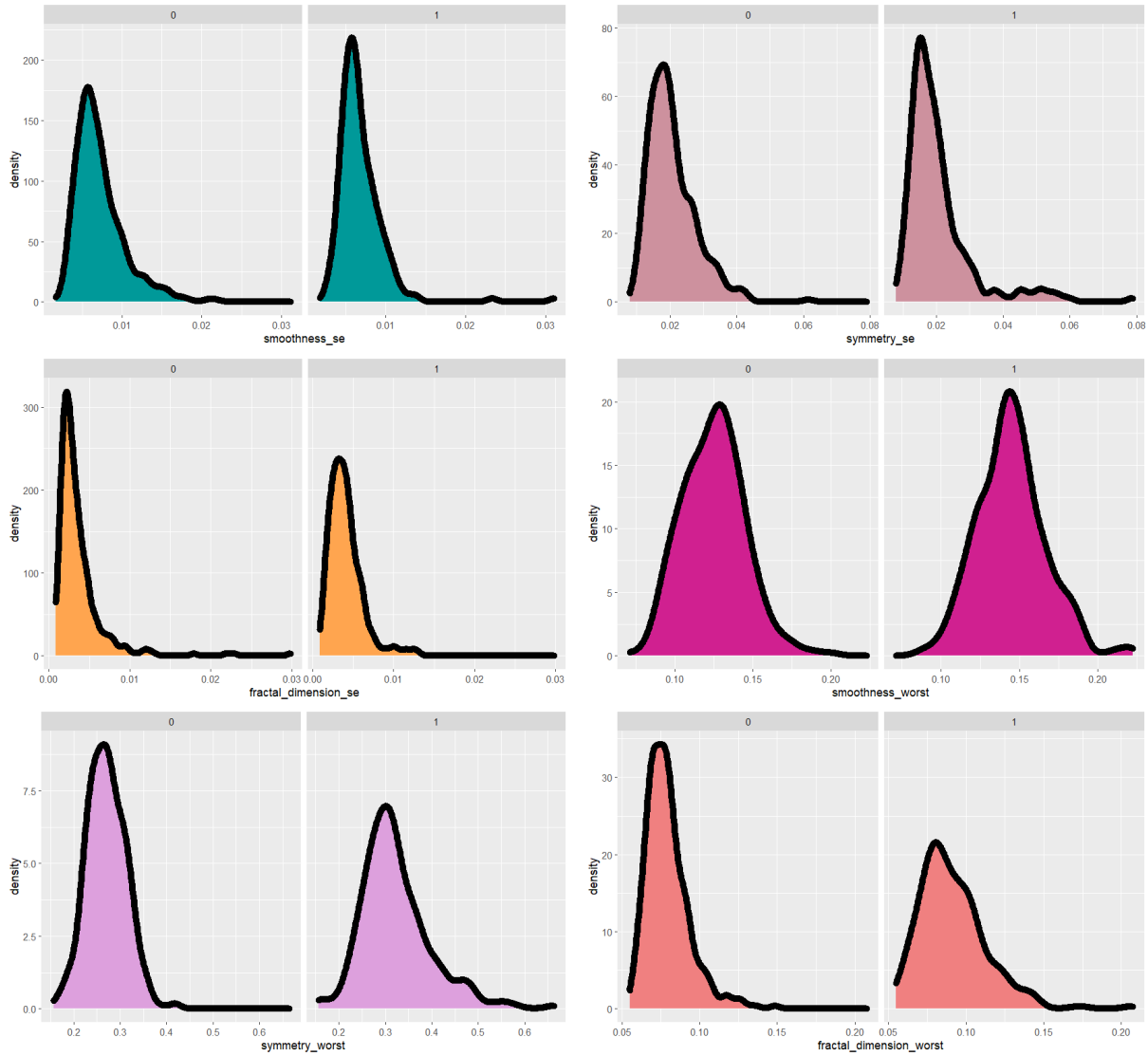
```
  B   M
357 212
```

The pie graphic clearly shows that more patients, roughly 62.7 percent have non-cancerous tumours.

Distribution of each variable with respect to diagnosis:

The above plots show the distribution of each variable with respect to diagnosis. Here 0 denotes M type and 1 denotes B type.





3.3 Classification Algorithms

Different machine learning models are suitable for various situations. In supervised learning, classification and regression are the two main categories of machine learning problems. We have a classification issue since we need to train the computer to distinguish between benign (0) and malignant(1) scans .Three machine learning models—Logistic Regression, Decision Tree, and Random Forest Classifier—will be tested on our data.

3.3.1 Logistic Regression

A sigmoid function is used in the statistical technique of logistic regression. Despite being a regression model, the methodology can be successfully applied as a classification method, particularly for binary classification problems (yes or no questions).

Advantages:

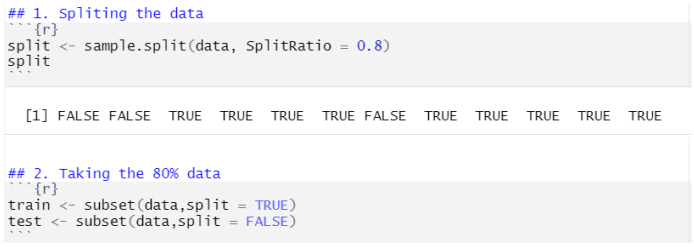
- Simple to execute, comprehend, and train.
- Works well when the dataset can be separated linearly.

Cons:

- The assumption that the dependent variable and the independent variables are linear is the biggest drawback.
- The performance of this approach is easily outperformed by more robust and compact algorithms like neural networks.

Steps for Fitting the Logistic Regression Model:

1. Create Training and Test Samples: Split the dataset into a training set to train the model on and a testing set to test the model on. Here we have used 80% of dataset as training set and remaining 20% as testing set.



```
## 1. Spliting the data
```{r}
split <- sample.split(data, SplitRatio = 0.8)
split
```

  [1] FALSE FALSE TRUE TRUE TRUE TRUE FALSE TRUE TRUE TRUE TRUE TRUE

2. Taking the 80% data
```{r}
train <- subset(data,split = TRUE)
test <- subset(data,split = FALSE)
```
```

2. Fit the Logistic Regression Model: The **glm** (general linear model) function and specify family="binomial" so that R fits a logistic regression model to the dataset.

```
## 3. Model fitting
```{r}
CanDet <- glm(diagnosis~.,family = binomial,data = train)
summary(CanDet)
```
```

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

Call:
glm(formula = diagnosis ~ ., family = binomial, data = train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.2084  -0.1106  -0.0152   0.0036   4.0355

Coefficients:
                          Estimate Std. Error z value Pr(>|z|)
(Intercept)             -4.409e+01  5.714e+00  -7.717 1.19e-14 ***
id                       1.092e-09  2.353e-09   0.464 0.642410
texture_mean             3.111e-01  8.030e-02   3.875 0.000107 ***
area_mean                2.202e-02  2.889e-03   7.622 2.50e-14 ***
symmetry_mean            6.935e+00  1.710e+01   0.406 0.685073
texture_se               1.526e+00  7.385e-01   2.067 0.038764 *
smoothness_se            1.695e+02  2.057e+02   0.824 0.409901
symmetry_se             -4.750e+00  5.844e+01  -0.081 0.935229
fractal_dimension_se    -9.304e+02  4.617e+02  -2.015 0.043862 *
smoothness_worst         7.247e+01  2.789e+01   2.598 0.009375 **
symmetry_worst           1.434e+01  9.778e+00   1.467 0.142504
fractal_dimension_worst  1.026e+02  5.346e+01   1.919 0.054929 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 751.44  on 568  degrees of freedom
Residual deviance: 110.52  on 557  degrees of freedom
AIC: 134.52
```

3. Assessing Model Fit: For logistic regression, there is no such R2 value. Instead, we can calculate a statistic called McFadden's R2, which has a range from 0 to just below 1. Values that are very close to 0 show that the model is completely unpredictive.

```
## 4. Assessing Model Fit
```{r}
library(pscl)
pscl::pR2(CanDet)["McFadden"]
```
```

```
fitting null model for pseudo-r2
 McFadden
0.8529209
```

4. Variable Importance: Additionally, we can use the **varImp** function from the caret package to determine the significance of each predictor variable in the model.

```
> caret::varImp(CanDet)
                          Overall
id                      0.4643325
texture_mean            3.8745185
area_mean               7.6216836
symmetry_mean           0.4055500
texture_se              2.0666883
smoothness_se           0.8240685
symmetry_se             0.0812683
fractal_dimension_se    2.0154034
smoothness_worst        2.5980714
symmetry_worst          1.4665292
fractal_dimension_worst 1.9194365
```

5. Model Diagnostics: Basically, this is to evaluate how well our model does on the test dataset.

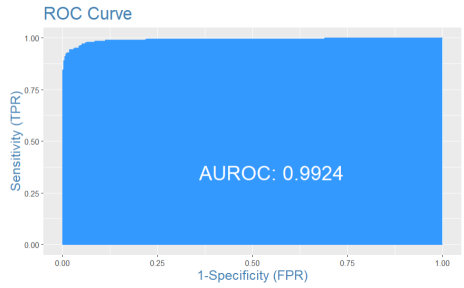
```{r}
library(InformationValue)
predicted <- predict(CanDet)
sensitivity(data$diagnosis, predicted)
```

[1] 0.9433962

```{r}
specificity(data$diagnosis, predicted)
```

[1] 0.9831933

```{r}
plotROC(data$diagnosis, predicted)
```




```{r}
misClassError(data$diagnosis, predicted, threshold=0.5)
```

[1] 0.0316

3.3.2 Random Forest Model

An algorithm used in ensemble approaches is called Random Forest. The final forecast is made by combining the estimates of various estimators using ensemble methods. An assortment of decision trees make up Random Forest (estimators). We're going to use randomForest for this project. Each tree in a classification problem casts a vote as to whether it believes the cancer scan is malignant (1) or benign (0), with the most common response being used as the outcome.

Advantages:

- Robust against outliers, little danger of overfitting
- Effective with huge datasets
- Compared to other algorithms, has a higher accuracy rating

Cons:

- When working with categorical data, it could be skewed.
- Ineffective for linear models with a large number of missing data.

Steps for Fitting the Random Forest Model:

1. Load the Necessary Packages: We need only one package (ie) randomForest. Also the we converted the data type of diagnosis variable to factor type.

```{r}
data <- transform(data,diagnosis=as.factor(diagnosis))
sapply(data, class)
```

```
                     id               diagnosis            texture_mean              area_mean 
              "integer"                "factor"               "numeric"               "numeric" 
        symmetry_mean             texture_se          smoothness_se             symmetry_se 
              "numeric"               "numeric"               "numeric"               "numeric" 
fractal_dimension_se         smoothness_worst        symmetry_worst fractal_dimension_worst 
              "numeric"               "numeric"               "numeric"               "numeric" 
```

2. Fit the Random Forest Model: To fit a random forest model in R we used the **randomForest()** function from the randomForest package.

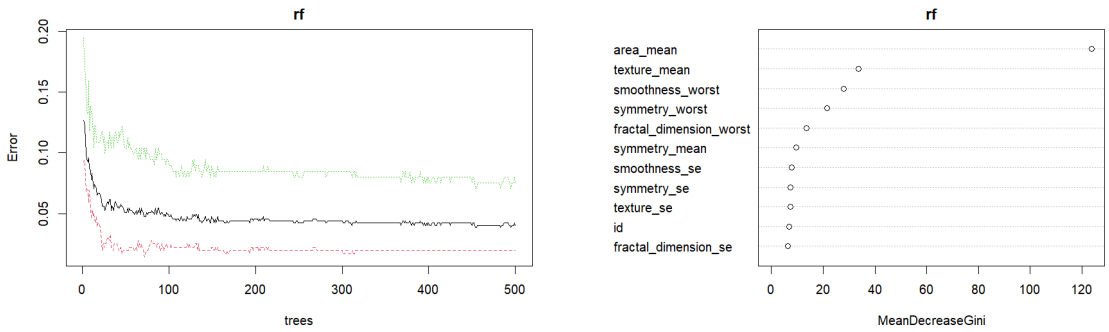
```{r}
library(party)
library(randomForest)
rf <- randomForest(
  diagnosis ~ .,
  data=data)
print(rf)
```

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

        OOB estimate of  error rate: 4.04%
Confusion matrix:
    0   1 class.error
0 350   7  0.01960784
1  16 196  0.07547170
```

3. Plot the test MSE by number of trees and produce variable importance plot: We also wanted test MSE plot based on the amount of trees utilised. Using the **varImpPlot()** method, you can produce a plot showing the relative weights of each predictor variable in the final model.

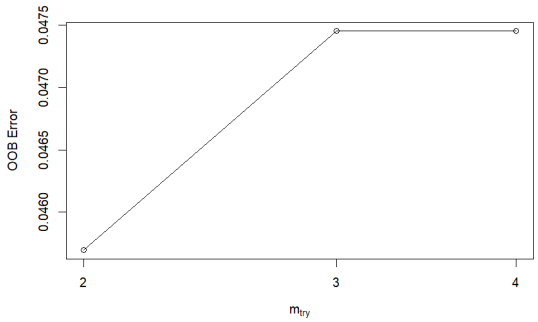
```{r}
plot(rf)
#produce variable importance plot
varImpPlot(rf)
```



4. Tune the Model: At each split, the randomForest() algorithm defaults to using 500 trees and (total predictors/3) randomly chosen predictors as candidates. Using the tuneRF() method, we may modify these parameters.

```
mtry <- tuneRF(data[-2],data$diagnosis, ntreeTry=500,
               stepFactor=1.5,improve=0.01, trace=TRUE, plot=TRUE)
best.m <- mtry[mtry[, 2] == min(mtry[, 2]), 1]
print(mtry)
print(best.m)
```

```
mtry = 3  OOB error = 4.75%
Searching left ...
mtry = 2        OOB error = 4.57%
0.03703704 0.01
Searching right ...
mtry = 4        OOB error = 4.75%
-0.03846154 0.01
      mtry   OOBError
2.OOB    2 0.04569420
3.OOB    3 0.04745167
4.OOB    4 0.04745167
[1] 2
```



5. Build model again using best mtry value: Here we check which variable has higher mean decrease accuracy or mean decrease gini score.

```
rf <-randomForest(diagnosis~.,data=data, mtry=best.m, importance=TRUE,ntree=500)
print(rf)
#Evaluate variable importance
importance(rf)
#Higher the value of mean decrease accuracy or mean decrease gini score , higher the importance of the
variable in the model. In the plot shown above, Area_mean is most important variable.
```

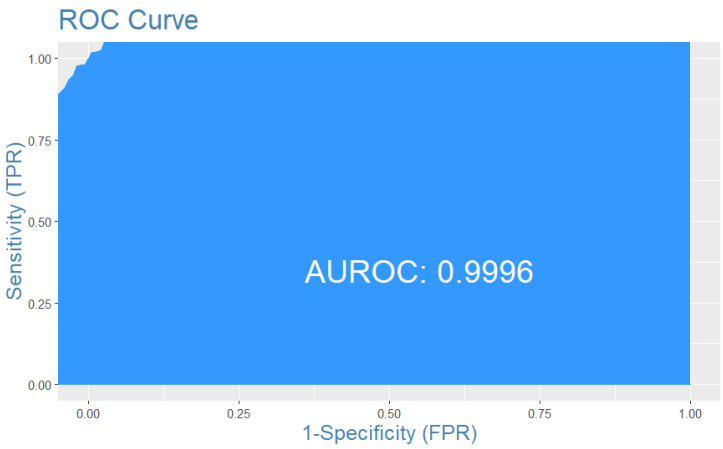
```
Call:
 randomForest(formula = diagnosis ~ ., data = data, mtry = best.m,      importance = TRUE, ntree = 500)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 2

        OOB estimate of  error rate: 4.04%
Confusion matrix:
    0   1 class.error
0 351   6  0.01680672
1  17 195  0.08018868
                          0         1 MeanDecreaseAccuracy MeanDecreaseGini
id                 5.464425  3.984541             6.829389        10.070879
texture_mean      22.245737 21.231673            28.136527        33.707336
area_mean         56.239259 59.508986            62.897991       107.277166
symmetry_mean      4.834708 10.877919            11.642067        12.650567
texture_se         8.413536  1.109584             7.672837         8.590305
smoothness_se      8.870895  2.141154             8.388563         9.541901
symmetry_se        7.498200  5.700205             9.675773        10.350996
fractal_dimension_se  9.073678  3.365084          9.878566         9.634459
smoothness_worst  19.370039 20.141039            25.378340        25.542219
symmetry_worst    14.579640 15.099807            19.984703        22.891335
fractal_dimension_worst  9.866585 11.751511      14.135946        15.514619
```

6. Model Diagnostics: Basically, this is to evaluate how well our model does on the test dataset.

```
```{r}
pred1=predict(rf,type = "prob")
library(ROCR)
perf = prediction(pred1[,2], data$diagnosis)
```
```

```
#Area under the curve
auc = performance(perf, "auc")
auc
# 2. True Positive and Negative Rate
pred3 = performance(perf, "tpr","fpr")
# 3. Plot the ROC curve
plotROC(test$diagnosis,pred1)
```



3.3.3 Decision Tree

A highly well-liked machine learning algorithm is the decision tree. Decision Tree uses a tree representation of the data to answer the machine learning problem. Each leaf node of the tree representation represents a class label, whereas each internal node stands for an attribute.Both regression and classification issues can be resolved with a decision tree approach.

Advantages:

- Compared to other algorithms decision trees requires less effort for data preparation during pre-processing.
- A decision tree does not require scaling of data as well.
- A decision tree does not require normalization of data.

Cons:

- Decision tree often involves higher time to train the model.

- Decision tree training is relatively expensive as the complexity and time has taken are more.

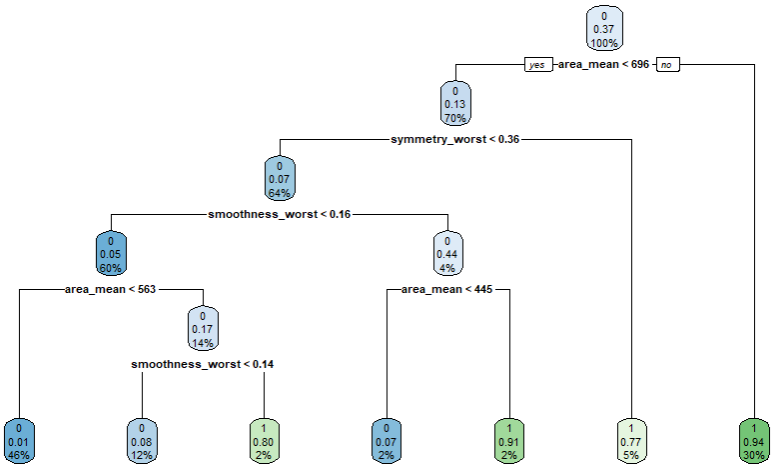
Steps for Fitting the Decision Tree:

1. Loading the necessary libraries: The two libraries used here rpart and rpart.plot

```r
library(rpart)
library(rpart.plot)
```

2. Using rpart.plot() create the tree. The optional features are configured to indicate the likelihood of the second class at 101. (useful for binary responses).

```r
fit <- rpart(diagnosis~., data = train, method = 'class')
rpart.plot(fit, extra = 106)
```



3. Model Diagnostics: Basically, this is to evaluate how well our model does on the test dataset. So we had done confusion matrix, Accuracy score and roc plot.

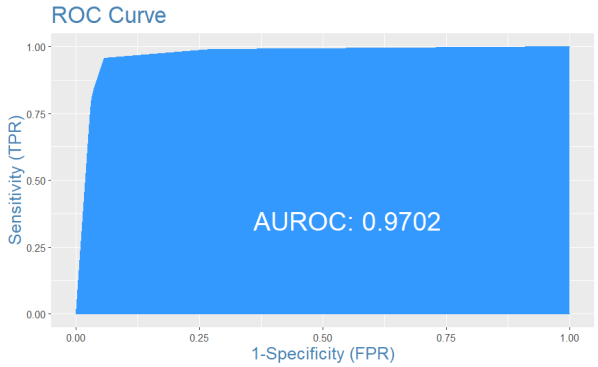
```r
prevarbre=predict(fit,newdata=test,type="prob")
previsions2=ifelse(prevarbre[,2]>0.5,"Yes","No")
table(previsions2,test$diagnosis)
```

```
previsions2   0   1
         No  337   9
         Yes  20 203
```

```r
CM_arbre<-table(prevarbre[,2]>0.5,test$diagnosis)
accuracy_arbre=(sum(diag(CM_arbre)))/sum(CM_arbre)
accuracy_arbre
```

```
[1] 0.9490334
```

```{r}
plotROC(test$diagnosis,prevarbre[,2])
```



Chapter 4

CONCLUSION

4.1 Logistic Regression Model

- The coefficients in the output indicate the average change in log odds of diagnosis. For example, a one unit increase in texture mean is associated with an average increase of 0.3111 in the log odds of diagnosis.

- The p-values in the output also give us an idea of how effective each predictor variable is at predicting the probability of diagnosis.We can see that texture-mean,area-mean,texture-se, fractal-dimension-se and smoothness-worst seem to be important predictors.

- R^2 is a common metric used in linear regression to measure how well a model fits the data. Higher numbers suggest better model fit; the range of this number is 0 to 1. However, logistic regression does not have an equivalent R^2 value. Instead, we can calculate the McFadden's R^2 metric, which has a range of 0 to just below 1. Values that are very close to 0 show that the model has no predictive ability.Using the pR^2 function from the pscl package, we can calculate McFadden's R^2 for our model. McFadden's R^2 has a high value of 0.8529209, indicating that our model has a strong ability to forecast the future and fits the data very well.

- Additionally, we can use the varImp function from the caret package to determine the significance of each predictor variable in the model.Greater values denote more significance.

The p-values from the model agree well with these findings. Area-mean, followed by texture-mean, texture-se, fractal-dimension-se, and smoothness-worst, is by far the most significant predictive variable.

- For the purpose of model evaluation, we have also calculated the sensitivity (also known as the "true positive rate") and specificity (sometimes known as the "true negative rate"), in addition to the overall misclassification error (which informs us of the proportion of all inaccurate classifications).

- For this model, the overall misclassification error rate is 3.16 percent. This particular model turns out to be particularly good at predicting whether a person will suffer from a B or M type tumour because, generally speaking, the lower this rate, the better the model is able to predict outcomes.

- As the prediction probability cutoff is dropped from 1 to 0, we may lastly plot the ROC (Receiver Operating Characteristic) Curve, which shows the percentage of true positives predicted by the model. Our model can predict outcomes more precisely the greater the AUC (area under the curve). The AUC is 0.9924, which is a very high value, as can be seen. This suggests that our approach is effective at predicting whether a person will suffer from M type or not.

4.2 Random Forest Model

- Fitting a random model with randomForest() was our first assignment. Additionally, we programmed to create a test MSE plot based on the number of trees used.

- The importance of each predictor variable in the final model is visualised using the varImpPlot() method. The average improvement in node purity of the regression trees based on splitting on the different predictors shown on the y-axis is shown on the x-axis. We can observe from the graphic that area-mean is the most significant predictor variable, closely followed by texture mean.

- We utilised the tuneRF() method along with the following criteria to identify the best model.
  - ntreeTry: The quantity of trees to construct.
  - stepFactor: The factor to raise by until the out-of-bag estimated error stops decreasing by a specific amount.
  - improve: The rate at which the step factor must be raised in order to reduce the out-of-bag error.
- This function produces the following plot, which displays the number of predictors used at each split when building the trees on the x-axis and the out-of-bag estimated error on the y-axis.
- Our model can predict outcomes more precisely the greater the AUC (area under the curve). The AUC is 0.9996, which is a very high value, as can be seen. This suggests that our approach is effective at predicting whether a person will suffer from M type or not.

4.3 Decision Tree

- The function **rpart** take the following arguments:
  - diagnosis $\sim$. : Formula of the Decision Trees
  - data = Dataset
  - method = 'class': Fit a binary model
- Rpart.plot(fit, extra= 106) creates a tree plot. The optional features are configured to 101 to show the likelihood of the second class (useful for binary responses).
- 37% of total population have cancer. 70% of them have area less than 696. Out of this 70% there is a 0.13 probability that they have cancer. Out of those 30% who have area mean greater than 696, there is a 0.94 probability that they don't have cancer.

- Out of the 70% who have area mean less than 696, 5% have symmetry worst more than 0.36. Out of these 5%, there is 0.77% chance that they d not have cancer. For the remaining 64%, there is 0.07 probability that they have cancer.

- From the 64%, 60% have smoothness worst less than 0.16 and have 0.05 probability to have cancer. For the remaining 4%, there is 0.44 probability to have cancer.

- Out of the 60%, there are 46% who have area mean less than 563 and have 0.01 probability to have cancer. And 14% have area mean greater than 563 and a probability of 0.17 to have cancer.

- Out of the 14% who have area mean greater than 563, 12% have smoothness worst less than 0.14 and a probability of 0.08 to have cancer. The remaining 2% have a 0.8 probability to not have cancer.

- From the 4% who have smoothness worst greater than 0.16, 2% have area mean less than 445 and have a 0.07 probability of having cancer. The remaining 2% have a 0.91 probability of not having cancer.

- The confusion matrix which shows false negative, false positive, true negative and true positive values corresponding to the decision tree model.

- The accuracy score for decision tree method is 0.9490334 and is obtain using the values from the confusion matrix.

- The AUROC for the decision tree model is found to be 0.9702 using plotROC() function. This shows that our method is successful in determining whether a person will suffer M type.

4.4 Result

All algorithms produced close results, with decision tree producing the lowest and random forest producing the highest. It's interesting to note that the various machine learning algorithms utilised in this study produced results with high accuracy, suggesting that these techniques could be used

as substitute predictive tools in the studies of breast cancer survival. The below table gives us a summary of the accuracy score using the ROC-AUC.

Logistic Regression Model	Random Forest Model	Decision Tree
0.9924	0.9996	0.9702