EDX Harvard Capstone Breast Cancer Prediction Project

Yin Thu Win

2025-05-29

1.1 Overview

This project is part of the “Choose-Your-Own” project from the HarvardX: PH125.9x Data Science Capstone course. It begins by outlining the project’s goals, followed by data preparation and setup. An exploratory data analysis (EDA) is conducted to understand the dataset and guide the development of a machine learning model to predict whether a breast cancer cell is benign or malignant. Various models are trained and evaluated, with results discussed in detail. The project concludes with final reflections on the findings and potential applications of the model in supporting breast cancer diagnosis.

1.2 Introduction

This project focuses on the classification of breast cancer cells using machine learning, specifically analyzing data from Fine Needle Aspiration (FNA) procedures. Breast cancer, one of the most prevalent cancers worldwide, causes over 400,000 deaths annually and is projected to rise significantly by 2030. Early detection is critical, and mammography followed by biopsy—such as FNA—is a common diagnostic path. In FNA, cell samples are extracted and analyzed microscopically, with software like ‘Xcyt’ used to define cell nuclei boundaries. This report evaluates various supervised learning algorithms—such as neural networks, logistic regression to determine the most accurate and efficient in predicting whether a tumor is benign or malignant. Metrics including accuracy, sensitivity, precision, and specificity are used for comparison. The integration of machine learning into healthcare offers powerful support for early diagnosis and clinical decision-making. As breast cancer data grows, so does the opportunity for AI-driven medical research and innovation.

1.3 Objectives

This report aims to develop machine learning models to predict whether breast cancer cells are benign or malignant. The dataset undergoes preprocessing, including transformation and dimensionality reduction, to improve analysis and reveal patterns. Models are evaluated using key metrics such as accuracy, sensitivity, and F1 score. The goal is to build a classifier that not only performs well overall but also minimizes false negatives, ensuring high sensitivity—critical for early cancer detection. Features are extracted from images of cell nuclei to support classification, helping determine the likelihood of malignancy and enhancing diagnostic support through data-driven methods.

2 Methods and Analysis

2.1 Data Analysis

2.1.1 Dataset

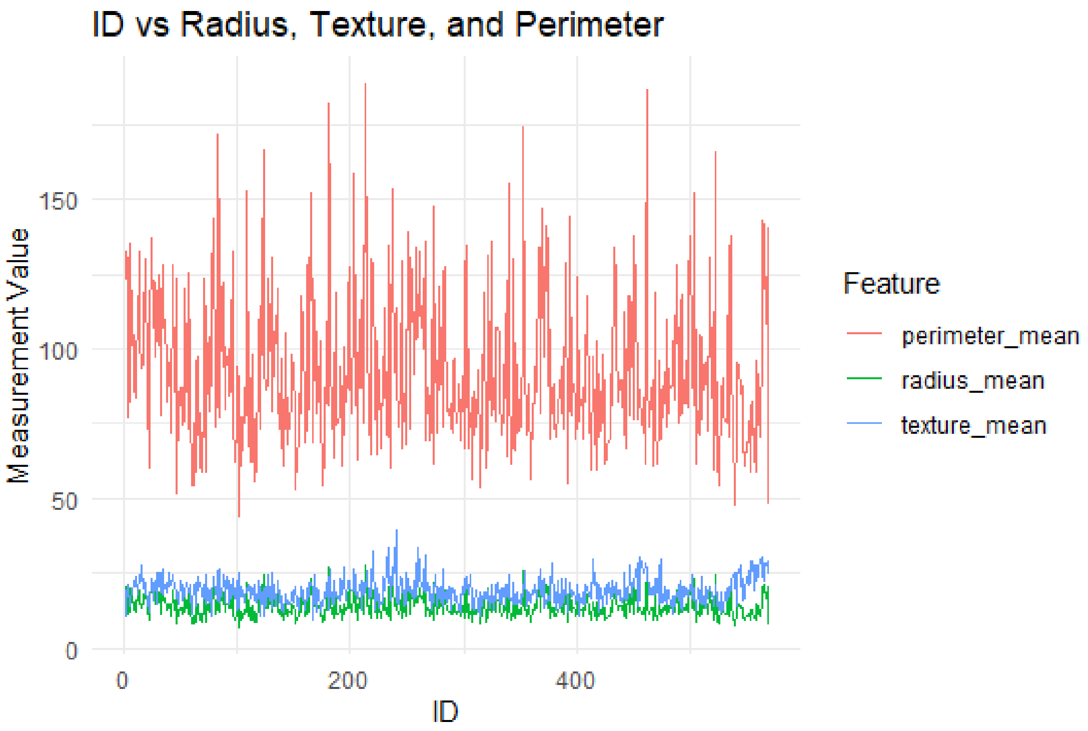
This report utilizes the Breast Cancer Wisconsin (Diagnostic) Dataset, originally created by Dr. William H. Wolberg at the University of Wisconsin Hospital in Madison. Collected in 1993, the dataset includes biopsy results from 569 patients and is widely used for research and machine learning applications in medical diagnosis. It contains detailed measurements of cell nuclei from breast mass samples to classify tumors as benign or malignant. The dataset, sourced from Kaggle, is provided in .csv format and was accessed through the author’s personal GitHub repository for this project.

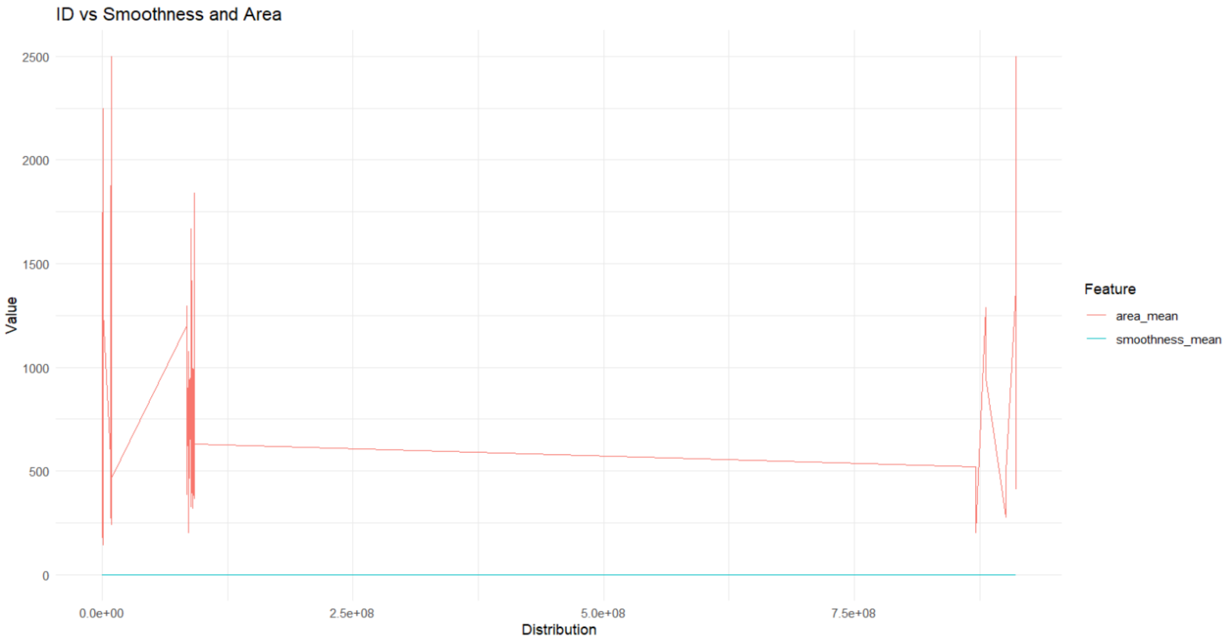
• [Wisconsin Breast Cancer Diagnostic Dataset] https://www.kaggle.com/uciml/breast-cancer-wisconsin-data/version/2

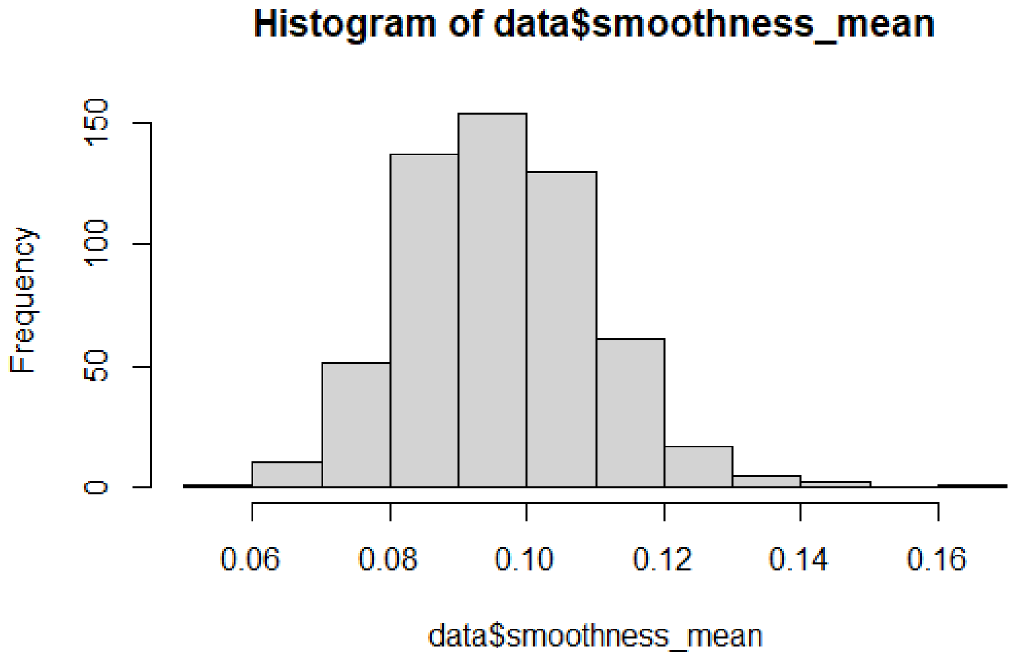
The .csv format file containing the data is loaded from my personal github account.

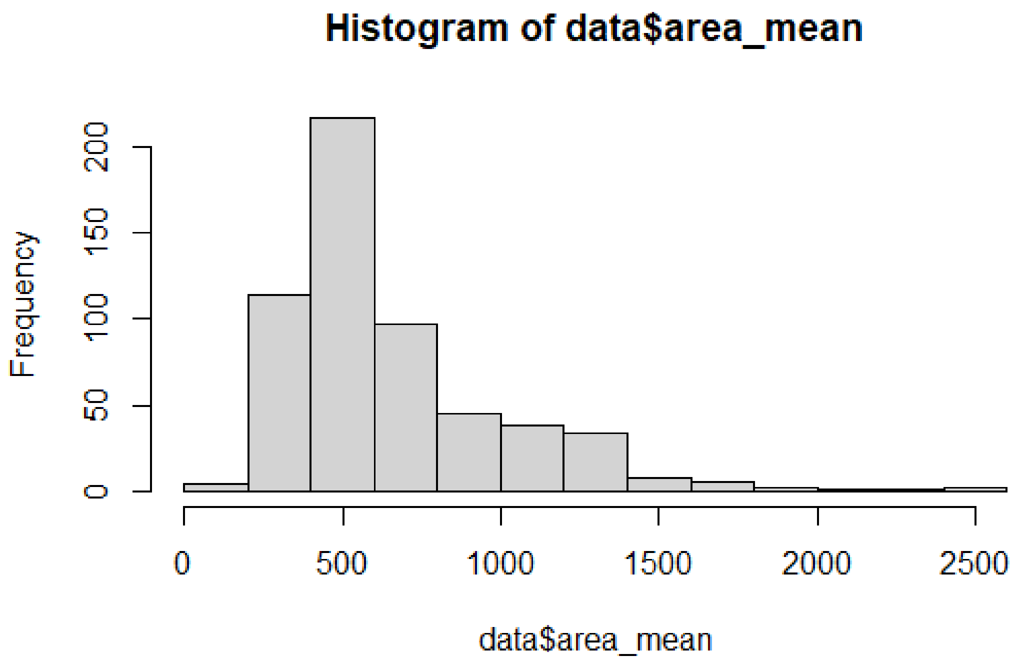
The dataset includes features that describe various characteristics of cell nuclei from breast tissue images, used to classify tumors as benign or malignant. Each sample is identified by an ID and labeled with a diagnosis (M = malignant, B = benign). Ten key features are calculated for each nucleus, including radius, texture, perimeter, area, smoothness, compactness, concavity, concave points, symmetry, and fractal dimension. For each feature, three statistics—mean, standard error, and worst (average of the three largest values)—were computed, resulting in 30 variables per case. The dataset contains 569 samples: 357 benign and 212 malignant, with histological confirmation.

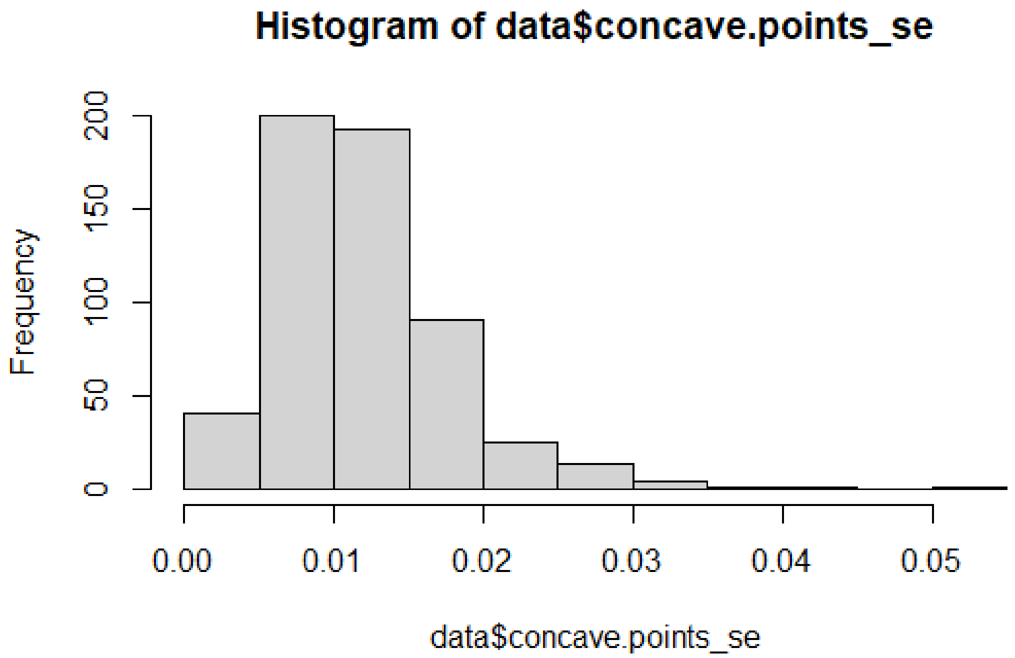
The column 33 is invalid.

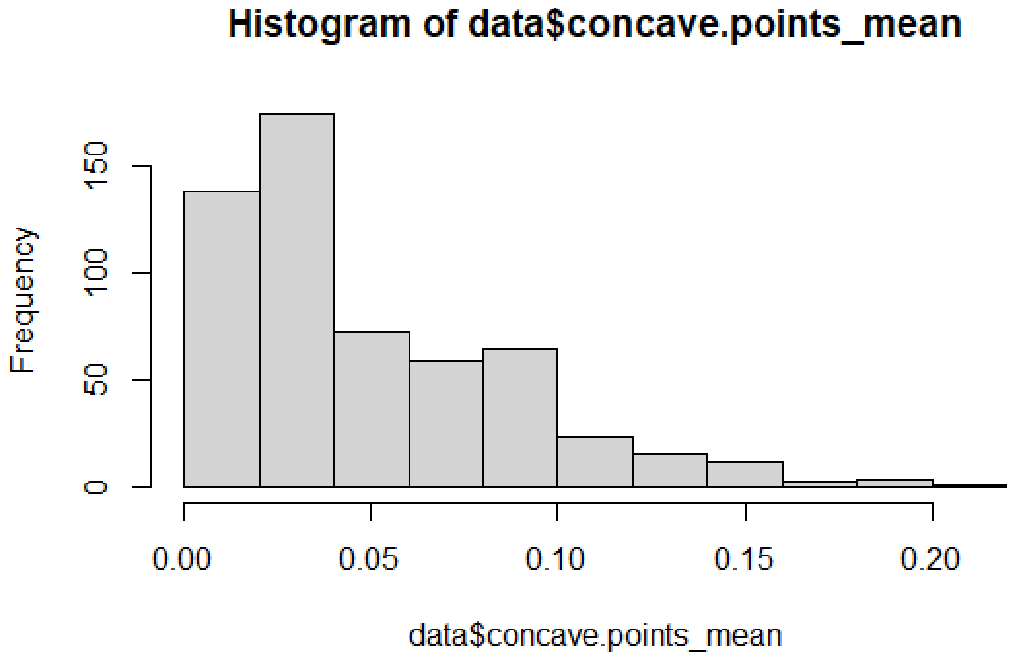

ID vs Radius, Texture, and Perimeter
Measurement Value
150
100
50
0
0
200
400
ID
Feature
perimeter_mean
radius_mean
texture_mean


ID vs Smoothness and Area
Value
2500
2000
1500
1000
500
0
0.0e+00
2.5e+08
5.0e+08
7.5e+08
Distribution
Feature
area_mean
smoothness_mean


Histogram of data$smoothness_mean
Frequency
0
50
100
150
0.06
0.08
0.10
0.12
0.14
0.16
data$smoothness_mean


Histogram of data$area_mean
Frequency
0
50
100
150
200
0
500
1000
1500
2000
2500
data$area_mean


Histogram of data$concave.points_se
Frequency
0
50
100
150
200
0.00
0.01
0.02
0.03
0.04
0.05
data$concave.points_se



Upon examining the dataset, we found that it contains 569 observations and 32 variables.

```
##        id             diagnosis  radius_mean      texture_mean
##  Min.   :     8670   B:357     Min.   : 6.981   Min.   : 9.71
##  1st Qu.:   869218   M:212     1st Qu.:11.700   1st Qu.:16.17
##  Median :   906024             Median :13.370   Median :18.84
##  Mean   : 30371831             Mean   :14.127   Mean   :19.29
##  3rd Qu.:  8813129             3rd Qu.:15.780   3rd Qu.:21.80
##  Max.   :911320502             Max.   :28.110   Max.   :39.28
##  perimeter_mean     area_mean      smoothness_mean   compactness_mean
##  Min.   : 43.79   Min.   : 143.5   Min.   :0.05263   Min.   :0.01938
##  1st Qu.: 75.17   1st Qu.: 420.3   1st Qu.:0.08637   1st Qu.:0.06492
##  Median : 86.24   Median : 551.1   Median :0.09587   Median :0.09263
##  Mean   : 91.97   Mean   : 654.9   Mean   :0.09636   Mean   :0.10434
##  3rd Qu.:104.10   3rd Qu.: 782.7   3rd Qu.:0.10530   3rd Qu.:0.13040
##  Max.   :188.50   Max.   :2501.0   Max.   :0.16340   Max.   :0.34540
##  concavity_mean    concave.points_mean symmetry_mean
fractal_dimension_mean
##  Min.   :0.00000   Min.   :0.00000     Min.   :0.1060   Min.   :0.04996
##  1st Qu.:0.02956   1st Qu.:0.02031     1st Qu.:0.1619   1st Qu.:0.05770
##  Median :0.06154   Median :0.03350     Median :0.1792   Median :0.06154
##  Mean   :0.08880   Mean   :0.04892     Mean   :0.1812   Mean   :0.06280
##  3rd Qu.:0.13070   3rd Qu.:0.07400     3rd Qu.:0.1957   3rd Qu.:0.06612
##  Max.   :0.42680   Max.   :0.20120     Max.   :0.3040   Max.   :0.09744
##    radius_se        texture_se       perimeter_se       area_se
##  Min.   :0.1115   Min.   :0.3602   Min.   : 0.757   Min.   :  6.802
##  1st Qu.:0.2324   1st Qu.:0.8339   1st Qu.: 1.606   1st Qu.: 17.850
##  Median :0.3242   Median :1.1080   Median : 2.287   Median : 24.530  
##  Mean   :0.4052   Mean   :1.2169   Mean   : 2.866   Mean   : 40.337  
##  3rd Qu.:0.4789   3rd Qu.:1.4740   3rd Qu.: 3.357   3rd Qu.: 45.190  
##  Max.   :2.8730   Max.   :4.8850   Max.   :21.980   Max.   :542.200  
##  smoothness_se      compactness_se      concavity_se     concave.points_se
##  Min.   :0.001713   Min.   :0.002252   Min.   :0.00000   Min.   :0.000000 
##  1st Qu.:0.005169   1st Qu.:0.013080   1st Qu.:0.01509   1st Qu.:0.007638 
##  Median :0.006380   Median :0.020450   Median :0.02589   Median :0.010930 
##  Mean   :0.007041   Mean   :0.025478   Mean   :0.03189   Mean   :0.011796 
##  3rd Qu.:0.008146   3rd Qu.:0.032450   3rd Qu.:0.04205   3rd Qu.:0.014710 
##  Max.   :0.031130   Max.   :0.135400   Max.   :0.39600   Max.   :0.052790 
##   symmetry_se       fractal_dimension_se  radius_worst   texture_worst  
##  Min.   :0.007882   Min.   :0.0008948    Min.   : 7.93   Min.   :12.02  
##  1st Qu.:0.015160   1st Qu.:0.0022480    1st Qu.:13.01   1st Qu.:21.08  
##  Median :0.018730   Median :0.0031870    Median :14.97   Median :25.41  
##  Mean   :0.020542   Mean   :0.0037949    Mean   :16.27   Mean   :25.68  
##  3rd Qu.:0.023480   3rd Qu.:0.0045580    3rd Qu.:18.79   3rd Qu.:29.72  
##  Max.   :0.078950   Max.   :0.0298400    Max.   :36.04   Max.   :49.54  
##  perimeter_worst    area_worst     smoothness_worst  compactness_worst
##  Min.   : 50.41   Min.   : 185.2   Min.   :0.07117   Min.   :0.02729  
##  1st Qu.: 84.11   1st Qu.: 515.3   1st Qu.:0.11660   1st Qu.:0.14720  
##  Median : 97.66   Median : 686.5   Median :0.13130   Median :0.21190  
##  Mean   :107.26   Mean   : 880.6   Mean   :0.13237   Mean   :0.25427  
##  3rd Qu.:125.40   3rd Qu.:1084.0   3rd Qu.:0.14600   3rd Qu.:0.33910  
##  Max.   :251.20   Max.   :4254.0   Max.   :0.22260   Max.   :1.05800  
##  concavity_worst  concave.points_worst symmetry_worst  
fractal_dimension_worst
##  Min.   :0.0000   Min.   :0.00000      Min.   :0.1565   Min.   :0.05504  
##  1st Qu.:0.1145   1st Qu.:0.06493      1st Qu.:0.2504   1st Qu.:0.07146  
##  Median :0.2267   Median :0.09993      Median :0.2822   Median :0.08004  
##  Mean   :0.2722   Mean   :0.11461      Mean   :0.2901   Mean   :0.08395  
##  3rd Qu.:0.3829   3rd Qu.:0.16140      3rd Qu.:0.3179   3rd Qu.:0.09208  
##  Max.   :1.2520   Max.   :0.29100      Max.   :0.6638   Max.   :0.20750

##         id diagnosis radius_mean texture_mean perimeter_mean area_mean
## 1   842302         M       17.99        10.38         122.80    1001.0
## 2   842517         M       20.57        17.77         132.90    1326.0
## 3 84300903         M       19.69        21.25         130.00    1203.0
## 4 84348301         M       11.42        20.38          77.58     386.1
## 5 84358402         M       20.29        14.34         135.10    1297.0
## 6   843786         M       12.45        15.70          82.57     477.1
##   smoothness_mean compactness_mean concavity_mean concave.points_mean
## 1         0.11840          0.27760         0.3001             0.14710
## 2         0.08474          0.07864         0.0869             0.07017
## 3         0.10960          0.15990         0.1974             0.12790
## 4         0.14250          0.28390         0.2414             0.10520
## 5         0.10030          0.13280         0.1980             0.10430
## 6         0.12780          0.17000         0.1578             0.08089
##   symmetry_mean fractal_dimension_mean radius_se texture_se perimeter_se
## 1        0.2419                0.07871    1.0950     0.9053        8.589
## 2        0.1812                0.05667    0.5435     0.7339        3.398
## 3        0.2069                0.05999    0.7456     0.7869        4.585
## 4        0.2597                0.09744    0.4956     1.1560        3.445
## 5        0.1809                0.05883    0.7572     0.7813        5.438
## 6        0.2087                0.07613    0.3345     0.8902        2.217
##   area_se smoothness_se compactness_se concavity_se concave.points_se
## 1  153.40      0.006399        0.04904      0.05373          0.01587
## 2   74.08      0.005225        0.01308      0.01860          0.01340
## 3   94.03      0.006150        0.04006      0.03832          0.02058
## 4   27.23      0.009110        0.07458      0.05661          0.01867
## 5   94.44      0.011490        0.02461      0.05688          0.01885
## 6   27.19      0.007510        0.03345      0.03672          0.01137
##   symmetry_se fractal_dimension_se radius_worst texture_worst
perimeter_worst
## 1     0.03003             0.006193        25.38         17.33
184.60
## 2     0.01389             0.003532        24.99         23.41
158.80
## 3     0.02250             0.004571        23.57         25.53
152.50
## 4     0.05963             0.009208        14.91         26.50
98.87
## 5     0.01756             0.005115        22.54         16.67
152.20
## 6     0.02165             0.005082        15.47         23.75
103.40
##   area_worst smoothness_worst compactness_worst concavity_worst
## 1     2019.0           0.1622            0.6656          0.7119
## 2     1956.0           0.1238            0.1866          0.2416
## 3     1709.0           0.1444            0.4245          0.4504
## 4      567.7           0.2098            0.8663          0.6869
## 5     1575.0           0.1374            0.2050          0.4000
## 6      741.6           0.1791            0.5249          0.5355
##   concave.points_worst symmetry_worst fractal_dimension_worst
## 1               0.2654         0.4601                 0.11890
## 2               0.1860         0.2750                 0.08902
## 3               0.2430         0.3613                 0.08758
## 4               0.2575         0.6638                 0.17300
## 5               0.1625         0.2364                 0.07678
## 6               0.1741         0.3985                 0.12440
```

```
## 'data.frame':    569 obs. of  32 variables:
##  $ id                     : int  842302 842517 84300903 84348301 84358402
843786 844359 84458202 844981 84501001 ...
##  $ diagnosis              : Factor w/ 2 levels "B","M": 2 2 2 2 2 2 2 2 2
2 ...
##  $ 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 ...
```

We need to check whether the dataset contains any missing values:

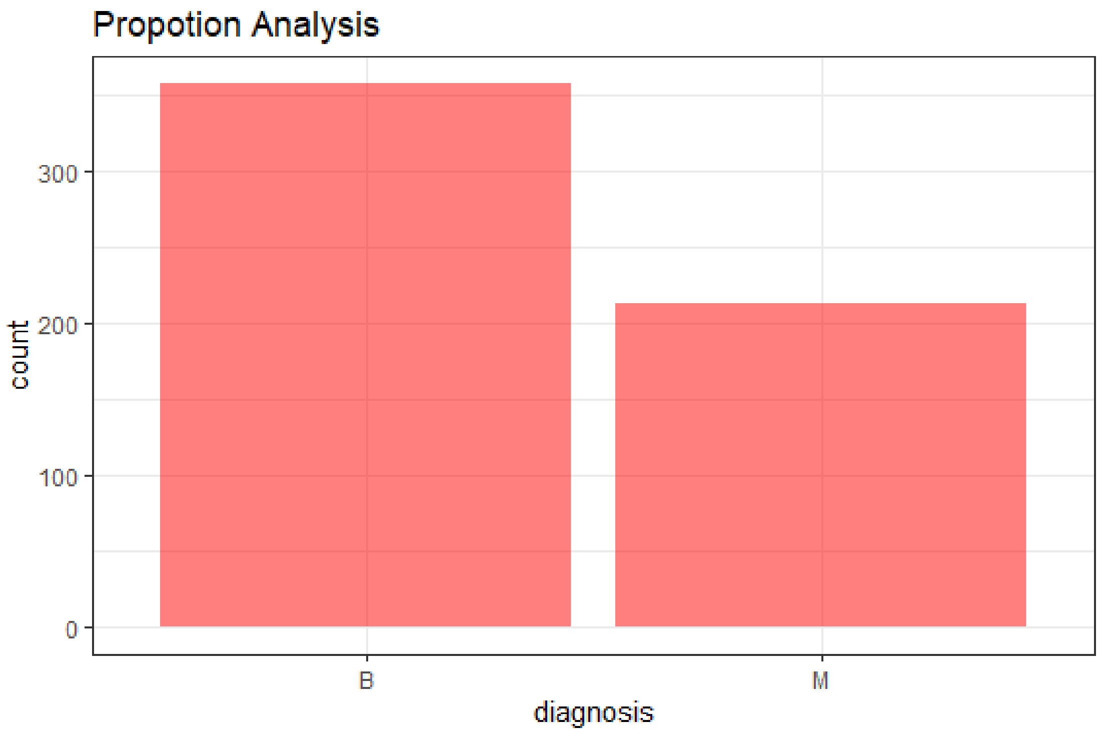
```
## 
##         B         M 
## 0.6274165 0.3725835
```

The proportion plot also confirms that the target variable is slightly imbalanced.

```
## $id
## [1] 0
## 
## $diagnosis
## [1] 0
## 
## $radius_mean
## [1] 0
## 
## $texture_mean
## [1] 0
## 
## $perimeter_mean
## [1] 0
```

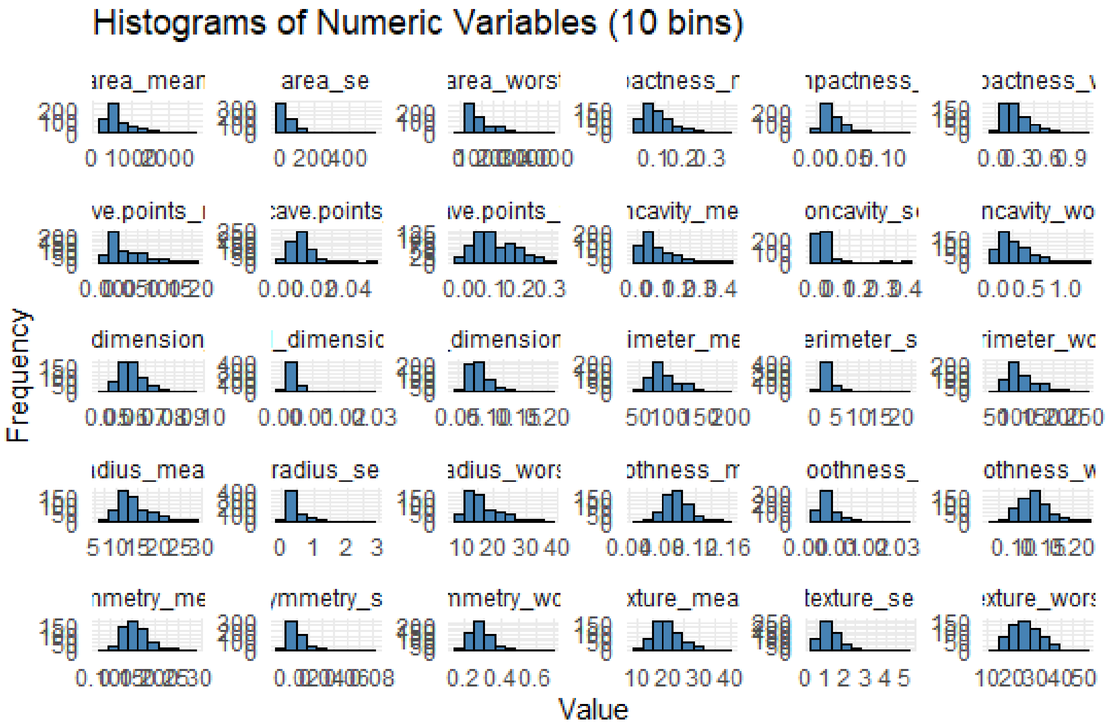
```
##
## $area_mean
## [1] 0
##
## $smoothness_mean
## [1] 0
##
## $compactness_mean
## [1] 0
##
## $concavity_mean
## [1] 0
##
## $concave.points_mean
## [1] 0
##
## $symmetry_mean
## [1] 0
##
## $fractal_dimension_mean
## [1] 0
##
## $radius_se
## [1] 0
##
## $texture_se
## [1] 0
##
## $perimeter_se
## [1] 0
##
## $area_se
## [1] 0
##
## $smoothness_se
## [1] 0
##
## $compactness_se
## [1] 0
##
## $concavity_se
## [1] 0
##
## $concave.points_se
## [1] 0
##
## $symmetry_se
## [1] 0
##
## $fractal_dimension_se
## [1] 0
##
## $radius_worst
## [1] 0
##
## $texture_worst
## [1] 0
##
## $perimeter_worst
## [1] 0
##
## $area_worst
## [1] 0
##
## $smoothness_worst
## [1] 0
##
## $compactness_worst
## [1] 0
##
## $concavity_worst
## [1] 0
##
## $concave.points_worst
## [1] 0
##
## $symmetry_worst
## [1] 0
##
## $fractal_dimension_worst
## [1] 0
```

The analysis shows that there are no missing (NA) values in the dataset. However, the class distribution is slightly imbalanced, as revealed by the proportion analysis:

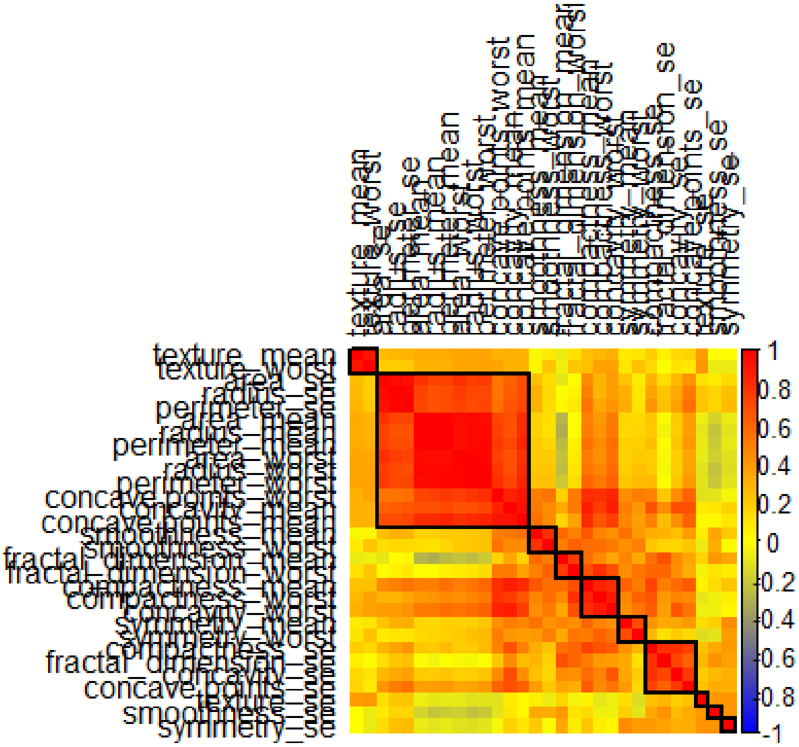


Most variables in the dataset are normally distributed, as shown in the plot below.

:



We now need to check for correlations between variables, as many machine learning algorithms assume that predictor variables are independent of one another.



As illustrated in the plot, many variables in the dataset are highly correlated with one another. This can negatively impact the performance of certain machine learning models, which often perform better when redundant or highly correlated features are removed. The caret package in R offers the findCorrelation function, which analyzes the correlation matrix and identifies variables that can be safely removed to reduce multicollinearity. Removing such correlated features helps improve model performance and stability.

```
## Indices of highly correlated features:

##  [1]  7  8 23 21  3 24  1 13 14  2
```

Choosing the right features in a dataset can be the key difference between mediocre performance with long training times and excellent performance with efficient training.

```
## [1] 22
```

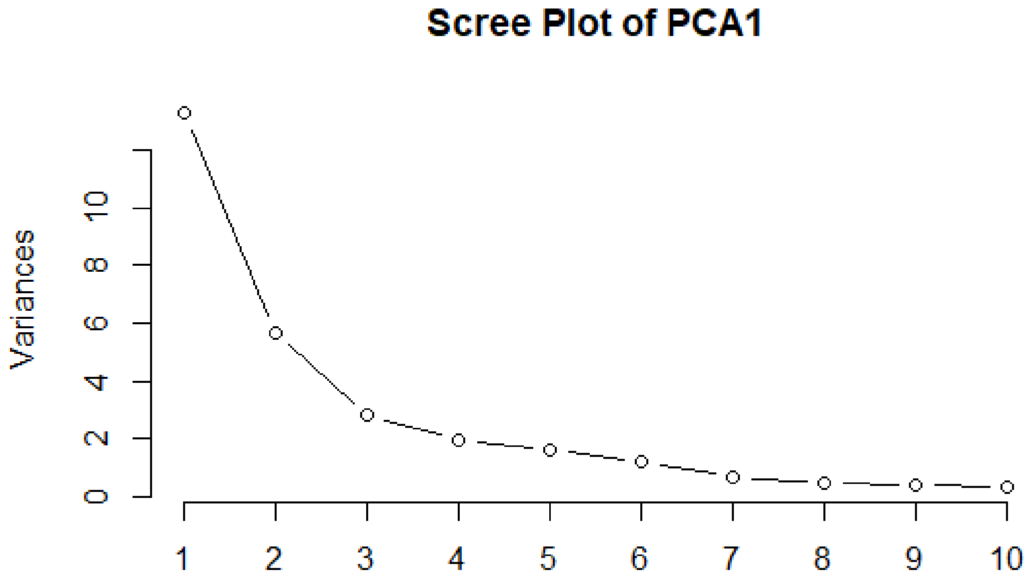
Right now 22 Variables and reduce of 10.

3 Modelling Approach

3.1. Modelling

Principal Component Analysis (PCA).

To reduce redundancy and enhance relevance, Principal Component Analysis (PCA) was applied using the prcomp function. PCA helps address the challenge of analyzing complex data with many correlated variables, which can strain memory and computation. It reduces the dimensionality of the dataset while preserving as much variance as possible. This is achieved by transforming the original correlated features into a new set of orthogonal variables called principal components (PCs). These components are ranked by the amount of variance they capture, allowing for more efficient analysis while minimizing information loss in clustering and classification tasks.

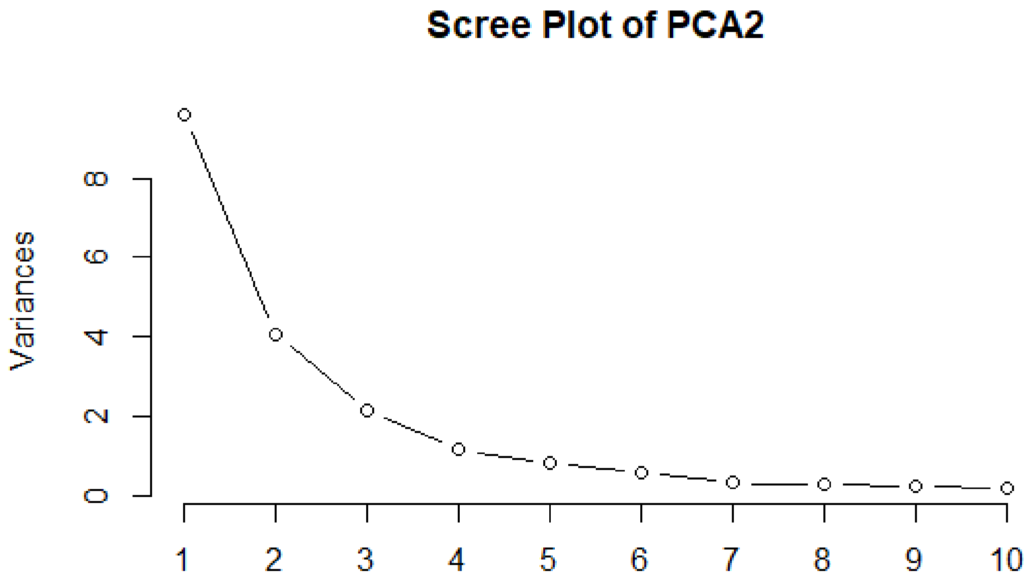


Scree Plot of PCA1

```
## Importance of components:
##                           PC1    PC2     PC3     PC4     PC5     PC6
PC7
## Standard deviation     3.6444 2.3857 1.67867 1.40735 1.28403 1.09880
0.82172
## Proportion of Variance 0.4427 0.1897 0.09393 0.06602 0.05496 0.04025
0.02251
## Cumulative Proportion  0.4427 0.6324 0.72636 0.79239 0.84734 0.88759
0.91010
##                            PC8    PC9    PC10   PC11    PC12    PC13
PC14
## Standard deviation     0.69037 0.6457 0.59219 0.5421 0.51104 0.49128
0.39624
## Proportion of Variance 0.01589 0.0139 0.01169 0.0098 0.00871 0.00805
0.00523
## Cumulative Proportion  0.92598 0.9399 0.95157 0.9614 0.97007 0.97812
0.98335
##                           PC15    PC16    PC17    PC18    PC19    PC20
```

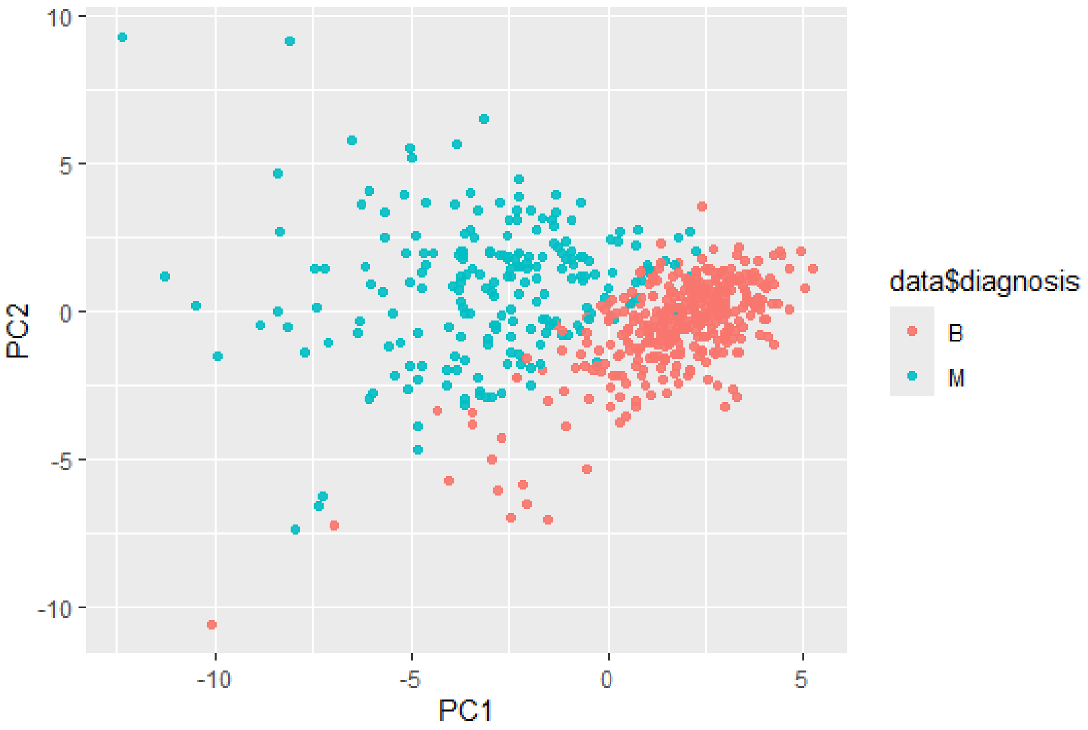
```
PC21
## Standard deviation     0.30681 0.28260 0.24372 0.22939 0.22244 0.17652
0.1731
## Proportion of Variance 0.00314 0.00266 0.00198 0.00175 0.00165 0.00104
0.0010
## Cumulative Proportion  0.98649 0.98915 0.99113 0.99288 0.99453 0.99557
0.9966
##                           PC22    PC23   PC24    PC25    PC26    PC27
PC28
## Standard deviation     0.16565 0.15602 0.1344 0.12442 0.09043 0.08307
0.03987
## Proportion of Variance 0.00091 0.00081 0.0006 0.00052 0.00027 0.00023
0.00005
## Cumulative Proportion  0.99749 0.99830 0.9989 0.99942 0.99969 0.99992
0.99997
##                           PC29    PC30
## Standard deviation     0.02736 0.01153
## Proportion of Variance 0.00002 0.00000
## Cumulative Proportion  1.00000 1.00000
```

As shown in the table above, the first two components explain 0.6324 of the variance. To explain more than 95% of the variance, we need 10 principal components, and 17 components are required to explain over 99% of the variance.

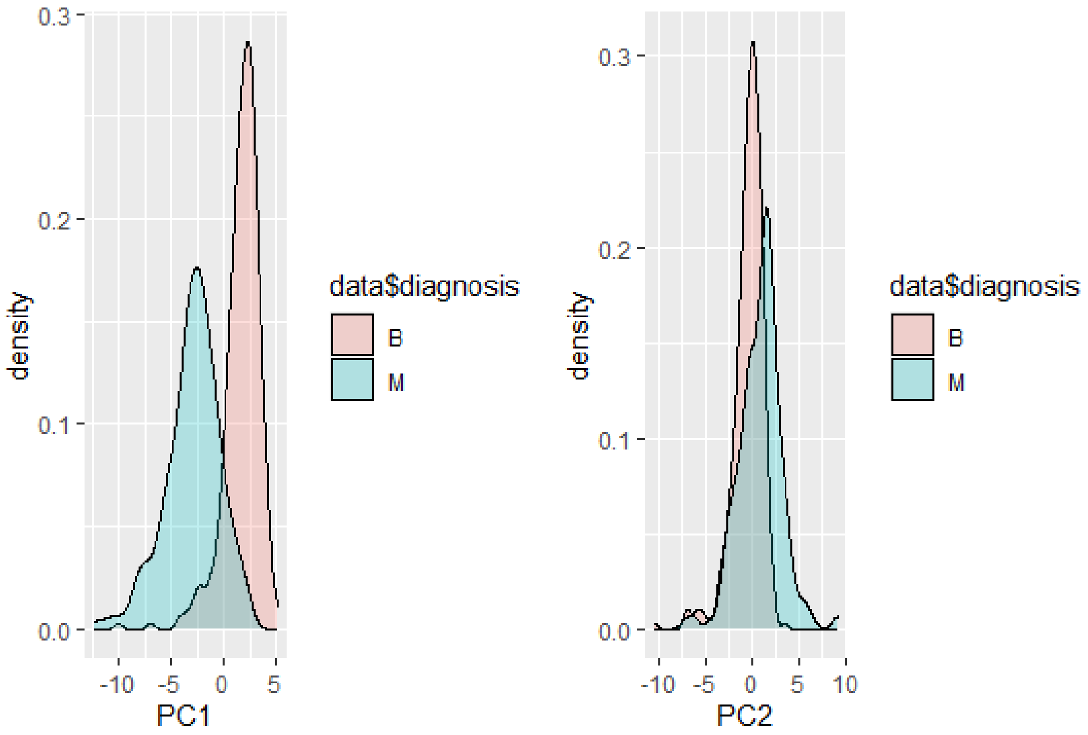


```
## Importance of components:
##                           PC1    PC2    PC3    PC4     PC5     PC6     PC7
## Standard deviation     3.0980 2.0196 1.4663 1.0845 0.91561 0.77019 0.57227
## Proportion of Variance 0.4799 0.2039 0.1075 0.0588 0.04192 0.02966 0.01637
## Cumulative Proportion  0.4799 0.6838 0.7913 0.8501 0.89205 0.92171 0.93808
##                            PC8     PC9    PC10    PC11    PC12    PC13
PC14
## Standard deviation     0.53641 0.50898 0.45726 0.36641 0.31778 0.28802
0.21369
## Proportion of Variance 0.01439 0.01295 0.01045 0.00671 0.00505 0.00415
0.00228
## Cumulative Proportion  0.95247 0.96542 0.97588 0.98259 0.98764 0.99179
0.99407
##                          PC15    PC16    PC17    PC18    PC19    PC20
## Standard deviation     0.1846 0.15579 0.15393 0.14782 0.09636 0.07375
## Proportion of Variance 0.0017 0.00121 0.00118 0.00109 0.00046 0.00027
## Cumulative Proportion  0.9958 0.99699 0.99817 0.99926 0.99973 1.00000
```

The table above demonstrates that 95% of the variance in the transformed dataset (dt2) is explained by the first 8 principal components.



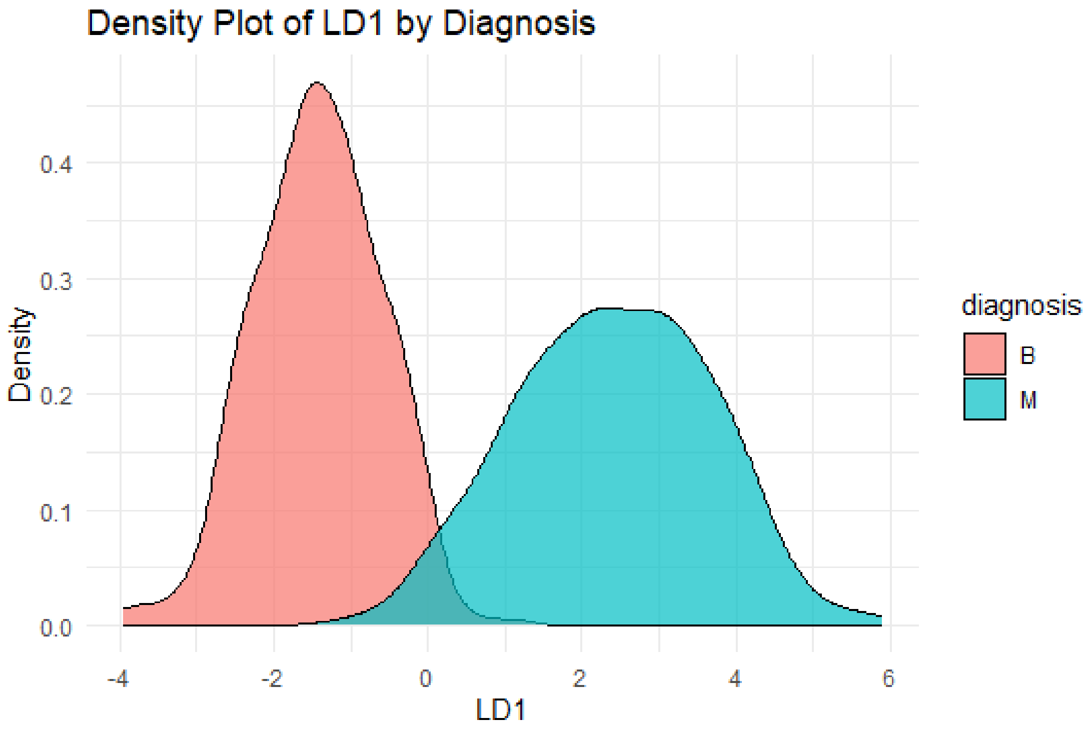
The data for the first two components can be easily separated into two classes. This is due to the relatively small variance explained by these components, making the separation straightforward.



Linear Discriminant Analysis (LDA) Another approach is to use Linear Discriminant Analysis (LDA) instead of PCA. Unlike PCA, LDA takes class labels into account and can often yield better results.

The key feature of LDA is that it models the distribution of predictors separately for each response class, then applies Bayes' Theorem to estimate the class probabilities. It's important to note that LDA assumes each class follows a normal distribution, with a class-specific mean and a shared variance across classes.

```
## Call:
## lda(diagnosis ~ ., data = data, center = TRUE, scale = TRUE)
##
## Prior probabilities of groups:
##         B         M
## 0.6274165 0.3725835
##
## Group means:
##         id radius_mean texture_mean perimeter_mean area_mean
smoothness_mean
## B 26543825    12.14652     17.91476       78.07541  462.7902
0.09247765
## M 36818050    17.46283     21.60491      115.36538  978.3764
0.10289849
##   compactness_mean concavity_mean concave.points_mean symmetry_mean
## B       0.08008462     0.04605762          0.02571741      0.174186
## M       0.14518778     0.16077472          0.08799000      0.192909
##   fractal_dimension_mean radius_se texture_se perimeter_se  area_se
## B             0.06286739 0.2840824   1.220380     2.000321 21.13515
## M             0.06268009 0.6090825   1.210915     4.323929 72.67241
##   smoothness_se compactness_se concavity_se concave.points_se symmetry_se
## B   0.007195902     0.02143825   0.02599674       0.009857653  0.02058381
## M   0.006780094     0.03228117   0.04182401       0.015060472  0.02047240
##   fractal_dimension_se radius_worst texture_worst perimeter_worst
area_worst
## B          0.003636051     13.37980      23.51507        87.00594
558.8994
## M          0.004062406     21.13481      29.31821       141.37033
1422.2863
##   smoothness_worst compactness_worst concavity_worst concave.points_worst
## B        0.1249595         0.1826725       0.1662377           0.07444434
## M        0.1448452         0.3748241       0.4506056           0.18223731
##   symmetry_worst fractal_dimension_worst
## B      0.2702459              0.07944207
## M      0.3234679              0.09152995
##
## Coefficients of linear discriminants:
##                                   LD1
## id                      -2.512117e-10
## radius_mean             -1.080876e+00
## texture_mean             2.338408e-02
## perimeter_mean           1.172707e-01
## area_mean                1.595690e-03
## smoothness_mean          5.251575e-01
## compactness_mean        -2.094197e+01
## concavity_mean           6.955923e+00
## concave.points_mean      1.047567e+01
## symmetry_mean            4.938898e-01
## fractal_dimension_mean  -5.937663e-02
## radius_se                2.101503e+00
## texture_se              -3.979869e-02
## perimeter_se            -1.121814e-01
## area_se                 -4.083504e-03
## smoothness_se            7.987663e+01
## compactness_se           1.387026e-01
## concavity_se            -1.768261e+01
## concave.points_se        5.350520e+01
## symmetry_se              8.143611e+00
## fractal_dimension_se    -3.431356e+01
## radius_worst             9.677207e-01
## texture_worst            3.540591e-02
## perimeter_worst         -1.204507e-02
## area_worst              -5.012127e-03
## smoothness_worst         2.612258e+00
## compactness_worst        3.636892e-01
## concavity_worst          1.880699e+00
## concave.points_worst     2.218189e+00
## symmetry_worst           2.783102e+00
## fractal_dimension_worst  2.117830e+01
```



3.2. Model creation

We will split the modified dataset into training (80%) and testing (20%) sets to build machine learning classification models. These models will be used to predict whether a cancer cell is benign or malignant.

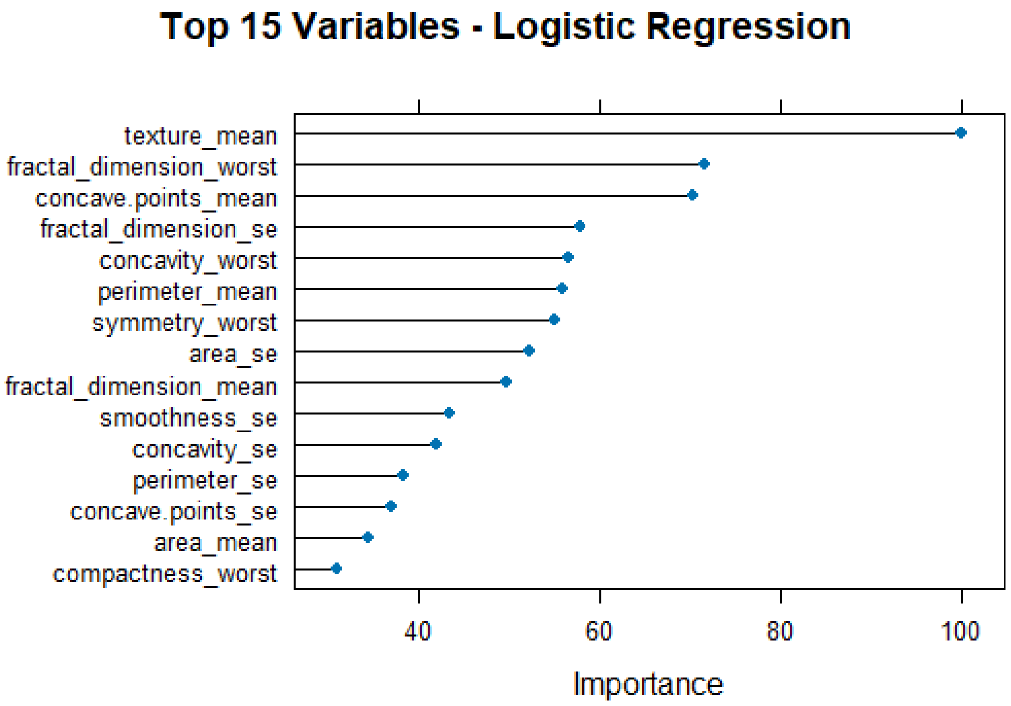
3.2.1 Logistic Regression Model

Logistic Regression is a widely used algorithm for binary classification tasks, such as distinguishing between classes labeled 0 and 1. It models the probability of a binary outcome based on one or more predictor (independent) variables or features.

```
## Confusion Matrix and Statistics
## 
##             Reference
## Prediction   B  M
##          B 67  0
##          M  4 42
## 
##                Accuracy : 0.9646
##                  95% CI : (0.9118, 0.9903)
##     No Information Rate : 0.6283
##     P-Value [Acc > NIR] : <2e-16
## 
##                   Kappa : 0.9257
```

```
##
##  Mcnemar's Test P-Value : 0.1336
##
##             Sensitivity : 1.0000
##             Specificity : 0.9437
##          Pos Pred Value : 0.9130
##          Neg Pred Value : 1.0000
##              Prevalence : 0.3717
##          Detection Rate : 0.3717
##    Detection Prevalence : 0.4071
##       Balanced Accuracy : 0.9718
##
##        'Positive' Class : M
##
```

The most important variables that permit the best prediction and contribute the most to the model are the following:



We can note the accuracy with such model. We will later describe better these metrics, where: Sensitivity (recall) represent the true positive rate: the proportions of actual positives correctly identified. Specificity is the true negative rate: the proportion of actual negatives correctly identified. Accuracy is the general score of the classifier model performance as it is the ratio of how many samples are correctly classified to all samples. F1 score: the harmonic mean of precision and sensitivity. Accuracy and F1 score would be used to compare the result with the benchmark model. Precision: the number of correct positive results divided by the number of all positive results returned by the classifier.

The following variables are the most significant contributors to the model’s predictive performance and play a key role in achieving accurate predictions:

3.2.2. Neural Network with PCA Model

Artificial Neural Networks (ANNs) are a class of mathematical algorithms inspired by the structure and function of biological neural networks. An ANN consists of interconnected nodes (called neurons) and connections between them (called synapses). Input data is passed through these weighted synapses to the neurons, where computations are performed. The results are then either forwarded to other neurons in subsequent layers or used to produce the final output.

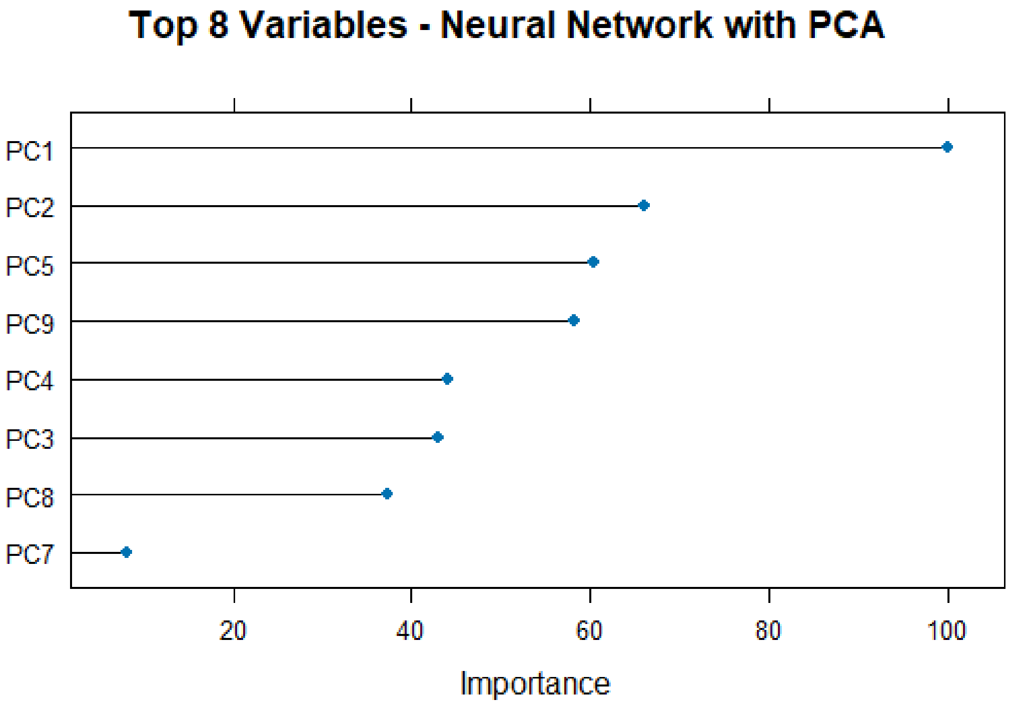
Neural networks learn by adjusting the weights of these connections based on the input data. Through training, the model iteratively updates the weights to minimize prediction errors. Once the network is fully trained, it can be used to classify new data points or, in the case of regression tasks, predict continuous values.

One of the key strengths of neural networks is their ability to model highly complex relationships without the need for extensive feature engineering. They can function effectively as “black box” models, handling raw or minimally processed input data. When combined with deep learning architectures (multi-layer networks), even more sophisticated patterns and representations can be learned, opening up powerful possibilities for advanced data analysis and prediction.

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction  B  M
##          B 70  0
##          M  1 42
## 
##                Accuracy : 0.9912
##                  95% CI : (0.9517, 0.9998)
##     No Information Rate : 0.6283
##     P-Value [Acc > NIR] : <2e-16
## 
##                   Kappa : 0.9811
## 
##  Mcnemar's Test P-Value : 1
## 
##             Sensitivity : 1.0000
##             Specificity : 0.9859
##          Pos Pred Value : 0.9767
##          Neg Pred Value : 1.0000
##              Prevalence : 0.3717
##          Detection Rate : 0.3717
##    Detection Prevalence : 0.3805
##       Balanced Accuracy : 0.9930
## 
```

```
##        'Positive' Class : M
##
```

The most influential variables that contribute significantly to the model’s predictive performance are as follows:



3.2.3. Neural Network with LDA Model

We will now create training and test sets from the LDA-transformed data generated in the previous sections.

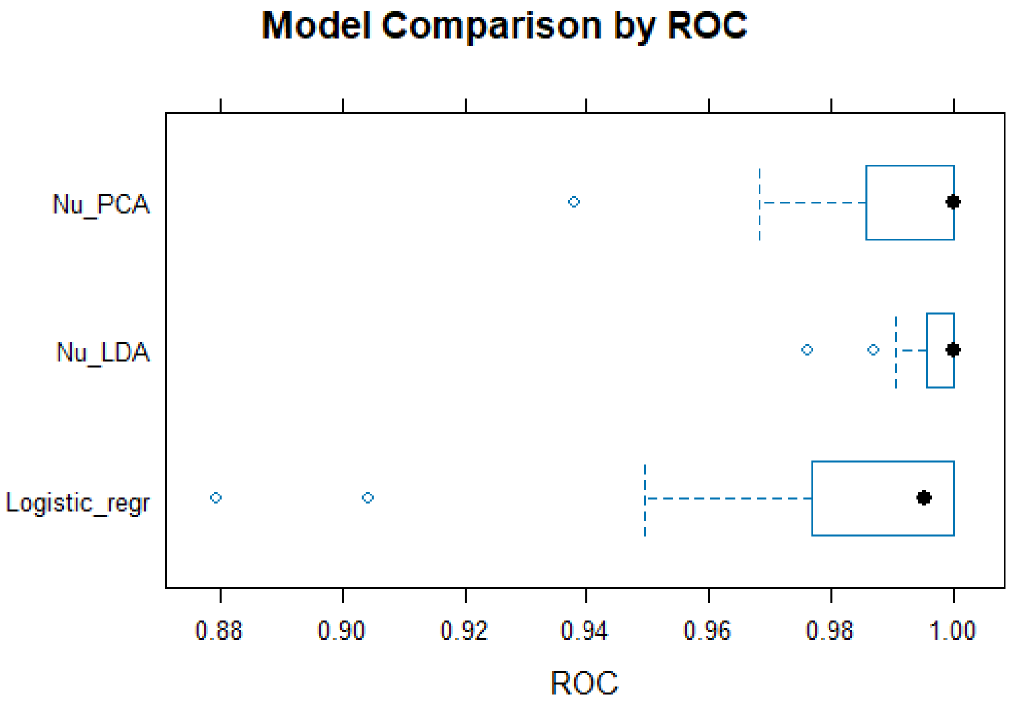
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  B  M
##          B 70  1
##          M  1 41
##
##                Accuracy : 0.9823
##                  95% CI : (0.9375, 0.9978)
##     No Information Rate : 0.6283
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.9621
##
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 0.9762
##             Specificity : 0.9859
##          Pos Pred Value : 0.9762
##          Neg Pred Value : 0.9859
##              Prevalence : 0.3717
##          Detection Rate : 0.3628
##    Detection Prevalence : 0.3717
##       Balanced Accuracy : 0.9811
##
##        'Positive' Class : M
##
```

4. Results

We can now proceed to compare and evaluate the results based on the calculations presented above.

```
##
## Call:
## summary.resamples(object = mdls_results)
##
## Models: Logistic_regr, Nu_PCA, Nu_LDA
## Number of resamples: 15
##
## ROC
##                    Min.   1st Qu.    Median      Mean 3rd Qu. Max. NA's
## Logistic_regr 0.8793860 0.9766746 0.9952153 0.9769431       1    1    0
## Nu_PCA        0.9377990 0.9858453 1.0000000 0.9899309       1    1    0
## Nu_LDA        0.9760766 0.9956140 1.0000000 0.9963052       1    1    0
##
## Sens
##                    Min.   1st Qu.    Median      Mean 3rd Qu. Max. NA's
## Logistic_regr 0.8421053 0.9473684 0.9473684 0.9580702       1    1    0
## Nu_PCA        0.8947368 1.0000000 1.0000000 0.9859649       1    1    0
## Nu_LDA        0.9473684 0.9736842 1.0000000 0.9859649       1    1    0
##
## Spec
##                    Min.   1st Qu.    Median      Mean 3rd Qu. Max. NA's
## Logistic_regr 0.8181818 0.9090909 0.9166667 0.9414141       1    1    0
## Nu_PCA        0.8181818 0.9090909 0.9166667 0.9409091       1    1    0
## Nu_LDA        0.8181818 0.9583333 1.0000000 0.9590909       1    1    0
```

As shown in the following plot, Logistic Regression models exhibit significant variability in performance, depending on the sample being processed.



The Neural Network with LDA model achieved a strong Area Under the ROC Curve (AUC), though with some variability. The ROC (Receiver Operating Characteristic) curve is a graphical representation of a classification model's performance across all possible classification thresholds. The AUC quantifies the overall ability of the model to distinguish between classes, regardless of the threshold used.

It's important to note that the default classification threshold is typically set at 0.5. However, in imbalanced datasets like this one, a threshold of 0.5 may not yield optimal results. Adjusting the threshold can significantly improve model performance, particularly in terms of sensitivity or specificity, depending on the clinical priority.

	Logistic_regr	Nu_PCA	Nu_LDA
Sensitivity	1.0000000	1.0000000	0.9761905
Specificity	0.9436620	0.9859155	0.9859155
Pos Pred Value	0.9130435	0.9767442	0.9761905
Neg Pred Value	1.0000000	1.0000000	0.9859155
Precision	0.9130435	0.9767442	0.9761905
Recall	1.0000000	1.0000000	0.9761905
F1	0.9545455	0.9882353	0.9761905
Prevalence	0.3716814	0.3716814	0.3716814
Detection Rate	0.3716814	0.3716814	0.3628319
Detection Prevalence	0.4070796	0.3805310	0.3716814
Balanced Accuracy	0.9718310	0.9929577	0.9810530

5. Discussion

We will now describe the metrics that we will compare in this section.

Accuracy is our starting point. It is the number of correct predictions made divided by the total number of predictions made, multiplied by 100 to turn it into a percentage.

Precision is the number of True Positives divided by the number of True Positives and False Positives. Put another way, it is the number of positive predictions divided by the total number of positive class values predicted. It is also called the Positive Predictive Value (PPV). A low precision can also indicate a large number of False Positives.

Recall (Sensitivity) is the number of True Positives divided by the number of True Positives and the number of False Negatives. Put another way it is the number of positive predictions divided by the number of positive class values in the test data. It is also called Sensitivity or the True Positive Rate. Recall can be thought of as a measure of a classifiers completeness. A low recall indicates many False Negatives.

The F1 Score is the 2 x ((precision x recall) / (precision + recall)). It is also called the F Score or the F Measure. Put another way, the F1 score conveys the balance between the precision and the recall.

The Neural Network combined with LDA achieved the highest sensitivity for detecting malignant breast cancer cases and also demonstrated a strong F1 score, making it the most effective model overall.

```
##                  metric   best_model     value
## 1           Sensitivity       Nu_PCA 1.0000000
## 2           Specificity       Nu_PCA 0.9859155
## 3        Pos Pred Value       Nu_PCA 0.9767442
## 4        Neg Pred Value Logistic_regr 1.0000000
## 5             Precision       Nu_PCA 0.9767442
## 6                Recall Logistic_regr 1.0000000
## 7                    F1       Nu_PCA 0.9882353
## 8            Prevalence Logistic_regr 0.3716814
## 9        Detection Rate       Nu_PCA 0.3716814
## 10 Detection Prevalence Logistic_regr 0.4070796
## 11    Balanced Accuracy       Nu_PCA 0.9929577
```

6. Conclusion & Recommendation

This paper approaches the Wisconsin Breast Cancer Diagnosis problem as a pattern classification task. Several machine learning models were evaluated, with the optimal model selected based on a combination of high accuracy and a low false-negative rate—reflected by high sensitivity.

The Neural Network combined with Principal Component Analysis (PCA) yielded the best performance, achieving an F1 score of 0.9882, a sensitivity of 1.000, and a balanced accuracy of 0.9930.

For the future work, it is recommended to deploy the model using SVM and Randomforest and comparison of the models performance for the innovation of the variety of methods.

7. References

1.Irizarry, R., 2019.Introduction To Data Science.[online] Rafalab.github.io. Available at:<https://rafalab.dfci.harvard.edu/dsbook/>

2.”UCI Machine Learning Repository: Breast Cancer Data Set.” [Online]. Available: https://www.kaggle.com/datasets/uciml/breast-cancer-wisconsin-data?select=data.csv

3.https://www.geeksforgeeks.org/boosting-in-machine-learning-boosting-and-adaboost/

4.“Introduction to Machine Learning with Python” https://www.oreilly.com/library/view/introduction-to-machine/9781449369880/

5.“ Fundamentals of Machine Learning for Predictive Data Analytics: Algorithms, Worked Examples, and Case Studies by John D. Kelleher, Brian Mac Namee, and Aoife D’Arcy” https://mitpress.mit.edu/9780262029445/fundamentals-of-machine-learning-for-predictive-data-analytics/

6.H.Asri, H. Mousannif, H. A. Moatassime, and T.Noel, ‘Using Machine Learning Algorithms for Breast Cancer Risk Prediction and Diagnosis’, Procedia Computer Science, vol.83, pp. 1064–1069, 2016,doi:10.1016/j.procs.2016.04.224.

7.Y.khoudfi and M.Bahaj, Applying Best Machine Learning Algorithms for Breast Cancer Prediction and Classification, 978-1-5386- 4225-2/18/ ©2018 IEEE.