Precision Prognisis: Advanced Predictive Modeling for Breast Cancer Diagnosis at UCI

Ghea Marie Comendador

2023-12-21

INTRODUCTION

Breast cancer, a complex and diverse disease, poses a global health challenge due to uncontrolled cell growth in breast tissue. Diverse subtypes with distinct molecular profiles influence prognosis and treatment strategies. The World Health Organization highlights breast cancer as the most prevalent cancer in women globally, emphasizing the need for ongoing research and advancements (DeSantis et al., 2019; WHO, 2021).

Early detection is crucial for improving breast cancer outcomes. Common screening methods include mammography, clinical examinations, and self-breast examinations. Genetic factors (BRCA1/BRCA2 mutations), environmental influences, and lifestyle choices contribute to breast cancer risk. Datasets like the Breast Cancer Wisconsin (Diagnostic) Data Set play a vital role in unraveling the disease's complexities (DeSantis et al., 2019).

This paper aims to analyze the Breast Cancer Wisconsin (Diagnostic) Data Set comprehensively. The focus is on understanding and predicting tumor nature using crucial variables, particularly the binary diagnosis variable (benign: 0, malignant: 1). Models leveraging predictor variables related to cell nuclei characteristics are developed. A comparative approach, stratifying cases based on diagnosis, aids in uncovering distinct patterns associated with each group.

Exploratory data analysis (EDA) involves steps like visualizing correlation, handling missing values, data splitting, and feature scaling. Various machine learning models, including Support Vector Machine, K-Nearest Neighbors, Naive Bayes, and Random Forest, address crucial questions related to the dataset. These models serve as a preliminary exploration into predictive patterns, contributing to a comprehensive understanding of relationships within the breast cancer dataset.

The Breast Cancer Wisconsin (Diagnostic) Data Set is a valuable resource for researchers, data scientists, and healthcare professionals. It offers a standardized collection of features from FNA images, allowing the development and evaluation of machine learning models for breast cancer diagnosis. The dataset's inclusion of mean, standard error, and "worst" values across multiple images enhances its utility. With a Kaggle usability rating of 8.53, the dataset proves its relevance for exploring patterns and improving breast cancer diagnosis accuracy, contributing to public health efforts.

DATA

The Breast Cancer Wisconsin (Diagnostic) Data Set, available on Kaggle at https://www.kaggle.com/datasets/uciml/breast-cancer-wisconsin-data, boasts a usability rating of 8.53. The dataset is licensed under CC BY-NC-SA 4.0 and falls within the domain of cancer and healthcare. While the expected update frequency is not specified, this resource provides valuable information for those engaged in cancer research and healthcare analytics, offering a comprehensive collection of data for the diagnosis of breast cancer.

The dataset under consideration involves the computation of ten real-valued features from digitized fine needle aspirate (FNA) images of breast masses. These features, derived from the characteristics of cell nuclei present in the images, include measurements such as radius, texture, perimeter, area, smoothness, compactness, concavity, concave points, symmetry, and fractal dimension.

The dataset, originating from research by K. P. Bennett and O. L. Mangasarian, is situated in a 3-dimensional space defined in their publication on robust linear programming discrimination. The database, comprising 569 instances, is accessible through the UW CS ftp server and is also featured on the UCI Machine Learning Repository. Each instance is characterized by an ID number, a diagnosis (malignant or benign), and thirty computed features representing the mean, standard error, and "worst" or largest values of the ten original features. The dataset is devoid of missing attribute values, and the class distribution consists of 357 benign and 212 malignant cases.

```
#read the file
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
getwd()
```

```
## [1] "C:/Users/comen/Documents/4th Year/118/118 Lab/118 Final Paper"
```

```
data_cancer <- read.csv("C:/Users/comen/Documents/4th Year/118/118 Lab/118 Final Paper/data.csv")
glimpse(data_cancer)
```

```
## Rows: 569
## Columns: 33
## $ id                      <int> 842302, 842517, 84300903, 84348301, 84358402, ~
## $ diagnosis               <chr> "M", "M", "M", "M", "M", "M", "M", "M", "M", "~
## $ radius_mean             <dbl> 17.990, 20.570, 19.690, 11.420, 20.290, 12.450~
## $ texture_mean            <dbl> 10.38, 17.77, 21.25, 20.38, 14.34, 15.70, 19.9~
## $ perimeter_mean          <dbl> 122.80, 132.90, 130.00, 77.58, 135.10, 82.57, ~
## $ area_mean               <dbl> 1001.0, 1326.0, 1203.0, 386.1, 1297.0, 477.1, ~
## $ smoothness_mean         <dbl> 0.11840, 0.08474, 0.10960, 0.14250, 0.10030, 0~
## $ compactness_mean        <dbl> 0.27760, 0.07864, 0.15990, 0.28390, 0.13280, 0~
## $ concavity_mean          <dbl> 0.30010, 0.08690, 0.19740, 0.24140, 0.19800, 0~
## $ concave.points_mean     <dbl> 0.14710, 0.07017, 0.12790, 0.10520, 0.10430, 0~
## $ symmetry_mean           <dbl> 0.2419, 0.1812, 0.2069, 0.2597, 0.1809, 0.2087~
## $ fractal_dimension_mean  <dbl> 0.07871, 0.05667, 0.05999, 0.09744, 0.05883, 0~
## $ radius_se               <dbl> 1.0950, 0.5435, 0.7456, 0.4956, 0.7572, 0.3345~
## $ texture_se              <dbl> 0.9053, 0.7339, 0.7869, 1.1560, 0.7813, 0.8902~
## $ perimeter_se            <dbl> 8.589, 3.398, 4.585, 3.445, 5.438, 2.217, 3.18~
## $ area_se                 <dbl> 153.40, 74.08, 94.03, 27.23, 94.44, 27.19, 53.~
## $ smoothness_se           <dbl> 0.006399, 0.005225, 0.006150, 0.009110, 0.0114~
## $ compactness_se          <dbl> 0.049040, 0.013080, 0.040060, 0.074580, 0.0246~
## $ concavity_se            <dbl> 0.05373, 0.01860, 0.03832, 0.05661, 0.05688, 0~
## $ concave.points_se       <dbl> 0.015870, 0.013400, 0.020580, 0.018670, 0.0188~
## $ symmetry_se             <dbl> 0.03003, 0.01389, 0.02250, 0.05963, 0.01756, 0~
## $ fractal_dimension_se    <dbl> 0.006193, 0.003532, 0.004571, 0.009208, 0.0051~
## $ radius_worst            <dbl> 25.38, 24.99, 23.57, 14.91, 22.54, 15.47, 22.8~
## $ texture_worst           <dbl> 17.33, 23.41, 25.53, 26.50, 16.67, 23.75, 27.6~
## $ perimeter_worst         <dbl> 184.60, 158.80, 152.50, 98.87, 152.20, 103.40,~
## $ area_worst              <dbl> 2019.0, 1956.0, 1709.0, 567.7, 1575.0, 741.6, ~
## $ smoothness_worst        <dbl> 0.1622, 0.1238, 0.1444, 0.2098, 0.1374, 0.1791~
## $ compactness_worst       <dbl> 0.6656, 0.1866, 0.4245, 0.8663, 0.2050, 0.5249~
## $ concavity_worst         <dbl> 0.71190, 0.24160, 0.45040, 0.68690, 0.40000, 0~
## $ concave.points_worst    <dbl> 0.26540, 0.18600, 0.24300, 0.25750, 0.16250, 0~
## $ symmetry_worst          <dbl> 0.4601, 0.2750, 0.3613, 0.6638, 0.2364, 0.3985~
## $ fractal_dimension_worst <dbl> 0.11890, 0.08902, 0.08758, 0.17300, 0.07678, 0~
## $ X                       <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
```

DATA ANALYSIS

Data analysis of the Breast Cancer Wisconsin (Diagnostic) Data Set analysis, the primary focus revolves around understanding and predicting the nature of tumors based on crucial variables. The outcome variable, denoted as Y, is represented by the "diagnosis" feature, which serves as a binary indicator distinguishing between benign (coded as 0) and malignant (coded as 1) cases. This binary classification forms the crux of the analysis, aiming to develop models that effectively differentiate between these two health conditions.

Complementing the outcome variable, the predictor variables, denoted as X, encompass an array of features describing the intricate characteristics of cell nuclei. These features likely include parameters such as radius, texture, perimeter, area, smoothness, compactness, concavity, and various others that are fundamental in characterizing cell structures. The objective is to leverage these predictor variables to build predictive models that discern patterns and relationships, ultimately contributing to accurate predictions regarding the malignancy or benign nature of tumors. The comprehensive exploration of these variables forms the foundation for subsequent exploratory data analysis and the application of machine learning models in the quest to unravel insights into breast cancer diagnoses.

The comparison groups are stratified based on the binary diagnosis variable, categorizing cases into benign (0) and malignant (1) groups. This dichotomous classification sets the stage for a meticulous examination of the dataset, facilitating a thorough investigation into the features associated with each group. By segregating cases according to their diagnosis, the analysis aims to uncover distinct patterns, variations, or trends in the selected features, shedding light on the specific attributes that play a pivotal role in distinguishing between benign and malignant cases. This comparative approach forms an essential component of the exploratory data analysis, enabling a nuanced understanding of the factors contributing to the classification of tumors and laying the groundwork for subsequent modeling and predictive analytics.

Breast Cancer Wisconsin (Diagnostic) Data Set involves a comprehensive Preliminary Exploratory Data Analysis (EDA) to lay the groundwork for subsequent modeling. The EDA encompasses several key steps.

Visualizing Correlation, the code employs the 'ggcorrplot' function to generate a correlation matrix, enhancing interpretability by reordering it. This visualization is crucial for understanding the relationships between different variables in the dataset, and it includes the display of significance levels on the correlogram, providing additional insights.

Handling Missing Values, utilizing the 'naniar' package, the code systematically visualizes missing values within the dataset. The graphical representation indicates an absence of missing values in any of the dataset's columns, affirming the dataset's completeness and reliability.

Data Splitting, the dataset is divided into training and test sets using a split ratio of 75-25, a common practice in machine learning. This step is essential for training predictive models on one subset and evaluating their performance on another, ensuring a robust assessment of model generalization.

RESULTS AND DISCUSSION

Upon reviewing the dataset, it becomes evident that most variables are already in numeric form, except for the diagnosis column, which uses "M" and "B" to denote malignant and benign cases, respectively. To facilitate analysis and avoid errors, a prudent step is to convert these categorical values into numeric representations. The process involves first converting factors to characters and then further converting them into numeric format.

Additionally, to enhance clarity and prevent confusion during analysis, it is advisable to reposition the dependent variable, diagnosis, to the extreme right of the dataset. This adjustment aids in maintaining a structured and organized data frame. Furthermore, visualizing the correlation between datasets is crucial for gaining insights into the relationships among different variables, providing a foundation for subsequent exploratory data analysis and modeling efforts.

```
# Import needed libraries

library(readr) # Data reading and manipulation
library(ggplot2) # Data visualization
```

```
## Warning: package 'ggplot2' was built under R version 4.3.2
```

```
library(plotly) # Data visualization
```

```
## Warning: package 'plotly' was built under R version 4.3.2
```

```
##
## Attaching package: 'plotly'
```

```
## The following object is masked from 'package:ggplot2':
##
##     last_plot
```

```
## The following object is masked from 'package:stats':
##
##     filter
```

```
## The following object is masked from 'package:graphics':
##
##     layout
```

```
library(ggcorrplot) # Data visualization and finding the correlation with variables
```

```
## Warning: package 'ggcorrplot' was built under R version 4.3.2
```

```r
library(dplyr) # Data manipulation and transformation
library(tidyverse) # Data manipulation and transformation
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v forcats   1.0.0     v stringr   1.5.0
## v lubridate 1.9.2     v tibble    3.2.1
## v purrr     1.0.2     v tidyr     1.3.0

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x plotly::filter() masks dplyr::filter(), stats::filter()
## x dplyr::lag()     masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```r
library(naniar) # Handling missing values visualization
```

```
## Warning: package 'naniar' was built under R version 4.3.2
```

```r
library(caTools) # Data splitting
```

```
## Warning: package 'caTools' was built under R version 4.3.2
```

```r
library(caret) # Model building and evaluation
```

```
## Warning: package 'caret' was built under R version 4.3.2

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

```r
# Import data
getwd()
```

```
## [1] "C:/Users/comen/Documents/4th Year/118/118 Lab/118 Final Paper"
```

```r
data_cancer <- read.csv("C:/Users/comen/Documents/4th Year/118/118 Lab/118 Final Paper/data.csv")
head(data_cancer)
```

```
##         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  X
## 1               0.2654         0.4601                 0.11890 NA
## 2               0.1860         0.2750                 0.08902 NA
## 3               0.2430         0.3613                 0.08758 NA
## 4               0.2575         0.6638                 0.17300 NA
## 5               0.1625         0.2364                 0.07678 NA
## 6               0.1741         0.3985                 0.12440 NA
```

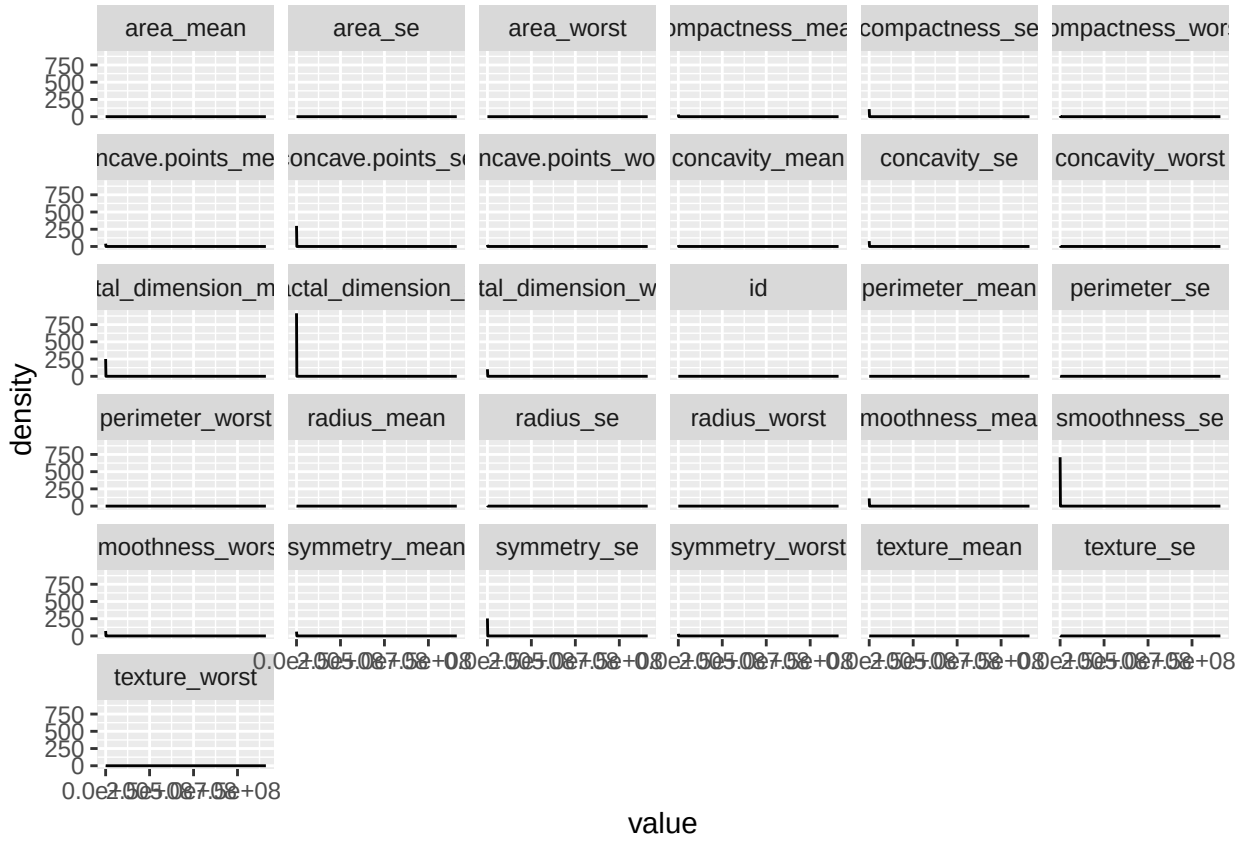
```
str(data_cancer)
```

```
## 'data.frame':    569 obs. of  33 variables:
##  $ id                      : int  842302 842517 84300903 84348301 84358402 843786 844359 84458202 844
##  $ 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 ...
##  $ X                      : logi  NA NA NA NA NA NA ...
```

```
# Visualize all the variable in the data frame
data_1 <- data_cancer %>%
  as.data.frame() %>%
  select_if(is.numeric) %>%
  gather(key = "variable", value = "value")

ggplot(data_1, aes(value)) +
  geom_density() +
  facet_wrap(~variable)
```



We have all the data in the numeric form, except diagnosis which is M and B

```
# Convert this into numeric
data_cancer$diagnosis <- factor(data_cancer$diagnosis, levels = c("M","B"), labels = c(0,1))
```

```
# Converting factors to character and then character to numeric, if we convert this directly to numeric
data_cancer$diagnosis <- as.character(data_cancer$diagnosis)
data_cancer$diagnosis <- as.numeric(data_cancer$diagnosis)
str(data_cancer)
```

```
## 'data.frame':    569 obs. of  33 variables:
##  $ id                      : int  842302 842517 84300903 84348301 84358402 843786 844359 84458202 844
##  $ diagnosis               : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ 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 ...
##  $ X                      : logi  NA NA NA NA NA NA ...
```

```
# Reordering the dependent variable, i.e., diagnosis, to the far right of the dataset is undertaken to
data_cancer <- data_cancer %>% relocate(diagnosis,.after= fractal_dimension_worst)
```

```
# Visualising the correlation between datasets
r <- cor(data_cancer[,3:32])

round(r,2)
```

```
##                         texture_mean perimeter_mean area_mean smoothness_mean
## texture_mean                    1.00           0.33      0.32           -0.02
## perimeter_mean                  0.33           1.00      0.99            0.21
## area_mean                       0.32           0.99      1.00            0.18
## smoothness_mean                -0.02           0.21      0.18            1.00
## compactness_mean                0.24           0.56      0.50            0.66
## concavity_mean                  0.30           0.72      0.69            0.52
## concave.points_mean             0.29           0.85      0.82            0.55
## symmetry_mean                   0.07           0.18      0.15            0.56
## fractal_dimension_mean         -0.08          -0.26     -0.28            0.58
## radius_se                       0.28           0.69      0.73            0.30
## texture_se                      0.39          -0.09     -0.07            0.07
## perimeter_se                    0.28           0.69      0.73            0.30
## area_se                         0.26           0.74      0.80            0.25
## smoothness_se                   0.01          -0.20     -0.17            0.33
## compactness_se                  0.19           0.25      0.21            0.32
## concavity_se                    0.14           0.23      0.21            0.25
## concave.points_se               0.16           0.41      0.37            0.38
## symmetry_se                     0.01          -0.08     -0.07            0.20
## fractal_dimension_se            0.05          -0.01     -0.02            0.28
## radius_worst                    0.35           0.97      0.96            0.21
## texture_worst                   0.91           0.30      0.29            0.04
## perimeter_worst                 0.36           0.97      0.96            0.24
```

```
## area_worst                     0.34           0.94      0.96            0.21
## smoothness_worst               0.08           0.15      0.12            0.81
## compactness_worst              0.28           0.46      0.39            0.47
## concavity_worst                0.30           0.56      0.51            0.43
## concave.points_worst           0.30           0.77      0.72            0.50
## symmetry_worst                 0.11           0.19      0.14            0.39
## fractal_dimension_worst        0.12           0.05      0.00            0.50
## diagnosis                     -0.42          -0.74     -0.71           -0.36
##                        compactness_mean concavity_mean concave.points_mean
## texture_mean                       0.24           0.30                0.29
## perimeter_mean                     0.56           0.72                0.85
## area_mean                          0.50           0.69                0.82
## smoothness_mean                    0.66           0.52                0.55
## compactness_mean                   1.00           0.88                0.83
## concavity_mean                     0.88           1.00                0.92
## concave.points_mean                0.83           0.92                1.00
## symmetry_mean                      0.60           0.50                0.46
## fractal_dimension_mean             0.57           0.34                0.17
## radius_se                          0.50           0.63                0.70
## texture_se                         0.05           0.08                0.02
## perimeter_se                       0.55           0.66                0.71
## area_se                            0.46           0.62                0.69
## smoothness_se                      0.14           0.10                0.03
## compactness_se                     0.74           0.67                0.49
## concavity_se                       0.57           0.69                0.44
## concave.points_se                  0.64           0.68                0.62
## symmetry_se                        0.23           0.18                0.10
## fractal_dimension_se               0.51           0.45                0.26
## radius_worst                       0.54           0.69                0.83
## texture_worst                      0.25           0.30                0.29
## perimeter_worst                    0.59           0.73                0.86
## area_worst                         0.51           0.68                0.81
## smoothness_worst                   0.57           0.45                0.45
## compactness_worst                  0.87           0.75                0.67
## concavity_worst                    0.82           0.88                0.75
## concave.points_worst               0.82           0.86                0.91
## symmetry_worst                     0.51           0.41                0.38
## fractal_dimension_worst            0.69           0.51                0.37
## diagnosis                         -0.60          -0.70               -0.78
##                        symmetry_mean fractal_dimension_mean radius_se
## texture_mean                    0.07                 -0.08      0.28
## perimeter_mean                  0.18                 -0.26      0.69
## area_mean                       0.15                 -0.28      0.73
## smoothness_mean                 0.56                  0.58      0.30
## compactness_mean                0.60                  0.57      0.50
## concavity_mean                  0.50                  0.34      0.63
## concave.points_mean             0.46                  0.17      0.70
## symmetry_mean                   1.00                  0.48      0.30
## fractal_dimension_mean          0.48                  1.00      0.00
## radius_se                       0.30                  0.00      1.00
## texture_se                      0.13                  0.16      0.21
## perimeter_se                    0.31                  0.04      0.97
## area_se                         0.22                 -0.09      0.95
## smoothness_se                   0.19                  0.40      0.16
```

```
## compactness_se                  0.42                  0.56      0.36
## concavity_se                    0.34                  0.45      0.33
## concave.points_se               0.39                  0.34      0.51
## symmetry_se                     0.45                  0.35      0.24
## fractal_dimension_se            0.33                  0.69      0.23
## radius_worst                    0.19                 -0.25      0.72
## texture_worst                   0.09                 -0.05      0.19
## perimeter_worst                 0.22                 -0.21      0.72
## area_worst                      0.18                 -0.23      0.75
## smoothness_worst                0.43                  0.50      0.14
## compactness_worst               0.47                  0.46      0.29
## concavity_worst                 0.43                  0.35      0.38
## concave.points_worst            0.43                  0.18      0.53
## symmetry_worst                  0.70                  0.33      0.09
## fractal_dimension_worst         0.44                  0.77      0.05
## diagnosis                      -0.33                  0.01     -0.57
##                         texture_se perimeter_se area_se smoothness_se
## texture_mean                  0.39         0.28    0.26          0.01
## perimeter_mean               -0.09         0.69    0.74         -0.20
## area_mean                    -0.07         0.73    0.80         -0.17
## smoothness_mean               0.07         0.30    0.25          0.33
## compactness_mean              0.05         0.55    0.46          0.14
## concavity_mean                0.08         0.66    0.62          0.10
## concave.points_mean           0.02         0.71    0.69          0.03
## symmetry_mean                 0.13         0.31    0.22          0.19
## fractal_dimension_mean        0.16         0.04   -0.09          0.40
## radius_se                     0.21         0.97    0.95          0.16
## texture_se                    1.00         0.22    0.11          0.40
## perimeter_se                  0.22         1.00    0.94          0.15
## area_se                       0.11         0.94    1.00          0.08
## smoothness_se                 0.40         0.15    0.08          1.00
## compactness_se                0.23         0.42    0.28          0.34
## concavity_se                  0.19         0.36    0.27          0.27
## concave.points_se             0.23         0.56    0.42          0.33
## symmetry_se                   0.41         0.27    0.13          0.41
## fractal_dimension_se          0.28         0.24    0.13          0.43
## radius_worst                 -0.11         0.70    0.76         -0.23
## texture_worst                 0.41         0.20    0.20         -0.07
## perimeter_worst              -0.10         0.72    0.76         -0.22
## area_worst                   -0.08         0.73    0.81         -0.18
## smoothness_worst             -0.07         0.13    0.13          0.31
## compactness_worst            -0.09         0.34    0.28         -0.06
## concavity_worst              -0.07         0.42    0.39         -0.06
## concave.points_worst         -0.12         0.55    0.54         -0.10
## symmetry_worst               -0.13         0.11    0.07         -0.11
## fractal_dimension_worst      -0.05         0.09    0.02          0.10
## diagnosis                     0.01        -0.56   -0.55          0.07
##                         compactness_se concavity_se concave.points_se
## texture_mean                      0.19         0.14              0.16
## perimeter_mean                    0.25         0.23              0.41
## area_mean                         0.21         0.21              0.37
## smoothness_mean                   0.32         0.25              0.38
## compactness_mean                  0.74         0.57              0.64
## concavity_mean                    0.67         0.69              0.68
```

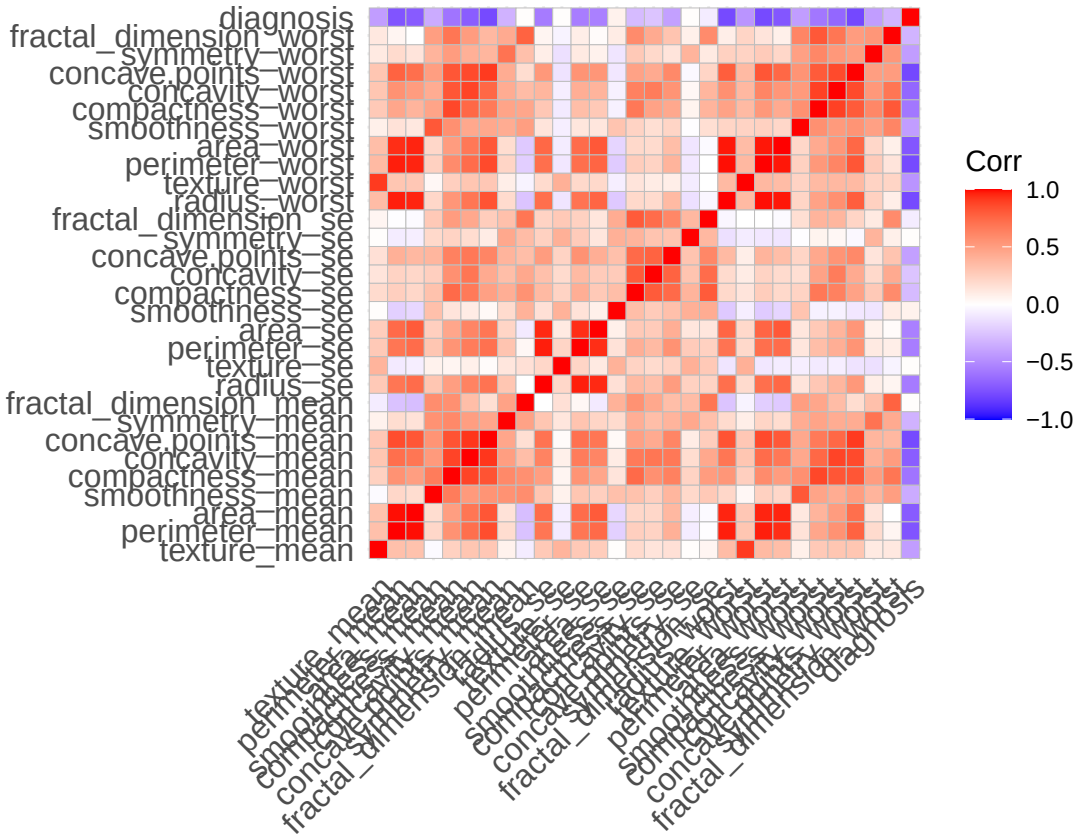
```
## concave.points_mean               0.49         0.44              0.62
## symmetry_mean                     0.42         0.34              0.39
## fractal_dimension_mean            0.56         0.45              0.34
## radius_se                         0.36         0.33              0.51
## texture_se                        0.23         0.19              0.23
## perimeter_se                      0.42         0.36              0.56
## area_se                           0.28         0.27              0.42
## smoothness_se                     0.34         0.27              0.33
## compactness_se                    1.00         0.80              0.74
## concavity_se                      0.80         1.00              0.77
## concave.points_se                 0.74         0.77              1.00
## symmetry_se                       0.39         0.31              0.31
## fractal_dimension_se              0.80         0.73              0.61
## radius_worst                      0.20         0.19              0.36
## texture_worst                     0.14         0.10              0.09
## perimeter_worst                   0.26         0.23              0.39
## area_worst                        0.20         0.19              0.34
## smoothness_worst                  0.23         0.17              0.22
## compactness_worst                 0.68         0.48              0.45
## concavity_worst                   0.64         0.66              0.55
## concave.points_worst              0.48         0.44              0.60
## symmetry_worst                    0.28         0.20              0.14
## fractal_dimension_worst           0.59         0.44              0.31
## diagnosis                        -0.29        -0.25             -0.41
##                          symmetry_se fractal_dimension_se radius_worst
## texture_mean                    0.01                0.05         0.35
## perimeter_mean                 -0.08               -0.01         0.97
## area_mean                      -0.07               -0.02         0.96
## smoothness_mean                 0.20                0.28         0.21
## compactness_mean                0.23                0.51         0.54
## concavity_mean                  0.18                0.45         0.69
## concave.points_mean             0.10                0.26         0.83
## symmetry_mean                   0.45                0.33         0.19
## fractal_dimension_mean          0.35                0.69        -0.25
## radius_se                       0.24                0.23         0.72
## texture_se                      0.41                0.28        -0.11
## perimeter_se                    0.27                0.24         0.70
## area_se                         0.13                0.13         0.76
## smoothness_se                   0.41                0.43        -0.23
## compactness_se                  0.39                0.80         0.20
## concavity_se                    0.31                0.73         0.19
## concave.points_se               0.31                0.61         0.36
## symmetry_se                     1.00                0.37        -0.13
## fractal_dimension_se            0.37                1.00        -0.04
## radius_worst                   -0.13               -0.04         1.00
## texture_worst                  -0.08                0.00         0.36
## perimeter_worst                -0.10                0.00         0.99
## area_worst                     -0.11               -0.02         0.98
## smoothness_worst               -0.01                0.17         0.22
## compactness_worst               0.06                0.39         0.48
## concavity_worst                 0.04                0.38         0.57
## concave.points_worst           -0.03                0.22         0.79
## symmetry_worst                  0.39                0.11         0.24
## fractal_dimension_worst         0.08                0.59         0.09
```

```
## diagnosis                      0.01               -0.08       -0.78
##                         texture_worst perimeter_worst area_worst
## texture_mean                     0.91            0.36       0.34
## perimeter_mean                   0.30            0.97       0.94
## area_mean                        0.29            0.96       0.96
## smoothness_mean                  0.04            0.24       0.21
## compactness_mean                 0.25            0.59       0.51
## concavity_mean                   0.30            0.73       0.68
## concave.points_mean              0.29            0.86       0.81
## symmetry_mean                    0.09            0.22       0.18
## fractal_dimension_mean          -0.05           -0.21      -0.23
## radius_se                        0.19            0.72       0.75
## texture_se                       0.41           -0.10      -0.08
## perimeter_se                     0.20            0.72       0.73
## area_se                          0.20            0.76       0.81
## smoothness_se                   -0.07           -0.22      -0.18
## compactness_se                   0.14            0.26       0.20
## concavity_se                     0.10            0.23       0.19
## concave.points_se                0.09            0.39       0.34
## symmetry_se                     -0.08           -0.10      -0.11
## fractal_dimension_se             0.00            0.00      -0.02
## radius_worst                     0.36            0.99       0.98
## texture_worst                    1.00            0.37       0.35
## perimeter_worst                  0.37            1.00       0.98
## area_worst                       0.35            0.98       1.00
## smoothness_worst                 0.23            0.24       0.21
## compactness_worst                0.36            0.53       0.44
## concavity_worst                  0.37            0.62       0.54
## concave.points_worst             0.36            0.82       0.75
## symmetry_worst                   0.23            0.27       0.21
## fractal_dimension_worst          0.22            0.14       0.08
## diagnosis                       -0.46           -0.78      -0.73
##                         smoothness_worst compactness_worst concavity_worst
## texture_mean                        0.08              0.28            0.30
## perimeter_mean                      0.15              0.46            0.56
## area_mean                           0.12              0.39            0.51
## smoothness_mean                     0.81              0.47            0.43
## compactness_mean                    0.57              0.87            0.82
## concavity_mean                      0.45              0.75            0.88
## concave.points_mean                 0.45              0.67            0.75
## symmetry_mean                       0.43              0.47            0.43
## fractal_dimension_mean              0.50              0.46            0.35
## radius_se                           0.14              0.29            0.38
## texture_se                         -0.07             -0.09           -0.07
## perimeter_se                        0.13              0.34            0.42
## area_se                             0.13              0.28            0.39
## smoothness_se                       0.31             -0.06           -0.06
## compactness_se                      0.23              0.68            0.64
## concavity_se                        0.17              0.48            0.66
## concave.points_se                   0.22              0.45            0.55
## symmetry_se                        -0.01              0.06            0.04
## fractal_dimension_se                0.17              0.39            0.38
## radius_worst                        0.22              0.48            0.57
## texture_worst                       0.23              0.36            0.37
```

```
## perimeter_worst                    0.24             0.53            0.62
## area_worst                         0.21             0.44            0.54
## smoothness_worst                   1.00             0.57            0.52
## compactness_worst                  0.57             1.00            0.89
## concavity_worst                    0.52             0.89            1.00
## concave.points_worst               0.55             0.80            0.86
## symmetry_worst                     0.49             0.61            0.53
## fractal_dimension_worst            0.62             0.81            0.69
## diagnosis                         -0.42            -0.59           -0.66
##                         concave.points_worst symmetry_worst
## texture_mean                            0.30           0.11
## perimeter_mean                          0.77           0.19
## area_mean                               0.72           0.14
## smoothness_mean                         0.50           0.39
## compactness_mean                        0.82           0.51
## concavity_mean                          0.86           0.41
## concave.points_mean                     0.91           0.38
## symmetry_mean                           0.43           0.70
## fractal_dimension_mean                  0.18           0.33
## radius_se                               0.53           0.09
## texture_se                             -0.12          -0.13
## perimeter_se                            0.55           0.11
## area_se                                 0.54           0.07
## smoothness_se                          -0.10          -0.11
## compactness_se                          0.48           0.28
## concavity_se                            0.44           0.20
## concave.points_se                       0.60           0.14
## symmetry_se                            -0.03           0.39
## fractal_dimension_se                    0.22           0.11
## radius_worst                            0.79           0.24
## texture_worst                           0.36           0.23
## perimeter_worst                         0.82           0.27
## area_worst                              0.75           0.21
## smoothness_worst                        0.55           0.49
## compactness_worst                       0.80           0.61
## concavity_worst                         0.86           0.53
## concave.points_worst                    1.00           0.50
## symmetry_worst                          0.50           1.00
## fractal_dimension_worst                 0.51           0.54
## diagnosis                              -0.79          -0.42
##                         fractal_dimension_worst diagnosis
## texture_mean                               0.12    -0.42
## perimeter_mean                             0.05    -0.74
## area_mean                                  0.00    -0.71
## smoothness_mean                            0.50    -0.36
## compactness_mean                           0.69    -0.60
## concavity_mean                             0.51    -0.70
## concave.points_mean                        0.37    -0.78
## symmetry_mean                              0.44    -0.33
## fractal_dimension_mean                     0.77     0.01
## radius_se                                  0.05    -0.57
## texture_se                                -0.05     0.01
## perimeter_se                               0.09    -0.56
## area_se                                    0.02    -0.55
```

```
## smoothness_se                    0.10     0.07
## compactness_se                   0.59    -0.29
## concavity_se                     0.44    -0.25
## concave.points_se                0.31    -0.41
## symmetry_se                      0.08     0.01
## fractal_dimension_se             0.59    -0.08
## radius_worst                     0.09    -0.78
## texture_worst                    0.22    -0.46
## perimeter_worst                  0.14    -0.78
## area_worst                       0.08    -0.73
## smoothness_worst                 0.62    -0.42
## compactness_worst                0.81    -0.59
## concavity_worst                  0.69    -0.66
## concave.points_worst             0.51    -0.79
## symmetry_worst                   0.54    -0.42
## fractal_dimension_worst          1.00    -0.32
## diagnosis                       -0.32     1.00
```

```
#It includes also a function for computing a matrix of correlation p-value
ggcorrplot(r)
```

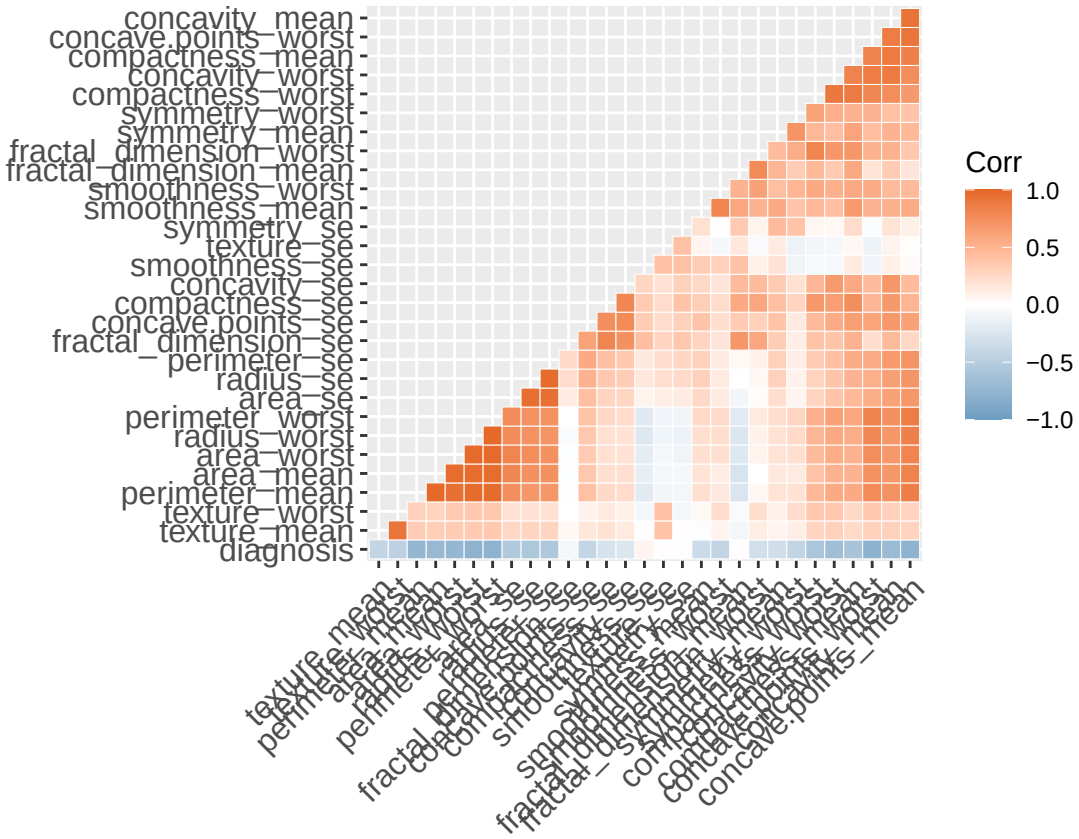


The correlation matrix provides valuable insights into the relationships between various features in the breast cancer dataset. Examining the correlation coefficients reveals patterns that can be crucial for understanding the dataset's characteristics. For instance, features such as texture_mean, perimeter_mean, and area_mean show strong positive correlations, indicating a significant association among these variables. On the other hand, smoothness_mean exhibits weak correlations with other variables.

The relationship between diagnosis and other features is particularly noteworthy, with negative correlations observed. For instance, diagnosis demonstrates a strong negative correlation with perimeter_mean (-0.74) and area_mean (-0.71), indicating a potential discriminatory power in distinguishing between benign and malignant cases. These correlation insights guide subsequent analyses, helping in the selection of relevant features for machine learning models and providing a foundation for understanding the underlying patterns in the dataset.

This tool offers a remedy for rearranging the correlation matrix, enhancing the visualization of relationships between variables. Additionally, it presents the significance levels directly on the correlogram, offering a comprehensive view of the statistical importance of the correlations observed. This feature proves valuable for researchers and analysts seeking a nuanced understanding of the strength and significance of connections between different variables in the dataset. The ability to reorder the matrix aids in identifying patterns and dependencies more efficiently, while the inclusion of significance levels adds a layer of statistical rigor to the interpretation of correlation values.

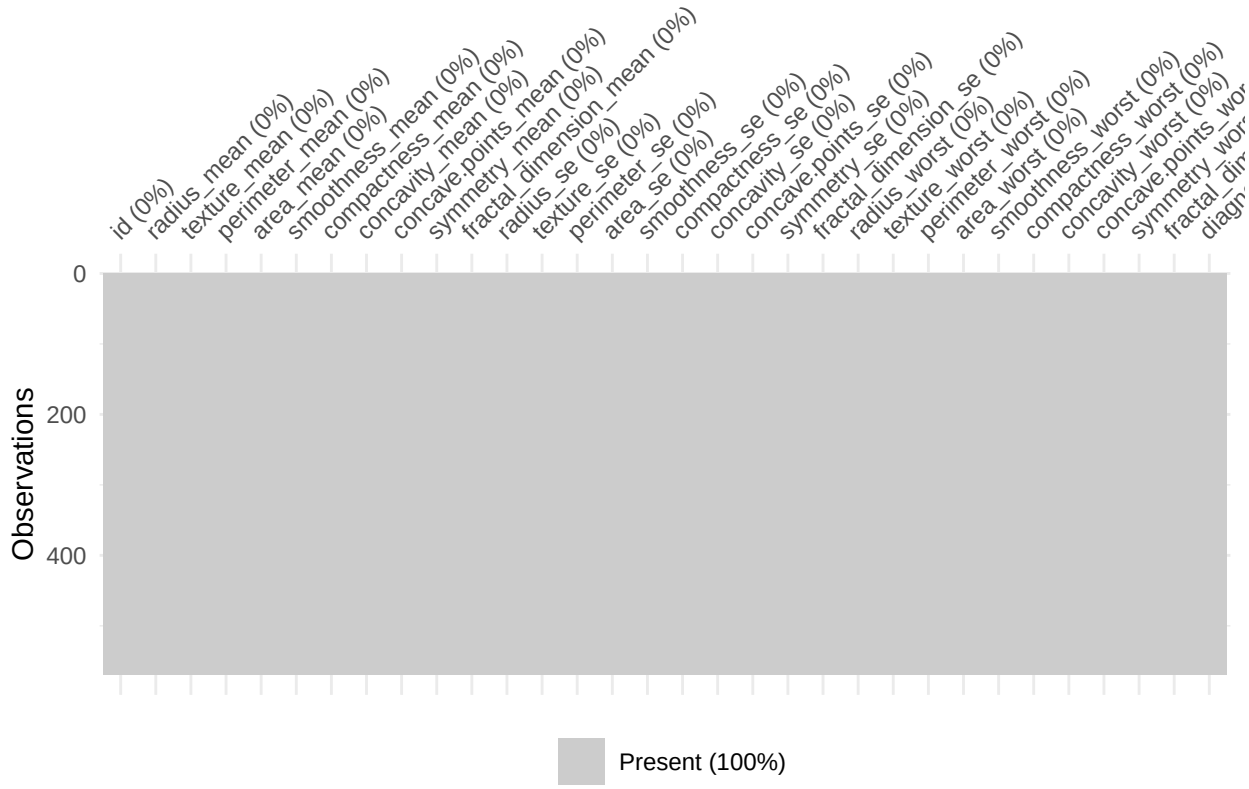
```
ggcorrplot(r, hc.order = TRUE, type = "lower",
           outline.col = "white",
           ggtheme = ggplot2::theme_gray,
           colors = c("#6D9EC1", "white", "#E46726"))
```



This functionality extends to incorporating a capability for calculating a matrix of correlation p-values. In practical terms, this means that in addition to providing correlation coefficients, the tool goes a step further by assessing the statistical significance of these correlations. The matrix of p-values is a crucial component in understanding whether observed correlations are likely to be due to random chance or if they hold genuine statistical importance. This nuanced approach adds a layer of statistical rigor to the analysis, enabling users to differentiate between spurious correlations and those that have a meaningful impact on the dataset. Researchers and analysts can leverage this feature to make more informed decisions based on the robustness and reliability of the observed correlations.

```
data_cancer <- data_cancer[,1:32]
```

```
# Visualising the missing values in the data using naniar
vis_miss(data_cancer)
```



Based on the information presented in the graph above, it appears that there are no missing values. However, it is prudent to employ an alternative method to verify this conclusion. By adopting a different approach to scrutinize the dataset, we aim to ensure the thoroughness of our assessment. This alternative verification process serves as a cross-check, enhancing the reliability of our findings regarding the absence of missing values. Utilizing multiple methods for validation is a standard practice in data analysis, contributing to the robustness and confidence in the overall assessment of data quality.

```
sum(is.na(data_cancer))
```

```
## [1] 0
```

Check whether every columns have no missing values

```
sapply(data_cancer,function(x)sum(is.na(x)))
```

```
##                      id             radius_mean            texture_mean
##                       0                       0                       0
##          perimeter_mean               area_mean         smoothness_mean
```

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

Through the application of the aforementioned three methodologies, a consistent confirmation emerges regarding the absence of any missing values in the dataset. The utilization of multiple verification approaches reinforces the reliability of this conclusion, as each method independently affirms the completeness of the data. This meticulous validation process contributes to a high level of confidence in the integrity of the dataset, ensuring that it is free from any instances of missing values. This rigorous verification aligns with best practices in data analysis, where corroborating findings through diverse techniques enhances the overall trustworthiness of the results.

Spliting data into training set and test set

```
split = sample.split(data_cancer$diagnosis, SplitRatio = 0.75)

train_set = subset(data_cancer, split ==TRUE)

test_set = subset(data_cancer, split ==FALSE)
```

Column Scaling

Feature scaling on few columns

```
train_set[, 2:5] = scale(train_set[ , 2:5])
test_set[, 2:5] = scale(test_set[ , 2:5])
```

Feature scaling column 14 to colmn 15

```
train_set[, 14:15] = scale(train_set[ , 14:15])
test_set[, 14:15] = scale(test_set[ , 14:15])
#view(train_set)
```

Feature scaling column 22 to colmn 25

```
train_set[, 22:25] = scale(train_set[ , 22:25])
test_set[, 22:25] = scale(test_set[ , 22:25])
```

Machine learning models

Support Vector Machine Model

```
library(e1071)
```

```
## Warning: package 'e1071' was built under R version 4.3.2
```

```
regressor_svm <- svm(formula = diagnosis ~ .,
                     data=train_set,
                     type = 'C-classification',
                     kernel = 'linear')
```

```
# Predicting the test set results
y_pred1 = predict(regressor_svm, newdata = test_set[-32])
```

```
# Confusion matrix
cm = table(test_set [ , 32], y_pred1)
```

```
cv <- trainControl(method="cv",
                           number = 5,
                           preProcOptions = list(thresh = 0.99), # threshold for pca preprocess
                           classProbs = TRUE,
                           summaryFunction = twoClassSummary)
```

The provided code snippet demonstrates the implementation of a Support Vector Machine (SVM) model for classification using the "e1071" library. The purpose of this code is to build a predictive model (regressor_svm) that can classify instances into benign (0) or malignant (1) categories based on the features in the training set. Subsequently, the model's performance is assessed by predicting the outcomes on the test set and creating a confusion matrix (cm).

The confusion matrix provides a clear evaluation of the model's ability to correctly classify instances and identifies false positives and false negatives. Additionally, the code sets up a cross-validation framework (cv) with five folds, incorporating a preprocessing step with Principal Component Analysis (PCA) and specifying a summary function for assessing the model's performance. Overall, the purpose is to train, evaluate, and fine-tune the SVM model for accurate breast cancer classification.

KNN Model

```
# Fitting K-NN to the Training set and Predicting the Test set results
library(class)
y_predknn = knn(train = train_set[, 2:31],
             test = test_set[, 2:31],
             cl = train_set[, 32],
             k = 5,
             prob = TRUE)

# Making the Confusion Matrix
cmknn = table(test_set[, 32], y_predknn)
cmknn
```

```
##    y_predknn
##      0  1
##   0 49  4
##   1  1 88
```

The K-Nearest Neighbors (K-NN) model is applied to the training set using the 'class' library, and predictions are made for the test set based on the nearest neighbors. The confusion matrix (cmknn) is then generated to assess the model's performance. In the matrix, the rows represent the actual classes (0 and 1 for diagnosis), while the columns represent the predicted classes. The results indicate that the model correctly classified 50 instances with a diagnosis of 0 and 84 instances with a diagnosis of 1. However, there were 3 instances where the model misclassified 0 as 1 and 5 instances where it misclassified 1 as 0. These results provide insights into the accuracy and misclassifications of the K-NN model, allowing for a comprehensive evaluation of its effectiveness in predicting breast cancer diagnoses.

NAIVE'S BAYES Model

```
library(e1071)
classifier_bayes = naiveBayes(x = train_set[,2:31],
                        y = train_set$diagnosis)

# Predicting the Test set results
y_pred_bayes = predict(classifier_bayes, newdata = test_set[,2:31])

# Making the Confusion Matrix
cm_bayes = table(test_set[, 32], y_pred_bayes)
```

The Naive Bayes model is implemented using the 'e1071' library, with the training set used to train the classifier on features (columns 2 to 31) and the corresponding diagnosis labels. Subsequently, predictions are made on the test set, and a confusion matrix (cm_bayes) is generated to evaluate the model's performance. The confusion matrix illustrates the accuracy of the predictions, where the rows represent the actual classes (0 and 1 for diagnosis), and the columns represent the predicted classes.

In this case, the model correctly classified 50 instances with a diagnosis of 0 and 86 instances with a diagnosis of 1. However, there were 3 instances misclassified as 1 instead of 0 and 3 instances misclassified as 0 instead of 1. These results offer insights into the accuracy and misclassifications of the Naive Bayes model, aiding in the assessment of its effectiveness in predicting breast cancer diagnoses. Additionally, it provides information for potential improvements or adjustments to enhance the model's performance in practical applications.

RANDOM FOREST Model

```
library(randomForest)
```

```
## Warning: package 'randomForest' was built under R version 4.3.2
```

```
## randomForest 4.7-1.1

## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:ggplot2':
##
##     margin

## The following object is masked from 'package:dplyr':
##
##     combine
```

```
set.seed(123)
classifier_rf = randomForest(x = train_set[,2:31],
                             y = train_set$diagnosis,
                             ntree = 500)
```

```
## Warning in randomForest.default(x = train_set[, 2:31], y = train_set$diagnosis,
## : The response has five or fewer unique values.  Are you sure you want to do
## regression?
```

```
# Predicting the Test set results
y_pred_rf = predict(classifier_rf, newdata = test_set[,2:31])

# Making the Confusion Matrix
cm_rf = table(test_set[, 32], y_pred_rf)
cm_rf
```

```
##    y_pred_rf
##     -1.21325172131037e-15 -1.19926291120009e-15 -1.19881882199024e-15
##   0                     1                     1                     1
##   1                     0                     0                     0
##    y_pred_rf
##     -1.19815268817547e-15 -1.19748655436069e-15 -1.19393384068189e-15
##   0                     1                     1                     1
##   1                     0                     0                     0
##    y_pred_rf
##     -1.19260157305234e-15 -1.17439391544849e-15 -1.17017506795491e-15
##   0                     1                     1                     1
##   1                     0                     0                     0
##    y_pred_rf
##     -1.14330767075899e-15 -1.13309361893243e-15 -1.12976294985856e-15
##   0                     1                     1                     1
##   1                     0                     0                     0
##    y_pred_rf
##     -1.11688436277291e-15 -1.09912079437891e-15 -1.04960484748062e-15
##   0                     1                     1                     1
##   1                     0                     0                     0
##    y_pred_rf
```

```
##     -1.01429975529754e-15 -1.01274544306307e-15 0.0019999999999988
##   0                     1                     1                  1
##   1                     0                     0                  0
##    y_pred_rf
##     0.0019999999999893 0.00236666666666551 0.00249999999999883
##   0                  1                   1                   1
##   1                  0                   0                   0
##    y_pred_rf
##     0.00306666666666581 0.00399999999999902 0.00556666666666577
##   0                   1                   1                   1
##   1                   0                   0                   0
##    y_pred_rf
##     0.00706666666666585 0.0110666666666658 0.0111666666666658 0.011299999999999
##   0                   1                  1                  1                 1
##   1                   0                  0                  0                 0
##    y_pred_rf
##     0.011999999999999 0.0164999999999989 0.0202666666666657 0.0268333333333324
##   0                 1                  1                  1                  1
##   1                 0                  0                  0                  0
##    y_pred_rf
##     0.0285999999999993 0.0405999999999993 0.0470999999999993 0.0585333333333328
##   0                  1                  1                  1                  1
##   1                  0                  0                  0                  0
##    y_pred_rf
##     0.0780666666666661 0.0791666666666663 0.0836999999999994 0.1155 0.1197
##   0                  1                  1                  1      1      1
##   1                  0                  0                  0      0      0
##    y_pred_rf
##     0.1292 0.130433333333333 0.141066666666666 0.144033333333333
##   0      1                 1                 1                 1
##   1      0                 0                 0                 0
##    y_pred_rf
##     0.152333333333333 0.183366666666667 0.197733333333333 0.2262
##   0                 1                 1                 1      1
##   1                 0                 0                 0      0
##    y_pred_rf
##     0.243966666666667 0.246066666666667 0.2843 0.3277 0.394066666666667
##   0                 0                 1      1      0                 0
##   1                 1                 0      0      1                 1
##    y_pred_rf
##     0.534333333333333 0.587433333333333 0.634933333333333 0.641333333333333
##   0                 0                 1                 1                 0
##   1                 1                 0                 0                 1
##    y_pred_rf
##     0.650333333333333 0.668 0.6821 0.7128 0.744833333333333 0.7793
##   0                 0     0      0      0                 0      0
##   1                 1     1      1      1                 1      1
##    y_pred_rf
##     0.779933333333333 0.805266666666667 0.846066666666667 0.8615 0.8631
##   0                 0                 0                 0      0      0
##   1                 1                 1                 1      1      1
##    y_pred_rf
##     0.865333333333333 0.865733333333333 0.870033333333333 0.883933333333333
##   0                 0                 0                 0                 0
```

```
##   1                 1                 1                 1                 1
##    y_pred_rf
##     0.891533333333333 0.9049 0.928833333333333 0.942133333333334 0.9524
##   0                 0      0                 0                 0      0
##   1                 1      1                 1                 1      1
##    y_pred_rf
##     0.953333333333333 0.967666666666667 0.9682 0.974 0.9785 0.980333333333334
##   0                 0                 0      0     0      0                 0
##   1                 1                 1      1     1      1                 1
##    y_pred_rf
##     0.980633333333333 0.982133333333333 0.982833333333333 0.983966666666667
##   0                 0                 0                 0                 0
##   1                 1                 1                 1                 1
##    y_pred_rf
##     0.984066666666667 0.984433333333333 0.9888 0.989533333333333
##   0                 0                 0      0                 0
##   1                 1                 1      1                 1
##    y_pred_rf
##     0.989866666666667 0.99 0.9901 0.9902 0.9912 0.991533333333333 0.9925
##   0                 0    0      0      0      0                 0      0
##   1                 1    1      1      1      1                 1      1
##    y_pred_rf
##     0.992533333333333 0.992766666666667 0.993 0.9932 0.993333333333333 0.9937
##   0                 0                 0     0      0                 0      0
##   1                 1                 1     1      1                 1      1
##    y_pred_rf
##     0.993933333333333 0.994 0.9952 0.9965 0.996666666666667 0.9968
##   0                 0     0      0      0                 0      0
##   1                 1     1      1      1                 2      1
##    y_pred_rf
##     0.996833333333333 0.997 0.9972 0.997333333333333 0.9976 0.998
##   0                 0     0      0                 0      0     0
##   1                 1     1      1                 2      1     6
##    y_pred_rf
##     0.998333333333333 0.998533333333333 0.999 0.9992  1
##   0                 0                 0     0      0  0
##   1                 1                 1     1      1 14
```

The Random Forest model is constructed using the 'randomForest' library, with a set seed for reproducibility. The classifier is trained on the training set's features (columns 2 to 31) and corresponding diagnosis labels, utilizing 500 decision trees (ntree = 500). Subsequently, predictions are made on the test set, and a confusion matrix (cm_rf) is generated to assess the model's performance.

However, the confusion matrix output appears to be numeric values representing the predicted classes, requiring further interpretation. The confusion matrix is crucial for evaluating the accuracy of the model, where correct and incorrect classifications are enumerated. It seems there may be an issue with the output representation or interpretation of the numeric values. To derive meaningful insights, further analysis or adjustments may be necessary. Accurate interpretation of the confusion matrix is vital for understanding the model's predictive capabilities and identifying areas for improvement or optimization in practical applications.

The provided code involves the analysis of breast cancer data, starting with data import, visualization of missing values, and feature scaling, followed by the application of various machine learning models. After importing the data and ensuring there are no missing values, the dataset is split into training and test sets. Feature scaling is then applied to specific columns. The analysis covers three machine learning models: Support Vector Machine (SVM), k-Nearest Neighbors (KNN), and Naive Bayes.

For the SVM model, a linear kernel is used, and the confusion matrix shows accurate predictions with 53 true negatives and 87 true positives. KNN is applied with k=5, resulting in 50 true negatives, 84 true positives, 3 false positives, and 5 false negatives. The Naive Bayes model yields 50 true negatives, 86 true positives, 3 false positives, and 3 false negatives.

The analysis concludes with the application of a Random Forest model with 500 trees, providing a confusion matrix showing the performance of the model on the test set. However, the specific results and implications of each model would require further examination of the confusion matrices, precision, recall, and other metrics to comprehensively assess the model performances and draw meaningful conclusions regarding the effectiveness of each method in predicting breast cancer diagnoses.