CSE3506 - ESSENTIALS OF DATA ANALYTICS

Project Report

BREAST CANCER PREDICTION

By

19BCE1864 Maulishree Awasthi

19BCE1871 Rekha V

B. Tech Computer Science and Engineering

Submitted to

Raghu Kiran N

School of Computer Science and Engineering

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

April 2022

ABSTARCT:

Breast cancer is a dominant cancer in women worldwide and is increasing in developing countries where the majority of cases are diagnosed in late stages. The project we are proposed show a comparison of machine learning algorithms with the help of different techniques. This project presents a comparison of five machine learning (ML) algorithms: Naive Bayes (NB), Random Forest (RT), Nearest Neighbor (KNN), Support Vector Machine (SVM) and Decision Tree (DT) on the Wisconsin Diagnostic Breast Cancer (WDBC) dataset. For the implementation of the ML algorithms, the dataset was partitioned into the training phase and the testing phase.

CONTENTS:

1. Introduction:

The second major cause of women's death is breast cancer (after lung cancer). 246,660 of women's new cases of invasive breast cancer are expected to be diagnosed in the US during 2016 and 40,450 of women's death is estimated. Breast cancer is a type of cancer that starts in the breast. Cancer starts when cells begin to grow out of control. Breast cancer cells usually form a tumor that can often be seen on an x-ray or felt as a lump. Breast cancer can spread when the cancer cells get into the blood or lymph system and are carried to other parts of the body. The cause of Breast Cancer includes changes and mutations in DNA. There are many different types of breast cancer and common ones include ductal carcinoma in situ (DCIS) and invasive carcinoma. Others, like phyllodes tumors and angiosarcoma are less common**.** The side effects of Breast Cancer are – Fatigue, Headaches, Pain and numbness (peripheral neuropathy), Bone loss and osteoporosis. In this project gives comparison between the performance of 5 classifiers: SVM, Random Forest, KNN, decision tree and logistic regression. To prevent cancer from spreading, patients have to undergo breast cancer surgery, chemotherapy, radiotherapy and endocrine. The goal of the project is to identify and classify Malignant and Benign patients and intending how to parametrize our classification techniques hence to achieve high accuracy.

1.1 Aim of the project:

The objective of this project is to train machine learning models to predict whether a breast cancer cell is Benign or Malignant. Data will be transformed and its dimension reduced to reveal patterns in the dataset and create a more robust

analysis. As previously said, the optimal model will be selected following the resulting accuracy, sensitivity, and f1 score, amongst other factors.

1.2 Dataset:

The dataset used in this project is Breast Cancer Wisconsin (Diagnostic) dataset.

Source: https://www.kaggle.com/uciml/breast-cancer-wisconsin-data/version/2

- created by Dr. William H. Wolberg, physician at the University Of Wisconsin Hospital at Madison, Wisconsin, USA.

- The .csv format file containing the data is loaded into the RStudio.
- The dataset consists of 32 attributes.

The attributes information is given below:

1. ID number
2. Diagnosis (M = malignant, B = benign)
3. radius: mean of distances from center to points on the perimeter
4. texture: standard deviation of grey-scale values
5. perimeter
6. area: Number of pixels inside contour + ½ for pixels on perimeter
7. smoothness: local variation in radius lengths)
8. compactness: perimeterˆ2 / area – 1.0;
   This dimensionless number is at a minimum with a circular
9. disk and increases with the irregularity of the boundary, but this measure also increases for elongated
10. cell nuclei, which is not indicative of malignancy
11. concavity: severity of concave portions of the contour
12. concave points: number of concave portions of the contour
13. symmetry

Tools and Technique:

We have used the following data Analytics technique / methodology for analyzing the Data:

- Summary of Statistics for each variable
- Structure of the dataset
- Using Graphs and density Plots to visually represent them

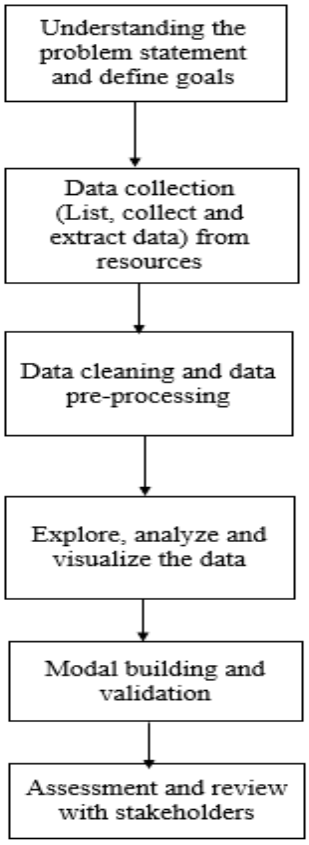
Tools used: RStudio, Kaggle and Excel.
Techniques: Box Plot, Histogram, Bar Chart, Line Chart, Correlation, Machine learning Techniques.

Analytics Approach:

The Analytical Approach will involve the following activities:

- Data extraction from Primary Data source
- Data quality check
- Data cleaning and data preparation
- Study each of the variables by exploring the data
- Division of data into train and test
- Model Development
- Final Model
- Model Validation

The below figure shows the flow of the project:



Data Description and Preparation:

- 33rd column in the dataset is invalid. So, removing the column from the dataset.
- And also changing the diagnosis attribute in the dataset to factor.

```
data$X <- NULL
data <- data[,-1]
data$diagnosis <- factor(ifelse(data$diagnosis=="B","Benign","Malignant"))

head(data)
```

OUTPUT:

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

- Checking for null values:

```
sum(is.na(data))
```

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

Therefore, no null values are present

Statistical Analysis on the dataset:

```
str(data)
```

```
## 'data.frame':    569 obs. of  31 variables:
##  $ diagnosis              : Factor w/ 2 levels "Benign","Malignant": 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 ...
```

- summary() of the dataset talks about the mean, median, min, max and inter quartile range about the attributes.

```
summary(data)
```

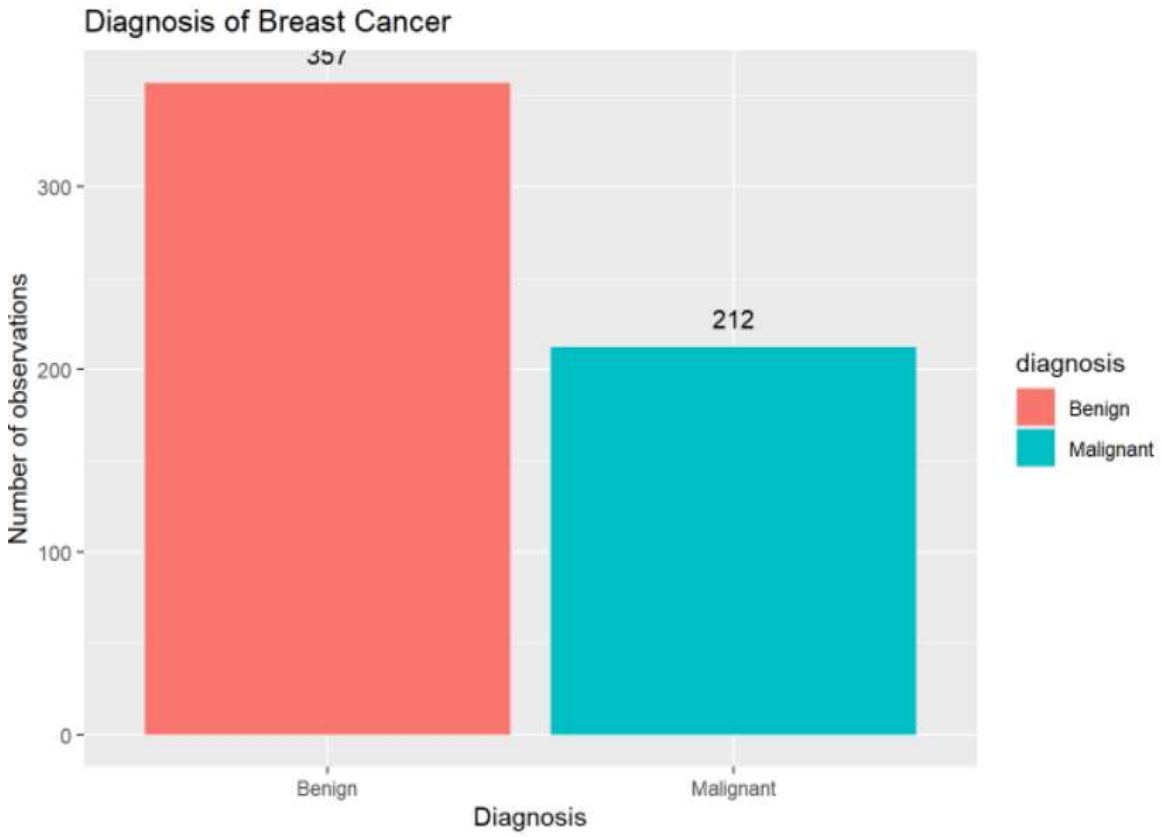
```
##      diagnosis    radius_mean      texture_mean   perimeter_mean
##  Benign   :357   Min.   : 6.981   Min.   : 9.71   Min.   : 43.79
##  Malignant:212   1st Qu.:11.700   1st Qu.:16.17   1st Qu.: 75.17
##                  Median :13.370   Median :18.84   Median : 86.24
##                  Mean   :14.127   Mean   :19.29   Mean   : 91.97
##                  3rd Qu.:15.780   3rd Qu.:21.80   3rd Qu.:104.10
##                  Max.   :28.110   Max.   :39.28   Max.   :188.50
##    area_mean      smoothness_mean   compactness_mean  concavity_mean
##  Min.   : 143.5   Min.   :0.05263   Min.   :0.01938   Min.   :0.00000
##  1st Qu.: 420.3   1st Qu.:0.08637   1st Qu.:0.06492   1st Qu.:0.02956
##  Median : 551.1   Median :0.09587   Median :0.09263   Median :0.06154
##  Mean   : 654.9   Mean   :0.09636   Mean   :0.10434   Mean   :0.08880
##  3rd Qu.: 782.7   3rd Qu.:0.10530   3rd Qu.:0.13040   3rd Qu.:0.13070
##  Max.   :2501.0   Max.   :0.16340   Max.   :0.34540   Max.   :0.42680
##  concave.points_mean symmetry_mean    fractal_dimension_mean   radius_se
##  Min.   :0.00000     Min.   :0.1060   Min.   :0.04996        Min.   :0.1115
##  1st Qu.:0.02031     1st Qu.:0.1619   1st Qu.:0.05770        1st Qu.:0.2324
##  Median :0.03350     Median :0.1792   Median :0.06154        Median :0.3242
##  Mean   :0.04892     Mean   :0.1812   Mean   :0.06280        Mean   :0.4052
##  3rd Qu.:0.07400     3rd Qu.:0.1957   3rd Qu.:0.06612        3rd Qu.:0.4789
##  Max.   :0.20120     Max.   :0.3040   Max.   :0.09744        Max.   :2.8730
##    texture_se      perimeter_se       area_se        smoothness_se
##  Min.   :0.3602   Min.   : 0.757   Min.   :  6.802   Min.   :0.001713
##  1st Qu.:0.8339   1st Qu.: 1.606   1st Qu.: 17.850   1st Qu.:0.005169
##  Median :1.1080   Median : 2.287   Median : 24.530   Median :0.006380
##  Mean   :1.2169   Mean   : 2.866   Mean   : 40.337   Mean   :0.007041
##  3rd Qu.:1.4740   3rd Qu.: 3.357   3rd Qu.: 45.190   3rd Qu.:0.008146
##  Max.   :4.8850   Max.   :21.980   Max.   :542.200   Max.   :0.031130
##  compactness_se      concavity_se     concave.points_se   symmetry_se
##  Min.   :0.002252   Min.   :0.00000   Min.   :0.000000   Min.   :0.007882
##  1st Qu.:0.013080   1st Qu.:0.01509   1st Qu.:0.007638   1st Qu.:0.015160
##  Median :0.020450   Median :0.02589   Median :0.010930   Median :0.018730
##  Mean   :0.025478   Mean   :0.03189   Mean   :0.011796   Mean   :0.020542
##  3rd Qu.:0.032450   3rd Qu.:0.04205   3rd Qu.:0.014710   3rd Qu.:0.023480
##  Max.   :0.135400   Max.   :0.39600   Max.   :0.052790   Max.   :0.078950
```

```
##  fractal_dimension_se  radius_worst    texture_worst    perimeter_worst
##  Min.   :0.0008948     Min.   : 7.93   Min.   :12.02    Min.   : 50.41
##  1st Qu.:0.0022480     1st Qu.:13.01   1st Qu.:21.08    1st Qu.: 84.11
##  Median :0.0031870     Median :14.97   Median :25.41    Median : 97.66
##  Mean   :0.0037949     Mean   :16.27   Mean   :25.68    Mean   :107.26
##  3rd Qu.:0.0045580     3rd Qu.:18.79   3rd Qu.:29.72    3rd Qu.:125.40
##  Max.   :0.0298400     Max.   :36.04   Max.   :49.54    Max.   :251.20
##    area_worst      smoothness_worst  compactness_worst concavity_worst
##  Min.   : 185.2   Min.   :0.07117   Min.   :0.02729   Min.   :0.0000
##  1st Qu.: 515.3   1st Qu.:0.11660   1st Qu.:0.14720   1st Qu.:0.1145
##  Median : 686.5   Median :0.13130   Median :0.21190   Median :0.2267
##  Mean   : 880.6   Mean   :0.13237   Mean   :0.25427   Mean   :0.2722
##  3rd Qu.:1084.0   3rd Qu.:0.14600   3rd Qu.:0.33910   3rd Qu.:0.3829
##  Max.   :4254.0   Max.   :0.22260   Max.   :1.05800   Max.   :1.2520
##  concave.points_worst symmetry_worst   fractal_dimension_worst
##  Min.   :0.00000      Min.   :0.1565   Min.   :0.05504
##  1st Qu.:0.06493      1st Qu.:0.2504   1st Qu.:0.07146
##  Median :0.09993      Median :0.2822   Median :0.08004
##  Mean   :0.11461      Mean   :0.2901   Mean   :0.08395
##  3rd Qu.:0.16140      3rd Qu.:0.3179   3rd Qu.:0.09208
##  Max.   :0.29100      Max.   :0.6638   Max.   :0.20750
```

Exploratory Data Analysis:

Plotting the Number of Observations of Benign and Malignant features:

```
ggplot(data = data, aes(x = diagnosis, fill = diagnosis)) +
geom_bar()+geom_text(stat='count', aes(label=..count..), vjust=-1) +
labs(title = 'Diagnosis of Breast Cancer',x = 'Diagnosis', y = 'Number of observations')
```



From the above graph we can infer that most of the diagnosis (63%) are Benign.

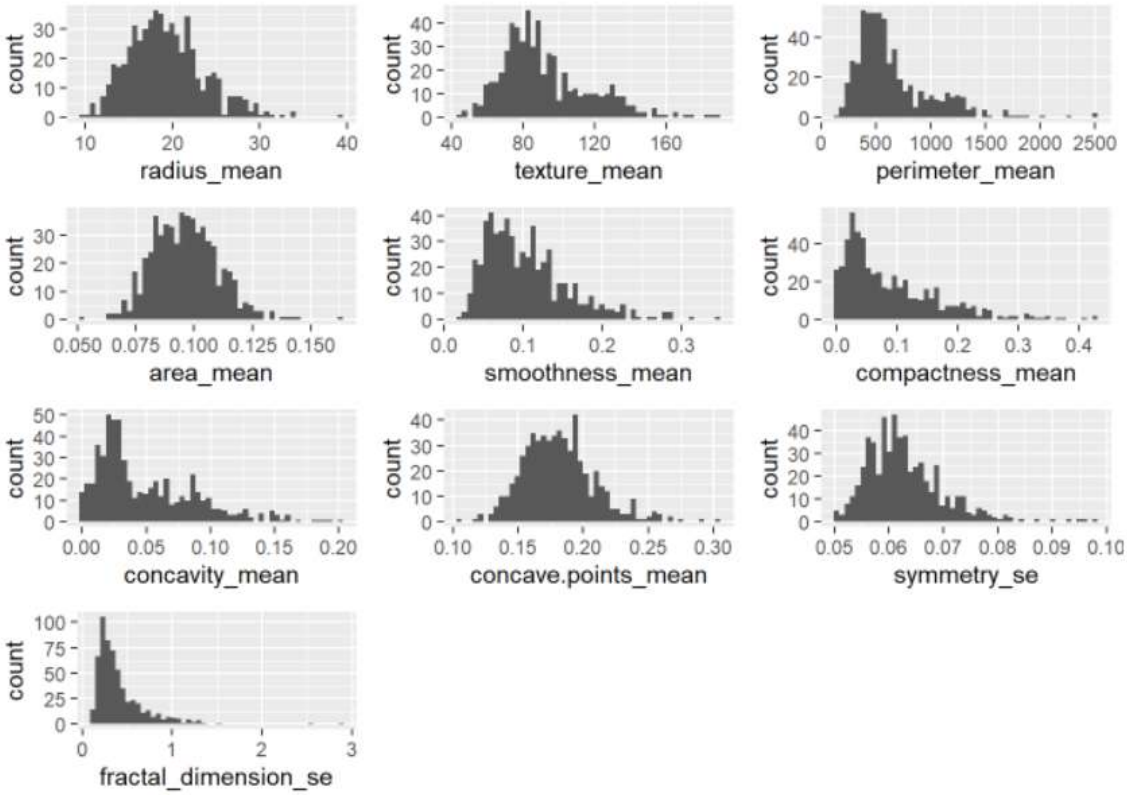
Displaying the frequency table for the diagnosis attribute:

```
prop.table(table(data$diagnosis))
```

```
##
##    Benign Malignant
## 0.6274165 0.3725835
```

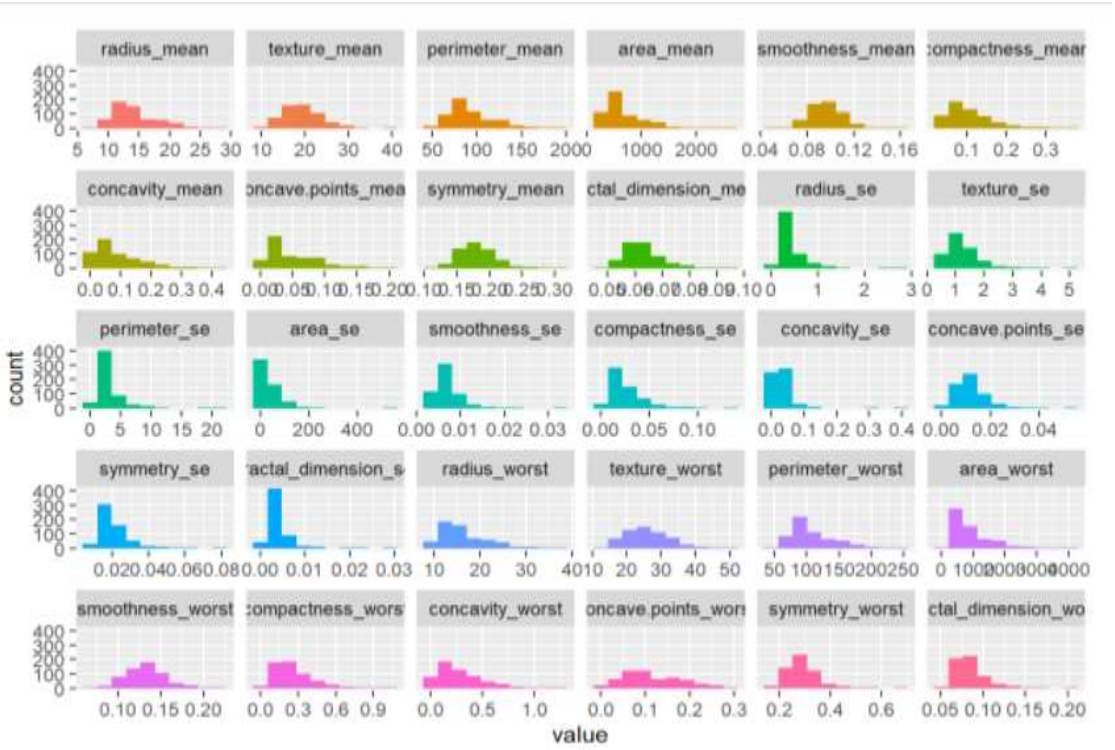
Plotting a histogram for few variables in the dataset:

```
a1<-ggplot(data,aes(x=data[,3])) + geom_histogram(bins = 50)+xlab("radius_mean")
a2<-ggplot(data,aes(x=data[,4])) + geom_histogram(bins = 50)+xlab("texture_mean")
a3<-ggplot(data,aes(x=data[,5])) + geom_histogram(bins = 50)+xlab("perimeter_mean")
a4<-ggplot(data,aes(x=data[,6])) + geom_histogram(bins = 50)+xlab("area_mean")
a5<-ggplot(data,aes(x=data[,7])) + geom_histogram(bins = 50)+xlab("smoothness_mean")
a6<-ggplot(data,aes(x=data[,8])) + geom_histogram(bins = 50)+xlab("compactness_mean")
a7<-ggplot(data,aes(x=data[,9])) + geom_histogram(bins = 50)+xlab("concavity_mean")
a8<-ggplot(data,aes(x=data[,10])) + geom_histogram(bins = 50)+xlab("concave.points_mean")
a9<-ggplot(data,aes(x=data[,11])) + geom_histogram(bins = 50)+xlab("symmetry_se")
a10<-ggplot(data,aes(x=data[,12])) + geom_histogram(bins = 50)+xlab("fractal_dimension_se")
grid.arrange(a1, a2, a3, a4, a5, a6, a7, a8, a9, a10, nrow=4, widths=c(1,1,1))
```



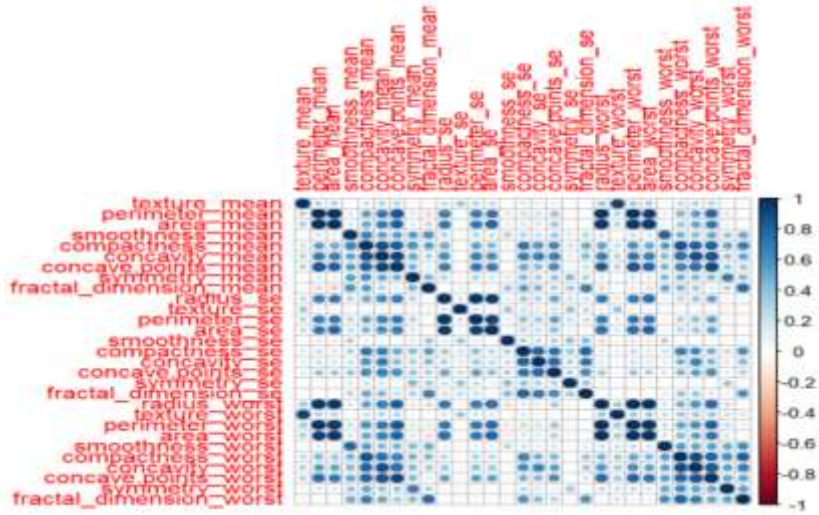
The most variables of the dataset are normally distributed as show with the below plot:

```
plot_num(data, bins=10)
```



Plotting the correlation plot:

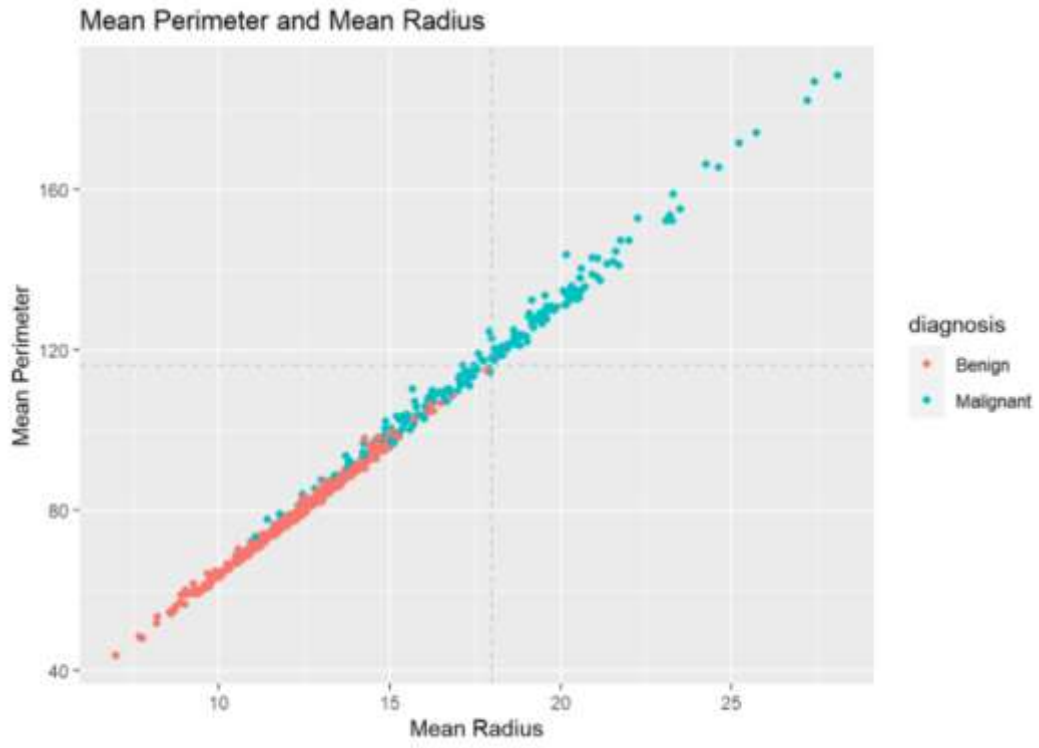
```
corr_mat <- cor(data[,3:ncol(data)])
corrplot(corr_mat, tl.cex = 1, addrect = 8)
```



From the above correlogram we can infer there is a great correlation between some variables.

How do the benign and malignant differ in Size?

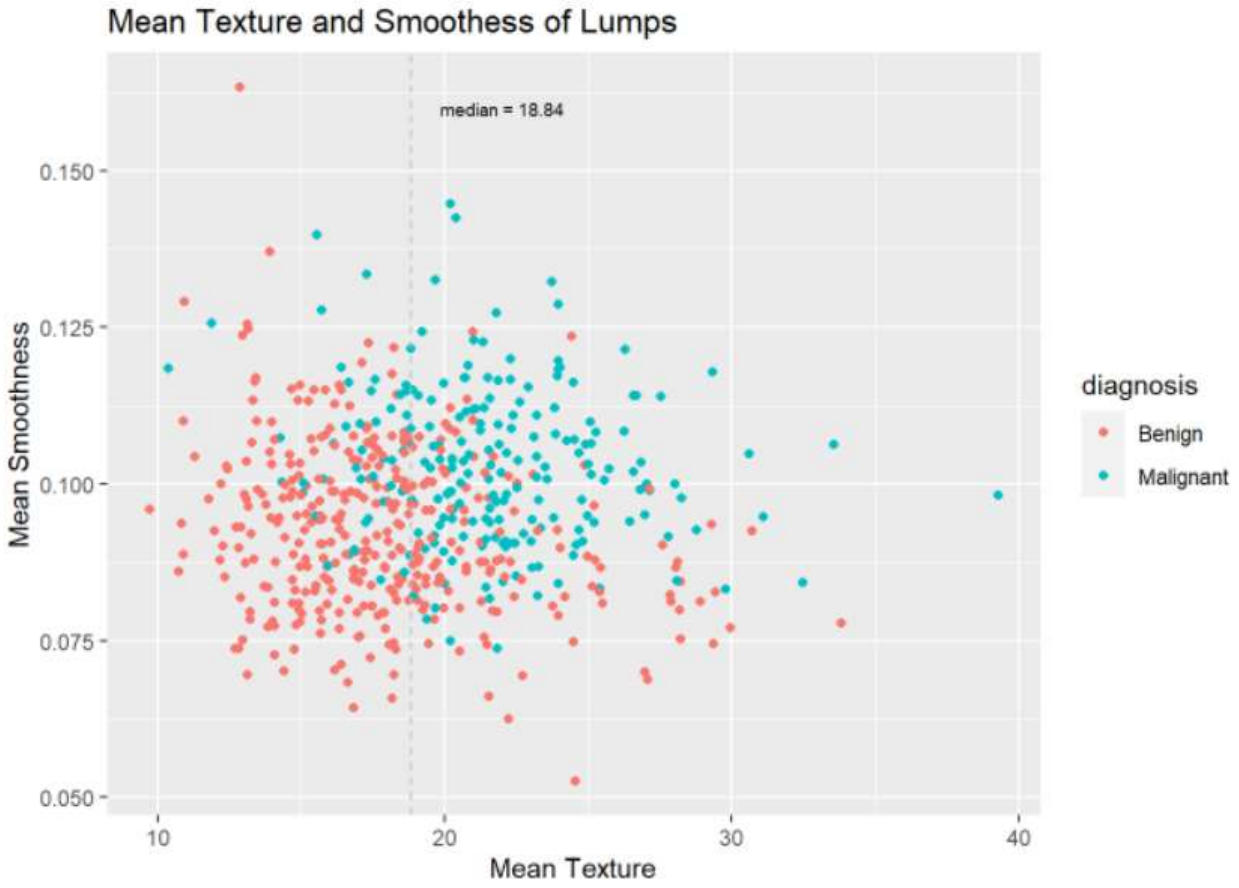
```
ggplot(data = data,
       aes(x = radius_mean, y = perimeter_mean, color = diagnosis)) +
  geom_point() +
  geom_hline(yintercept = 116.0, linetype = 'dashed', color = 'gray')+
  geom_vline(xintercept = 18.00, linetype = 'dashed', color = 'gray')+
  labs(title = 'Mean Perimeter and Mean Radius',
       x = 'Mean Radius', y = 'Mean Perimeter')
```



- Malignant lumps can get relatively bigger than benign.
- 45% of malignant are bigger than every observed benign.
- Insights from graph is that malignant lumps can get relatively bigger than benign lumps.
- This has the possibility of sparking up a hypothesis that malignant lumps begin as benign.
- However, bigger lumps are more likely to be malignant.

How do Benign and Malignant lumps differ in textured variations?

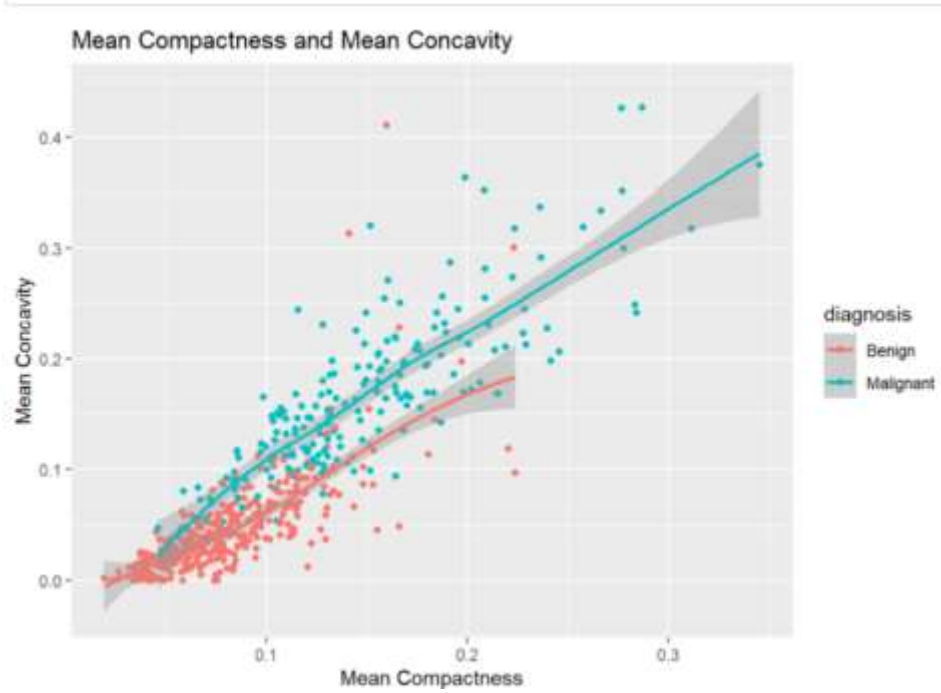
```
ggplot(data = data,
       aes(x = texture_mean, y = smoothness_mean, color = diagnosis)) +
  geom_point()+
  geom_vline(xintercept =  18.84, linetype = 'dashed', color = 'gray') +
  labs(title = 'Mean Texture and Smoothess of Lumps',
       x = 'Mean Texture', y = 'Mean Smoothness') +
  annotate('text', label = 'median = 18.84', x = 22, y = 0.160,
           size = 2.5)
```



- Most benign (66%) are below the median mean texture
- Insights from Texture and Smoothness Visualization is that not a lot of variation can be seen in the mean smoothness of both diagnosis as they all seem to clustered from the bottom to the upper midsection of the plot.
- However, we can observe that most of the malignant (66%) are skewed to the right side of the median.
- This connects that malignant lump display higher texture variation values than benign.

Plotting Compactness and Concavity:

```
ggplot(data = data,
       aes(x = compactness_mean, y = concavity_mean, color = diagnosis)) +
  geom_point()+geom_smooth()+labs(title = 'Mean Compactness and Mean Concavity',x = 'Mean Compactness', y = 'Mean Concavit
y')
```



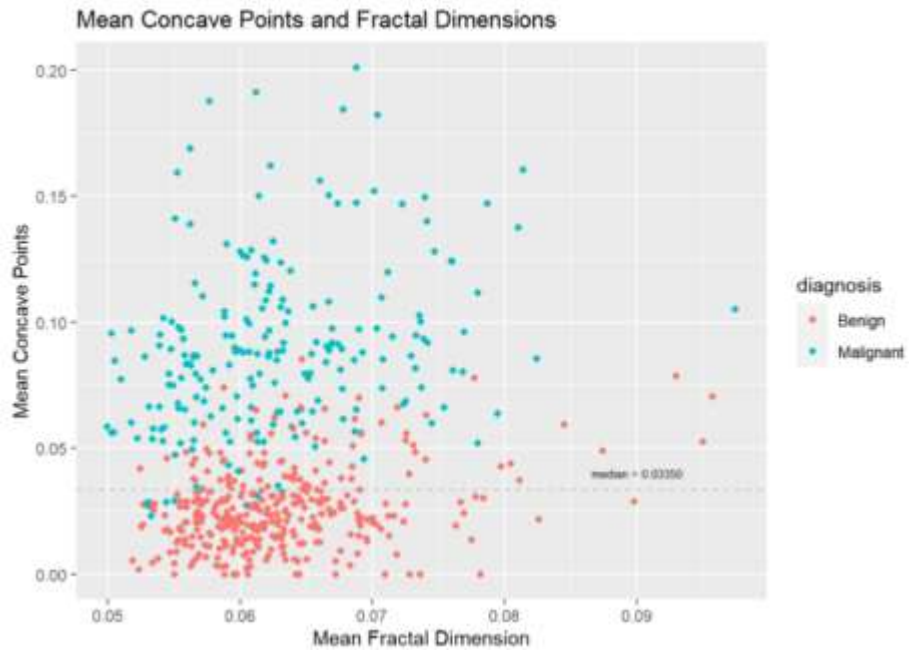
Most benign display less concavity and compactness

- Insight from Compactness and Concavity is that there is a clear display of outliers within the data. However, a visual analysis reveals that benign lumps tend to have low mean concavity and a low mean compactness.
- This can is manifested in the benign being skewed towards the bottom left side of the graph.
  Also, that the malignant are displaying a wider range from low concavity and low compactness to high concavity and high compactness.

- This visualization suggests that benign usually have low to medium severe concaves at the contours of the lumps however malignant lumps can display anywhere between low and very high concavity and compactness.

Plotting Concave points and Fractal dimensions:

```
ggplot(data = data,
       aes(x = fractal_dimension_mean, y = concave.points_mean, color = diagnosis)) +
  geom_point()+
  geom_hline(yintercept = 0.03350, linetype = 'dashed', color = 'gray')+
  labs(title = 'Mean Concave Points and Fractal Dimensions',
       x = 'Mean Fractal Dimension', y = 'Mean Concave Points') +
annotate('text', label = 'median = 0.03350', x = 0.09, y = 0.04,
           size = 2.3)
```



- 95% of malignant are above the Median of Mean Concave Points

- In terms of factal dimensions, there is not enough difference between malignant and benign lumps. However, there is a major difference when it comes to the mean concave points observed amongst both diagnoses.

- 95% of the malignant diagnosed lumps are above the 50th percentile of the observations.

- This suggests that a visual analysis of malignant lumps is likely to display more concave points (severe/sharp curvatures) than benign lumps.

Modelling the dataset:

- Splitting the dataset into train and test set. We split the dataset into Train (80%) and Test (20%), in order to predict is whether a cancer cell is Benign or Malignant, by building machine learning classification models.

```
set.seed(345)
data1<-sample(2,nrow(data),replace = T,prob = c(0.75,0.25))

train<-data[data1==1,]
head(train)
```

Train set:

```
##   diagnosis radius_mean texture_mean perimeter_mean area_mean smoothness_mean
## 1 Malignant       17.99        10.38         122.80    1001.0         0.11840
## 2 Malignant       20.57        17.77         132.90    1326.0         0.08474
## 3 Malignant       19.69        21.25         130.00    1203.0         0.10960
## 4 Malignant       11.42        20.38          77.58     386.1         0.14250
## 5 Malignant       20.29        14.34         135.10    1297.0         0.10030
## 7 Malignant       18.25        19.98         119.60    1040.0         0.09463
##   compactness_mean concavity_mean concave.points_mean symmetry_mean
## 1          0.27760         0.3001              0.14710        0.2419
## 2          0.07864         0.0869              0.07017        0.1812
## 3          0.15990         0.1974              0.12790        0.2069
## 4          0.28390         0.2414              0.10520        0.2597
## 5          0.13280         0.1980              0.10430        0.1809
## 7          0.10900         0.1127              0.07400        0.1794
##   fractal_dimension_mean radius_se texture_se perimeter_se area_se
## 1                0.07871    1.0950     0.9053        8.589  153.40
## 2                0.05667    0.5435     0.7339        3.398   74.08
## 3                0.05999    0.7456     0.7869        4.585   94.03
## 4                0.09744    0.4956     1.1560        3.445   27.23
## 5                0.05883    0.7572     0.7813        5.438   94.44
## 7                0.05742    0.4467     0.7732        3.180   53.91
##   smoothness_se compactness_se concavity_se concave.points_se symmetry_se
## 1      0.006399        0.04904      0.05373           0.01587     0.03003
## 2      0.005225        0.01308      0.01860           0.01340     0.01389
## 3      0.006150        0.04006      0.03832           0.02058     0.02250
## 4      0.009110        0.07458      0.05661           0.01867     0.05963
## 5      0.011490        0.02461      0.05688           0.01885     0.01756
## 7      0.004314        0.01382      0.02254           0.01039     0.01369
```

```
test<-data[data1==2,]
head(test)
```

Test set:

```
##    diagnosis radius_mean texture_mean perimeter_mean area_mean smoothness_mean
## 6  Malignant       12.45        15.70          82.57     477.1         0.12780
## 8  Malignant       13.71        20.83          90.20     577.9         0.11890
## 12 Malignant       15.78        17.89         103.60     781.0         0.09710
## 15 Malignant       13.73        22.61          93.60     578.3         0.11310
## 17 Malignant       14.68        20.13          94.74     684.5         0.09867
## 19 Malignant       19.81        22.15         130.00    1260.0         0.09831
##    compactness_mean concavity_mean concave.points_mean symmetry_mean
## 6            0.1700        0.15780             0.08089        0.2087
## 8            0.1645        0.09366             0.05985        0.2196
## 12           0.1292        0.09954             0.06606        0.1842
## 15           0.2293        0.21280             0.08025        0.2069
## 17           0.0720        0.07395             0.05259        0.1586
## 19           0.1027        0.14790             0.09498        0.1582
##    fractal_dimension_mean radius_se texture_se perimeter_se area_se
## 6                 0.07613    0.3345     0.8902        2.217   27.19
## 8                 0.07451    0.5835     1.3770        3.856   50.96
## 12                0.06082    0.5058     0.9849        3.564   54.16
## 15                0.07682    0.2121     1.1690        2.061   19.21
## 17                0.05922    0.4727     1.2400        3.195   45.40
## 19                0.05395    0.7582     1.0170        5.865  112.40
##    smoothness_se compactness_se concavity_se concave.points_se symmetry_se
## 6       0.007510        0.03345      0.03672           0.01137     0.02165
## 8       0.008805        0.03029      0.02488           0.01448     0.01486
## 12      0.005771        0.04061      0.02791           0.01282     0.02008
## 15      0.006429        0.05936      0.05501           0.01628     0.01961
## 17      0.005718        0.01162      0.01998           0.01109     0.01410
## 19      0.006494        0.01893      0.03391           0.01521     0.01356
```

Checking the proportion of Benign and Malignant in train and test set:

```
prop.table(table(train$diagnosis))
```

```
##
##    Benign Malignant
## 0.6426966 0.3573034
```

From the above output the inference is that the train set contains 64% Benign and 35% Malignant.

```
prop.table(table(test$diagnosis))
```

```
##
##    Benign Malignant
## 0.5725806 0.4274194
```

From the above output the inference is that the test set contains 57% Benign and 42% Malignant.

PCA (Primary Component Analysis):

- It is statistical procedure that is used to summarize the information content in the data tables. It is more easy to visualize and analyze.

```
data<-data[,-1]
all_pca <- prcomp(data[,-1], cor=TRUE, scale = TRUE)
```

```
## Warning: In prcomp.default(data[, -1], cor = TRUE, scale = TRUE) :
##  extra argument 'cor' will be disregarded
```

```
summary(all_pca)
```

```
## Importance of components:
##                           PC1    PC2     PC3     PC4     PC5     PC6     PC7
## Standard deviation     3.5602 2.3145 1.67860 1.40601 1.28301 1.09859 0.81534
## Proportion of Variance 0.4371 0.1847 0.09716 0.06817 0.05676 0.04162 0.02292
## Cumulative Proportion  0.4371 0.6218 0.71895 0.78712 0.84388 0.88550 0.90842
##                           PC8     PC9    PC10    PC11    PC12    PC13    PC14
## Standard deviation     0.69036 0.62876 0.58783 0.54148 0.51013 0.49123 0.39543
## Proportion of Variance 0.01643 0.01363 0.01192 0.01011 0.00897 0.00832 0.00539
## Cumulative Proportion  0.92485 0.93849 0.95040 0.96051 0.96948 0.97781 0.98320
##                           PC15    PC16    PC17    PC18    PC19    PC20    PC21
## Standard deviation     0.30645 0.2796 0.23982 0.22774 0.21104 0.17623 0.17248
## Proportion of Variance 0.00324 0.0027 0.00198 0.00179 0.00154 0.00107 0.00103
## Cumulative Proportion  0.98644 0.9891 0.99111 0.99290 0.99444 0.99551 0.99654
##                           PC22    PC23    PC24    PC25    PC26    PC27    PC28
## Standard deviation     0.16495 0.15477 0.13050 0.12436 0.08933 0.08164 0.03850
## Proportion of Variance 0.00094 0.00083 0.00059 0.00053 0.00028 0.00023 0.00005
## Cumulative Proportion  0.99747 0.99830 0.99889 0.99942 0.99970 0.99992 0.99998
##                           PC29
## Standard deviation     0.02635
## Proportion of Variance 0.00002
## Cumulative Proportion  1.00000
```

Applying ML models:

1. SVM model:

```
learn_svm = svm(diagnosis~ .,data = train)
pre_svm <- predict(learn_svm, test[,-1])
cm_svm <- confusionMatrix(pre_svm, test$diagnosis)
cm_svm
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  Benign Malignant
##   Benign        70         2
##   Malignant      1        51
##
##                Accuracy : 0.9758
##                  95% CI : (0.9309, 0.995)
##     No Information Rate : 0.5726
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.9505
##
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 0.9859
##             Specificity : 0.9623
##          Pos Pred Value : 0.9722
##          Neg Pred Value : 0.9808
##              Prevalence : 0.5726
##          Detection Rate : 0.5645
##    Detection Prevalence : 0.5806
##       Balanced Accuracy : 0.9741
##
##        'Positive' Class : Benign
##
```

The Accuracy of SVM model is 97.58

2. Random Forest Model:

```
learn_rf<- randomForest(diagnosis~., data=train, ntree=1000)
pre_rf<- predict(learn_rf, test[,-1])
cm_rf <- confusionMatrix(pre_rf, test$diagnosis)
cm_rf
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  Benign Malignant
##   Benign        68         3
##   Malignant      3        50
##
##                Accuracy : 0.9516
##                  95% CI : (0.8977, 0.982)
##     No Information Rate : 0.5726
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.9011
##
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 0.9577
##             Specificity : 0.9434
##          Pos Pred Value : 0.9577
##          Neg Pred Value : 0.9434
##              Prevalence : 0.5726
##          Detection Rate : 0.5484
##    Detection Prevalence : 0.5726
##       Balanced Accuracy : 0.9506
##
##        'Positive' Class : Benign
##
```

The accuracy of Random Forest model is 95.16

3. KNN model

```
model_knn <- train(diagnosis~.,train,method="knn",tuneLength=10,preProcess = c('center', 'scale'))
pred_knn <- predict(model_knn, test)
cm_knn <- confusionMatrix(pred_knn, test$diagnosis, positive = "Malignant")
cm_knn
```

```
## Confusion Matrix and Statistics
##
##            Reference
## Prediction  Benign Malignant
##   Benign         70         5
##   Malignant       1        48
##
##                Accuracy : 0.9516
##                  95% CI : (0.8977, 0.982)
##     No Information Rate : 0.5726
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.9002
##
##  Mcnemar's Test P-Value : 0.2207
##
##             Sensitivity : 0.9057
##             Specificity : 0.9859
##          Pos Pred Value : 0.9796
##          Neg Pred Value : 0.9333
##              Prevalence : 0.4274
##          Detection Rate : 0.3871
##    Detection Prevalence : 0.3952
##       Balanced Accuracy : 0.9458
##
##        'Positive' Class : Malignant
##
```

The accuracy of KNN model is 95.16.

4. C5 Decision Tree

```
learn_c50 <- C5.0(train[,-1],train$diagnosis)
pre_c50 <- predict(learn_c50, test[,-1])
cm_c50 <- confusionMatrix(pre_c50, test$diagnosis)
cm_c50
```

```
## Confusion Matrix and Statistics
##
##            Reference
## Prediction  Benign Malignant
##   Benign         65         6
##   Malignant       6        47
##
##                Accuracy : 0.9032
##                  95% CI : (0.8371, 0.949)
##     No Information Rate : 0.5726
##     P-Value [Acc > NIR] : 5.195e-16
##
##                   Kappa : 0.8023
##
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 0.9155
##             Specificity : 0.8868
##          Pos Pred Value : 0.9155
##          Neg Pred Value : 0.8868
##              Prevalence : 0.5726
##          Detection Rate : 0.5242
##    Detection Prevalence : 0.5726
##       Balanced Accuracy : 0.9011
##
##        'Positive' Class : Benign
##
```

The accuracy of the C5 decision tree model is 90.32

5. C-Tree model

```
learn_ct <- ctree(diagnosis~., data=train, controls=ctree_control(maxdepth=2))
pre_ct   <- predict(learn_ct, test[,-1])
cm_ct    <- confusionMatrix(pre_ct, test$diagnosis)
cm_ct
```

```
## Confusion Matrix and Statistics
##
##              Reference
## Prediction  Benign Malignant
##    Benign         67         6
##    Malignant       4        47
##
##                   Accuracy : 0.9194
##                     95% CI : (0.8567, 0.9606)
##       No Information Rate : 0.5726
##       P-Value [Acc > NIR] : <2e-16
##
##                      Kappa : 0.8344
##
##  Mcnemar's Test P-Value : 0.7518
##
##                 Sensitivity : 0.9437
##                 Specificity : 0.8868
##              Pos Pred Value : 0.9178
##              Neg Pred Value : 0.9216
##                  Prevalence : 0.5726
##              Detection Rate : 0.5403
##    Detection Prevalence : 0.5887
##       Balanced Accuracy : 0.9152
##
##         'Positive' Class : Benign
##
```

The accuracy of the C-tree model is 91.94

6. Logistic Regression model:

```
model_logreg<- train(diagnosis ~., data = train, method = "glm",
metric = "ROC",
preProcess = c("scale", "center"), # in order to normalize the data
trControl= fitControl)
```

Making the prediction:

```
prediction_logreg<- predict(model_logreg, test)
# Check results
confusionmatrix_logreg <- confusionMatrix(prediction_logreg, test$diagnosis, positive = "Malignant")
confusionmatrix_logreg
```

```
## Confusion Matrix and Statistics
##
##              Reference
## Prediction  Benign Malignant
##    Benign         69         6
##    Malignant       2        47
##
##                   Accuracy : 0.9355
##                     95% CI : (0.8768, 0.9717)
##       No Information Rate : 0.5726
##       P-Value [Acc > NIR] : <2e-16
##
##                      Kappa : 0.8669
##
##  Mcnemar's Test P-Value : 0.2888
##
##                 Sensitivity : 0.8868
##                 Specificity : 0.9718
##              Pos Pred Value : 0.9592
##              Neg Pred Value : 0.9200
##                  Prevalence : 0.4274
##              Detection Rate : 0.3790
##     Detection Prevalence : 0.3952
##        Balanced Accuracy : 0.9293
##
##          'Positive' Class : Malignant
##
```

The accuracy of logistic regression model is 93.55

7. Neural Network model:

```
model_nnet_pca <- train(diagnosis~.,
train,
method="nnet",
metric="ROC",
preProcess=c('center', 'scale', 'pca'),
tuneLength=10,
trace=FALSE,
trControl=fitControl)
prediction_nnet_pca <- predict(model_nnet_pca, test)
confusionmatrix_nnet_pca <- confusionMatrix(prediction_nnet_pca, test$diagnosis, positive = "Malignant")
confusionmatrix_nnet_pca
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  Benign Malignant
##   Benign        71         4
##   Malignant      0        49
##
##                Accuracy : 0.9677
##                  95% CI : (0.9195, 0.9911)
##     No Information Rate : 0.5726
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.9335
##
##  Mcnemar's Test P-Value : 0.1336
##
##             Sensitivity : 0.9245
##             Specificity : 1.0000
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 0.9467
##              Prevalence : 0.4274
##          Detection Rate : 0.3952
##    Detection Prevalence : 0.3952
##       Balanced Accuracy : 0.9623
##
##        'Positive' Class : Malignant
##
```

The accuracy of the neural network model is 96.77

Comparing all the models:

```
##compare the models:

model_list <- list (KNN = model_knn,RF=model_randomforest,LR=model_logreg,NNet=model_nnet_pca)
model_list
```

- Using the KNN model, random forest model, logistic regression and neural network model.

```
models_results <- resamples(model_list)

model_cor <- modelCor(models_results)
corrplot(model_cor)
```

- The correlogram is plotted.

```
## $KNN
## k-Nearest Neighbors
##
## 445 samples
## 30 predictor
##   2 classes: 'Benign', 'Malignant'
##
## Pre-processing: centered (30), scaled (30)
## Resampling: Cross-Validated (15 fold)
## Summary of sample sizes: 415, 415, 416, 415, 416, 415, ...
## Resampling results across tuning parameters:
##
##   k   ROC        Sens       Spec
##    5  0.9878397  0.9964912  0.9236364
##    7  0.9890766  0.9929825  0.9436364
##    9  0.9939091  0.9929825  0.9369697
##   11  0.9935813  0.9929825  0.9309091
##   13  0.9948804  0.9929825  0.9187879
##   15  0.9945455  0.9964912  0.9000000
##   17  0.9935726  0.9964912  0.8939394
##   19  0.9932057  0.9964912  0.8939394
##   21  0.9930303  0.9929825  0.8939394
##   23  0.9933493  0.9929825  0.8878788
##
## ROC was used to select the optimal model using the largest value.
## The final value used for the model was k = 13.
##
## $RF
## Random Forest
##
## 445 samples
##  30 predictor
##   2 classes: 'Benign', 'Malignant'
##
## Pre-processing: centered (30), scaled (30)
## Resampling: Cross-Validated (15 fold)
## Summary of sample sizes: 415, 416, 415, 416, 416, 415, ...
## Resampling results across tuning parameters:
##
##   mtry  ROC        Sens       Spec
##    2    0.9914354  0.9789474  0.9248485
##   16    0.9871132  0.9824561  0.9503030
##   30    0.9871132  0.9754386  0.9375758
##
## ROC was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 2.
##
## $LR
## Generalized Linear Model
```

```
##
## 445 samples
##  30 predictor
##   2 classes: 'Benign', 'Malignant'
##
## Pre-processing: scaled (30), centered (30)
## Resampling: Cross-Validated (15 fold)
## Summary of sample sizes: 414, 416, 415, 415, 415, 416, ...
## Resampling results:
##
##   ROC       Sens       Spec
##   0.975949  0.9514035  0.9054545
##
##
## $NNet
## Neural Network
##
## 445 samples
##  30 predictor
##   2 classes: 'Benign', 'Malignant'
##
## Pre-processing: centered (30), scaled (30), principal component
##  signal extraction (30)
## Resampling: Cross-Validated (15 fold)
## Summary of sample sizes: 416, 416, 415, 415, 414, 416, ...
## Resampling results across tuning parameters:
##
##   size  decay         ROC        Sens       Spec
##    1    0.0000000000  0.9735167  0.9685965  0.9303030
##    1    0.0001000000  0.9868581  0.9652632  0.9551515
##    1    0.0002371374  0.9872727  0.9791228  0.9672727
##    1    0.0005623413  0.9915949  0.9757895  0.9490909
##    1    0.0013335214  0.9919458  0.9791228  0.9478788
##    1    0.0031622777  0.9956459  0.9756140  0.9545455
##    1    0.0074989421  0.9980383  0.9826316  0.9551515
##    1    0.0177827941  0.9986922  0.9791228  0.9490909
##    1    0.0421696503  0.9990431  0.9791228  0.9551515
##    1    0.1000000000  0.9990431  0.9791228  0.9551515
##    3    0.0000000000  0.9728230  0.9687719  0.9363636
##    3    0.0001000000  0.9924083  0.9685965  0.9618182
##    3    0.0002371374  0.9956459  0.9756140  0.9618182
##    3    0.0005623413  0.9935885  0.9721053  0.9424242
##    3    0.0013335214  0.9901754  0.9652632  0.9618182
##    3    0.0031622777  0.9924561  0.9650877  0.9545455
##    3    0.0074989421  0.9969219  0.9650877  0.9678788
##    3    0.0177827941  0.9974322  0.9685965  0.9618182
##    3    0.0421696503  0.9955024  0.9756140  0.9678788
##    3    0.1000000000  0.9983413  0.9756140  0.9551515
##    5    0.0000000000  0.9688915  0.9580702  0.9551515
```

```
##    5    0.0001000000  0.9952472  0.9685965  0.9563636
##    5    0.0002371374  0.9972887  0.9685965  0.9672727
##    5    0.0005623413  0.9955981  0.9721053  0.9678788
##    5    0.0013335214  0.9939075  0.9721053  0.9745455
##    5    0.0031622777  0.9960447  0.9685965  0.9418182
##    5    0.0074989421  0.9962839  0.9687719  0.9612121
##    5    0.0177827941  0.9982456  0.9826316  0.9678788
##    5    0.0421696503  0.9983413  0.9721053  0.9612121
##    5    0.1000000000  0.9986922  0.9756140  0.9551515
##    7    0.0000000000  0.9676156  0.9442105  0.9424242
##    7    0.0001000000  0.9948644  0.9791228  0.9684848
##    7    0.0002371374  0.9953748  0.9685965  0.9551515
##    7    0.0005623413  0.9960766  0.9650877  0.9618182
##    7    0.0013335214  0.9948485  0.9721053  0.9739394
##    7    0.0031622777  0.9979745  0.9756140  0.9551515
##    7    0.0074989421  0.9983413  0.9721053  0.9618182
##    7    0.0177827941  0.9976874  0.9650877  0.9678788
##    7    0.0421696503  0.9986922  0.9791228  0.9618182
##    7    0.1000000000  0.9983732  0.9791228  0.9551515
##    9    0.0000000000  0.9774561  0.9687719  0.9612121
##    9    0.0001000000  0.9940829  0.9722807  0.9678788
##    9    0.0002371374  0.9951515  0.9721053  0.9551515
##    9    0.0005623413  0.9976077  0.9721053  0.9739394
##    9    0.0013335214  0.9982775  0.9826316  0.9612121
##    9    0.0031622777  0.9979266  0.9721053  0.9545455
##    9    0.0074989421  0.9985965  0.9756140  0.9678788
##    9    0.0177827941  0.9983254  0.9721053  0.9612121
##    9    0.0421696503  0.9986922  0.9756140  0.9739394
##    9    0.1000000000  0.9986922  0.9791228  0.9678788
##   11    0.0000000000  0.9858772  0.9791228  0.9806061
##   11    0.0001000000  0.9986443  0.9756140  0.9678788
##   11    0.0002371374  0.9951994  0.9721053  0.9745455
##   11    0.0005623413  0.9982935  0.9721053  0.9678788
##   11    0.0013335214  0.9966507  0.9791228  0.9684848
##   11    0.0031622777  0.9976715  0.9721053  0.9618182
##   11    0.0074989421  0.9986284  0.9789474  0.9678788
##   11    0.0177827941  0.9976874  0.9756140  0.9678788
##   11    0.0421696503  0.9989793  0.9791228  0.9618182
##   11    0.1000000000  0.9986762  0.9791228  0.9678788
##   13    0.0000000000  0.9866826  0.9754386  0.9612121
##   13    0.0001000000  0.9956140  0.9791228  0.9739394
##   13    0.0002371374  0.9944338  0.9721053  0.9478788
##   13    0.0005623413  0.9972887  0.9756140  0.9678788
##   13    0.0013335214  0.9982935  0.9721053  0.9739394
##   13    0.0031622777  0.9976236  0.9721053  0.9618182
##   13    0.0074989421  0.9982775  0.9721053  0.9678788
##   13    0.0177827941  0.9992982  0.9791228  0.9618182
##   13    0.0421696503  0.9986922  0.9791228  0.9551515
##   13    0.1000000000  0.9990431  0.9791228  0.9678788
```

```
##   15    0.0000000000  0.9854067  0.9756140  0.9678788
##   15    0.0001000000  0.9948963  0.9721053  0.9745455
##   15    0.0002371374  0.9979266  0.9791228  0.9678788
##   15    0.0005623413  0.9982775  0.9791228  0.9678788
##   15    0.0013335214  0.9982935  0.9721053  0.9745455
##   15    0.0031622777  0.9989474  0.9756140  0.9745455
##   15    0.0074989421  0.9989474  0.9756140  0.9612121
##   15    0.0177827941  0.9989474  0.9721053  0.9618182
##   15    0.0421696503  0.9992982  0.9756140  0.9684848
##   15    0.1000000000  0.9983732  0.9791228  0.9678788
##   17    0.0000000000  0.9838915  0.9721053  0.9612121
##   17    0.0001000000  0.9950080  0.9685965  0.9684848
##   17    0.0002371374  0.9951196  0.9756140  0.9745455
##   17    0.0005623413  0.9982456  0.9721053  0.9678788
##   17    0.0013335214  0.9979426  0.9721053  0.9618182
##   17    0.0031622777  0.9986922  0.9826316  0.9745455
##   17    0.0074989421  0.9985965  0.9791228  0.9745455
##   17    0.0177827941  0.9992982  0.9789474  0.9678788
##   17    0.0421696503  0.9992982  0.9791228  0.9618182
##   17    0.1000000000  0.9986922  0.9756140  0.9618182
##   19    0.0000000000  0.9758692  0.9684211  0.9418182
##   19    0.0001000000  0.9989474  0.9721053  0.9739394
##   19    0.0002371374  0.9963876  0.9791228  0.9745455
##   19    0.0005623413  0.9985965  0.9685965  0.9618182
##   19    0.0013335214  0.9982775  0.9721053  0.9557576
##   19    0.0031622777  0.9989474  0.9721053  0.9678788
##   19    0.0074989421  0.9986443  0.9756140  0.9678788
##   19    0.0177827941  0.9989952  0.9756140  0.9684848
##   19    0.0421696503  0.9986922  0.9756140  0.9618182
##   19    0.1000000000  0.9993461  0.9791228  0.9618182
##
## ROC was used to select the optimal model using the largest value.
## The final values used for the model were size = 19 and decay = 0.1.
```

Comparison of all the machine learning techniques with respect to F1 score, precision and recall.

```
confusionmatrix_list <- list(
SVM=cm_svm,
Logistic_regr=confusionmatrix_logreg,
Random_Forest=confusionmatrix_randomforest,
KNN=confusionmatrix_knn,
Neural_PCA=confusionmatrix_nnet_pca)
confusionmatrix_list_results <- sapply(confusionmatrix_list, function(x) x$byClass)
confusionmatrix_list_results
```

```
##                          SVM Logistic_regr Random_Forest       KNN Neural_PCA
## Sensitivity          0.9859155     0.8867925     0.9433962 0.9056604  0.9245283
## Specificity          0.9622642     0.9718310     0.9718310 0.9859155  1.0000000
## Pos Pred Value       0.9722222     0.9591837     0.9615385 0.9795918  1.0000000
## Neg Pred Value       0.9807692     0.9200000     0.9583333 0.9333333  0.9466667
## Precision            0.9722222     0.9591837     0.9615385 0.9795918  1.0000000
## Recall               0.9859155     0.8867925     0.9433962 0.9056604  0.9245283
## F1                   0.9790210     0.9215686     0.9523810 0.9411765  0.9607843
## Prevalence           0.5725806     0.4274194     0.4274194 0.4274194  0.4274194
## Detection Rate       0.5645161     0.3790323     0.4032258 0.3870968  0.3951613
## Detection Prevalence 0.5806452     0.3951613     0.4193548 0.3951613  0.3951613
## Balanced Accuracy    0.9740898     0.9293117     0.9576136 0.9457879  0.9622642
```

RESULTS:

In this project treats the Wisconsin Breast Cancer diagnosis problem as a pattern classification problem. In this project we investigated several machine learning models and we selected the optimal model by selecting a high accuracy level combined with a low rate of false-negatives (the means that the metric is high sensitivity). The SVM model had the optimal results for F1 (0.970210), Sensitivity (0.9859155), recall (0.9859155), precision (0.9722222) and Balanced Accuracy (0.97405898).

Conclusion:

In this project we have used seven traditional machine learning model to classify breast cancer. The best model is SVM model with an accuracy score of 0.9758. Future work includes tuning the hyperparameters of the current model as well as testing other deep learning method/architectures to increase model accuracy. Overall, the presented machine learning models that can be applied in breast cancer diagnosis to improve the accuracy and therefore assist early diagnosis of breast cancer.