VIT®

Vellore Institute of Technology

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

School of Computer Science and Engineering

J Component report

Programme : B.Tech CSE

Course Title : DATA VISUALIZATION

Course Code : CSE3020

Slot : F1

Title: BREAST CANCER PREDICTION

Team Members:

Hima Rani Mathews – 19BCE1532

Kamalika Gunasekaran – 19BCE1588

Submitted to:

Parvathi R

Sign:

Date: 27/04/2022

DECLARATION BY THE CANDIDATE

I hereby declare that the report titled "***Breast Cancer Prediction***" submitted by **Kamalika Gunasekaran (19BCE1588)** and **Hima Rani Mathews (19BCE1532)** to VIT Chennai is a record of bona-fide work undertaken by me under the supervision of Dr. Parvathi R**, SCOPE, Vellore Institute of Technology, Chennai.**

Hima Rani Mathews **Kamalika Gunasekaran**

ACKNOWLEDGEMENT

We wish to express our sincere thanks and deep sense of gratitude to our project guide, **Dr. Parvathi R,** School of Computer Science and Engineering for her consistent encouragement and valuable guidance offered to us throughout the course of the project work.

We are extremely grateful to **Dr. R. Ganesan, Dean,** School of Computer Science and Engineering (SCOPE), Vellore Institute of Technology, Chennai, for extending the facilities of the school towards our project and for his unstinting support.

We express our thanks to our **Head of the Department** for his support throughout the course of this project.

We also take this opportunity to thank all the faculty of the school for their support and their wisdom imparted to us throughout the courses.

We thank our parents, family, and friends for bearing with us throughout the course of our project and for the opportunity they provided us in undergoing this course in such a prestigious institution.

BONAFIDE CERTIFICATE

Certified that this project report entitled "***Breast Cancer Prediction***" is a bona-fide work of **Kamalika Gunasekaran (19BCE1588)** and **Hima Rani Mathews (19BCE1532)** carried out the "J"-Project work under my supervision and guidance for **CSE3020- Data Visualization.**

Dr. Parvathi R
SCOPE

TABLE OF CONTENTS

ABSTRACT

Breast cancer affects the majority of women worldwide, and it is the second most common cause of death among women. However, if cancer is detected early and treated properly, it is possible to be cured of the condition. Early detection of breast cancer can dramatically improve the prognosis and chances of survival by allowing patients to receive timely clinical therapy. Furthermore, precise benign tumour classification can help patients avoid unneeded treatment. In this paper, we explore machine learning models that can be applied to help increasing the accuracy of the diagnosis of breast cancer. The main problem of the project is to detect breast cancer based on a set of features calculated from a digitized image of the Fine Needle Aspiration (FNA) of a breast mass from a patient. We present a diagnosis model using both traditional and deep learning machine learning models. Classic machine learning models including Naïve bayes, Rpart, Random Forest, AdaBoost, SVM and Tune SVM are tested on the Breast Cancer Wisconsin dataset. The Wisconsin Breast Cancer dataset is obtained from a prominent machine learning database named UCI machine learning database. The performance of the study is measured with respect to accuracy, sensitivity, specificity, negative predictive value and positive predictive value. The results reveal that the SVM tune has obtained the highest accuracy of 98.83 whereas 94.15%, 95.25%, 97.66%, 97.75% and 98.2% are accuracies obtained by Naïve Bayes, RPart, Random Forest, AdaBoost and SVM respectively.

CHAPTER – 1
OVERVIEW

1.1 Introduction

Breast cancer is one of the most common cancers in women and the second leading cause of women's cancer death. Despite the lack of effective treatment, the low accuracy of diagnosis is also a major cause of the high incidence and mortality of breast cancer. Mammography is a traditional method used for diagnosing breast cancer. According to UC Health's report, only 78% of breast cancer can be accurately diagnosed by mammography. Many cases such as doctors' negligence or incompetence in addition to a mammography error may also result in a late diagnosis or misdiagnosis, which can be considered a cause of breast cancer death. In the long term, early-stage diagnosis could significantly increase the survival rate of breast cancer, therefore, it is important to improve the accuracy of breast cancer diagnosis. Machine learning has been applied in medical diagnosis in a large number of papers. In order to increase the accuracy of breast cancer diagnosis, we aim to use machine learning models and choose the model with higher performance.

Breast Cancer Wisconsin is a widely used dataset provided by UC Irvine machine learning repository. In this paper, we will train our models using this dataset. The input of our algorithm is a set of features calculated from a digitized image of the Fine Needle Aspiration (FNA) of a breast mass from a patient. We will then use six traditional methods including Naïve bayes, Rpart, Random Forest, AdaBoost, SVM and Tune SVM to predict whether the case is benign or malignant.

1.2 Aim of Project

The objective of this report 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. We will later define these metrics. We can use machine learning method to extract the features of cancer cell nuclei image and classify them. It would be helpful to determine whether a given sample appears to be Benign ("B") or Malignant ("M"). The machine learning models that we will applicate in this report try to create a classifier that provides a high accuracy level combined with a low rate of false-negatives (high sensitivity).

1.3 Dataset

The present report covers the Breast Cancer Wisconsin (Diagnostic). DataSet (https://www.kaggle.com/datasets/uciml/breast-cancer-wisconsin-data) created by Dr. William H. Wolberg, physician at the University Of Wisconsin Hospital at Madison, Wisconsin, USA. The data used for this project was collected in 1993 by the University of Wisconsin and it is composed by the biopsy result of 569 patients in Wisconsin Hospital.

The dataset's features describe characteristics of the cell nuclei on the image. The features information are specified below:

Attribute Information:

1. ID number
2. Diagnosis (M = malignant, B = benign)

Ten features were computed for each cell nucleus:

1. radius: mean of distances from center to points on the perimeter
2. texture: standard deviation of grey-scale values
3. perimeter
4. area: Number of pixels inside contour + ½ for pixels on perimeter

5. smoothness: local variation in radius lengths)

6. compactness (perimeter^2 / area - 1.0)

7. concavity: severity of concave portions of the contour

8. concave points: number of concave portions of the contour

9. symmetry

10. fractal dimension: “coastline approximation” - 1; a higher value corresponds a less regular contour and thus to a higher probability of malignancy.

The mean, standard error and “worst” or largest (mean of the three largest values) of these features were computed for each image, resulting in 30 variables. From this diagnosis, 357 of the cases were classified as benign tumours and 212 were considered malignant tumours. All cancers and some of the benign masses were histologically confirmed.

CHAPTER – 2
LITERATURE SURVEY

Ali Al Bataineh in his study on a Comparative Analysis of Non-Linear ML algorithms showed comparative study of prediction between five models built on algorithms like Multilayer Perceptron (MLP), KNN, Navie Bayes etc on WDBC dataset. His worked showed that MLP performed best with an accuracy of 99.12% on training dataset.

Another significant work in this field is done by JianHuangLai, Chee-Keong Kwoh and their team developed a new ensemble clustering technique and termed is as MDEC. They showed that large random population of varied metrics can be beneficial for clustering using ensemble. The framework they built was combination of three clustering algorithms using consensus functions. They conducted experiments over more than30 real world data-set of high dimensionality.

Pragya Chauhan and Amit Swami in their work of prediction of breast cancer using algorithm which wasbased onensemble approach and genetic algorithmsuggested a model where they found that the cancer prediction is a field which is open for research. ML algorithms were used for detection and prognosis of cancer. Linear model, naive bayes, CARTetc. methods were used for prediction.

A work was done in prediction of cancer and lymph nodes with tumour marker by Hsiao-Lin Hwa and his team. They used serum samples of femalepatients with and without breast cancer. Then they used logistic regression of univariate and multivariate for evaluation. According to their serum level of TPS had the best value for prediction, with combination of various medical parameter or biomarkers and integrating them with logistic regression raised the model sensitivity and accuracy significantly.

CHAPTER – 3

METHOD AND ANALYSIS

3.1 ANALYSING THE DATASET

```
wbcd <- read.csv("data.csv")
head(wbcd,10)
```

```
##          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
## 7    844359         M       18.25        19.98         119.60    1040.0
## 8  84458202         M       13.71        20.83          90.20     577.9
## 9    844981         M       13.00        21.82          87.50     519.8
## 10 84501001         M       12.46        24.04          83.97     475.9
##    smoothness_mean compactness_mean concavity_mean concave.points_mean
## 1          0.11840          0.27760        0.30010             0.14710
## 2          0.08474          0.07864        0.08690             0.07017
## 3          0.10960          0.15990        0.19740             0.12790
## 4          0.14250          0.28390        0.24140             0.10520
## 5          0.10030          0.13280        0.19800             0.10430
## 6          0.12780          0.17000        0.15780             0.08089
## 7          0.09463          0.10900        0.11270             0.07400
## 8          0.11890          0.16450        0.09366             0.05985
## 9          0.12730          0.19320        0.18590             0.09353
## 10         0.11860          0.23960        0.22730             0.08543
##    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
## 7         0.1794                0.05742    0.4467     0.7732        3.180
## 8         0.2196                0.07451    0.5835     1.3770        3.856
## 9         0.2350                0.07389    0.3063     1.0020        2.406
## 10        0.2030                0.08243    0.2976     1.5990        2.039
##    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
## 7    53.91      0.004314        0.01382      0.02254           0.01039
## 8    50.96      0.008805        0.03029      0.02488           0.01448
```

- By observing our dataset, we found that it contains 569 observations with 32 variables.

```
dim(wbcd)
```

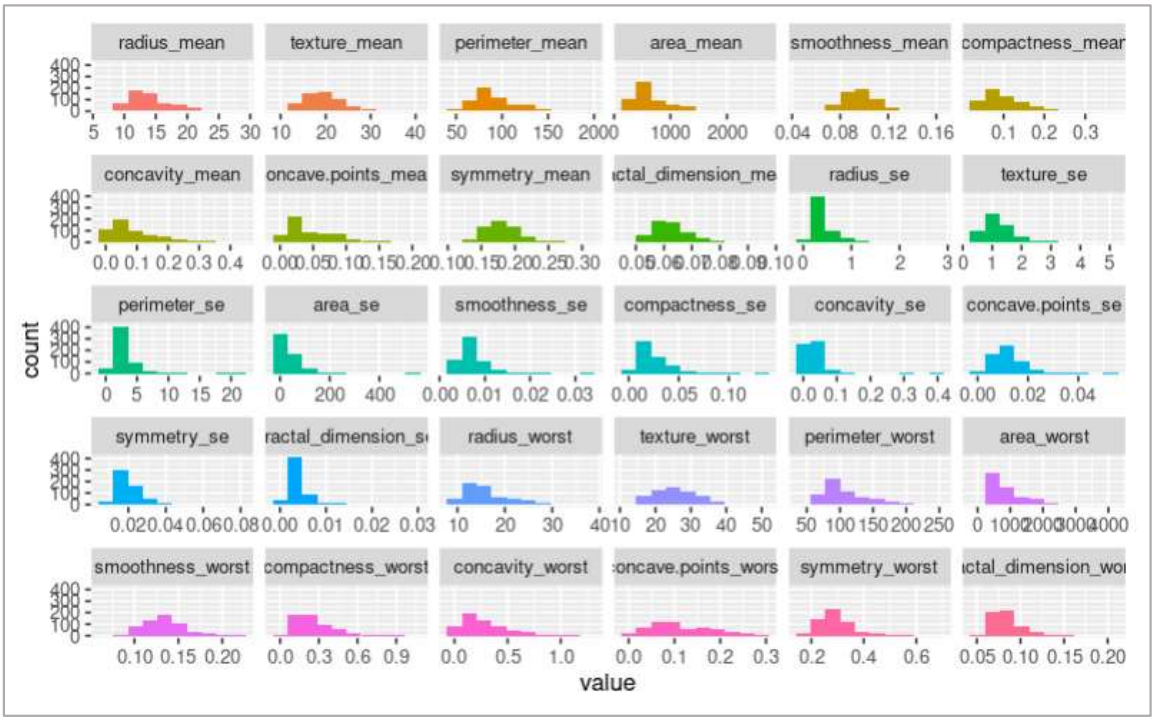
```
## [1] 569  33
```

```
str(wbcd)
```

```
## 'data.frame':    569 obs. of  33 variables:
##  $ id                     : int  842302 842517 84300903 84348301 84358402 843786 844359 84458202 844981 84501001 ...
##  $ diagnosis              : chr  "M" "M" "M" "M" ...
##  $ radius_mean            : num  18 20.6 19.7 11.4 20.3 ...
##  $ texture_mean           : num  10.4 17.8 21.2 20.4 14.3 ...
##  $ perimeter_mean         : num  122.8 132.9 130 77.6 135.1 ...
##  $ area_mean              : num  1001 1326 1203 386 1297 ...
##  $ smoothness_mean        : num  0.1184 0.0847 0.1096 0.1425 0.1003 ...
##  $ compactness_mean       : num  0.2776 0.0786 0.1599 0.2839 0.1328 ...
##  $ concavity_mean         : num  0.3001 0.0869 0.1974 0.2414 0.198 ...
##  $ concave.points_mean    : num  0.1471 0.0702 0.1279 0.1052 0.1043 ...
##  $ symmetry_mean          : num  0.242 0.181 0.207 0.26 0.181 ...
##  $ fractal_dimension_mean : num  0.0787 0.0567 0.06 0.0974 0.0588 ...
##  $ radius_se              : num  1.095 0.543 0.746 0.496 0.757 ...
##  $ texture_se             : num  0.905 0.734 0.787 1.156 0.781 ...
##  $ perimeter_se           : num  8.59 3.4 4.58 3.44 5.44 ...
##  $ area_se                : num  153.4 74.1 94 27.2 94.4 ...
##  $ smoothness_se          : num  0.0064 0.00522 0.00615 0.00911 0.01149 ...
##  $ compactness_se         : num  0.049 0.0131 0.0401 0.0746 0.0246 ...
##  $ concavity_se           : num  0.0537 0.0186 0.0383 0.0566 0.0569 ...
##  $ concave.points_se      : num  0.0159 0.0134 0.0206 0.0187 0.0188 ...
##  $ symmetry_se            : num  0.03 0.0139 0.0225 0.0596 0.0176 ...
##  $ fractal_dimension_se   : num  0.00619 0.00353 0.00457 0.00921 0.00511 ...
##  $ radius_worst           : num  25.4 25 23.6 14.9 22.5 ...
##  $ texture_worst          : num  17.3 23.4 25.5 26.5 16.7 ...
##  $ perimeter_worst        : num  184.6 158.8 152.5 98.9 152.2 ...
##  $ area_worst             : num  2019 1956 1709 568 1575 ...
##  $ smoothness_worst       : num  0.162 0.124 0.144 0.21 0.137 ...
##  $ compactness_worst      : num  0.666 0.187 0.424 0.866 0.205 ...
##  $ concavity_worst        : num  0.712 0.242 0.45 0.687 0.4 ...
##  $ concave.points_worst   : num  0.265 0.186 0.243 0.258 0.163 ...
##  $ symmetry_worst         : num  0.46 0.275 0.361 0.664 0.236 ...
##  $ fractal_dimension_worst: num  0.1189 0.089 0.0876 0.173 0.0768 ...
##  $ X                      : logi  NA NA NA NA NA NA ...
```

- Visualising the distribution of different attribute using plot_num()

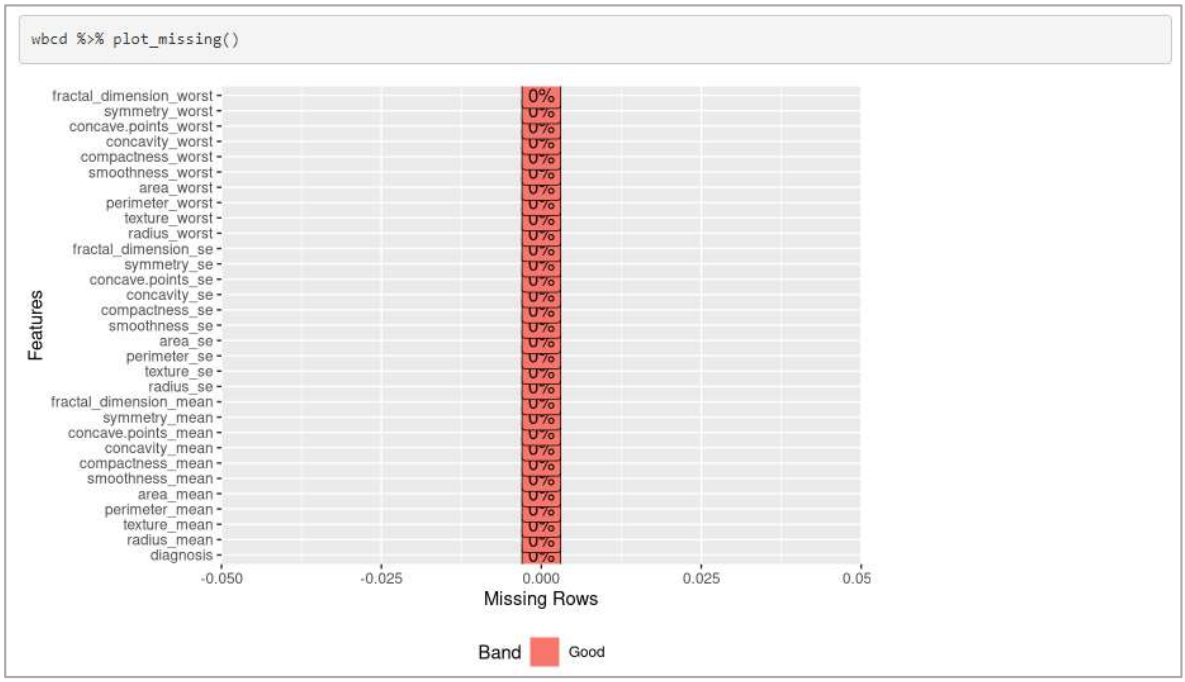
  plot_num(data %>% select(-id), bins=10)



3.2 DATA CLEANING AND PRE-PROCESSING

- No missing data were found in the dataset

```
wbcd %>% plot_missing()
```



- Pre-processed the Diagnosis column

```
wbcd$X <- NULL
wbcd <- wbcd[,-1]
wbcd$diagnosis <- factor(ifelse(wbcd$diagnosis=="B","Benign","Malignant"))
head(wbcd,10)
```

```
##    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
## 7  Malignant       18.25        19.98         119.60    1040.0         0.09463
## 8  Malignant       13.71        20.83          90.20     577.9         0.11890
## 9  Malignant       13.00        21.82          87.50     519.8         0.12730
## 10 Malignant       12.46        24.04          83.97     475.9         0.11860
```

	diagnosis <fct>	radius_mean <dbl>	texture_mean <dbl>	perimeter_mean <dbl>	area_mean <dbl>	smoothness_mean <dbl>	compactness_mean <dbl>
1	Malignant	17.99	10.38	122.80	1001.0	0.11840	0.27760
2	Malignant	20.57	17.77	132.90	1326.0	0.08474	0.07864
3	Malignant	19.69	21.25	130.00	1203.0	0.10960	0.15990
4	Malignant	11.42	20.38	77.58	386.1	0.14250	0.28390
5	Malignant	20.29	14.34	135.10	1297.0	0.10030	0.13280
6	Malignant	12.45	15.70	82.57	477.1	0.12780	0.17000
7	Malignant	18.25	19.98	119.60	1040.0	0.09463	0.10900
8	Malignant	13.71	20.83	90.20	577.9	0.11890	0.16450
9	Malignant	13.00	21.82	87.50	519.8	0.12730	0.19320
10	Malignant	12.46	24.04	83.97	475.9	0.11860	0.23960

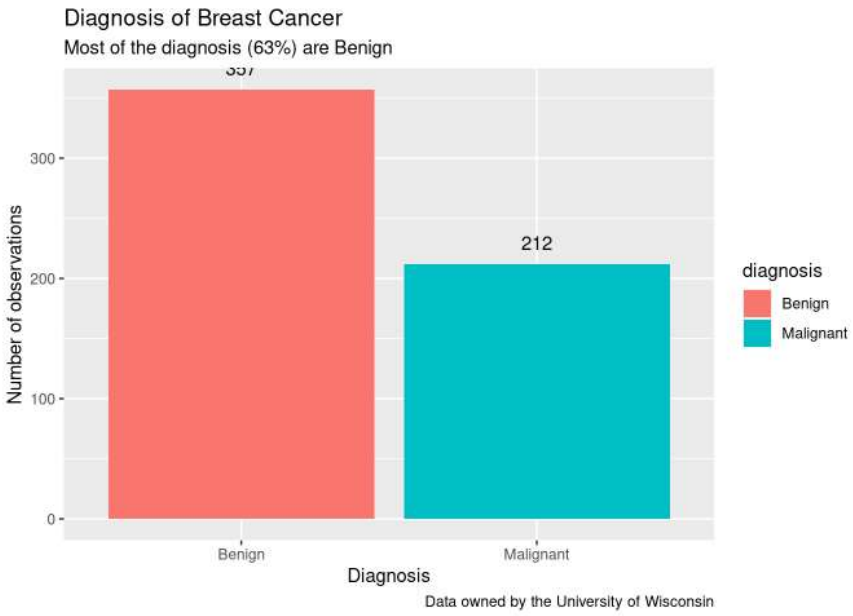
1-10 of 10 rows | 1-8 of 31 columns

3.3 DATA VISUALIZATION

(a) BAR PLOT

Malignant and Benign diagnosis visualization using ggplot() and geom_bar(). From the visualization we were able to find about 212 data were Malignant and rest 357 data were Benign.

```
#Insight into Breast Cancer Wisconsin- Data
ggplot(data = wbcd, aes(x = diagnosis, fill = diagnosis)) +
geom_bar()+
geom_text(stat='count', aes(label=..count..), vjust=-1) +
labs(title = 'Diagnosis of Breast Cancer',
     subtitle = 'Most of the diagnosis (63%) are Benign',
     caption = 'Data owned by the University of Wisconsin',
     x = 'Diagnosis', y = 'Number of observations')
```



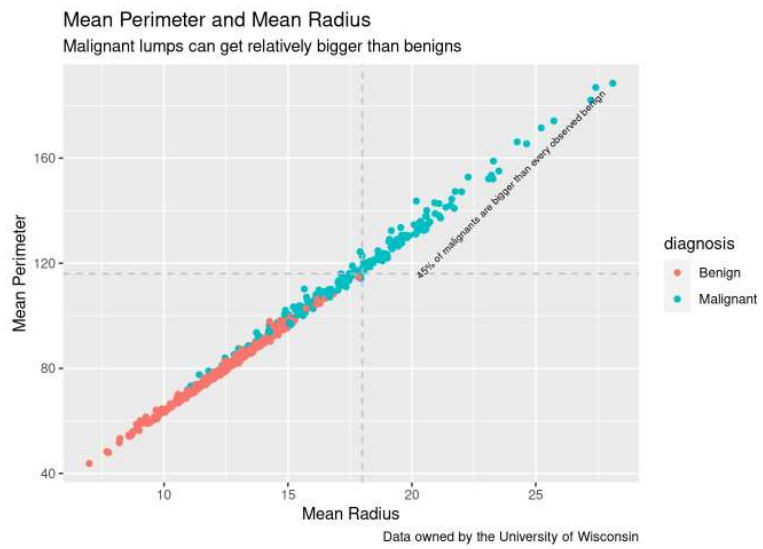
(b) SCATTER PLOTS

(i) Mean Perimeter and Mean Radius

Over here, we will be using the mean perimeter and the mean radius observed from the center of the lump to the perimeter. This will reveal how both types of lumps look in relative size.

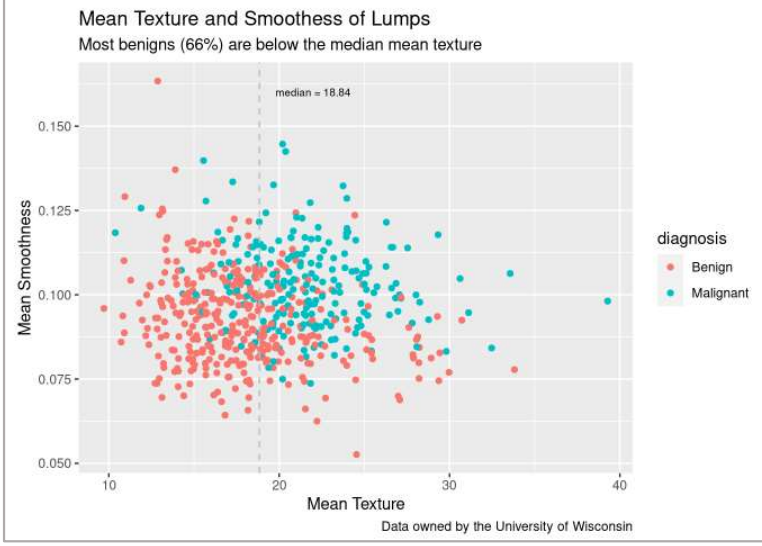
- **Insights:** Malignant lumps can get relatively bigger than benign lumps. This has the possibility of sparking up a hypothesis that malignant lumps begin as benigns.

```
ggplot(data = wbcd,
       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',
       subtitle = 'Malignant lumps can get relatively bigger than benigns',
       caption = 'Data owned by the University of Wisconsin',
       x = 'Mean Radius', y = 'Mean Perimeter') +
  annotate('text', x = 24, y = 150,
           label = '45% of malignants are bigger than every observed benign',
           size = 2.3, angle = 45)
```



(ii) Mean Texture and Smoothness of Lumps

```
ggplot(data = wbcd,
       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',
       subtitle = 'Most benigns (66%) are below the median mean texture',
       caption = 'Data owned by the University of Wisconsin',
       x = 'Mean Texture', y = 'Mean Smoothness') +
  annotate('text', label = 'median = 18.84', x = 22, y = 0.160,
           size = 2.5)
```

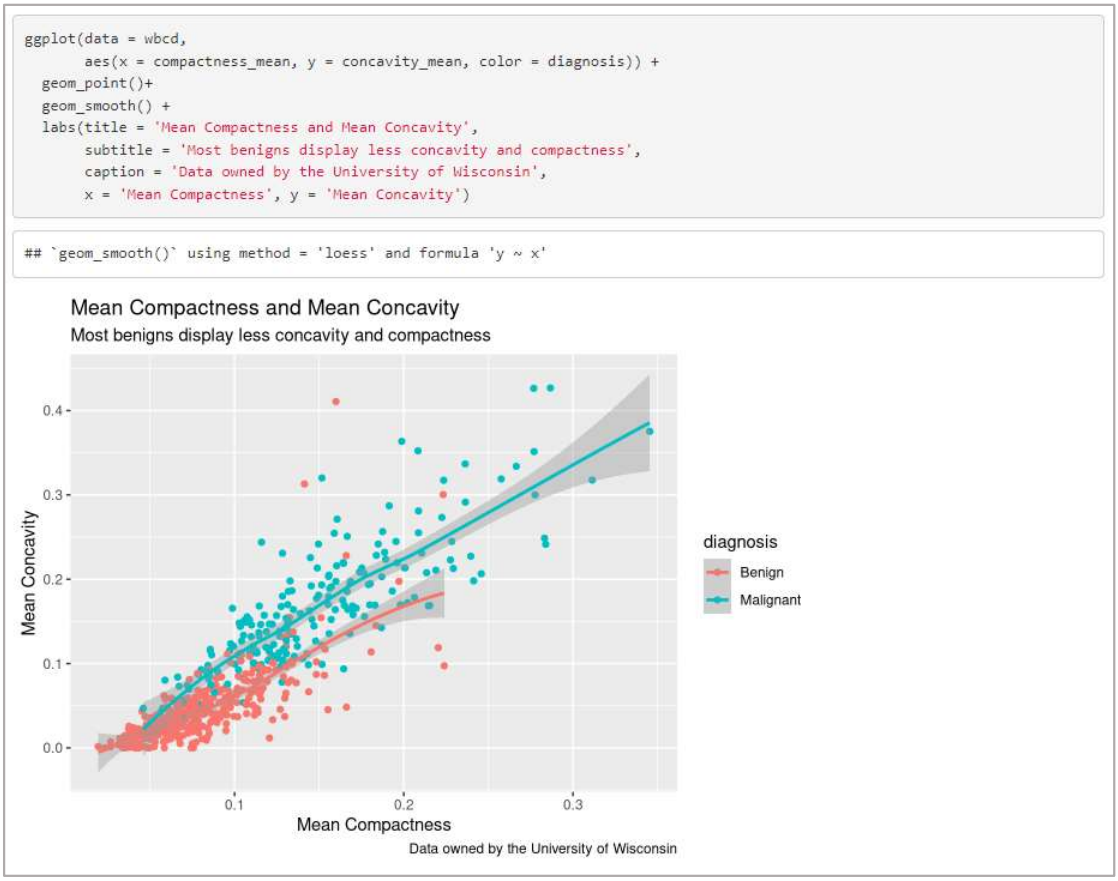


- **Insights**: Not a lot of variation can be seen in the mean smoothness of both diagnosis as they all seem to clustered from the bottom to the upper midsection of the plot. However, we can observe that most of the malignant (66%) are skewed to the right side of the median. This connotes that malignant lump display higher texture variation values than benign.

(iii) Compactness and Concavity

```
ggplot(data = wbcd,
       aes(x = compactness_mean, y = concavity_mean, color = diagnosis)) +
  geom_point()+
  geom_smooth() +
  labs(title = 'Mean Compactness and Mean Concavity',
       subtitle = 'Most benigns display less concavity and compactness',
       caption = 'Data owned by the University of Wisconsin',
       x = 'Mean Compactness', y = 'Mean Concavity')
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

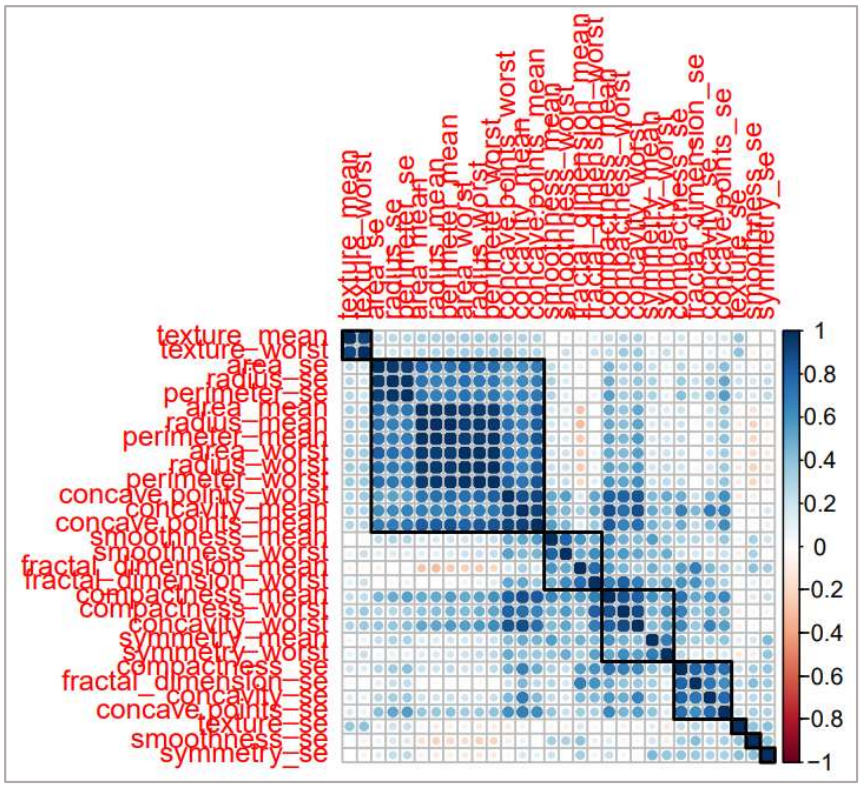


- **Insights**: There is a clear display of outliers within the data. However, a visual analysis reveals that benign lumps tend to have low mean concavity and a low mean compactness. This can is manifested in the benign being skewed towards the bottom left side of the graph. Notice that the malignant 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.

(c) CORRELATION PLOT

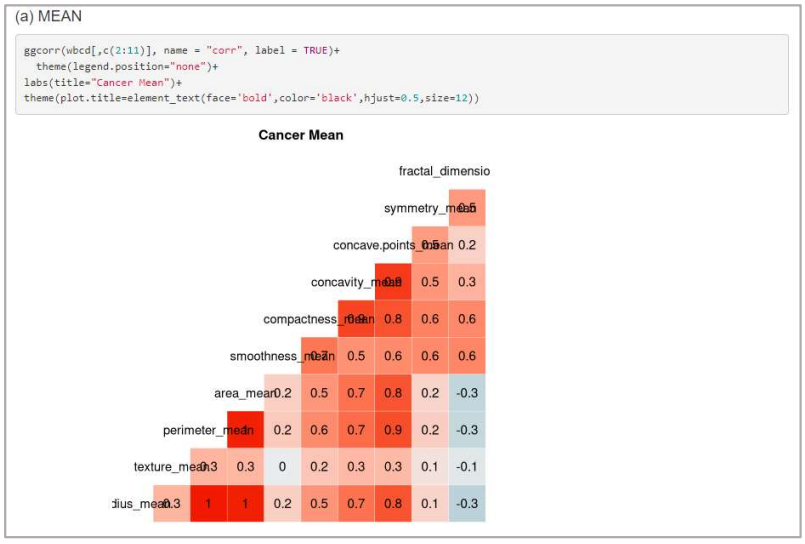
```
correlationMatrix <- cor(wbcd[,3:ncol(wbcd)])
corrplot(correlationMatrix, order = "hclust", tl.cex = 1, addrect = 8)
```

As shown by this plot, many variables are highly correlated with each others.



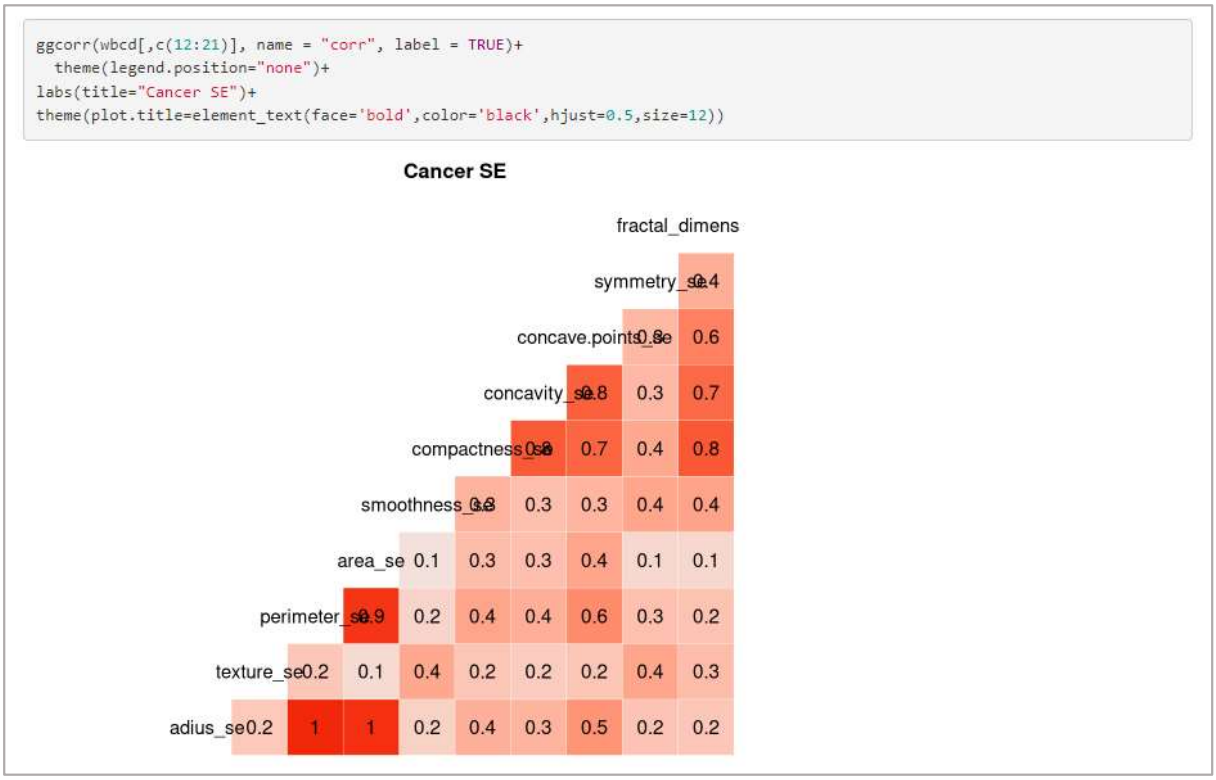
(i) Correlation Between Different Variables Using ggcorr Funtion

- Correlation plot of Mean category: perimeter mean and area mean have high correlation



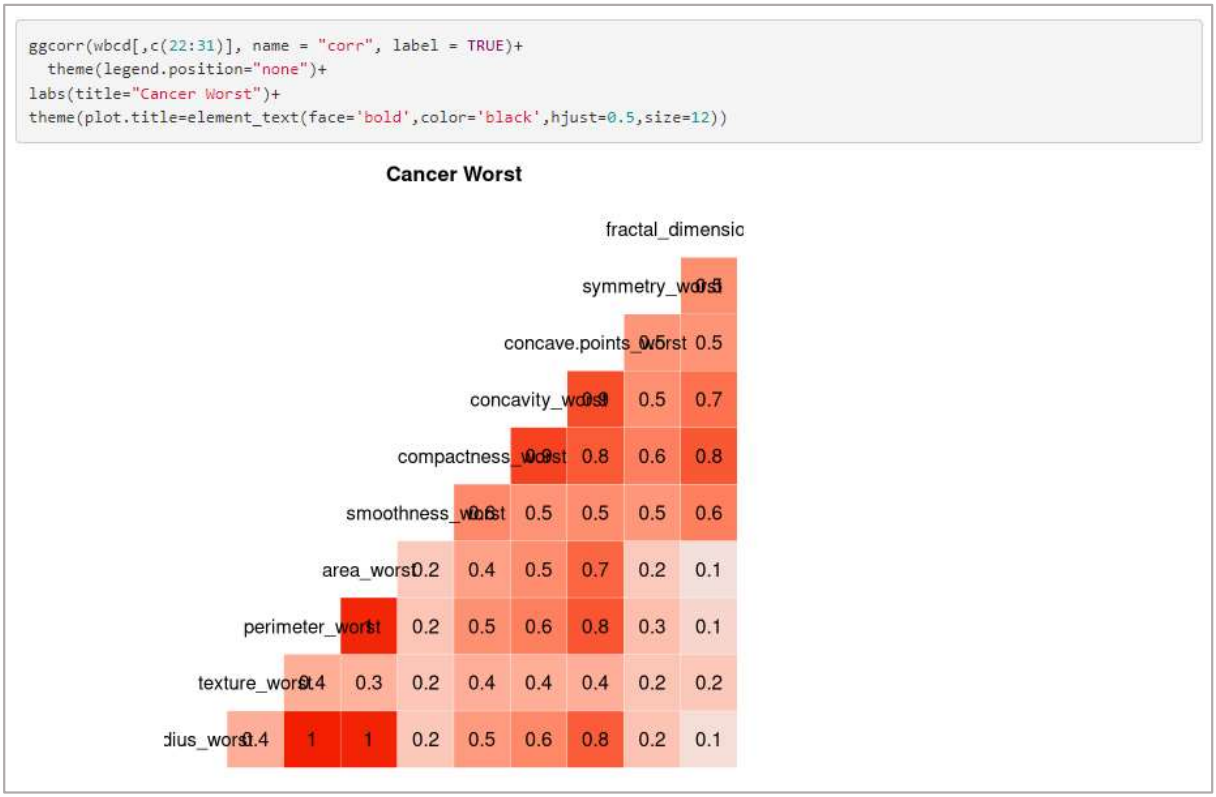
- Correlation plot of Standard Error category: perimeter SE and area SE have high correlation

```
ggcorr(wbcd[,c(12:21)], name = "corr", label = TRUE)+
  theme(legend.position="none")+
labs(title="Cancer SE")+
theme(plot.title=element_text(face='bold',color='black',hjust=0.5,size=12))
```



- Correlation plot of Worst category: perimeter worst and area worst have high correlation

```
ggcorr(wbcd[,c(22:31)], name = "corr", label = TRUE)+
  theme(legend.position="none")+
labs(title="Cancer Worst")+
theme(plot.title=element_text(face='bold',color='black',hjust=0.5,size=12))
```



3.4 Principal Component Analysis (PCA)

To avoid redundancy and relevancy, we used the function 'prcomp' to calculate the Principal Component Analysis (PCA) and select the rights components to avoid correlated variables that can be detrimental to our clustering analysis. One of the common problems in analysis of complex data comes from a large number of variables, which requires a large amount of memory and computation power. This is where PCA comes in. It is a technique to reduce the dimension of the feature space by feature extraction. The main idea of PCA is to reduce the dimensionality of a data set consisting of many variables correlated with each other, either heavily or lightly, while retaining the variation present in the dataset, up to the maximum extent.

The cumulative proportion from PC1 to PC6 is about 88.7%. (above 85%) It means that PC1~PC6 can explain 88.7% of the whole data.

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

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

```
summary(all_pca)
```

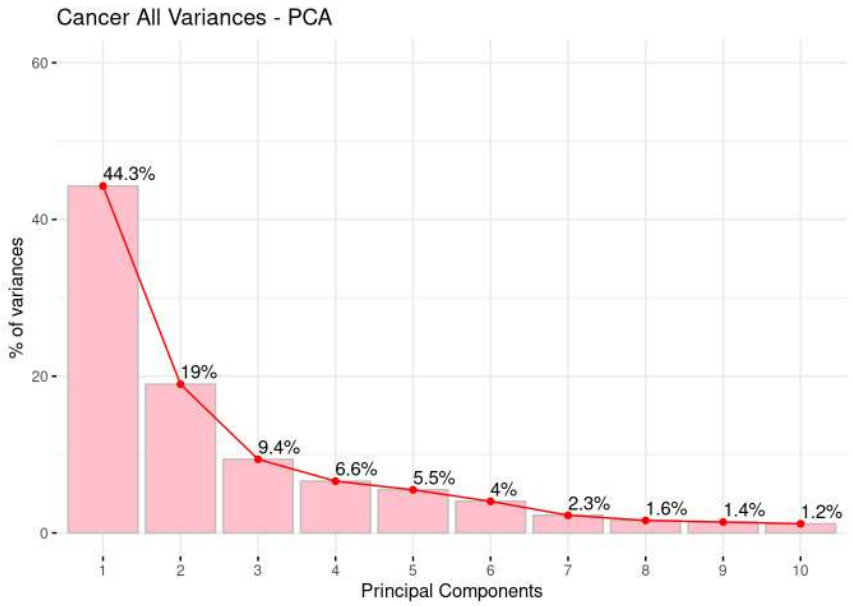
```
## Importance of components:
##                           PC1    PC2     PC3     PC4     PC5     PC6     PC7
## Standard deviation     3.6444 2.3857 1.67867 1.40735 1.28403 1.09880 0.82172
## Proportion of Variance 0.4427 0.1897 0.09393 0.06602 0.05496 0.04025 0.02251
## Cumulative Proportion  0.4427 0.6324 0.72636 0.79239 0.84734 0.88759 0.91010
##                           PC8    PC9    PC10   PC11    PC12    PC13    PC14
## Standard deviation     0.69037 0.6457 0.59219 0.5421 0.51104 0.49128 0.39624
## Proportion of Variance 0.01589 0.0139 0.01169 0.0098 0.00871 0.00805 0.00523
## Cumulative Proportion  0.92598 0.9399 0.95157 0.9614 0.97007 0.97812 0.98335
##                           PC15    PC16    PC17    PC18    PC19    PC20   PC21
## Standard deviation     0.30681 0.28260 0.24372 0.22939 0.22244 0.17652 0.1731
## Proportion of Variance 0.00314 0.00266 0.00198 0.00175 0.00165 0.00104 0.0010
## Cumulative Proportion  0.98649 0.98915 0.99113 0.99288 0.99453 0.99557 0.9966
##                           PC22    PC23   PC24    PC25    PC26    PC27    PC28
## Standard deviation     0.16565 0.15602 0.1344 0.12442 0.09043 0.08307 0.03987
## Proportion of Variance 0.00091 0.00081 0.0006 0.00052 0.00027 0.00023 0.00005
## Cumulative Proportion  0.99749 0.99830 0.9989 0.99942 0.99969 0.99992 0.99997
##                           PC29    PC30
## Standard deviation     0.02736 0.01153
## Proportion of Variance 0.00002 0.00000
## Cumulative Proportion  1.00000 1.00000
```

In case of Mean category, the cumulative proportion from PC1 to PC3 is about 88.7%., Standard Error, the cumulative proportion from PC1 to PC4 is about 86.7%. and Worst, the cumulative proportion from PC1 to PC3 is about 85.8%.

(a) ScreePlot Representation of whole data

The percentage of variability explained by the principal components can be ascertained through screeplot. In case of taking screeplot of whole data we see that the line lies at point PC6

```
fviz_eig(all_pca, addlabels=TRUE, ylim=c(0,60), geom = c("bar", "line"), barfill = "pink", barcolor="grey",linecolor = "red", ncp=10)+
labs(title = "Cancer All Variances - PCA",
         x = "Principal Components", y = "% of variances")
```

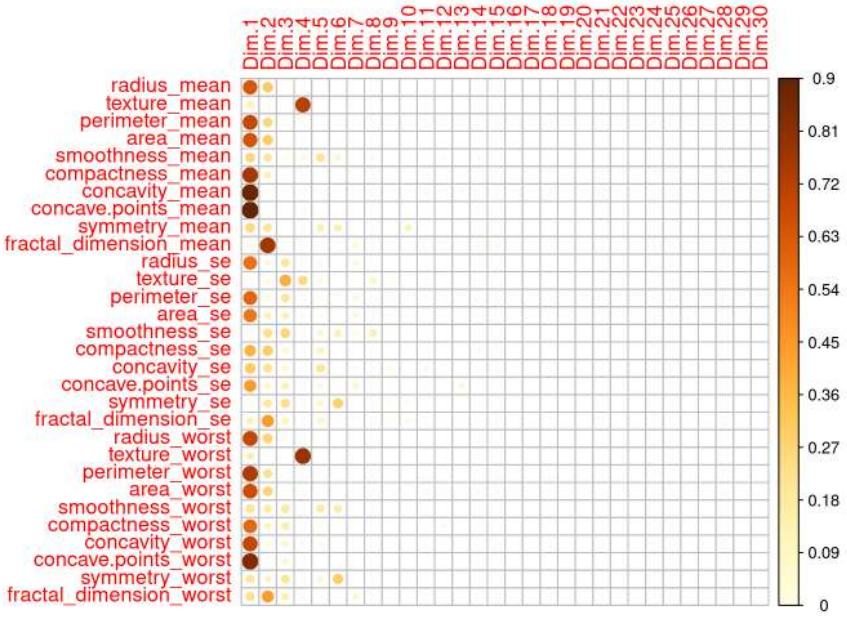


- Correlation between variables and PCA

```
library("corrplot")
```

```
## corrplot 0.92 loaded
```

```
corrplot(all_var$cos2, is.corr=FALSE)
```



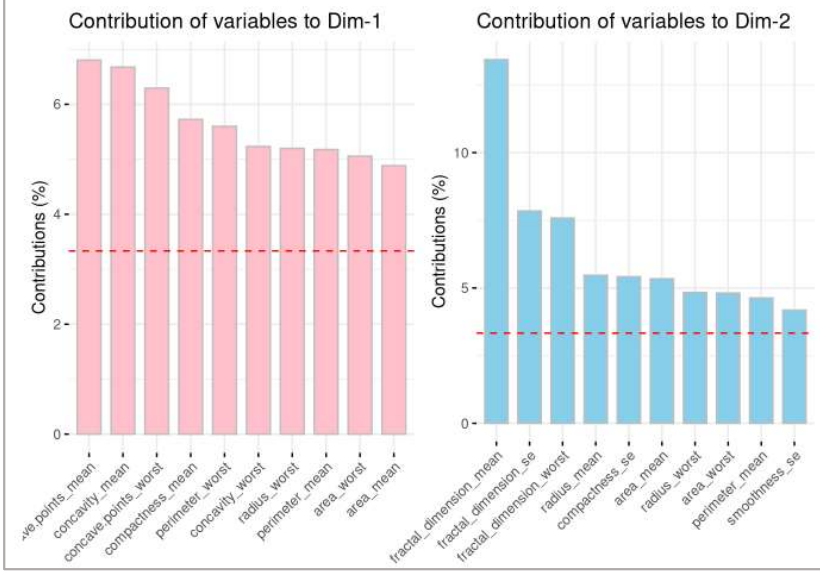
- Correlation of variables to PC1 & PC2

```
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'
```

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

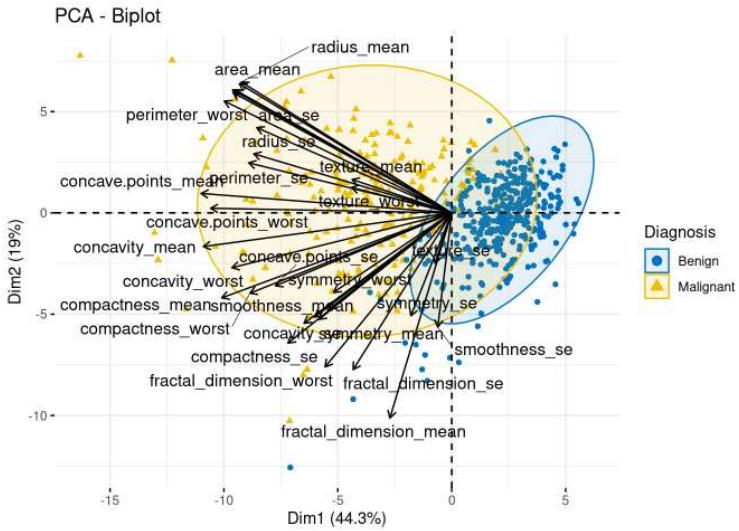
```
p1 <- fviz_contrib(all_pca, choice="var", axes=1, fill="pink", color="grey", top=10)
p2 <- fviz_contrib(all_pca, choice="var", axes=2, fill="skyblue", color="grey", top=10)
grid.arrange(p1,p2,ncol=2)
```



- Biplot of all pca

```
fviz_pca_biplot(all_pca, col.ind = wbcd$diagnosis, col="black",
                palette = "jco", geom = "point", repel=TRUE,
                legend.title="Diagnosis", addEllipses = TRUE)
```

```
## Warning: ggrepel: 4 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```



3.5 MODEL BUILDING

- We classified the dataset into train set (70%) and test set (30%)

```
nrows <- NROW(wbcd)
set.seed(218)                          ## fix random value
index <- sample(1:nrows, 0.7 * nrows)   ## shuffle and divide

#train <- wbcd                         ## 569 test data (100%)
train <- wbcd[index,]                  ## 398 test data (70%)
test <- wbcd[-index,]                  ## 171 test data (30%)

prop.table(table(train$diagnosis)) #proportion of diagnosis (Benign / Malignant)
```

```
##
##    Benign Malignant
## 0.6180905 0.3819095
```

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

```
##
##    Benign Malignant
## 0.6491228 0.3508772
```

APPLYING ML MODELS:

(1) NAÏVE BAYES

Naïve Bayes Classifier is one of the simple and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions. It is a probabilistic classifier, which means it predicts on the basis of the probability of an object.

```
library(e1071)

learn_nb <- naiveBayes(train[,-1], train$diagnosis)
pre_nb <- predict(learn_nb, test[,-1])
cm_nb <- confusionMatrix(pre_nb, test$diagnosis)
cm_nb
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  Benign Malignant
##   Benign       107         6
##   Malignant      4        54
##
##               Accuracy : 0.9415
##                 95% CI : (0.8951, 0.9716)
##    No Information Rate : 0.6491
##    P-Value [Acc > NIR] : <2e-16
##
##                  Kappa : 0.8706
##
##  Mcnemar's Test P-Value : 0.7518
##
##             Sensitivity : 0.9640
##             Specificity : 0.9000
##          Pos Pred Value : 0.9469
##          Neg Pred Value : 0.9310
##              Prevalence : 0.6491
##          Detection Rate : 0.6257
##    Detection Prevalence : 0.6608
##       Balanced Accuracy : 0.9320
##
##        'Positive' Class : Benign
##
```

Inference: In Naïve Bayes modelling we got an accuracy, sensitivity, specificity, positive prediction value and negative prediction value as 94.15%, 96.4%, 90.00%, 94.69% and 93.2% respectively.

(2) RECURSIVE PARTITIONING AND REGRESSION TREES (RPart)

Rpart is a powerful machine learning library in R that is used for building classification and regression trees.

```
library(rpart)
learn_rp <- rpart(diagnosis~.,data=train,control=rpart.control(minsplit=2))
pre_rp <- predict(learn_rp, test[,-1], type="class")
cm_rp  <- confusionMatrix(pre_rp, test$diagnosis)
cm_rp
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  Benign Malignant
##   Benign       108         5
##   Malignant      3        55
##
##               Accuracy : 0.9532
##                 95% CI : (0.9099, 0.9796)
##    No Information Rate : 0.6491
##    P-Value [Acc > NIR] : <2e-16
##
##                  Kappa : 0.8965
##
##  Mcnemar's Test P-Value : 0.7237
##
##             Sensitivity : 0.9730
##             Specificity : 0.9167
##          Pos Pred Value : 0.9558
##          Neg Pred Value : 0.9483
##              Prevalence : 0.6491
##          Detection Rate : 0.6316
##    Detection Prevalence : 0.6608
##       Balanced Accuracy : 0.9448
##
##        'Positive' Class : Benign
##
```

Inference: Using RPart modelling we got an accuracy, sensitivity, specificity, positive prediction value and negative prediction value as 95.32%, 97.30%, 91.67%, 95.58% and 94.83% respectively.

(3) RANDOM FOREST

Random forests or random decision forests is an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time. For classification tasks, the output of the random forest is the class selected by most trees. For regression tasks, the mean or average prediction of the individual trees is returned.

```
learn_rf <- randomForest(diagnosis~., data=train, ntree=500, proximity=T, importance=T)
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       111         4
##   Malignant      0        56
##
##               Accuracy : 0.9766
##                 95% CI : (0.9412, 0.9936)
##    No Information Rate : 0.6491
##    P-Value [Acc > NIR] : <2e-16
##
##                  Kappa : 0.9478
##
##  Mcnemar's Test P-Value : 0.1336
##
##             Sensitivity : 1.0000
##             Specificity : 0.9333
##          Pos Pred Value : 0.9652
##          Neg Pred Value : 1.0000
##              Prevalence : 0.6491
##          Detection Rate : 0.6491
##    Detection Prevalence : 0.6725
##       Balanced Accuracy : 0.9667
##
##        'Positive' Class : Benign
##
```

Inference: In Random Forest modelling we got an accuracy, sensitivity, specificity, positive prediction value and negative prediction value as 97.66%, 100%, 93.33%, 96.52%and 100% respectively.

(4) ADABOOST

AdaBoost also called Adaptive Boosting is a technique in Machine Learning used as an Ensemble Method. The most common algorithm used with AdaBoost is decision trees with one level that means with Decision trees with only 1 split. These trees are also called Decision Stumps. It builds a model and gives equal weights to all the data points. It then assigns higher weights to points that are wrongly classified. Now all the points which have higher weights are given more importance in the next model. It will keep training models until and unless a lower error is received.

Inference: In AdaBoost modelling we got an accuracy, sensitivity, specificity, positive prediction value and negative prediction value as 98.25%, 99.10%, 96.67%, 98.21%and 98.31% respectively.

```
library(rpart)
library(ada)
control <- rpart.control(cp = -1, maxdepth = 14,maxcompete = 1,xval = 0)
learn_ada <- ada(diagnosis~., data = train, test.x = train[,-1], test.y = train[,1], type = "gentle", control = control, ite
r = 70)
pre_ada <- predict(learn_ada, test[,-1])
cm_ada <- confusionMatrix(pre_ada, test$diagnosis)
cm_ada
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  Benign Malignant
##   Benign       110         2
##   Malignant      1        58
##
##               Accuracy : 0.9825
##                 95% CI : (0.9496, 0.9964)
##    No Information Rate : 0.6491
##    P-Value [Acc > NIR] : <2e-16
##
##                  Kappa : 0.9613
##
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 0.9910
##             Specificity : 0.9667
##          Pos Pred Value : 0.9821
##          Neg Pred Value : 0.9831
##              Prevalence : 0.6491
##          Detection Rate : 0.6433
##    Detection Prevalence : 0.6550
##       Balanced Accuracy : 0.9788
##
##        'Positive' Class : Benign
##
```

(5) SUPPORT VECTOR MACHINE

```
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       109         1
##   Malignant      2        59
##
##               Accuracy : 0.9825
##                 95% CI : (0.9496, 0.9964)
##    No Information Rate : 0.6491
##    P-Value [Acc > NIR] : <2e-16
##
##                  Kappa : 0.9616
##
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 0.9820
##             Specificity : 0.9833
##          Pos Pred Value : 0.9909
##          Neg Pred Value : 0.9672
##              Prevalence : 0.6491
##          Detection Rate : 0.6374
##    Detection Prevalence : 0.6433
##       Balanced Accuracy : 0.9827
##
##        'Positive' Class : Benign
##
```

Support Vector Machine (SVM) is a supervised machine learning algorithm used for both classification and regression. Though we say regression problems as well its best suited for classification. The objective of SVM algorithm is to find a hyperplane in an N-dimensional space that distinctly classifies the data points.

Inference: In SVM modelling we got an accuracy, sensitivity, specificity, positive prediction value and negative prediction value as 98.25%, 98.20%, 98.33%, 99.09%and 96.72% respectively.

(6) SUPPORT VECTOR MACHINE after TUNING

In SVM Tune, we choose 'gamma, cost' which shows best predict performance in SVM.

```
gamma <- seq(0,0.1,0.005)
cost <- 2^(0:5)
parms <- expand.grid(cost=cost, gamma=gamma)    ## 231

acc_test <- numeric()
accuracy1 <- NULL; accuracy2 <- NULL

for(i in 1:NROW(parms)){
        learn_svm <- svm(diagnosis~., data=train, gamma=parms$gamma[i], cost=parms$cost[i])
        pre_svm <- predict(learn_svm, test[,-1])
        accuracy1 <- confusionMatrix(pre_svm, test$diagnosis)
        accuracy2[i] <- accuracy1$overall[1]
}

acc <- data.frame(p= seq(1,NROW(parms)), cnt = accuracy2)

opt_p <- subset(acc, cnt==max(cnt))[1,]
sub <- paste("Optimal number of parameter is", opt_p$p, "(accuracy :", opt_p$cnt,") in SVM")

library(highcharter)
```

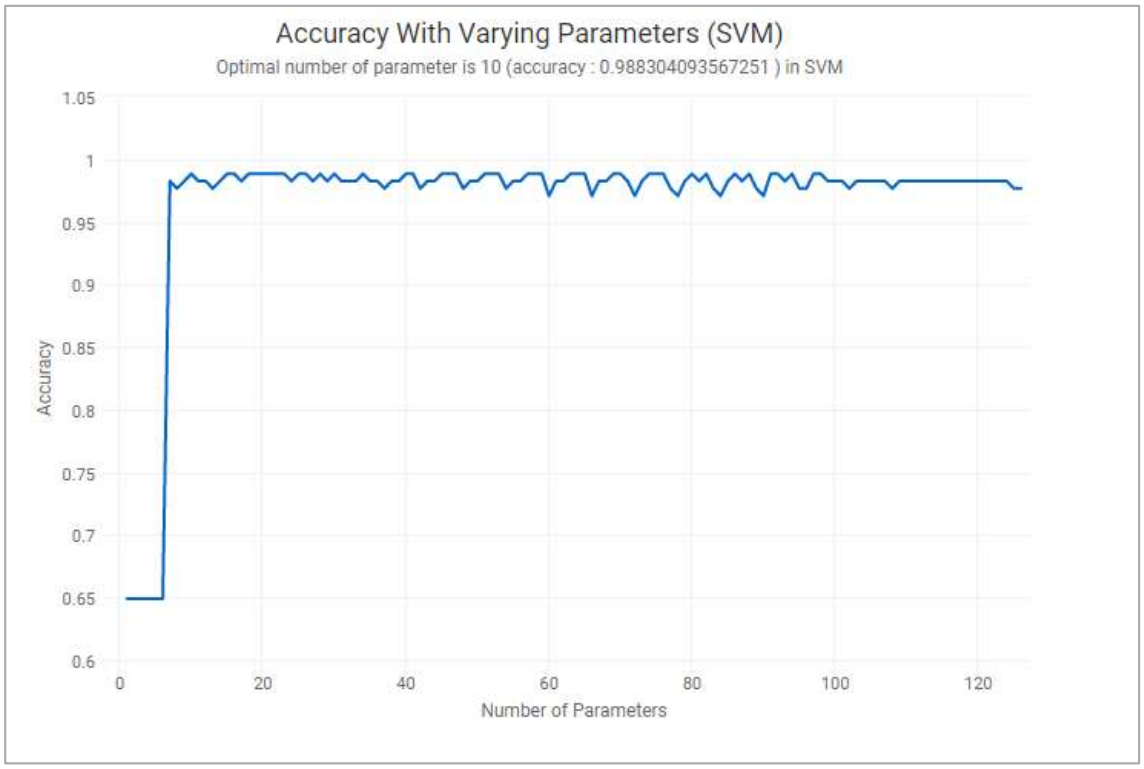
```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
## Highcharts (www.highcharts.com) is a Highsoft software product which is
```

```
## not free for commercial and Governmental use
```

```
hchart(acc, 'line', hcaes(p, cnt)) %>%
  hc_title(text = "Accuracy With Varying Parameters (SVM)") %>%
  hc_subtitle(text = sub) %>%
  hc_add_theme(hc_theme_google()) %>%
  hc_xAxis(title = list(text = "Number of Parameters")) %>%
  hc_yAxis(title = list(text = "Accuracy"))
```

Kernel method enables SVM to adapt to patterns of data, by nonlinearly mapping the data from original space into a higher dimensional space. As one of broadly used kernel, radial basis function (RBF) is utilized to enhance SVM flexibility and robustness to fit to the given data distritbuion. However, mixture use of SVM and RBF increases technical difficulty for data scientists to figure out optimal parameters (Gamma and C) and then to come up with optimized models.



Apply optimal parameters(gamma, cost) to show best predict performance in SVM

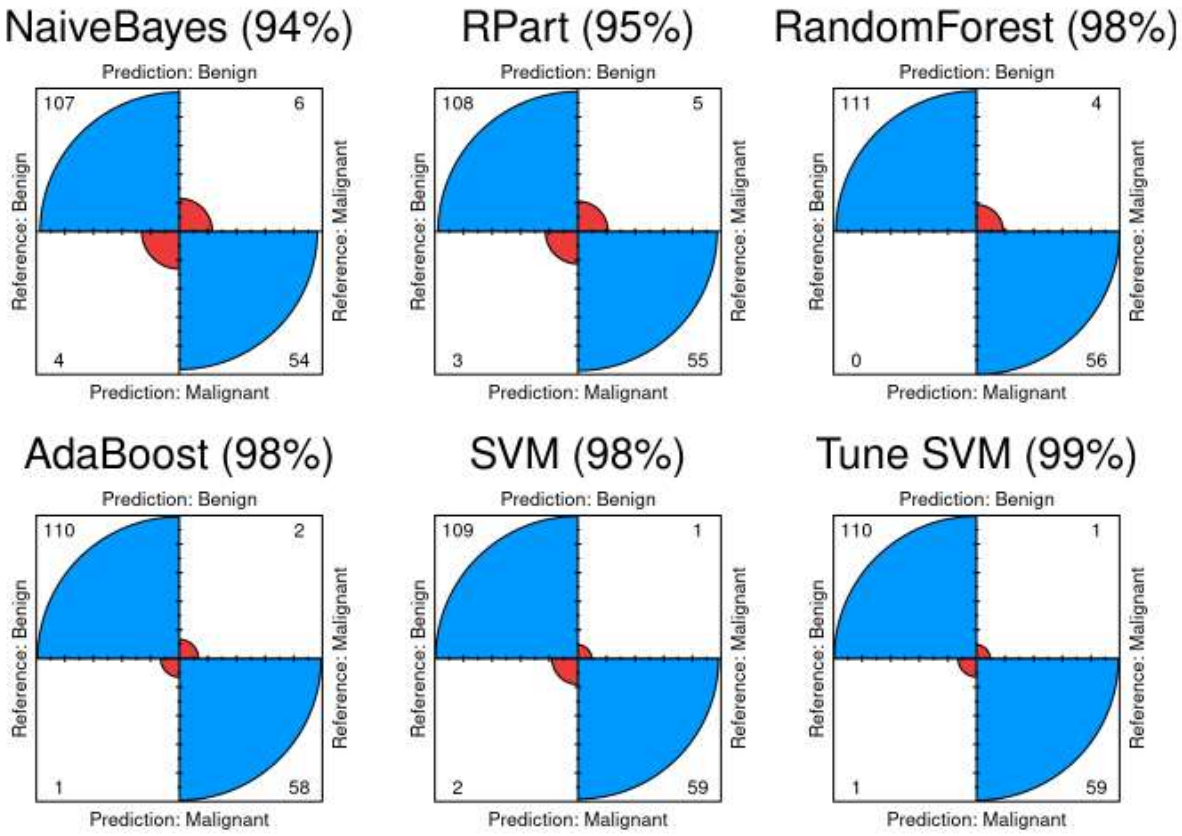
```
learn_imp_svm <- svm(diagnosis~., data=train, cost=parms$cost[opt_p$p], gamma=parms$gamma[opt_p$p])
pre_imp_svm <- predict(learn_imp_svm, test[,-1])
cm_imp_svm <- confusionMatrix(pre_imp_svm, test$diagnosis)
cm_imp_svm
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  Benign Malignant
##   Benign       110         1
##   Malignant      1        59
##
##                Accuracy : 0.9883
##                  95% CI : (0.9584, 0.9986)
##     No Information Rate : 0.6491
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.9743
##
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 0.9910
##             Specificity : 0.9833
##          Pos Pred Value : 0.9910
##          Neg Pred Value : 0.9833
##              Prevalence : 0.6491
##          Detection Rate : 0.6433
##    Detection Prevalence : 0.6491
##       Balanced Accuracy : 0.9872
##
##        'Positive' Class : Benign
##
```

Inference: In SVM tune we got a high accuracy, sensitivity, specificity, positive prediction value and negative prediction value as 98.83%, 99.10%, 98.33%, 99.10%and 98.33% respectively.

COMPARISION OF DIFFERNET MODELS

```
col <- c("#ed3b3b", "#0099ff")
par(mfrow=c(2,3))
fourfoldplot(cm_nb$table, color = col, conf.level = 0, margin = 1, main=paste("NaiveBayes (",round(cm_nb$overall[1]*10
0),"%)",sep=""))
fourfoldplot(cm_rp$table, color = col, conf.level = 0, margin = 1, main=paste("RPart (",round(cm_rp$overall[1]*100),"%)",sep
=""))
fourfoldplot(cm_rf$table, color = col, conf.level = 0, margin = 1, main=paste("RandomForest (",round(cm_rf$overall[1]*10
0),"%)",sep=""))
fourfoldplot(cm_ada$table, color = col, conf.level = 0, margin = 1, main=paste("AdaBoost (",round(cm_ada$overall[1]*10
0),"%)",sep=""))
fourfoldplot(cm_svm$table, color = col, conf.level = 0, margin = 1, main=paste("SVM (",round(cm_svm$overall[1]*100),"%)",sep
=""))
fourfoldplot(cm_imp_svm$table, color = col, conf.level = 0, margin = 1, main=paste("Tune SVM (",round(cm_imp_svm$overall[1]*
100),"%)",sep=""))
```



3.6 RESULT

Considering all 6 modelling techniques Tuned SVM had a high accuracy of 98.83%, Specificity of 98.33% and Sensitivity of 99.10%. Sensitivity refers to a test's ability to designate an individual with disease as positive. A highly sensitive test means that there are few false negative results, and thus fewer cases of disease are missed. The specificity of a test is its ability to designate an individual who does not have a disease as negative.

CHAPTER – 4
CONCLUSION

In conclusion, comparing all the six traditional models achieved a higher accuracy of around 98.83% for breast cancer diagnosis was found by tuning SVM model. Among all traditional models, the random forest model provided the best sensitivity rate (100%) to our dataset. In order to improve the performance, we experiment with feature selection using correlation matrix and feature importance. It results in better accuracies when removing certain features for certain model. We have performed Principal component Analysis on our data and tried to find the attributes having high interference on our data.

Future work includes tuning the hyperparameters of the current model as well as testing other deep learning method/architectures to increase model accuracy. Overall, we presented machine learning models that can be applied in breast cancer diagnosis to improve the accuracy and therefore assist early diagnosis of breast cancer.

CHAPTER – 5
REFERENCES

[1] Al Bataineh, Ali. "A comparative analysis of nonlinear machine learning algorithms for breast cancer detection." *International Journal of Machine Learning and Computing* 9, no. 3 (2019): 248-254.

[2] Huang, Dong, Chang-Dong Wang, Jian-Sheng Wu, Jian-Huang Lai, and Chee-Keong Kwoh. "Ultra-scalable spectral clustering and ensemble clustering." *IEEE Transactions on Knowledge and Data Engineering* 32, no. 6 (2019): 1212-1226.

[3] Chauhan, Pragya, and Amit Swami. "Breast cancer prediction using genetic algorithm based ensemble approach." In *2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, pp. 1-8. IEEE, 2018.

[4] Hwa, Hsiao-Lin, Wen-Hong Kuo, Li-Yun Chang, Ming-Yang Wang, Tao-Hsin Tung, King-Jen Chang, and Fon-Jou Hsieh. "Prediction of breast cancer and lymph node metastatic status with tumour markers using logistic regression models." *Journal of evaluation in clinical practice* 14, no. 2 (2008): 275-280.