

Pattern Recognition in Breast Cancer Using Machine Learning

Name: M.SARAVANAKUMAR

AUTHOR Name: M.SARAVANAKUMAR MSC(CS)

Designation of 1st Author: THIRUTHANGAL-626130

SUPERVISOR NAME :Dr.S.KANNAN MCA.,PhD

Name of Department of : COMPUTER SCIENCE

Name of organization: SCHOOL OF INFORMATION TECHNOLOGY

City: SIVAKASI

Country: INDIA

ABSTRACT

Breast Cancer is the most often identified cancer among women and major reason for increasing mortality rate among women. As the diagnosis of this disease manually takes long hours and the lesser availability of systems, there is a need to develop the automatic diagnosis system for early detection of cancer. Data mining techniques contribute a lot in the development of such system. For the classification of benign and malignant tumor we have used classification techniques of machine learning in which the machine is learned from the past data and can predict the category of new input. This paper is a relative study on the implementation of models using Logistic Regression, Support Vector Machine (SVM) and Latent Dirichlet Allocation (LDA) is done on the dataset taken from the UCI repository. With respect to the results of accuracy, precision, sensitivity, specificity and False Positive Rate the efficiency of each algorithm is measured and compared. Breast Cancer is the most leading malignancy affecting 2.1 million women each year which leads to greatest number of deaths among women. Early treatment not only helps to cure cancer but also helps in prevention of its recurrence. And hence this system mainly focuses on prediction of breast cancer where it uses different machine learning algorithms for creating models like decision tree, logistic regression, random forest which are applied on pre-processed data which suspects greater accuracy for prediction. Amongst all the models, Random Forest Classification leads to best accuracy with 98.6% techniques

1.INTRODUCTION

1.1 About the Project

According to World health organization, Breast cancer is the most frequent cancer among women and it is the second dangerous cancer after lung cancer. In 2018, from the research it is estimated that total 627,000 women lost their life due to breast cancer that is 15% of all cancer deaths among women. In case of any symptom, people visit to oncologist. Doctors can easily identify breast cancer by using Breast ultrasound, Diagnostic mammogram, Magnetic resonance imaging (MRI), Biopsy. Based on these test results, doctor may recommend further tests or therapy. Early detection is very crucial in breast cancer. If chances of cancer are predicted at early stage then survivability chances of patient may increase. An alternate way to identify breast cancer is using machine learning algorithms for prediction of abnormal tumor. Thus, the research is carried out for the proper diagnosis and categorization of patients into malignant and benign groups.

Three types of tumors are as follows:-

- a. Benign tumors are not cancerous they cannot spread or they can grow very slowly. And if doctors remove them, then they cannot cause any harm to the human body.
- b. In Premalignant tumors the cells are not cancerous but they have potential to become malignant.
- c. Malignant cells are cancerous and they can spread rapidly in body.

In machine learning, cancer classification can be done using benign or malignant cells could significantly improve our ability for prognosis in cancer patients, poor progress has been made for their application in the clinics. However, before gene expression profiling can be used in clinical practice, studies with larger data samples and more adequate validation are needed. And thus, our aim is to develop a prediction system that can predict chances of breast cancer on a huge data.

2.SYSTEM REQUIREMENT

2.1 Hardware Requirement

Processor	:	Dual Core Processor
RAM	:	2GB
Hard Disk	:	160GB
Keyboard	:	Logitech Multimedia Keyboard
Mouse	:	Logitech Scroll Mouse
Monitor	:	15” Color Monitor

2.2 Software requirement

Front End	:	R
Operation System	:	Windows 10 Ultimate
Word Processor	:	Ms Office 2007

2.3 About the Software

Introduction to R

R is a language and environment for statistical computing and graphics. It is a GNU project which is similar to the S language and environment which was developed at Bell Laboratories (formerly AT&T, now Lucent Technologies) by John Chambers and colleagues. R can be considered as a different implementation of S. There are some important differences, but much code written for S runs unaltered under R.

R provides a wide variety of statistical (linear and nonlinear modelling, classical statistical tests, time-series analysis, classification, clustering, ...) and graphical techniques, and is highly extensible. The S language is often the vehicle of choice for research in statistical methodology, and R provides an Open Source route to participation in that activity.

One of R's strengths is the ease with which well-designed publication-quality plots can be produced, including mathematical symbols and formulae where needed. Great care has been taken over the defaults for the minor design choices in graphics, but the user retains full control.

R is available as Free Software under the terms of the Free Software Foundation's GNU General Public License in source code form. It compiles and runs on a wide variety of UNIX platforms and similar systems (including FreeBSD and Linux), Windows and MacOS.

The R environment

R is an integrated suite of software facilities for data manipulation, calculation and graphical display. It includes

- an effective data handling and storage facility,
- a suite of operators for calculations on arrays, in particular matrices,
- a large, coherent, integrated collection of intermediate tools for data analysis,
- graphical facilities for data analysis and display either on-screen or on hardcopy, and
- a well-developed, simple and effective programming language which includes conditionals, loops, user-defined recursive functions and input and output facilities.

The term "environment" is intended to characterize it as a fully planned and coherent system, rather than an incremental accretion of very specific and inflexible tools, as is frequently the case with other data analysis software.

R, like S, is designed around a true computer language, and it allows users to add additional functionality by defining new functions. Much of the system is itself written in the R dialect of S, which makes it easy for users to follow the algorithmic choices made. For computationally-intensive tasks, C, C++ and Fortran code can be linked and called at run time. Advanced users can write C code to manipulate R objects directly.

Many users think of R as a statistics system. Users prefer to think of it of an environment within which statistical techniques are implemented. R can be extended (easily) via packages. There are about eight packages supplied with the R distribution and many more are available through the CRAN family of Internet sites covering a very wide range of modern statistics.

R has its own LaTeX-like documentation format, which is used to supply comprehensive documentation, both on-line in a number of formats and in hardcopy.

History

R is an implementation of the S programming language combined with lexical scoping semantics, inspired by Scheme. S was created by John Chambers in 1976, while at Bell Labs. There are some important differences, but much of the code written for S runs unaltered.

R was created by Ross Ihaka and Robert Gentleman at the University of Auckland, New Zealand, and currently developed by the R Development Core Team (of which Chambers is a member). R is named partly after the first names of the first two R authors and partly as a play on the name of S. The project was conceived in 1992, with an initial version released in 1995 and a stable beta version in 2000.

Statistical features

R and its libraries implement a wide variety of statistical and graphical techniques, including linear and nonlinear modelling, classical statistical tests, time-series analysis, classification, clustering, and others. R is easily extensible through functions and extensions, and the R community is noted for its active contributions in terms of packages. Many of R's standard functions are written in R itself, which makes it easy for users to follow the algorithmic choices made. For computationally intensive tasks, C, C++, and Fortran code can be linked and called at run time. Advanced users can write C, C++, Java, .NET or Python code to manipulate R objects directly. R is highly extensible through the use of user-submitted packages for specific functions or specific areas of study. Due to its S heritage, R has stronger object-oriented programming facilities than most statistical computing languages. Extending R is also eased by its lexical scoping rules.

Another strength of R is static graphics, which can produce publication-quality graphs, including mathematical symbols. Dynamic and interactive graphics are available through additional packages.

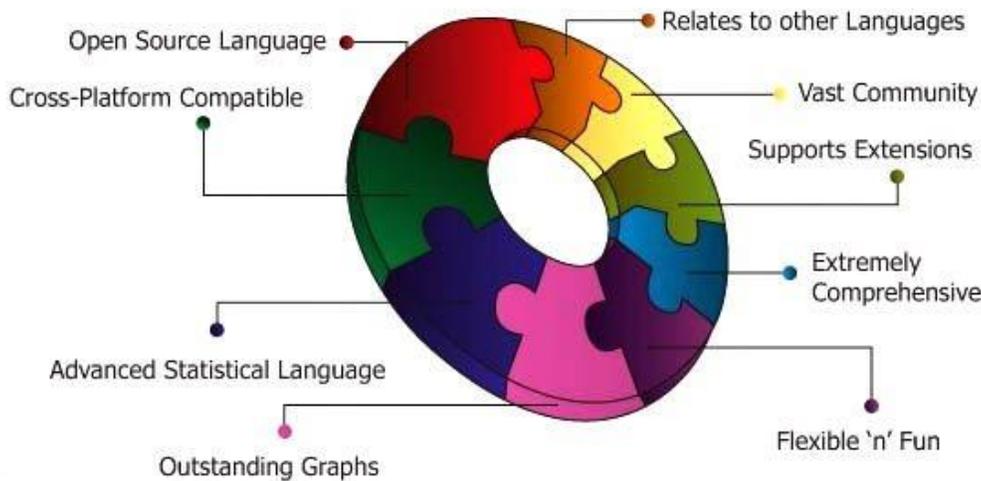
R has Rd, its own LaTeX-like documentation format, which is used to supply comprehensive documentation, both online in a number of formats and in hard copy.

Advantage of R Tool

There are a number of advantages of this data analytics tool, which make it so very popular amongst Data Scientists. Firstly, the fact that it is by far the most comprehensive statistical analysis package

available totally works in its favor. This tool strives to incorporate all of the standard statistical tests, models and analyses as well as provides for an effective language so as to manage and manipulate data.

One of the biggest advantages of this tool is the fact that it is entirely open sourced. This means that it can be downloaded very easily and is free of cost. This is mainly the reason why there are also communities, which strive to develop the various aspects of this tool. Currently, there are about some 19 developers, including practising professionals from the IT industry, who help in tweaking out this software. This is also the reason why most of the latest technological developments, are first to arrive on this software before they are seen anywhere else.



R Programming

When it comes to graphical representation, the related attributed to R are extremely exemplary. This is the reason why it is able to surpass most of the other statistical and graphical packages with great ease. The fact that it has no licencerestrictions, makes it literally the go-to software, for all of those who want to practice this in the earlier stages. It has over 4800 packages available, in its environment which belongto various repositories with specialization in various topics like econometrics, data mining, spatial analysis and bioinformatics.

The best part about R programming is that it is more of a user run software, which means that anyone is allowed to provide code enhancements and new packages. The quality of great packages on the R community environment is a testament to this very approach to developing a certain software by sharing and encouraging inputs. This tool is also compatible across platforms and thereby it runs on many operating systems as well as hardware. It can function with similar clarity for both the Linux as well as Microsoft Windows Operating

Systems. In addition to this, the fact that R can also work well with other data analytics tools like SAS, SPSS and MySQL, have resulted in a number of takers for this data analytics tool. Imarticus Learning The Data Science Prodegree powered by Genpact is one such course which offers both SAS and R along with the opportunity to be a Data Scientist at Genpact.

Purpose of using R

R is used by the best data scientists in the world. In surveys on Kaggle (the competitive machine learning platform), R is by far the most used machine learning tool. When professional machine learning practitioners were surveyed in 2015, again the most popular machine learning tool was R.

R is powerful because of the breadth of techniques it offers. Any techniques that user can think of for data analysis, visualization, sampling, supervised learning and model evaluation are provided in R. The platform has more techniques than any other that user will come across.

R is state-of-the-art because it is used by academics. One of the reasons why R has so many techniques is because academics that develop new algorithms are developing them in R and releasing them as R packages. This means that user can get access to state-of-the-art algorithms in R before other platforms. It also means that user can only access some algorithms in R until someone ports them to other platforms.

R is free because it is open source software. User can download it right now for free and it runs on any workstation platform user are likely to use.

R in Machine Learning

This introductory workshop on machine learning with R is aimed at participants who are not experts in machine learning (introductory material will be presented as part of the course), but have some familiarity with scripting in general and R in particular. The workshop will offer a hands-on overview of typical machine learning applications in R, including unsupervised (clustering, such as hierarchical and k-means clustering, and dimensionality reduction, such as principal component analysis) and supervised methods (classification and regression, such as k-nearest neighbour and linear regression). Users will also address questions such as model selection using cross-validation. The material has an important hands-on component and readers should have a computer running R 3.4.1 or later.

Overview of Machine Learning

In supervised learning (SML), the learning algorithm is presented with labelled example inputs, where the labels indicate the desired output. SML itself is composed of classification, where the output is categorical, and regression, where the output is numerical. In unsupervised learning (UML), no labels are provided, and the learning algorithm focuses solely on detecting structure in unlabelled input data.

Note that there are also semi-supervised learning approaches that use labelled data to inform unsupervised learning on the unlabelled data to identify and annotate new classes in the dataset (also called novelty detection).

Reinforcement learning, the learning algorithm performs a task using feedback from operating in a real or synthetic environment.

R Machine Learning Packages

R is an open-source language so people can contribute from anywhere in the world. User can use a Black Box in your code, which is written by someone else. In R, this Black Box is referred to as a package. The package is nothing but a pre-written code that can be used repeatedly by anyone.

1. CARAT

The package CARAT refers to classification and regression training. The task of this CARAT package is to integrate the training and prediction of a model. It is one of the best packages of R for machine learning as well as data science. The parameters can be searched by the integration of several functions to calculate the overall performance of a given model by using the grid search method of this package. After successful completion of all trials, the grid search finally finds the best combinations.

After installation of this package, the developer can run names (`getModelInfo()`) to see the 217 possible functions which can be run through only one function. For building a predictive model, the CARAT package uses a `train()` function. The syntax of this function:

```
train(formula, data, method)
```

2. RandomForest

RandomForest is one of the most popular R packages for machine learning. This R machine learning package can be employed for solving regression and classification tasks. Additionally, it can be used for training missing

values and outliers. This machine learning package with R generally is used to generate multiple numbers of decision trees. Basically, it takes random samples. And then, observations are given into the decision tree. Finally, the common output that comes from the decision tree is the ultimate output. The syntax of this function:

```
randomForest(formula=, data=)
```

3. e1071

This e1071 is one of the most widely used R packages for machine learning. Using this package, a developer can implement support vector machines (SVM), shortest path computation, bagged clustering, Naive Bayes classifier, short-time Fourier transform, fuzzy clustering, etc.

As an instance, for Breast data SVM syntax is:

```
svm(Species ~Sepal.Length + Sepal.Width, data=breast)
```

4. Rpart

Rpart stands for recursive partitioning and regression training. This R package for machine learning can be performed both tasks: classification and regression. It acts using a two-stage step. The output model a binary tree. The plot() function is used to plot the output result. Also, there is an alternative function, prp() function, that is more flexible and powerful than a basic plot() function.

The function rpart() used to establish a relationship between independent and dependent variables. The syntax is: `rpart(formula, data=, method=,control=)`

where the formula is the combination of independent and dependent variables, data is the name of the dataset, the method is the objective, and control is your system requirement.

5. ggplot2

Another one of the most elegant and aesthetic graphics framework R packages for data science is ggplot2. It's a system of creating graphics based on the grammar of graphics. The installation syntax for this data science package

3.SYSTEM ANALYSIS

3.1 Feasibility Study

A feasibility study will help prove to the entrepreneur, venture capitalists, lenders and investors the existence of the market, the liquidity of the business venture and the expected return on investment.

A feasibility study will help you identify the flaws, business challenges, strengths, weaknesses, opportunities, threats and unforeseen circumstances that might affect the success and sustainability of the business venture. The system should undergo three different categories of feasibility studies

- Economical Feasibility
- Technical Feasibility
- Operational Feasibility

3.2 Economical Feasibility

Analysis of a project's costs and revenues in an effort to determine whether or not it is logical and possible to complete. The project was developed to find the heart disease of the patient by analyzing the previous issues status which was stored as a dataset. The project will reduce lot of economical issue in reducing the cost for medical testing in knowing the percentage of predicting breast cancer range.

3.3 Technical Feasibility

The technical aspects for the development of the proposed project are well within the project team's capabilities to produce such a product. The present patient issue get updated towards the dataset and thus as per the environment changes, the issue changes can be analyzed. Any changes get loaded to the database in dynamic manner therefore all category of person can undergo prediction process.

3.4 Operational Feasibility

Operational feasibility is a measure of how well a proposed system solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development.

Any changes to the application can be updated at any time interval and thus all type of medical users can make use of the application in predicting the percentage of **breast cancer** for the patient.

3.5 Proposed System

In proposed system the overall range of scaly, or thickened, Nipple discharge, Irritated or itchy breasts, Change in breast color, Increase in breast size or shape (over a short period of time), Changes in touch (may feel hard, tender or warm), Peeling or flaking of the nipple skin, A breast lump or thickening, Redness or pitting of the breast skin are stored in the database. The value of affected range of each patient also gets stored as a dataset. The present range of patient can be analyzed with the previous dataset therefore the range of diabetes can be constructed in a graphical manner. The medicine should be provided for the patient as per the affected percentage which can be executed with data mining techniques.

Advantages

- The range of affected breast cancer patient can be analyzed by knowing the past patient dataset.
- The accuracy in predicting will be more and thus as per the affected range, the treatment can be given by the doctor in quick manner.
- The percentage of breast cancer patient can be monitored by constructing the model and accuracy factor.

3.6 Existing System

In existing system the prediction of diabetes can be recognized as per the symptoms such as A lump or mass in the breast, Swelling of all or part of the breast, even if no lump is felt, Skin irritation or dimpling, Breast or nipple pain, Nipple retraction (turning inward), The nipple or breast skin appears red, scaly, or thickened, Nipple discharge, Irritated or itchy breasts, Change in breast color, Increase in breast size or shape (over a short period of time), Changes in touch (may feel hard, tender or warm), Peeling or flaking of the nipple skin, A breast lump or thickening, Redness or pitting of the breast skin (like the skin of an orange). The doctor won't have a facility in analyzing the dataset of previous affected patient details therefore the percentage of cancer can't be determined in accurate manner. The medicine should be provided for the patient as per the affected percentage which can't be executed without data mining techniques.

Disadvantages

The treatment related to predicting breast cancer can be done with various testing and it won't give accurate result of range.

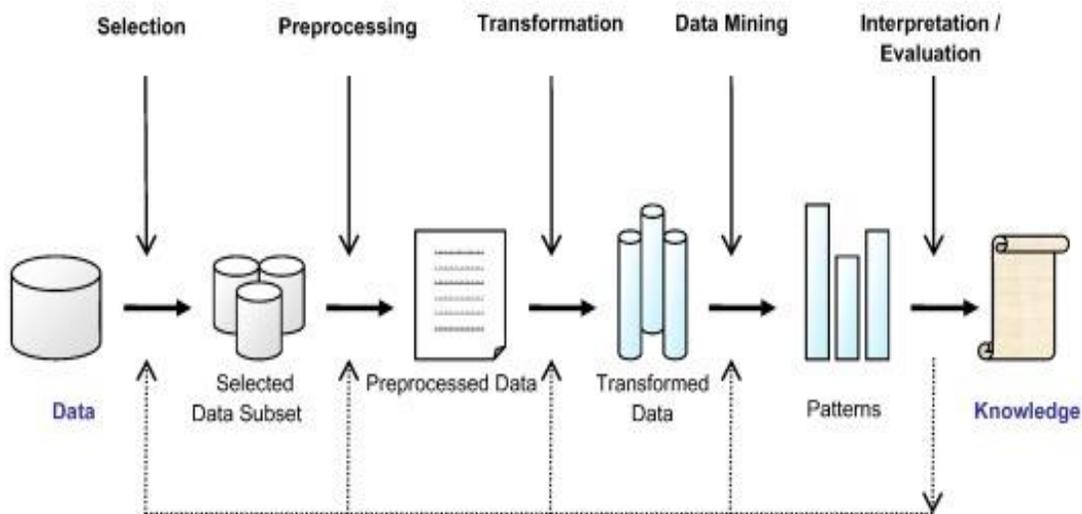
The percentage of affected range of cancer and symptoms of patient details are not maintained in the database.

The analyzing factor in calculating the occurrence can't be executed.

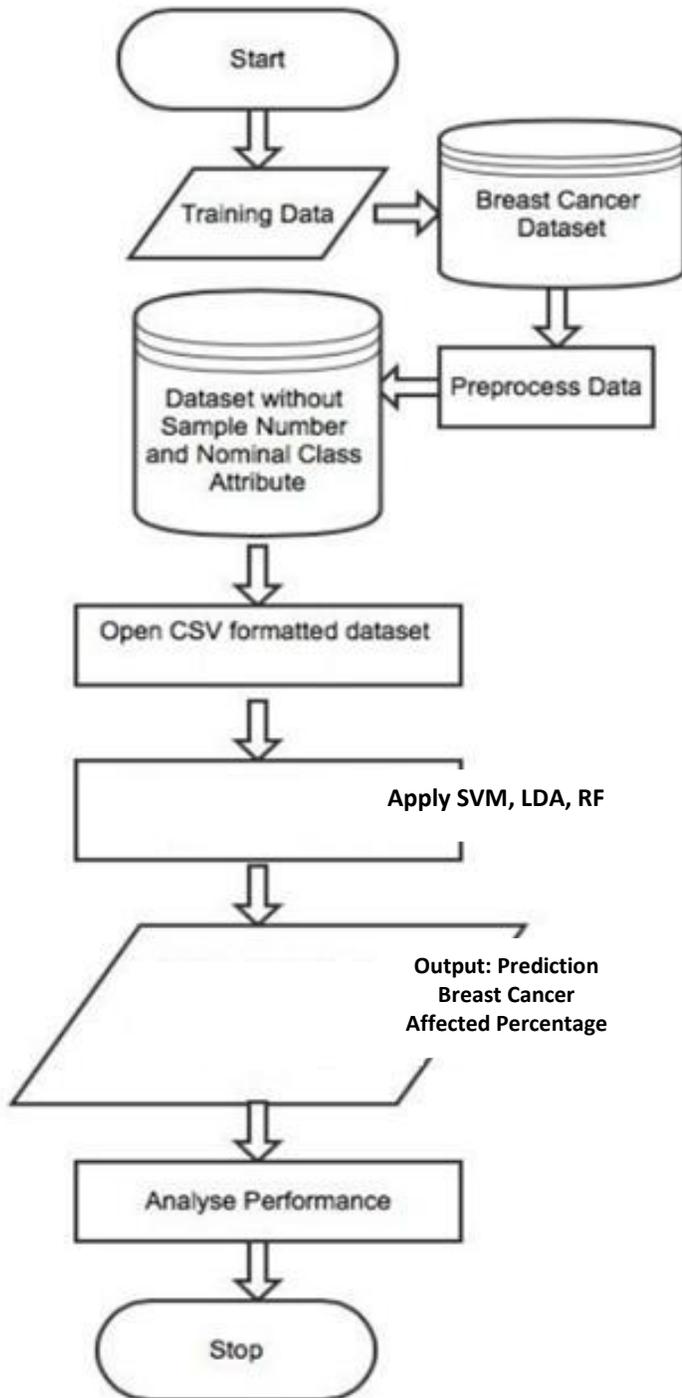
Data mining concept will help only in prediction where machine learning algorithm will study from the factor that not executed in existing.

4.SYSTEM DESIGN

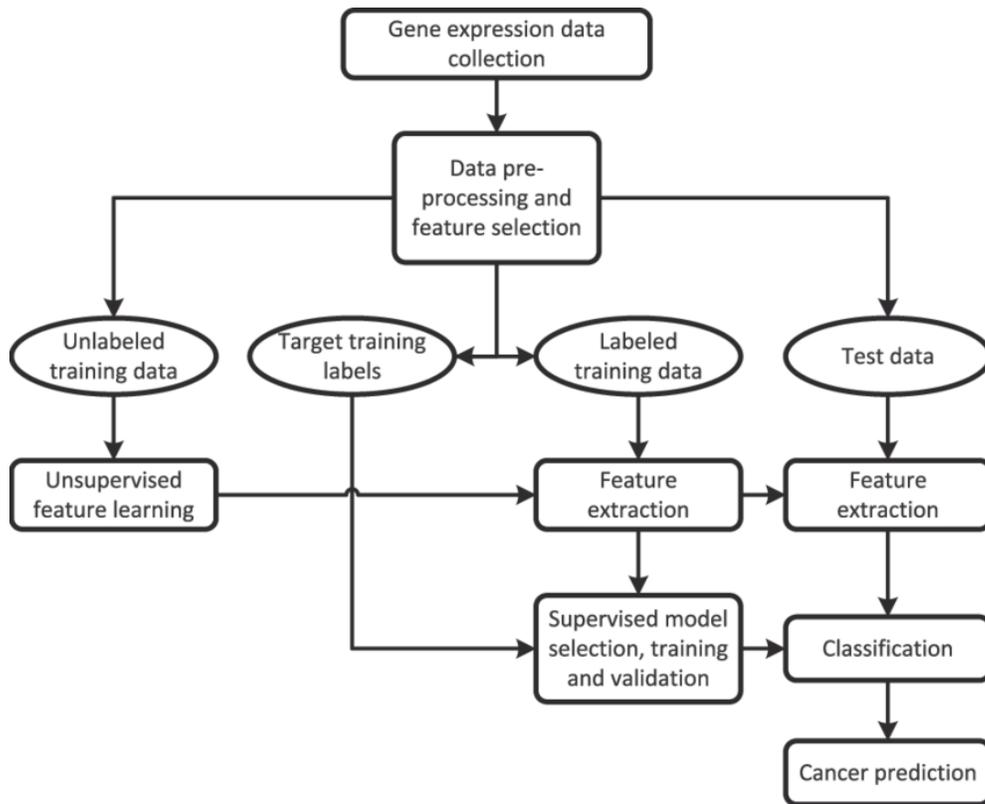
4.1 Data Flow Diagram



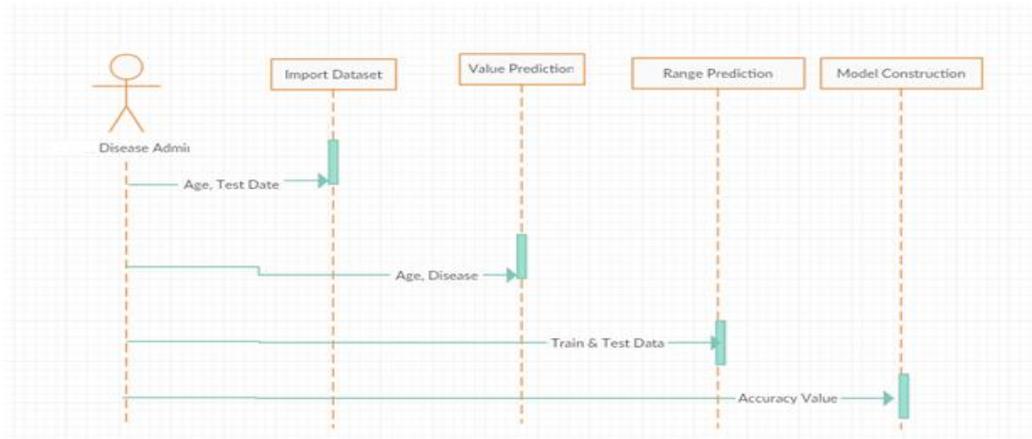
DATA FLOW DIAGRAM



Flow Chart



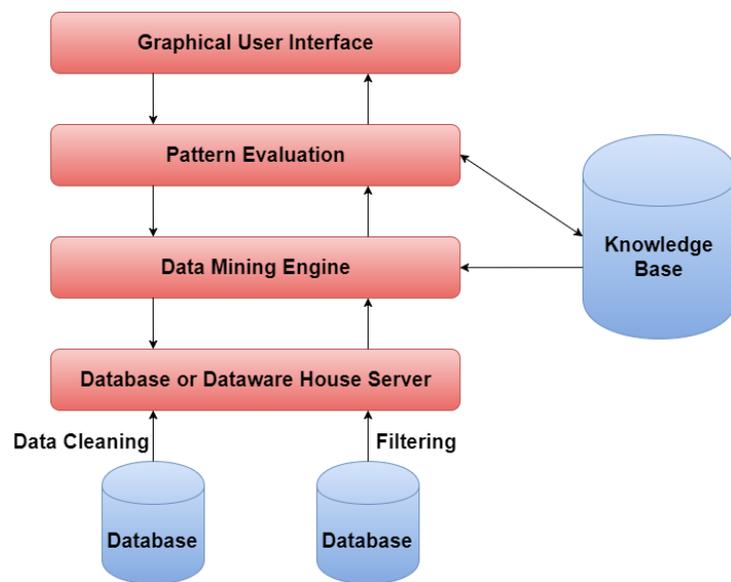
Cancer Prediction Work Flow



SEQUENCE DIAGRAM

4.2 Architecture of the System

A system architecture or systems architecture is the computational design that defines the structure and/or behavior of a system. An architecture description is a formal description of a system, organized in a way that supports reasoning about the structural properties of the system. It defines the system components or building blocks and provides a plan from which products can be procured and system developed that will work together to implement the overall system.



Architectural Diagram

5.SYSTEM IMPLEMENTATION

Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus it can be considered to be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective.

The implementation stage involves careful planning, investigation of the existing system and its constraints on implementation, designing of methods to achieve changeover and evaluation of changeover methods.

Implementation is the process of converting a new system design into operation. It is the phase that focuses on the user training, site preparation and file conversion for installing a candidate system. The important factor that should be considered here is that the conversion should not disrupt the function of the organization.

Thus this project is transferred from the design phase to the implementation. The implementation of the project is made successfully that the result by the proposed system is achieved. The breast cancer prediction can be analyzed which helps the user to predict the affected range in accurate manner. The testing and training data will split to find the accuracy of the dataset. The dataset can be used to analyze any category of shape, diameter, parameter and color data to find the level and range of breast cancer affected area prediction.

5.1 Module Description

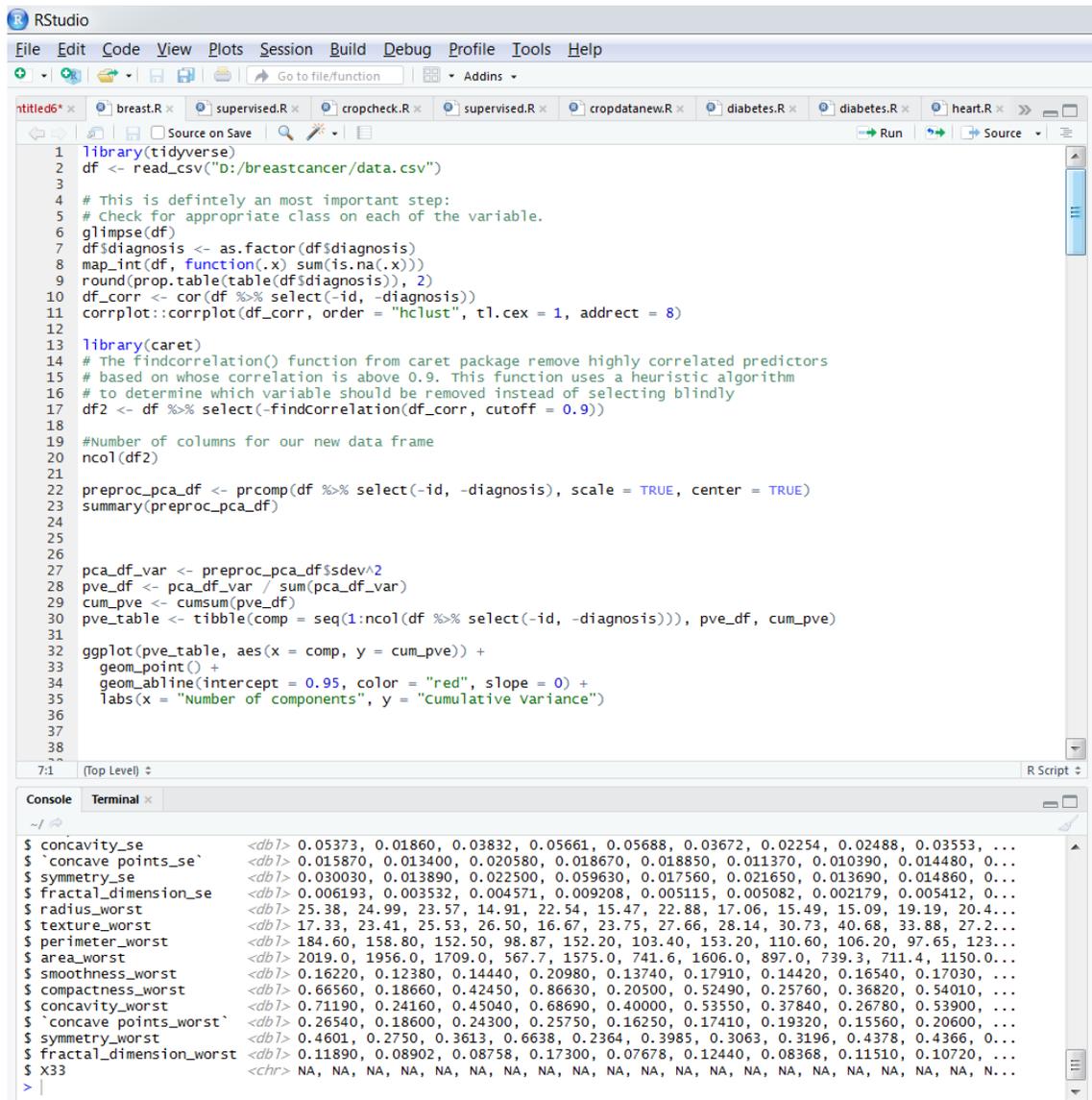
Import Data

The breast cancer dataset is stored as a .csv file which contains the data of overall breast cancer symptoms and range. As per the field comparison, the patient can be predicted the presence of breast cancer or not. The dataset uploaded to R and then prediction occurs.

Breast Cancer Dataset

```
## Observations: 569
## Variables: 32
## $ id <int> 842302, 842517, 84300903, 84348301, 84...
## $ diagnosis <chr> "M", "M", "M", "M", "M", "M", "M", "M"...
## $ radius_mean <dbl> 17.990, 20.570, 19.690, 11.420, 20.290...
## $ texture_mean <dbl> 10.38, 17.77, 21.25, 20.38, 14.34, 15...
## $ perimeter_mean <dbl> 122.80, 132.90, 130.00, 77.58, 135.10,...
## $ area_mean <dbl> 1001.0, 1326.0, 1203.0, 386.1, 1297.0,...
## $ smoothness_mean <dbl> 0.11840, 0.08474, 0.10960, 0.14250, 0...
## $ compactness_mean <dbl> 0.27760, 0.07864, 0.15990, 0.28390, 0...
## $ concavity_mean <dbl> 0.30010, 0.08690, 0.19740, 0.24140, 0...
## $ concave_points_mean <dbl> 0.14710, 0.07017, 0.12790, 0.10520, 0...
## $ symmetry_mean <dbl> 0.2419, 0.1812, 0.2069, 0.2597, 0.1809...
## $ fractal_dimension_mean <dbl> 0.07871, 0.05667, 0.05999, 0.09744, 0...
## $ radius_se <dbl> 1.0950, 0.5435, 0.7456, 0.4956, 0.7572...
## $ texture_se <dbl> 0.9053, 0.7339, 0.7869, 1.1560, 0.7813...
## $ perimeter_se <dbl> 8.589, 3.398, 4.585, 3.445, 5.438, 2.2...
## $ area_se <dbl> 153.40, 74.08, 94.03, 27.23, 94.44, 27...
## $ smoothness_se <dbl> 0.006399, 0.005225, 0.006150, 0.009110...
## $ compactness_se <dbl> 0.049040, 0.013080, 0.040060, 0.074580...
## $ concavity_se <dbl> 0.05373, 0.01860, 0.03832, 0.05661, 0...
## $ concave_points_se <dbl> 0.015870, 0.013400, 0.020580, 0.018670...
## $ symmetry_se <dbl> 0.03003, 0.01389, 0.02250, 0.05963, 0...
## $ fractal_dimension_se <dbl> 0.006193, 0.003532, 0.004571, 0.009208...
## $ radius_worst <dbl> 25.38, 24.99, 23.57, 14.91, 22.54, 15...
## $ texture_worst <dbl> 17.33, 23.41, 25.53, 26.50, 16.67, 23...
## $ perimeter_worst <dbl> 184.60, 158.80, 152.50, 98.87, 152.20,...
## $ area_worst <dbl> 2019.0, 1956.0, 1709.0, 567.7, 1575.0,...
## $ smoothness_worst <dbl> 0.1622, 0.1238, 0.1444, 0.2098, 0.1374...
## $ compactness_worst <dbl> 0.6656, 0.1866, 0.4245, 0.8663, 0.2050...
## $ concavity_worst <dbl> 0.71190, 0.24160, 0.45040, 0.68690, 0...
## $ concave_points_worst <dbl> 0.26540, 0.18600, 0.24300, 0.25750, 0...
## $ symmetry_worst <dbl> 0.4601, 0.2750, 0.3613, 0.6638, 0.2364...
## $ fractal_dimension_worst <dbl> 0.11890, 0.08902, 0.08758, 0.17300, 0...
```

Upload Breast Cancer CSV file



```
1 library(tidyverse)
2 df <- read_csv("D:/breastcancer/data.csv")
3
4 # This is definitely an most important step:
5 # Check for appropriate class on each of the variable.
6 glimpse(df)
7 df$diagnosis <- as.factor(df$diagnosis)
8 map_int(df, function(x) sum(is.na(x)))
9 round(prop.table(table(df$diagnosis)), 2)
10 df_corr <- cor(df %>% select(-id, -diagnosis))
11 corrplot::corrplot(df_corr, order = "hclust", tl.cex = 1, addrect = 8)
12
13 library(caret)
14 # The findCorrelation() function from caret package remove highly correlated predictors
15 # based on whose correlation is above 0.9. This function uses a heuristic algorithm
16 # to determine which variable should be removed instead of selecting blindly
17 df2 <- df %>% select(-findCorrelation(df_corr, cutoff = 0.9))
18
19 #Number of columns for our new data frame
20 ncol(df2)
21
22 preproc_pca_df <- prcomp(df %>% select(-id, -diagnosis), scale = TRUE, center = TRUE)
23 summary(preproc_pca_df)
24
25
26
27 pca_df_var <- preproc_pca_df$sdev^2
28 pve_df <- pca_df_var / sum(pca_df_var)
29 cum_pve <- cumsum(pve_df)
30 pve_table <- tibble(comp = seq(1:ncol(df %>% select(-id, -diagnosis))), pve_df, cum_pve)
31
32 ggplot(pve_table, aes(x = comp, y = cum_pve)) +
33   geom_point() +
34   geom_abline(intercept = 0.95, color = "red", slope = 0) +
35   labs(x = "Number of components", y = "Cumulative variance")
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

```
~|
$ concavity_se <dbl> 0.05373, 0.01860, 0.03832, 0.05661, 0.05688, 0.03672, 0.02254, 0.02488, 0.03553, ...
$ `concave points_se` <dbl> 0.015870, 0.013400, 0.020580, 0.018670, 0.018850, 0.011370, 0.010390, 0.014480, 0...
$ symmetry_se <dbl> 0.030030, 0.013890, 0.022500, 0.059630, 0.017560, 0.021650, 0.013690, 0.014860, 0...
$ fractal_dimension_se <dbl> 0.006193, 0.003532, 0.004571, 0.009208, 0.005115, 0.005082, 0.002179, 0.005412, 0...
$ radius_worst <dbl> 25.38, 24.99, 23.57, 14.91, 22.54, 15.47, 22.88, 17.06, 15.49, 15.09, 19.19, 20.4...
$ texture_worst <dbl> 17.33, 23.41, 25.53, 26.50, 16.67, 23.75, 27.66, 28.14, 30.73, 40.68, 33.88, 27.2...
$ perimeter_worst <dbl> 184.60, 158.80, 152.50, 98.87, 152.20, 103.40, 153.20, 110.60, 106.20, 97.65, 123...
$ area_worst <dbl> 2019.0, 1956.0, 1709.0, 567.7, 1575.0, 741.6, 1606.0, 897.0, 739.3, 711.4, 1150.0...
$ smoothness_worst <dbl> 0.16220, 0.12380, 0.14440, 0.20980, 0.13740, 0.17910, 0.14420, 0.16540, 0.17030, ...
$ compactness_worst <dbl> 0.66560, 0.18660, 0.42450, 0.86630, 0.20500, 0.52490, 0.25760, 0.36820, 0.54010, ...
$ concavity_worst <dbl> 0.71190, 0.24160, 0.45040, 0.68690, 0.40000, 0.53550, 0.37840, 0.26780, 0.53900, ...
$ `concave points_worst` <dbl> 0.26540, 0.18600, 0.24300, 0.25750, 0.16250, 0.17410, 0.19320, 0.15560, 0.20600, ...
$ symmetry_worst <dbl> 0.4601, 0.2750, 0.3613, 0.6638, 0.2364, 0.3985, 0.3063, 0.3196, 0.4378, 0.4366, 0...
$ fractal_dimension_worst <dbl> 0.11890, 0.08902, 0.08758, 0.17300, 0.07678, 0.12440, 0.08368, 0.11510, 0.10720, ...
$ x33 <chr> NA, N...
```

Categories Data

As per each field such as diagnosis, radius, perimeter, texture, smoothness, compactness, symmetry and dimension, the data get spitted which helps recognize the current patient is affected by breast cancer or not. If affected, the affected level will be predicted. The data get compared with each field and as per the match

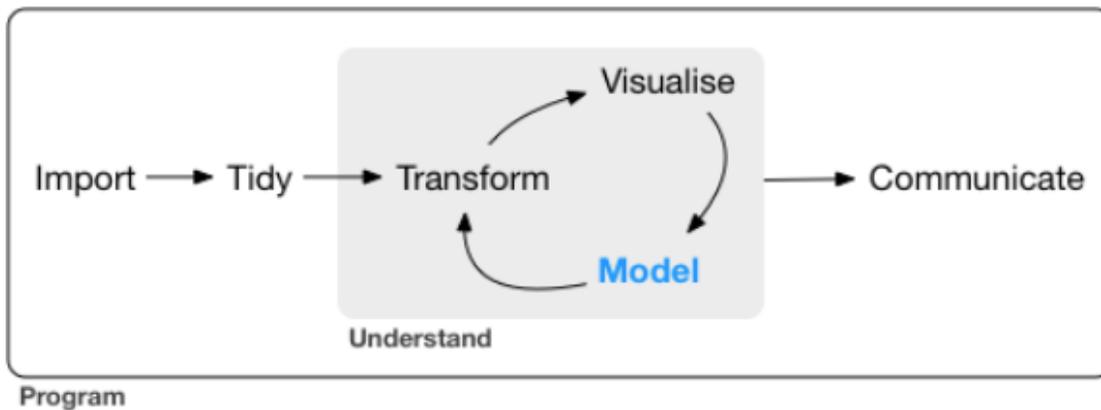
occurs, the percentage of accuracy calculated. The missing values also calculated. Here the value is zero therefore no missing values.

Predict Affected Values

All variables related to breast cancer are compared to predict the matches. The matches show the affected and non affected values. The details of current patient are applied into data mining which compares with all other dataset and find the affected range. As per the range, the doctor can provide medicine for quick recover.

Transformation of Data

It will remove the id and diagnosis variable, then we will also scale and center the variables. The important compounds for variable comparison will be analyzed and predicted. This will help to find the accuracy rate in quick time.



Preprocessing of Data

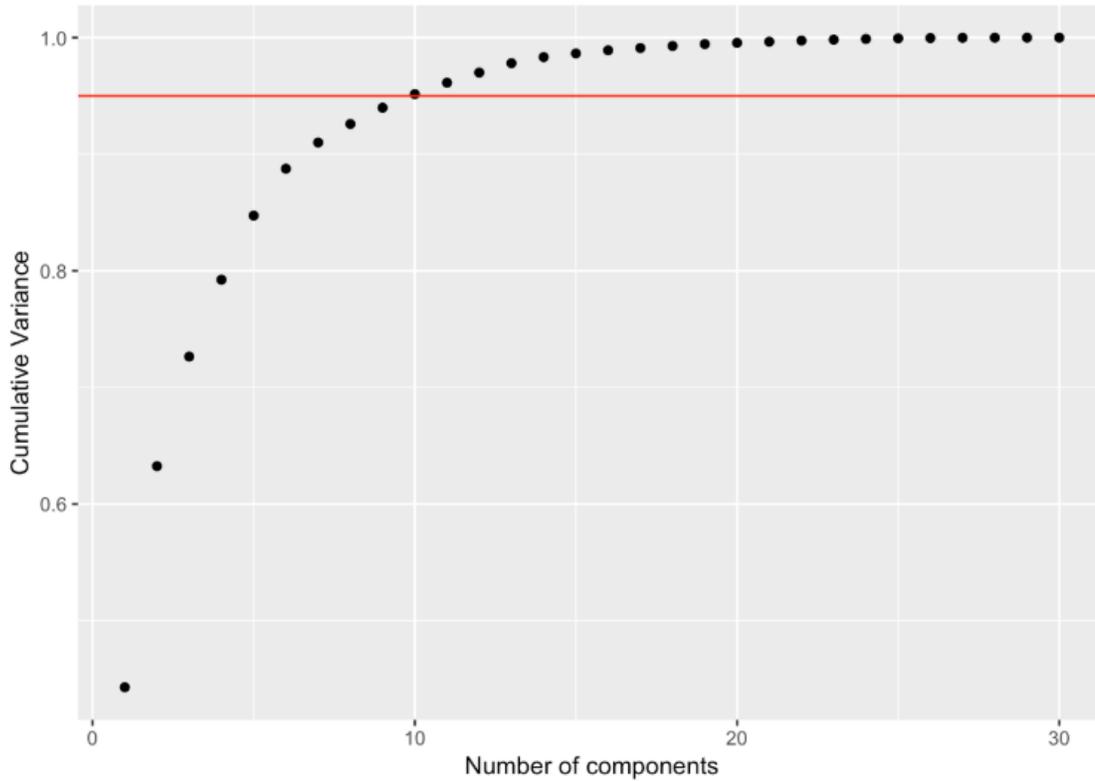
The preprocessing helps to find the standard deviation, proportion of variance and cumulative proportion. This values can be used to find the accuracy rate in each iteration. The patient data can be applied with many iteration to find the accuracy of breast cancer affected level. This will help the doctor to find the cancer is in preliminary or extreme level.

Preprocessing

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

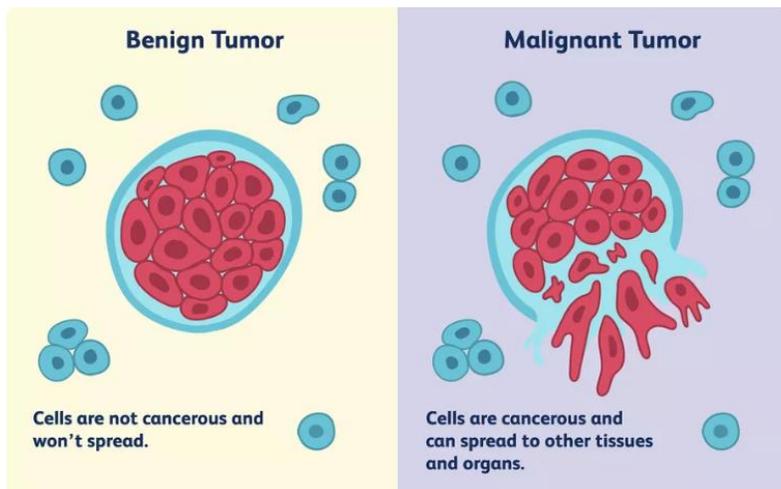
Proportion of variance

“Proportion of variance” is a generic term to mean a part of variance as a whole. For example, the total variance in any system is 100%, but there might be many different causes for the total variance — each of which have their own proportion associated with them. Here the number of cumulative variances and number of components are taken into account. As per the changes, the graph will be constructed to detect the changes of patient data while comparing with breast cancer dataset.

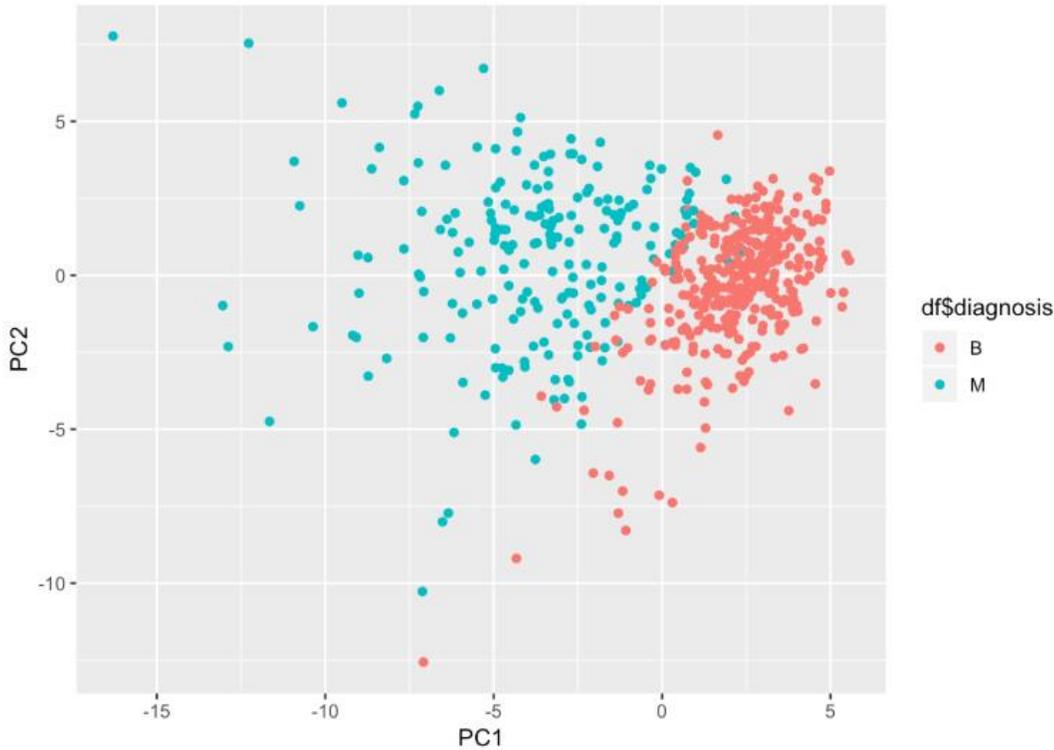


Diagnosis View

Its objective is to train a classifier model on cancer cells characteristics dataset to predict whether the cell is B = benign or M = malignant. Doctor will take is to find out whether it is malignant or benign, as this will affect your treatment plan. In short, the meaning of malignant is cancerous and the meaning of benign is non-cancerous.

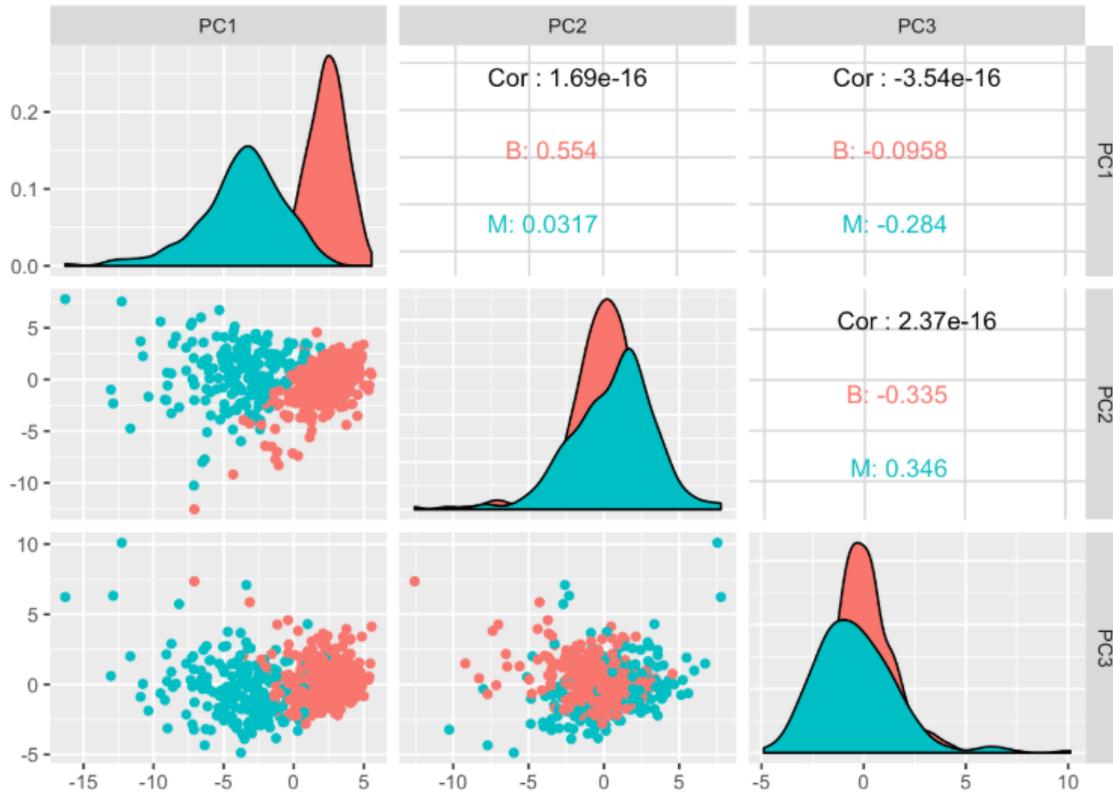


Predicting B and M in Diagnosis



Variable Detection Range

The package offers a single plotting interface for these analysis results and plots in a unified style using 'ggplot2'. The field such as diagnosis, radius, perimeter, texture, smoothness, compactness, symmetry and dimension are predicted as per the cancer level detected. The line gets constructed by fixing the value either at B or M.



Model Construction

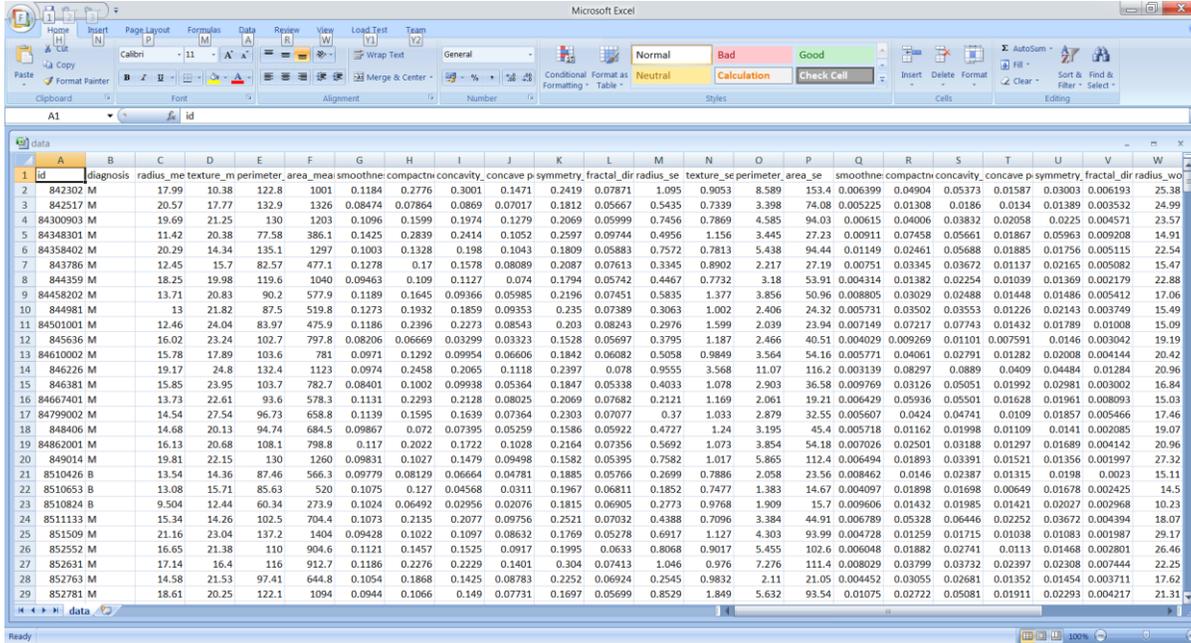
At this stage, users should have gathered enough information and insights on our data to choose appropriate modeling techniques.

Before users go ahead, users need to split the data into a training and testing set

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  B  M
##           B  71  2
##           M   0  40
##
##           Accuracy : 0.9823
##           95% CI : (0.9375, 0.9978)
##           No Information Rate : 0.6283
##           P-Value [Acc > NIR] : <2e-16
##
##           Kappa : 0.9617
##           Mcnemar's Test P-Value : 0.4795
##
##           Sensitivity : 0.9524
##           Specificity : 1.0000
##           Pos Pred Value : 1.0000
##           Neg Pred Value : 0.9726
##           Prevalence : 0.3717
##           Detection Rate : 0.3540
##           Detection Prevalence : 0.3540
##           Balanced Accuracy : 0.9762
##
##           'Positive' Class : M
##
```

5.2 Project Description

Breast Cancer Dataset



id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness	concavity	concave points	symmetry	fractal dimension	radius standard deviation	texture standard deviation	perimeter standard deviation	area standard deviation	smoothness standard deviation	compactness standard deviation	concavity standard deviation	concave points standard deviation	symmetry standard deviation	fractal dimension standard deviation	radius worst case
1	842302 M	17.99	10.38	122.8	1001	0.1184	0.2776	0.3001	0.1471	0.2419	0.07871	1.095	0.9053	8.589	153.4	0.006399	0.04904	0.05373	0.01587	0.03003	0.006193	25.38
2	842517 M	20.57	17.77	132.9	1326	0.08474	0.07864	0.0869	0.07017	0.1812	0.05667	0.5435	0.7339	3.398	74.08	0.005225	0.01308	0.0186	0.0134	0.01389	0.003532	24.99
3	8430093 M	19.69	21.25	130	1203	0.1096	0.1599	0.1974	0.1279	0.2069	0.05999	0.7456	0.7869	4.585	94.03	0.00615	0.04006	0.03832	0.02058	0.0225	0.004571	23.57
4	8434801 M	11.42	20.38	77.58	386.1	0.1425	0.2839	0.2414	0.1052	0.2597	0.09744	0.4956	1.156	3.445	27.23	0.00911	0.07458	0.05661	0.01867	0.05963	0.009208	14.91
5	84358402 M	20.29	14.34	135.1	1297	0.1003	0.1328	0.198	0.1043	0.1809	0.05883	0.7572	0.7813	5.438	94.44	0.01149	0.02461	0.05688	0.01885	0.01756	0.005115	22.54
6	843786 M	12.45	15.7	82.57	477.1	0.1278	0.17	0.1578	0.08089	0.2087	0.07613	0.3345	0.8902	2.217	27.19	0.00751	0.03345	0.03672	0.01137	0.02165	0.005082	15.47
7	844359 M	18.25	19.98	119.6	1040	0.09463	0.109	0.1127	0.074	0.1794	0.05742	0.4467	0.7732	3.18	53.91	0.004314	0.01382	0.02254	0.01039	0.01369	0.002179	22.88
8	84458202 M	13.71	20.83	90.2	577.9	0.1189	0.1645	0.09366	0.05985	0.2196	0.07451	0.5835	1.377	3.856	50.96	0.008805	0.03029	0.02488	0.01448	0.01486	0.005412	17.06
9	844981 M	13	21.82	87.5	519.8	0.1273	0.1932	0.1859	0.09353	0.235	0.07389	0.3063	1.002	2.406	24.32	0.005731	0.03502	0.03553	0.01226	0.02143	0.003749	15.49
10	84501001 M	12.46	24.04	83.97	475.9	0.1186	0.2396	0.2273	0.08543	0.203	0.08243	0.2976	1.599	2.039	23.94	0.007149	0.07217	0.07743	0.01432	0.01789	0.001008	15.09
11	845636 M	16.02	23.24	102.7	797.8	0.08206	0.06669	0.03299	0.03323	0.1528	0.05697	0.3795	1.187	2.466	40.51	0.004029	0.009269	0.01101	0.007591	0.0146	0.003042	19.19
12	84610002 M	15.78	17.89	103.6	781	0.0971	0.1292	0.09954	0.06606	0.1842	0.06082	0.5058	0.9849	3.564	54.16	0.005771	0.04061	0.02791	0.01282	0.02008	0.004144	20.42
13	846226 M	19.17	24.8	132.4	1123	0.0974	0.2458	0.2065	0.1118	0.2397	0.078	0.9555	3.568	11.07	116.2	0.003139	0.08297	0.0889	0.0409	0.04484	0.01284	20.96
14	846381 M	15.85	23.95	103.7	782.7	0.08401	0.1002	0.09938	0.05364	0.1847	0.05338	0.4033	1.078	2.903	36.58	0.009769	0.03126	0.05051	0.01992	0.02981	0.003002	16.84
15	84667401 M	13.73	22.61	93.6	578.3	0.1131	0.2293	0.2128	0.08025	0.2069	0.07682	0.2121	1.169	2.061	19.21	0.006429	0.05936	0.05501	0.01628	0.01961	0.008093	15.03
16	84799002 M	14.54	27.54	96.73	658.8	0.1139	0.1595	0.1639	0.07364	0.2303	0.07077	0.37	1.033	2.879	32.55	0.005607	0.0424	0.04741	0.0109	0.01857	0.005466	17.46
17	848406 M	14.68	20.13	94.74	684.5	0.09867	0.072	0.07395	0.05259	0.1586	0.05922	0.4727	1.24	3.195	45.4	0.005718	0.01162	0.01998	0.01109	0.0141	0.002085	19.07
18	84862001 M	16.13	20.68	108.1	798.8	0.117	0.2022	0.1722	0.1028	0.2164	0.07356	0.5692	1.073	3.854	54.18	0.007026	0.02501	0.03188	0.01297	0.01689	0.004142	20.96
19	849014 M	19.81	22.15	130	1260	0.09831	0.1027	0.1479	0.09498	0.1582	0.05395	0.7582	1.017	5.865	112.4	0.006494	0.01893	0.03391	0.01521	0.01356	0.001997	27.32
20	8510426 B	13.54	14.36	87.46	566.3	0.09779	0.08129	0.06664	0.04781	0.1885	0.05766	0.2699	0.7886	2.058	23.56	0.008462	0.0146	0.02387	0.01315	0.0198	0.0023	15.11
21	8510653 B	13.08	15.71	85.63	520	0.1075	0.127	0.04568	0.0311	0.1967	0.06811	0.1852	0.7477	1.383	14.67	0.004097	0.01898	0.01698	0.00649	0.01678	0.002425	14.5
22	8510824 B	9.504	12.44	60.34	273.9	0.1024	0.06492	0.02956	0.02076	0.1815	0.06905	0.2773	0.9768	1.909	15.7	0.009606	0.01432	0.01985	0.01421	0.02027	0.002968	10.23
23	8511133 M	15.34	14.26	102.5	704.4	0.1073	0.2135	0.2077	0.09756	0.2521	0.07032	0.4388	0.7096	3.384	44.91	0.006789	0.05328	0.06446	0.02252	0.03672	0.004394	18.07
24	851509 M	21.16	23.04	137.2	1404	0.09428	0.1022	0.1097	0.08632	0.1769	0.05278	0.6917	1.127	4.303	93.99	0.004728	0.01259	0.01715	0.01038	0.01083	0.001987	29.17
25	852552 M	16.65	21.38	110	904.6	0.1121	0.1457	0.1525	0.0917	0.1995	0.0633	0.8068	0.9017	5.455	102.6	0.006048	0.01882	0.02741	0.0113	0.01468	0.002801	26.46
26	852631 M	17.14	16.4	116	912.7	0.1186	0.2276	0.2229	0.1401	0.304	0.07413	1.046	0.976	7.276	111.4	0.008029	0.03799	0.03732	0.02397	0.02308	0.007444	22.25
27	852763 M	14.58	21.53	97.41	644.8	0.1054	0.1868	0.1425	0.08783	0.2252	0.06924	0.2545	0.9832	2.11	21.05	0.004452	0.03055	0.02681	0.01352	0.01454	0.003711	17.62
28	852781 M	18.61	20.25	122.1	1094	0.0944	0.1066	0.149	0.07731	0.1697	0.05699	0.8529	1.849	5.632	93.54	0.01075	0.02722	0.05081	0.01911	0.02293	0.004217	21.31

Variables

diagnosis

The diagnosis of breast tissues (M = malignant, B = benign)

radius_mean

mean of distances from center to points on the perimeter

texture_mean

standard deviation of gray-scale values

perimeter_mean

mean size of the core tumor

area_mean

smoothness_mean

mean of local variation in radius lengths

compactness_mean

mean of $\text{perimeter}^2 / \text{area} - 1.0$

concavity_mean

mean of severity of concave portions of the contour

concave points_mean

mean for number of concave portions of the contour

symmetry_mean

fractal_dimension_mean

mean for "coastline approximation" - 1

radius_se

standard error for the mean of distances from center to points on the perimeter

texture_se

standard error for standard deviation of gray-scale values

perimeter_se

area_se

smoothness_se

standard error for local variation in radius lengths

compactness_se

standard error for $\text{perimeter}^2 / \text{area} - 1.0$

concavity_se

standard error for severity of concave portions of the contour

concave points_se

standard error for number of concave portions of the contour

symmetry_se

fractal_dimension_se

standard error for "coastline approximation" - 1

radius_worst

"worst" or largest mean value for mean of distances from center to points on the perimeter

texture_worst

"worst" or largest mean value for standard deviation of gray-scale values

perimeter_worst

area_worst

smoothness_worst

"worst" or largest mean value for local variation in radius lengths

compactness_worst

"worst" or largest mean value for $\text{perimeter}^2 / \text{area} - 1.0$

concavity_worst

"worst" or largest mean value for severity of concave portions of the contour

concave points_worst

"worst" or largest mean value for number of concave portions of the contour

symmetry_worst

fractal_dimension_worst

"worst" or largest mean value for "coastline approximation" - 1

6.SYSTEM TESTING

Testing is vital to the success of the system. System Testing makes a logical assumption that, if all the parts of the system are correct, the goal will be achieved. Its basic function is to find the errors in the software by examining all possible loopholes. The goal of testing is to point out uncover requirements, design or coding errors or invalid acceptance or storage of data.

A Strategy for software testing integrates software test case design methods into a well planned series of steps that result in successful construction of software.

Testing is a set activity that can be planned in advance and conducted systematically. A number of software testing strategies have the following criteria:

- Test begins at module level and works outward the integration of computer-based system.
- Different testing techniques are appropriate at different points in time.
- The software developer and an independent test tool conduct testing.
- Testing and debugging are different activities, but debugging must be accommodated in any test strategy.

There are different types of system testing

- Unit Testing
- White/Black box testing
- Validation testing
- Integration testing

UNIT TESTING

In unit testing each module is tested individually. It focuses the verification efforts on the smallest unit of software in the module. This is known as module testing.

In this project all the modules are tested separately. For our project testing is performed at programming stage itself. In this testing each module in breast cancer prediction project is found to work satisfactorily with regard to the expected output from the module. There are some validations for fields. Here unit testing is performed now and then whenever a patch is delivered for each module.

INTEGRATION TESTING

Integration testing is the phase of software testing in which individual software modules are combined and tested as a group. The modules such as data upload, variable analyze and graph visualize in predicting breast cancer are tested separately and then interconnected with each other. The overall project will undergo integration testing to fulfill the requirement of the administrator.

In this project the parts of a module are tested first and then testing the sum of its parts, integration testing is done successfully

WHITE-BOX TESTING

White-box testing is sometimes called Glass-Box testing. Using this, the software engineer can derive test cases that guarantee that all independent paths within a module have been exercised at least once, exercise all logical decisions on their true and false sides, execute all loops at their boundaries within their operational bounds and exercise internal data structure to assure their validity. In this project each module is tested individually again tested with relative modules. Here, white-box testing is tested and all the boundaries are tested, logical decisions are exercised, all independent paths are checked.

BLACK-BOX TESTING

Black-box testing focuses on the functional requirements of the software. That is it enables the software to derive set of input conditions that will fully exercise all functional requirements for a program. Black box testing attempts to find errors in incorrect or missing functions, interface errors, errors in data structures or external database access, performance errors ,initialization and termination errors .

VALIDATION TESTING

At the culmination of Black-box testing, validation test begins. Validation testing can be defined in many ways but a simple definition is that validation succeeds when the software functions in a manner that can be reasonably used. After conducting the validation testing, one of the two possible condition exists whether the project performance conform to specifications or not. Depending on the output of this testing the project is accepted or rejected

7.FUTURE SCOPE

From different research studies, it was found that comparable accuracy can be achieved even with less number of features for prediction of breast cancer. Only the features that are selected by a particular feature selection technique will be the input for machine learning classifier, which overall reduces the computational complexity of the model. So future study includes the investigation to check whether number of features to be selected depends on factors like data set, standard deviation, correlation etc. or not. Moreover as a future direction, proposed to use some hybrid machine learning classifiers based on deep learning and extreme learning classifiers to compare and show the effectiveness of proposed algorithms. Further, proposed to use nature inspired algorithm for feature reductions and smooth identification of cancer from biological database.

BIBLIOGRAPHY

1. Ultrasound characterisation of breast masses, The Indian journal of radiology imaging by S. Gokhale., Vol. 19, pp. 242-249, 2009. K. Elissa, Title of paper if known, unpublished.
2. Breast Cancer Prediction Using Genetic Algorithm Based Ensemble Approach by Pragya Chauhan and Amit Swami, 18 October 2018
3. On Breast Cancer Detection: An Application of Machine Learning Algorithms on the Wisconsin Diagnostic Dataset by Abien Fred M. Agarap, 7 February 2019
4. Analysis of Machine Learning Techniques for Breast Cancer Prediction by the Priyanka Gupta and Prof. Shalini L of VIT university, vellore, 5 May 2018.
5. Breast Cancer Diagnosis by Dierent Machine Learning Methods Using Blood Analysis Data by the Muhammet Fatih Aslan, Yunus Celik , Kadir Sabani and Akif Durdu, 31 December, 2018

6. Performance Evaluation of Machine Learning Methods for Breast Cancer Prediction, by Yixuan Li, Zixuan Chen October 18, 2018
7. Breast Cancer Prediction and Detection Using Data Mining Classification Algorithms: A Comparative Study by Mumine Kaya Keles, Feb 2019
8. Breast Cancer Prediction Using Data Mining Method by Haifeng Wang and Sang Won Yoon, Department of Systems Science and Industrial Engineering State University of New York at Binghamton Binghamton, May 2015.
9. Machine Learning with Applications in Breast Cancer Diagnosis and Prognosis by Wenbin Yue , Zidong Wang, 9 May 2018

APPENDIX

A)Sample Pseducode

Using LDA

Linear Discriminant Analysis or LDA is a dimensionality reduction technique. It is used as a pre-processing step in Machine Learning and applications of pattern classification. The goal of LDA is to project the features in higher dimensional space onto a lower-dimensional space in order to avoid the curse of dimensionality and also reduce resources and dimensional costs. LDA is a supervised classification technique that is considered a part of crafting competitive machine learning models. This category of dimensionality reduction is used in areas like image recognition and predictive analysis in marketing. The LDA algorithm is applied to predict the range of affected range of patient Breast in comparing the cancer data.

Code to Apply LDA

```
preproc_lda_df <- MASS::lda(diagnosis ~., data = df, center = TRUE, scale = TRUE)
preproc_lda_df
```

```
## Call:
## lda(diagnosis ~ ., data = df, center = TRUE, scale = TRUE)
##
## Prior probabilities of groups:
##      B      M
## 0.6274165 0.3725835
##
## Group means:
##      id radius_mean texture_mean perimeter_mean area_mean
## B 26543825 12.14652 17.91476 78.07541 462.7902
## M 36818050 17.46283 21.60491 115.36538 978.3764
## smoothness_mean compactness_mean concavity_mean concave_points_mean
## B 0.09247765 0.08008462 0.04605762 0.02571741
## M 0.10289849 0.14518778 0.16077472 0.08799000
## symmetry_mean fractal_dimension_mean radius_se texture_se perimeter_se
## B 0.174186 0.06286739 0.2840824 1.220380 2.000321
## M 0.192909 0.06268009 0.6090825 1.210915 4.323929
## area_se smoothness_se compactness_se concavity_se concave_points_se
## B 21.13515 0.007195902 0.02143825 0.02599674 0.009857653
## M 72.67241 0.006780094 0.03228117 0.04182401 0.015060472
## symmetry_se fractal_dimension_se radius_worst texture_worst
## B 0.02058381 0.003636051 13.37980 23.51507
## M 0.02047240 0.004062406 21.13481 29.31821
## perimeter_worst area_worst smoothness_worst compactness_worst
## B 87.00594 558.8994 0.1249595 0.1826725
## M 141.37033 1422.2863 0.1448452 0.3748241
## concavity_worst concave_points_worst symmetry_worst
## B 0.1662377 0.07444434 0.2702459
## M 0.4506056 0.18223731 0.3234679
## fractal_dimension_worst
## B 0.07944207
## M 0.09152995
##
## Coefficients of linear discriminants:
##      LD1
## id -2.512117e-10
## radius_mean -1.080876e+00
## texture_mean 2.338408e-02
## perimeter_mean 1.172707e-01
## area_mean 1.595690e-03
```

Using Random Forest

Random forest algorithm is a supervised classification algorithm. As the name suggest, this algorithm creates the forest with a number of trees. In general, the more trees in the forest the more robust the forest looks like. In the same way in the random forest classifier, the higher the number of trees in the forest gives the high accuracy results. The construction gives better understanding of patient data.

Working of Random Forest

- Randomly select “**k**” features from total “**m**” features.
 1. Where $k \ll m$
- Among the “**k**” features, calculate the node “**d**” using the best split point.
- Split the node into **daughter nodes** using the **best split**.
- Repeat **1 to 3** steps until “**l**” number of nodes has been reached.
- Build forest by repeating steps **1 to 4** for “**n**” number times to create “**n**” **number of trees**

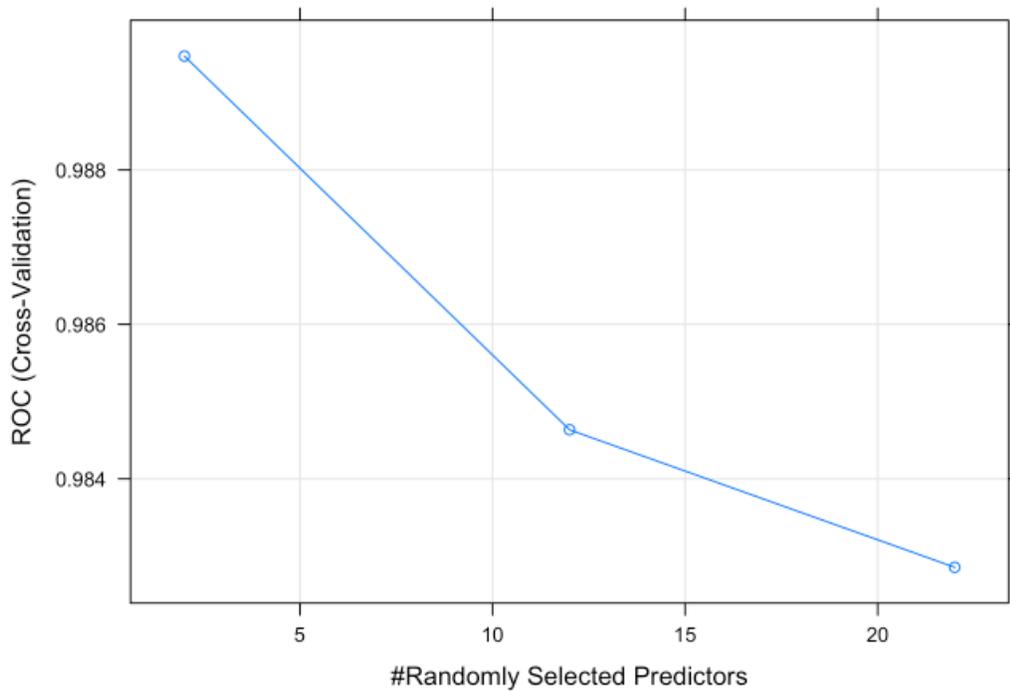
Code to Construct

```
model_rf_df <- train(diagnosis ~., data = df_training,  
  
                    method = "rf",  
  
                    metric = 'ROC',  
  
                    trControl = df_control)
```

```
prediction_rf_df <- predict(model_rf_df, df_testing)
```

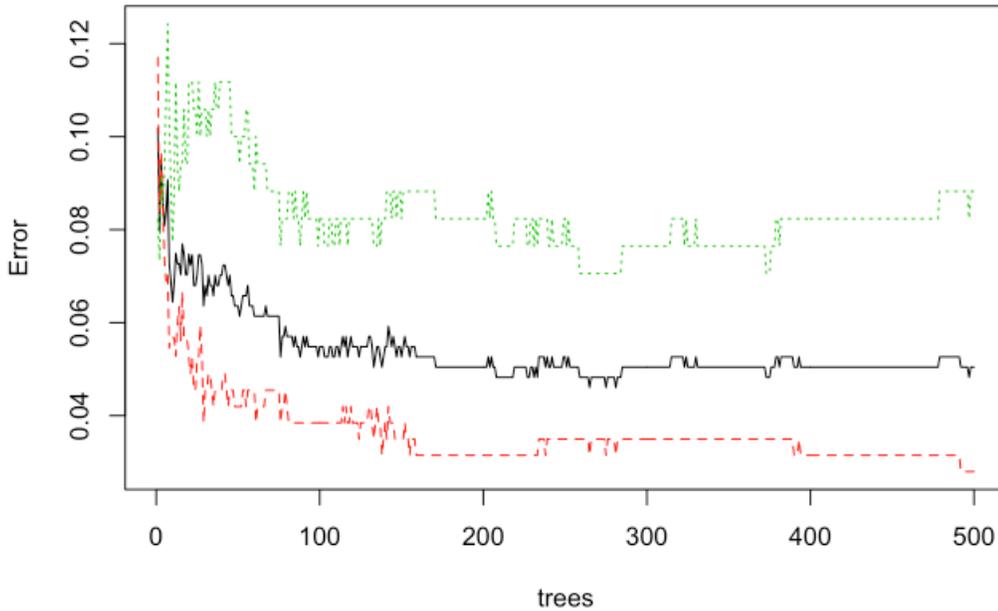
```
cm_rf_df <- confusionMatrix(prediction_rf_df, df_testing$diagnosis, positive = "M")
```

```
cm_rf_df
```



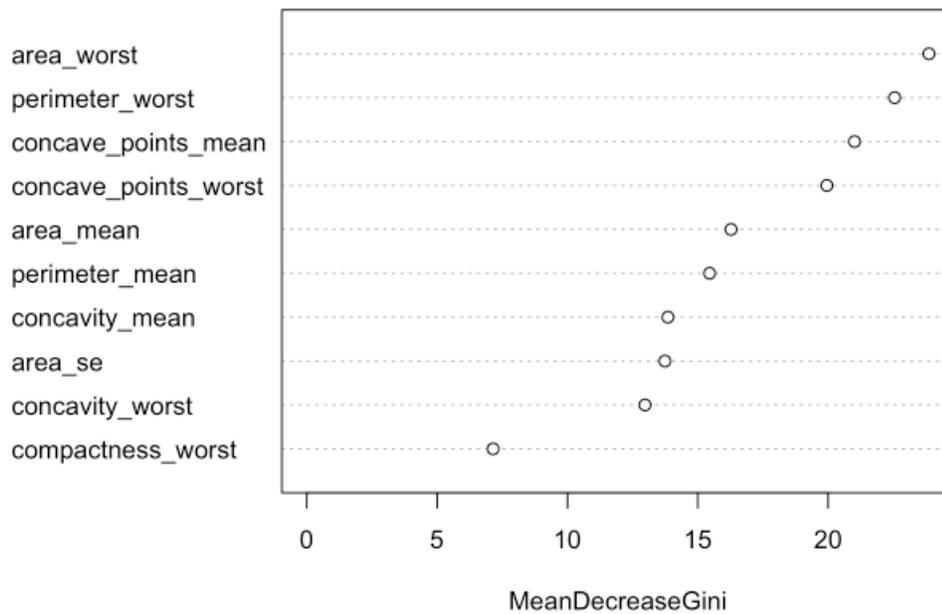
Final Model Construction

model_rf_df\$finalModel



Variable Selection using Random Forest

The 10 variables with the most predictive power



Using KNN

K-Nearest Neighbors (KNN) is one of the simplest algorithms used in Machine Learning for regression and classification problem. KNN algorithms use data and classify new data points based on similarity measures (e.g. distance function). Classification is done by a majority vote to its neighbors. The data is assigned to the class which has the nearest neighbors. As you increase the number of nearest neighbors, the value of k, accuracy might increase. The patient data will be analyzed one by one variable and at last the overall prediction range is calculated.

Working of KNN

Step-1: Select the number K of the neighbors

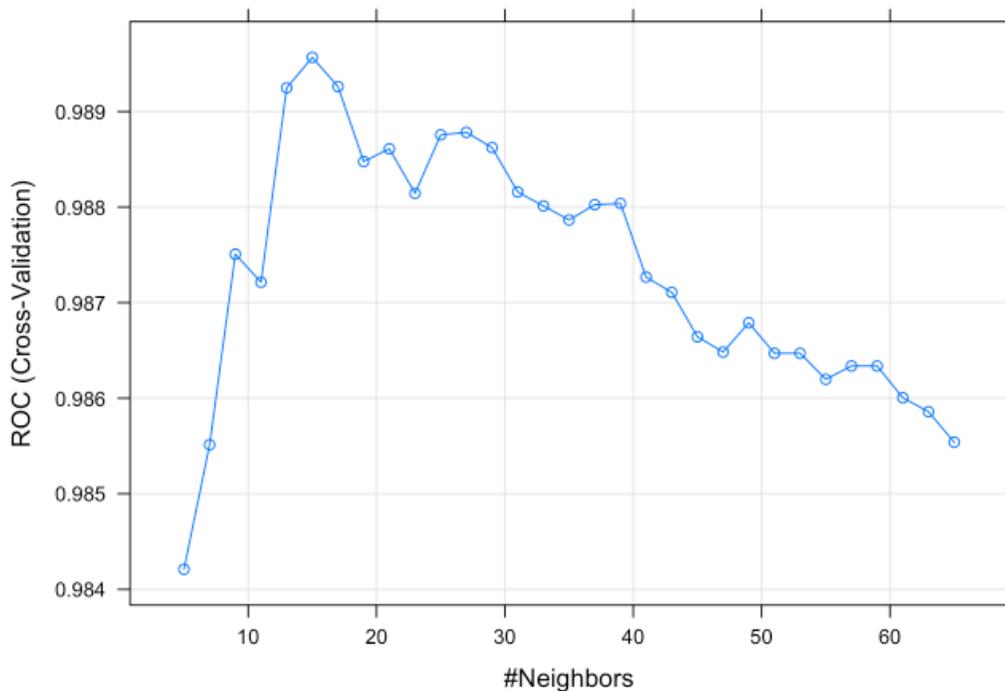
Step-2: Calculate the Euclidean distance of K number of neighbors

Step-3: Take the K nearest neighbors as per the calculated Euclidean distance.

Step-4: Among these k neighbors, count the number of the data points in each category.

Step-5: Assign the new data points to that category for which the number of the neighbor is maximum.

Step-6: Our model is ready.



Using SVM

Support Vector Machine or SVM algorithm is a simple yet powerful Supervised Machine Learning algorithm that can be used for building both regression and classification models. SVM algorithm can perform really well with both linearly separable and non-linearly separable datasets. Even with a limited amount of data, the support vector machine algorithm does not fail to show its magic. As per the data limitation, the patient data compared with the breast cancer dataset to determine the level of affected range.

Working of SVM

C: Keeping large values of C will indicate the SVM model to choose a smaller margin hyperplane. Small value of C will indicate the SVM model to choose a larger margin hyperplane.

kernel: It is the kernel type to be used in SVM model building. It can be 'linear', 'rbf', 'poly', or 'sigmoid'. The default value of kernel is 'rbf'.

degree: It's only considered in the case of polynomial kernel. It is the degree of the polynomial kernel function. The default value of degree is 3.

Code to apply SVM

```
set.seed(1815)
model_svm_df <- train(diagnosis ~., data = df_training, method = "svmLinear",
  metric = "ROC",
  preProcess = c("scale", "center"),
  trace = FALSE,
  trControl = df_control)

prediction_svm_df <- predict(model_svm_df, df_testing)
cm_svm_df <- confusionMatrix(prediction_svm_df, df_testing$diagnosis, positive = "M")
cm_svm_df
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  B  M
##           B 70  2
##           M  1 40
##
##           Accuracy : 0.9735
##           95% CI : (0.9244, 0.9945)
##           No Information Rate : 0.6283
##           P-Value [Acc > NIR] : <2e-16
##
##           Kappa : 0.9429
##           Mcnemar's Test P-Value : 1
##
##           Sensitivity : 0.9524
##           Specificity : 0.9859
##           Pos Pred Value : 0.9756
##           Neg Pred Value : 0.9722
##           Prevalence : 0.3717
##           Detection Rate : 0.3540
##           Detection Prevalence : 0.3628
##           Balanced Accuracy : 0.9691
##
##           'Positive' Class : M
##
```

Using Neural Network with LDA

LDA will work as a classifier and posteriorly it will reduce the dimensionality of the dataset and a neural network will perform the classification task, the results of both approaches will be compared afterwards.

Code to get accuracy

```
lda_training <- predict_lda_df[df_sampling_index, ]
lda_testing <- predict_lda_df[-df_sampling_index, ]
model_nnetlda_df <- train(diagnosis ~., lda_training,
```

```
method = "nnet",  
metric = "ROC",  
preProcess = c("center", "scale"),  
tuneLength = 10,  
trace = FALSE,  
trControl = df_control)
```

```
prediction_nnetlda_df <- predict(model_nnetlda_df, lda_testing)
```

```
cm_nnetlda_df <- confusionMatrix(prediction_nnetlda_df, lda_testing$diagnosis, positive = "M")
```

```
cm_nnetlda_df
```

```
## Confusion Matrix and Statistics  
##  
##           Reference  
## Prediction  B  M  
##           B 71  1  
##           M  0 41  
##  
##           Accuracy : 0.9912  
##           95% CI : (0.9517, 0.9998)  
##           No Information Rate : 0.6283  
##           P-Value [Acc > NIR] : <2e-16  
##  
##           Kappa : 0.981  
##           McNemar's Test P-Value : 1  
##  
##           Sensitivity : 0.9762  
##           Specificity : 1.0000  
##           Pos Pred Value : 1.0000  
##           Neg Pred Value : 0.9861  
##           Prevalence : 0.3717  
##           Detection Rate : 0.3628  
##           Detection Prevalence : 0.3628  
##           Balanced Accuracy : 0.9881  
##  
##           'Positive' Class : M  
##
```

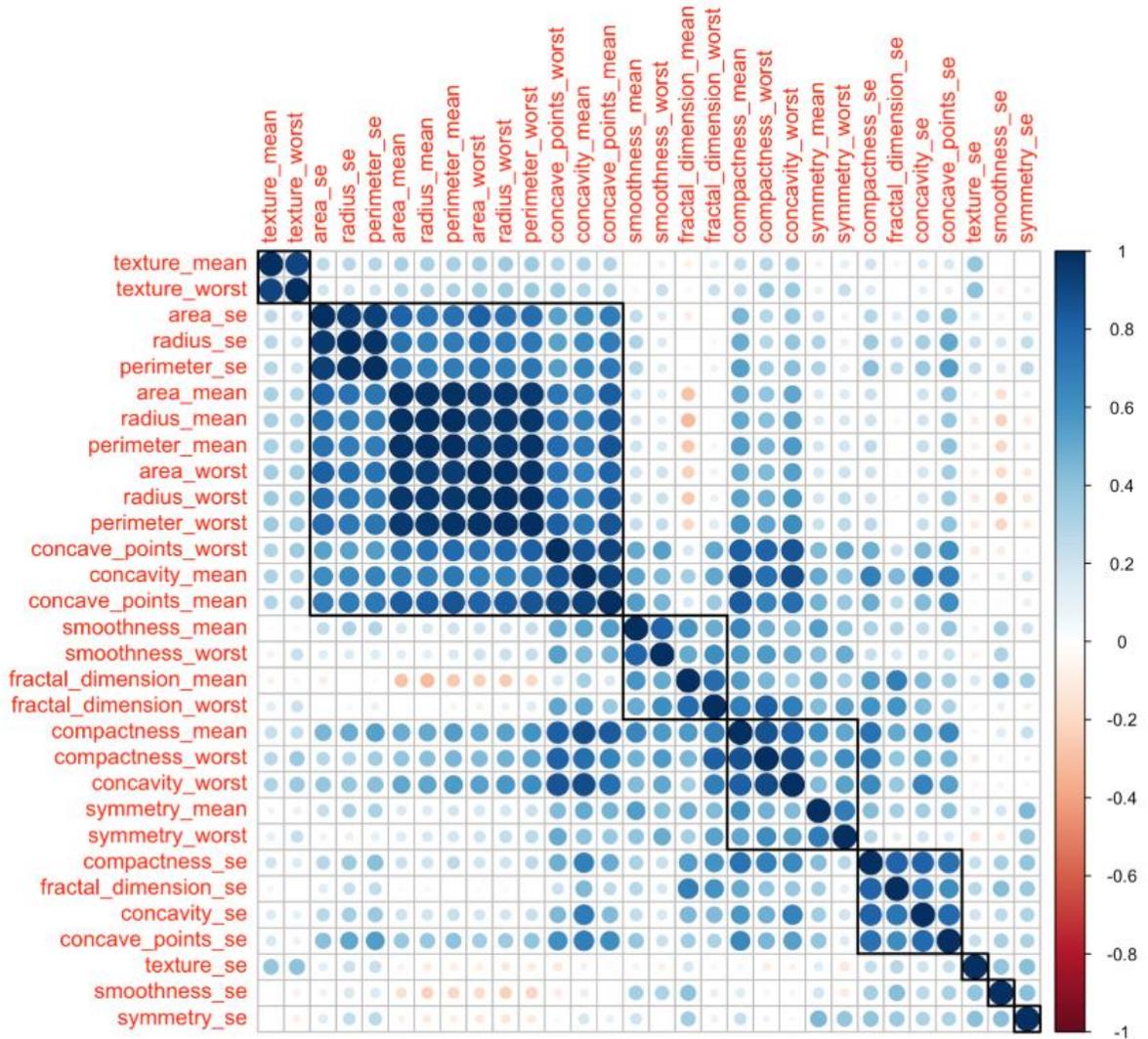
Accuracy Evaluation

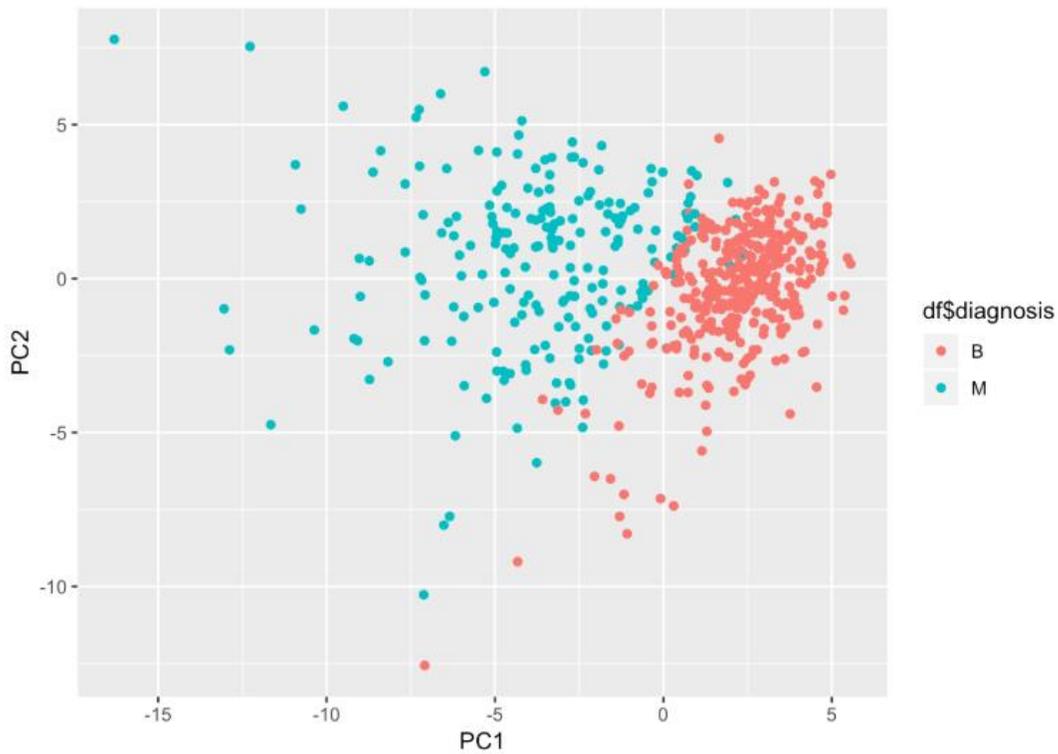
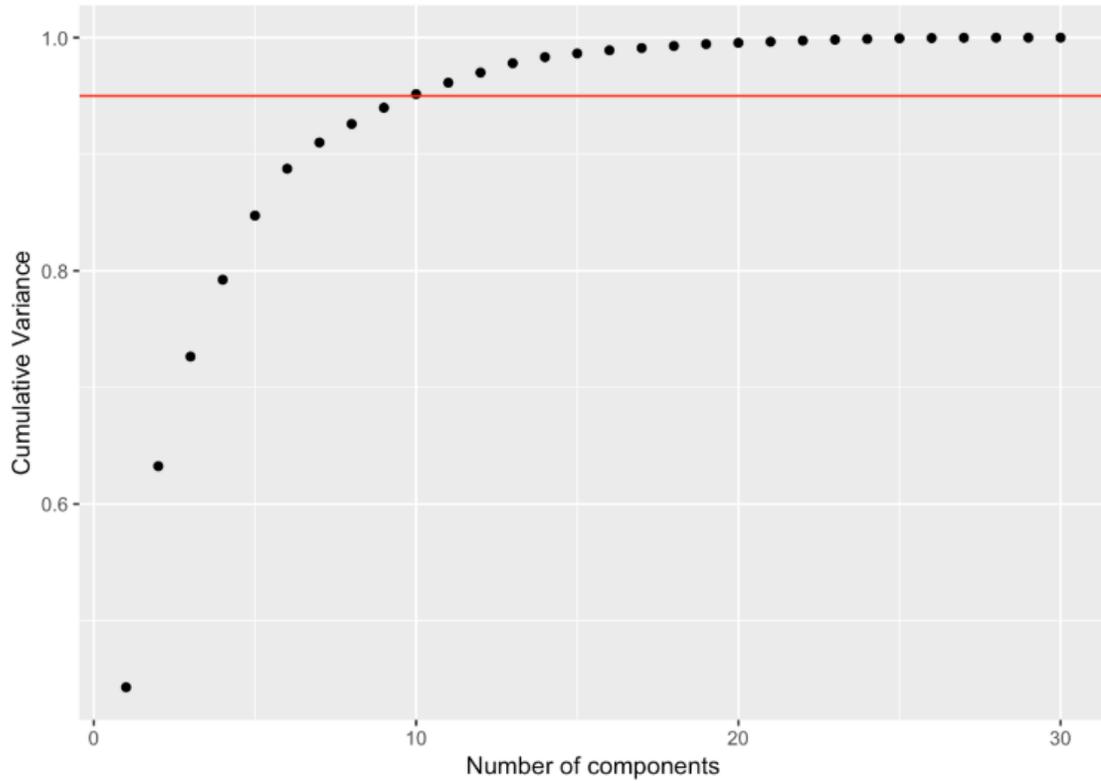
The logistic has to much variability for it to be reliable. The Random Forest and Neural Network with LDA pre-processing are giving the best results. The ROC metric measure the auc of the roc curve of each model. This metric is independent of any threshold. Let's remember how these models result with the testing dataset. Prediction classes are obtained by default with a threshold of 0.5 which could not be the best with an unbalanced dataset

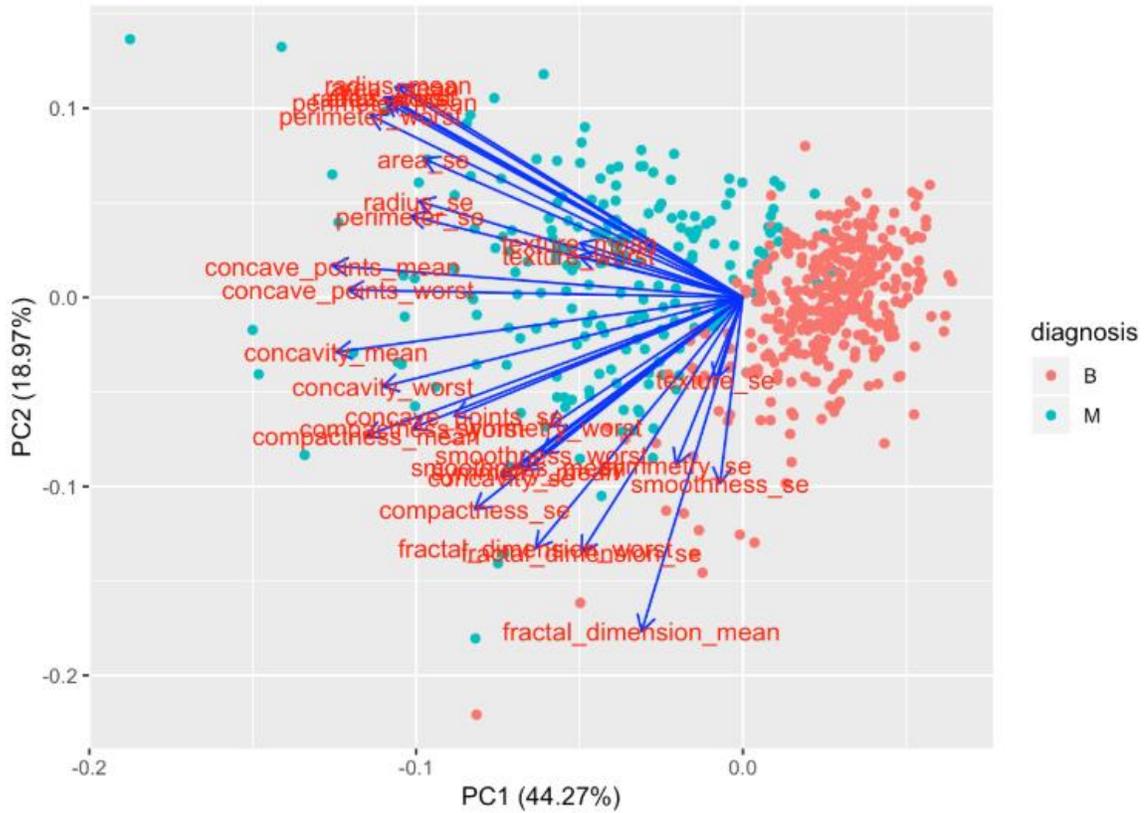
```
## # A tibble: 11 x 5
##   cm_rf cm_svm cm_logistic cm_nnet_LDA stat
##   <dbl> <dbl>    <dbl>    <dbl> <chr>
## 1 0.929 0.952    0.952    0.976 Sensitivity
## 2 1      1          1          1      Specificity
## 3 1      1          1          1      Pos Pred Value
## 4 0.959 0.973    0.973    0.986 Neg Pred Value
## 5 1      1          1          1      Precision
## 6 0.929 0.952    0.952    0.976 Recall
## 7 0.963 0.976    0.976    0.988 F1
## 8 0.372 0.372    0.372    0.372 Prevalence
## 9 0.345 0.354    0.354    0.363 Detection Rate
## 10 0.345 0.354    0.354    0.363 Detection Prevalence
## 11 0.964 0.976    0.976    0.988 Balanced Accuracy
```

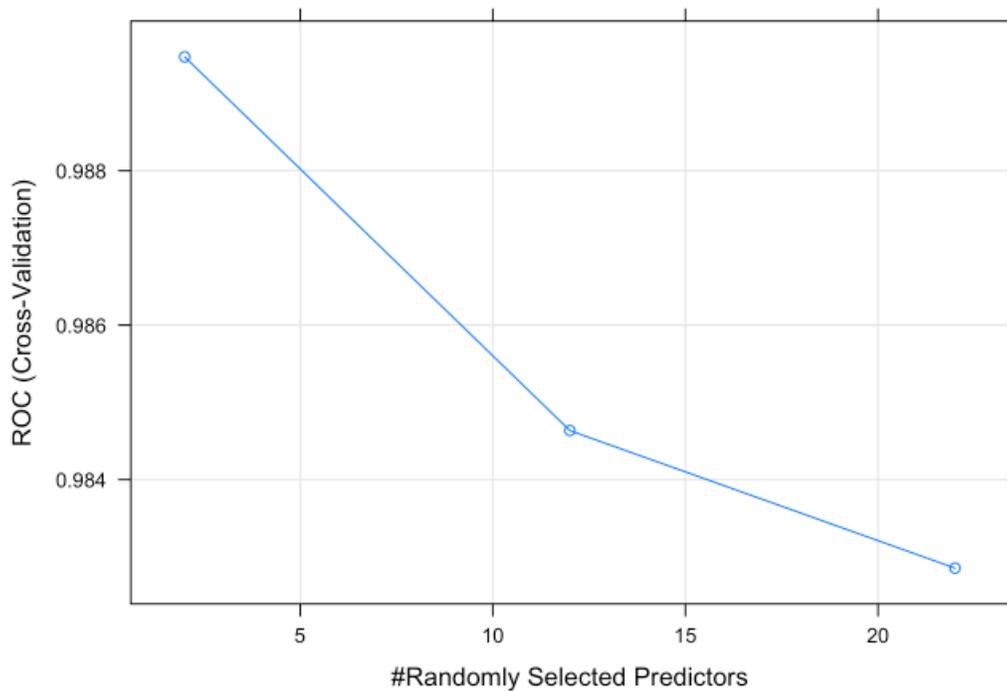
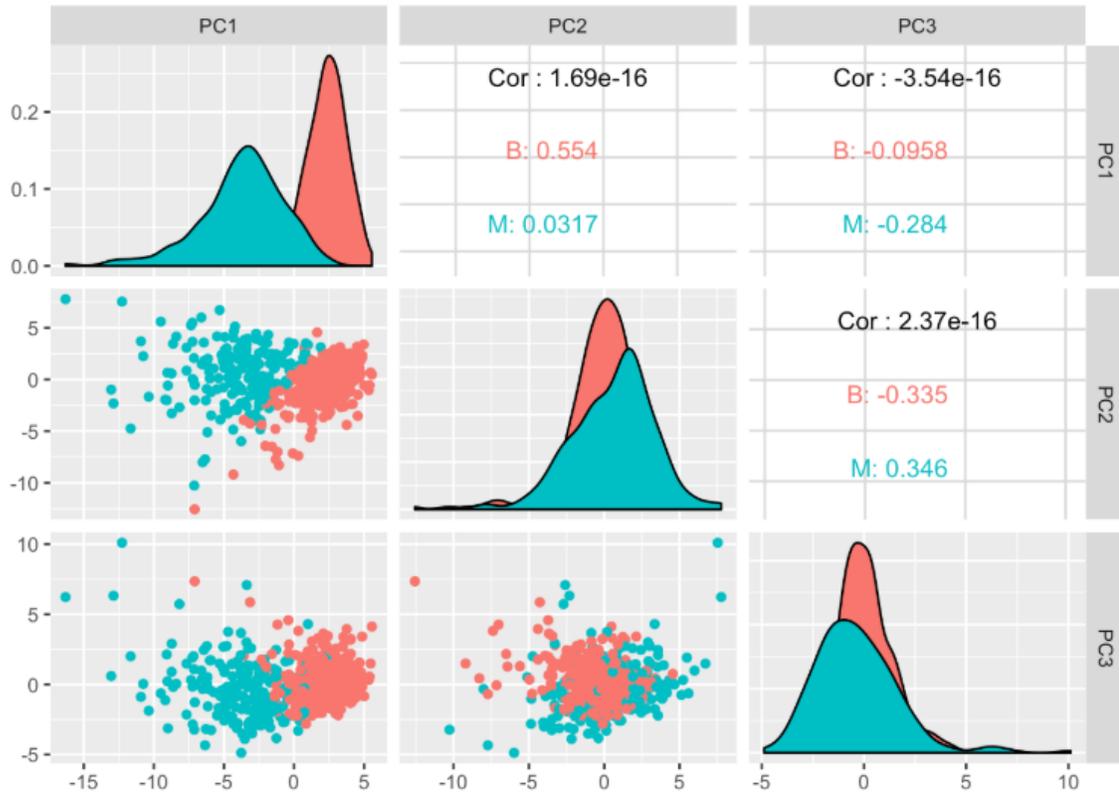
B) Sample Screen Shots

```
##          id          diagnosis          radius_mean
##          0              0              0
## texture_mean      perimeter_mean      area_mean
##          0              0              0
## smoothness_mean  compactness_mean      concavity_mean
##          0              0              0
## concave_points_mean  symmetry_mean  fractal_dimension_mean
##          0              0              0
##          radius_se      texture_se      perimeter_se
##          0              0              0
##          area_se      smoothness_se      compactness_se
##          0              0              0
##          concavity_se  concave_points_se      symmetry_se
##          0              0              0
## fractal_dimension_se      radius_worst      texture_worst
##          0              0              0
##          perimeter_worst      area_worst      smoothness_worst
##          0              0              0
##          compactness_worst      concavity_worst      concave_points_worst
##          0              0              0
##          symmetry_worst  fractal_dimension_worst
##          0              0
```

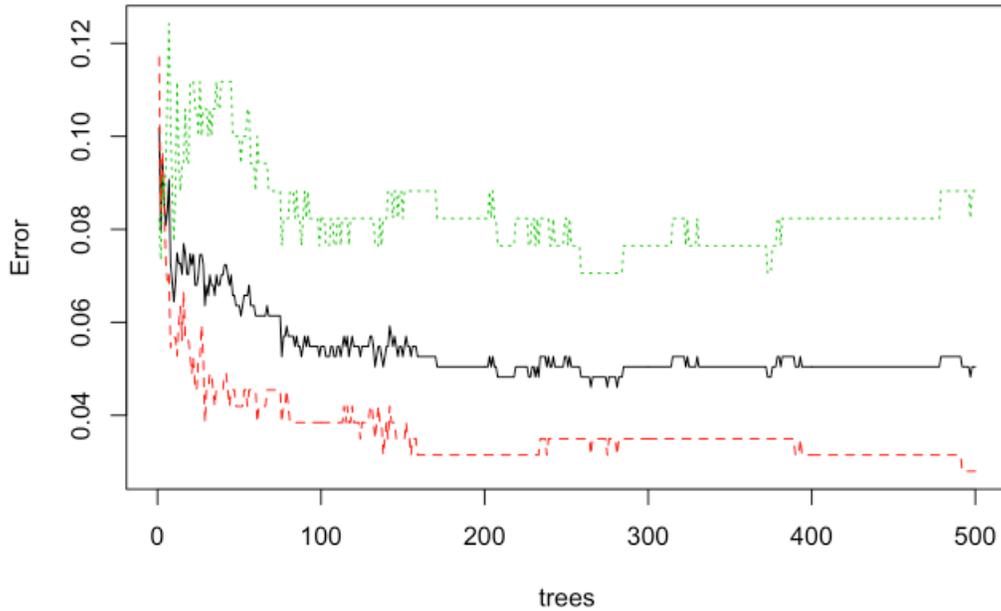




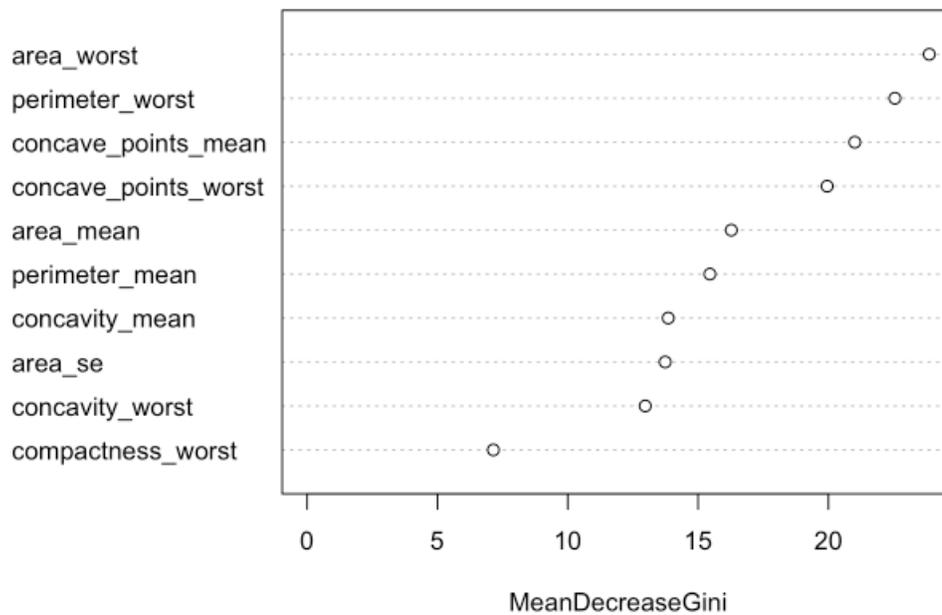


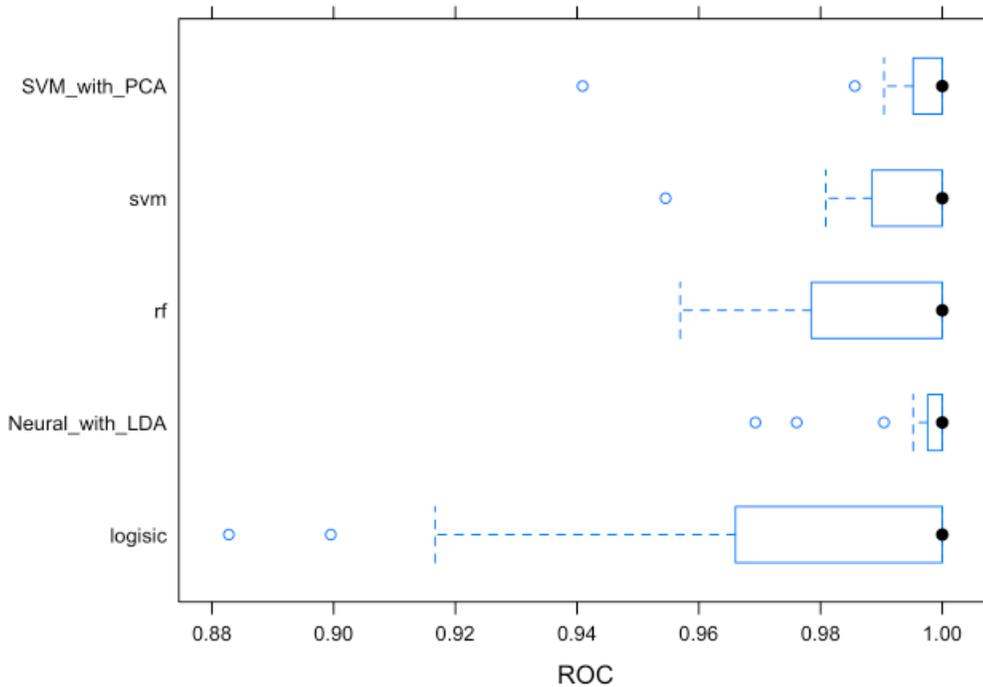
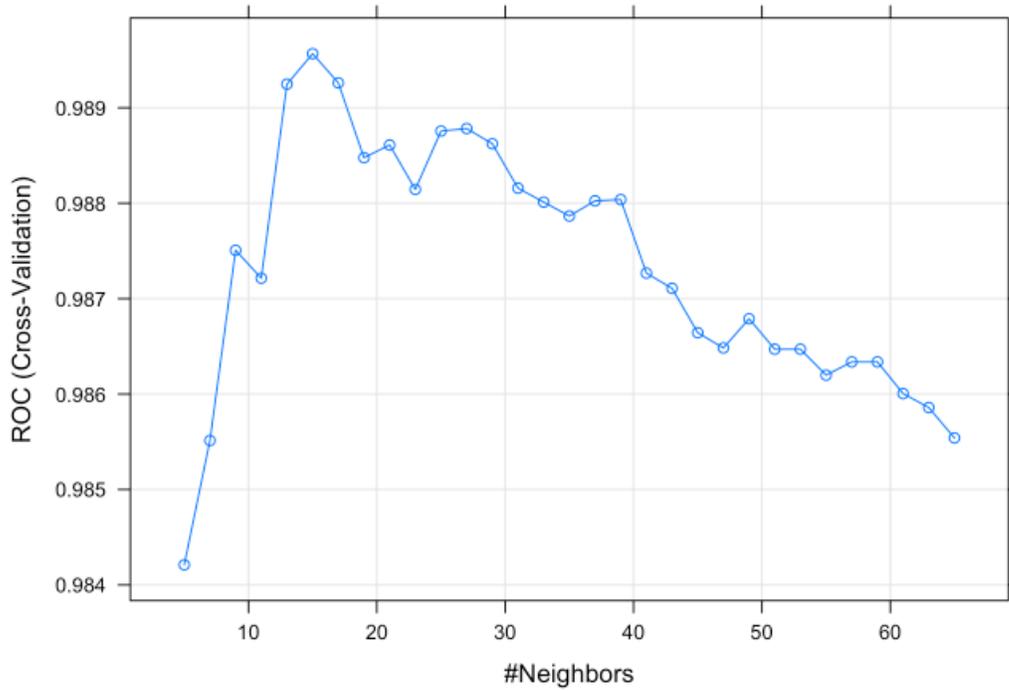


model_rf_df\$finalModel



The 10 variables with the most predictive power





CONCLUSION

The application was designed for medical field in predicting the breast cancer in accurate manner. As per the age and testing, the data get analyzed with the previous patient data which helps to get a solution of prediction level in accurate manner. The time consumption in fixing the level of affection will be more thus to overcome this factor the dataset of previous patient get matched and thus the affecting range will be predicted. The prediction can be done in short duration therefore the treatment can be provided for the patient in an immediate manner which will save many patient life. The best results for sensitivity (detection of breast cases) is LDA_NNET which also has a great F1 score. This analysis aims to observe which features are most helpful in predicting malignant or benign cancer and to see general trends that may aid us in model selection and hyper parameter selection. The goal is to classify whether the breast cancer is benign or malignant. To achieve this I have used machine learning classification methods to fit a function that can predict the discrete class of new input.