PREDICTIVE MODELLING OF DISEASE DATASETS

A Project report submitted in partial fulfillment of requirements for the Degree of M.Sc. (Statistics) with specialization in Industrial Statistics

By

Mr.Deore Vaibhav Manohar
Ms. Dhabu Trupti Appasaheb
Ms. Main Leena Prabhakar

(NAAC Re-accredited A Grade University with CGPA 3.11)

DEPARTMENT OF STATISTICS
SCHOOL OF MATHEMATICAL SCIENCES
NORTH MAHARASHTRA UNIVERSITY
JALGAON – 425001

(2017-2018)

CERTIFICATE

This is to certify that **Mr. Deore Vaibhav Manohar, Ms. Dhabu Trupti Appasaheb, Ms. Main Leena Prabhakar** students of M.Sc.(Statistics) with specialization in Industrial Statistics, at North Maharashtra University, Jalgaon have successfully completed their project work entitled **PREDICTIVE MODELLING OF DISEASE DATASETS** as a part of M.Sc. (Statistics) program under my guidance and supervision during the academic year 2017-2018.

(Prof. R. L. Shinde)
Project Guide

ACKNOWLEDGEMENT

On the completion of this project we must acknowledge from the core of our heart is none other than **Dr.R .L Shinde**, Head department of statistics, North Maharashtra University, Jalgaon and our project guide. His generous attitude and his comments have always guided us throughout and made our work very easy.

The successful completion of our work was possible due to the help, inspiration and guidance rendered by our project guide **Dr.R.L Shinde**. Under his supervision and guidance, we have learned a lot and made us comfortable during the difficult and uneasy situations of the project work. We are thankful to him for his supportive and encouraging nature during the work.

We would like to thank **Dr.K.K Kamalja , Prof.R.D Koshti and Prof.M.C. Patil** for their valuable guidance during the project work.

We also owe thank to friends and all non-teaching staff of Department of Statistics for providing us the required lab and other facilities.

Place: Jalgaon

Date:

Mr.Vaibhav M. Deore (Seat No:387274)

Ms.Trupti A. Dhabu(Seat No:387276)

Ms.Leena P. Main(Seat No:387287)

INDEX

Chapter 1: Overview of the Project

Introduction

As we know today the people of overall world are suffering from many more diseases and there is a need to treat them well and as early as possible. Many more techniques can be applied to such problems but if we have the past data regarding this we can apply our statistical tools and techniques to solve this problem very effectively. Hence we decided to work on this topic and tried to study, how we can apply our statistical knowledge over such problem. we have taken the two disease datasets and tried to analyze these using our statistical concepts like PCA, LDA and classification algorithm to attain classification and furthermore we tried to predict the values of unknown objects which may come to us.

Objective / Aim of the project

- To study and apply the various statistical concepts Which we have learnt in academic program.
- To be familiar with various softwares and the advance techniques of classification.
- To predict the values of response variables based on the disease data
- To make prediction of disease so that one can go through the corresponding treatment as early as possible and simultaneously the consumption of time and money can be achieved.
- To get experience of how to handle the real life datasets.

Motivation

Being a student of M.sc –II (statistics) with specialization in industrial statistics we were interested in study how we can solve the problems where our basic statistical techniques fails. we were then interested to study the advance techniques of classification and some multivariate techniques.

Chapter 2: Statistical Concepts and Tools Used

Chi-square test for independence of attributes

The chi square test of independence is used to determine if there is significant relationship between two nominal (categorical) variables and the frequency of each category for one nominal variable is compared across the categories of the second nominal variable.

Null hypothesis: there is no association between two variables.

Vs

Alternative hypothesis: there is association between two variables.

Interpretation: If both the p-values are greater than 0.05 then there is no evidence that association between two variables.

Correlation and correlation plot

correlation is a statistical measure that indicates the extent to which two or more variables fluctuate together. The correlation coefficient r measures the strength and direction of a linear relationship between two variables on a scatterplot. I. e correlation plot. the r is always lies between +1 to -1.

Principal component analysis

It is the statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principle components often known as a dimensionality reduction technique.

If there are observations with variables then the no of district principal components is min (n-1, p). this transformation is defined in such away that the first principal component has the largest possible variance. And each succeeding component in turn has the highest variance possible under the constraint that is it is orthogonal to the preceding components. The resulting vectors are an I=uncorrelated orthogonal basis set.

PCA is the simplest of the true eigenvector-based multivariate analyses. It is the tool used as a tool in exploratory data analysis and for making predictive models.

Linear discriminant analysis

LDA is the generalization of **Fisher's linear discriminant**, a method used in statistics, pattern recognition and machine learning to find a linear combination of features that characterizes or separates two or more classes of objects or events. The resulting combination may be used as a linear classifier, or, more commonly, for dimensionality reduction before later classification.

LDA is closely related to analysis of variance (ANOVA) and regression analysis, which also attempt to express one dependent variable as a linear combination of other features or measurements. However, ANOVA uses categorical independent variables and a continuous dependent variable,Logistic regression and probity regression are more similar to LDA than ANOVA is, as they also explain a categorical variable by the values of continuous independent variables. These other methods are preferable in applications where it is not reasonable to assume that the independent variables are normally distributed, which is a fundamental assumption of the LDA method. LDA works when the measurements made on independent variables for each observation are continuous quantities.

J48 Algorithm:

Classification is the process of building a model of classes from a set of records that contain class labels. Decision Tree Algorithm is to find out the way the attributes-vector behaves for a number of instances.This algorithm generates the rules for the prediction of the target variable. With the help of tree classification algorithm, the critical distribution of the data is easily understandable.

J48 is an extension of ID3. The additional features of J48 are accounting for missing values, decision trees pruning, continuous attribute value ranges, derivation of rules, etc. In the WEKA data mining tool, J48 is an open source Java implementation of the C4.5 algorithm. In other algorithms the classification is performed recursively till every single leaf is pure, that is the classification of the data should be as perfect as possible.

This algorithm it generates the rules from which particular identity of that data is generated. The objective is progressively generalization of a decision tree until it gains equilibrium of flexibility and accuracy.

Basic Steps in the Algorithm:

(i) In case the instances belong to the same class the tree represents a leaf so the leaf is returned by labeling with the same class.

(ii) The potential information is calculated for every attribute, given by a test on the attribute. Then the gain in information is calculated that would result from a test on the attribute.

(iii) Then the best attribute is found on the basis of the present selection criterion and that attribute selected for branching.

Counting Gain

This process uses the "Entropy" which is a measure of the data disorder. The Entropy is calculated by

$$\text{Information Gain= } I(p,n) = \frac{-p}{p+n}\,\boldsymbol{log_2}\left(\frac{-p}{p+n}\right) - \frac{n}{p+n}\boldsymbol{log_2}\left(\frac{n}{p+n}\right)$$

$$\text{Entropy(A)} = \sum_{i=1}^{v} \frac{p_i+n_i}{p+n} I(p,n)$$

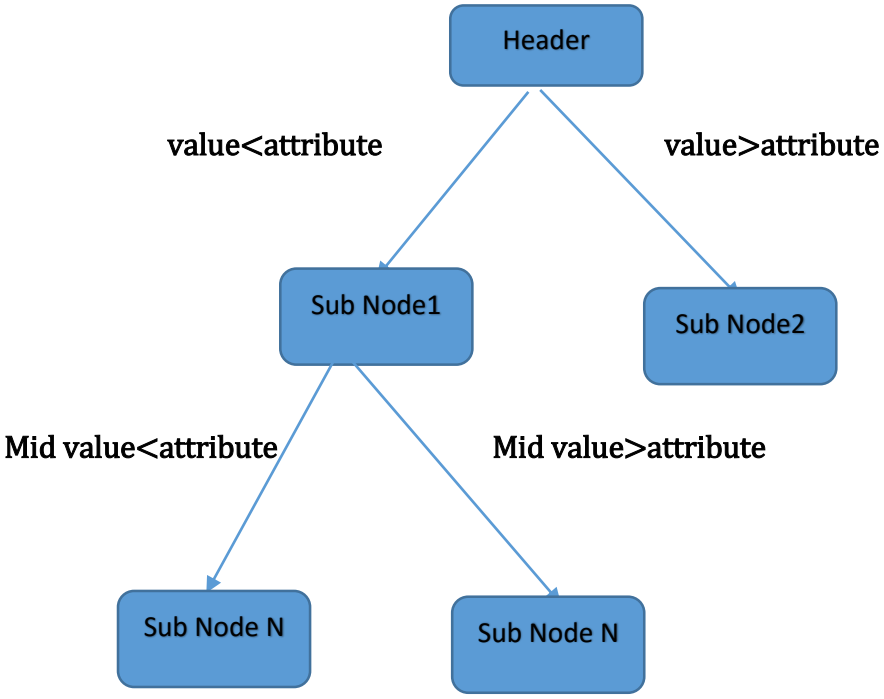
$$\text{Gain(A)} = I(p,n) - \text{Entropy(A)}$$

And Gain is After the tree is fully constructed, this algorithm performs the pruning of the tree. After its construction drives back through the tree and challenges to remove branches that are not helping in reaching the leaf nodes.

After the tree is fully constructed, this algorithm performs the pruning of the tree. after its construction drives back through the tree and challenges to remove branches that are not helping in reaching the leaf nodes.

|

The structure of j48 Decision Tree is as follow



Features of the Algorithm

1. Both the discrete and continuous attributes are handled by this algorithm. A threshold value is decided by for handling continuous attributes. This value divides the data list into those who have their attribute value below the threshold and those having more than or equal to it.
2. This algorithm also handles the missing values in the training data. After the tree is fully constructed, this algorithm performs the pruning of the tree. after its construction drives back
3. through the tree and challenges to remove branches that are not helping in reaching the leaf nodes.

Statistical tools / software used

- **Minitab 17**
- **Rstudio**

  Packages used:

  1. **MASS**
  2. **corrplot**
  3. **Psych**
  4. **Roc**
- **Weka**
- **MS-Excel**

Chapter 3: Overview of Weka Software

1.LAUNCHING WEKA

The Weka GUI Chooser (class weka.gui.GUIChooser) provides a starting point for launching Weka's main GUI applications and supporting tools. If one prefers a MDI ("multiple document interface") appearance, then this is provided by analternative launcher called "Main" (class weka.gui.Main).The GUI Chooser consists of four buttons—one for each of the four major
Weka applications—and four menus.

Fig :1 weka window

The buttons can be used to start the following applications:

• **Explorer**: An environment for exploring data with WEKA (the rest of
this documentation deals with this application in more detail).

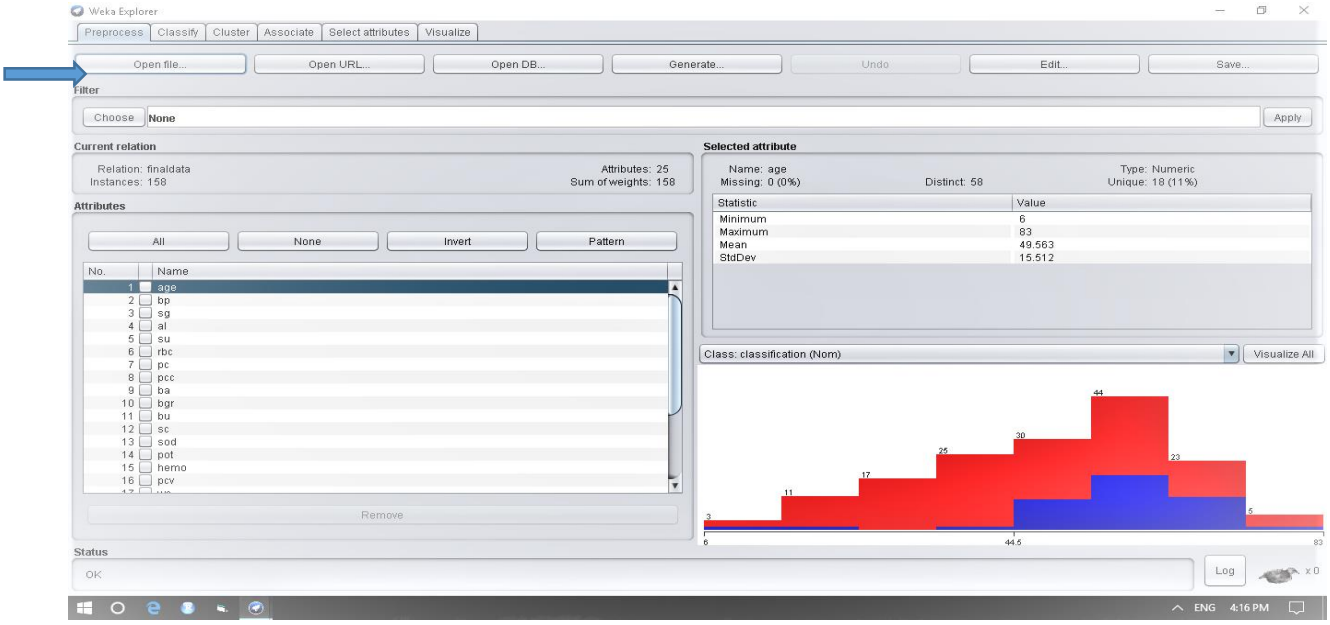
• **Experimenter**: An environment for performing experiments and conducting statistical tests between learning schemes.

• **Knowledge Flow:** This environment supports essentially the same functions as the Explorer but with a drag-and-drop interface. One advantage is that it supports incremental learning.

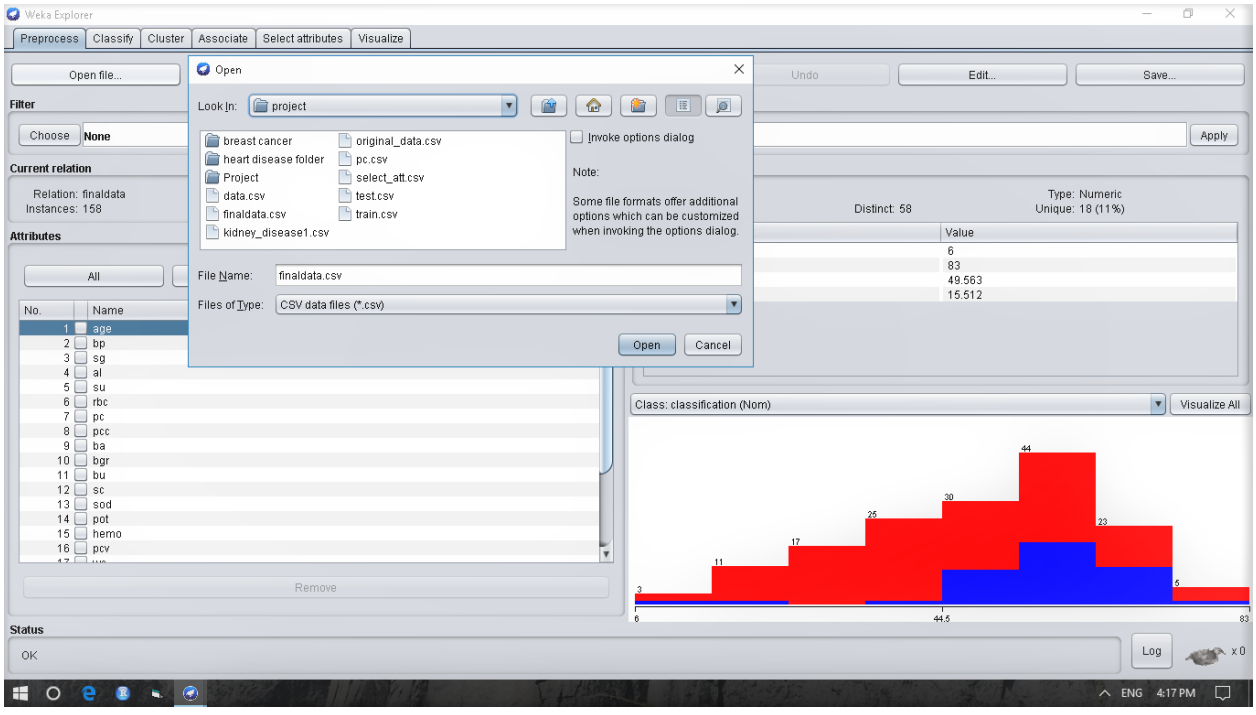
• **Simple CLI :** Provides a simple command-line interface that allows direct execution of WEKA commands for operating systems that do not provide their own command line interface

PREPROCESSING

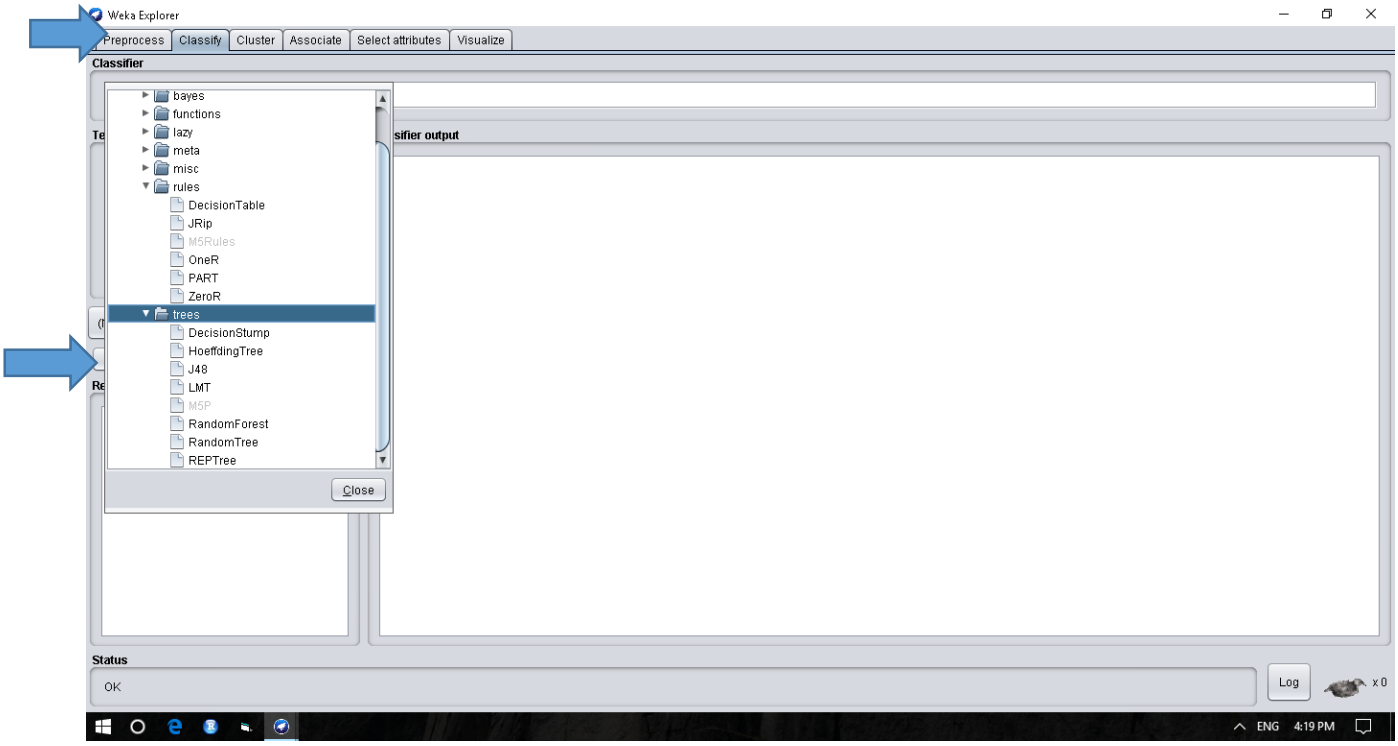
How to import data



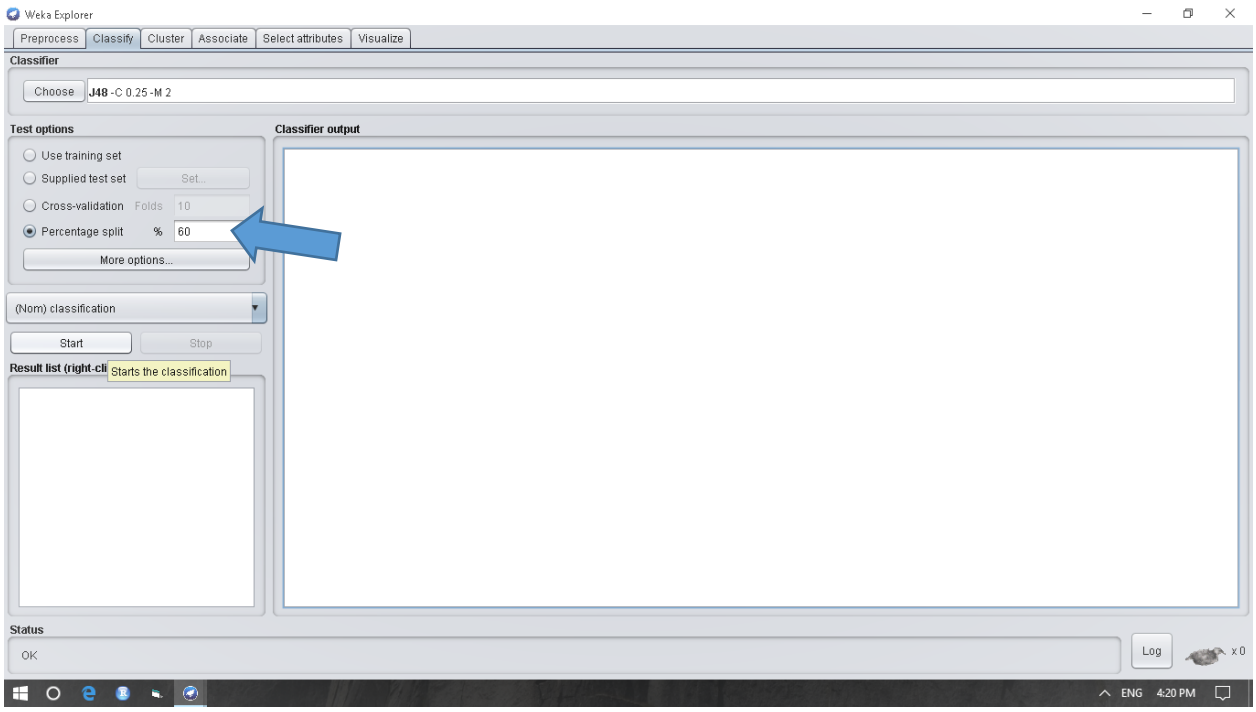
- To import the file click on the " *open file*" and choose the path in which the data file is contained.



Classification

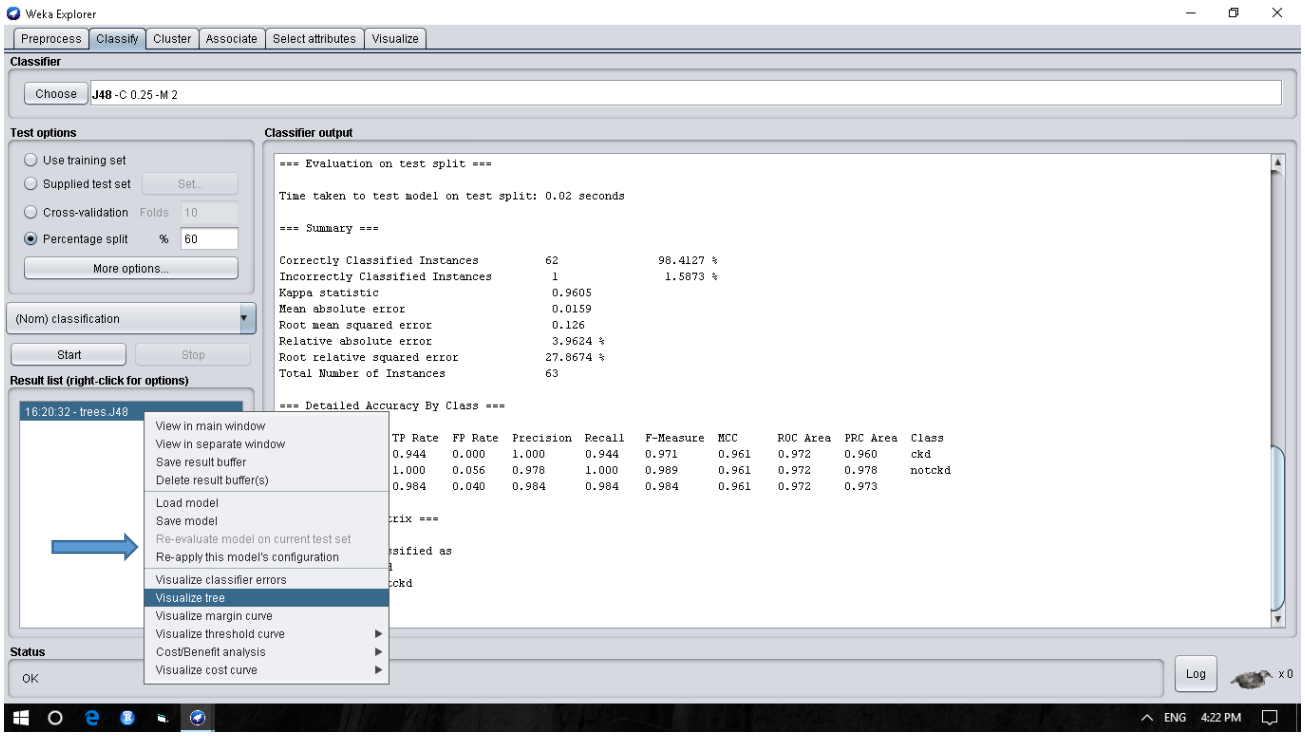


- To choose J48 algorithm go to the classify menu and choose the corresponding algorithm.

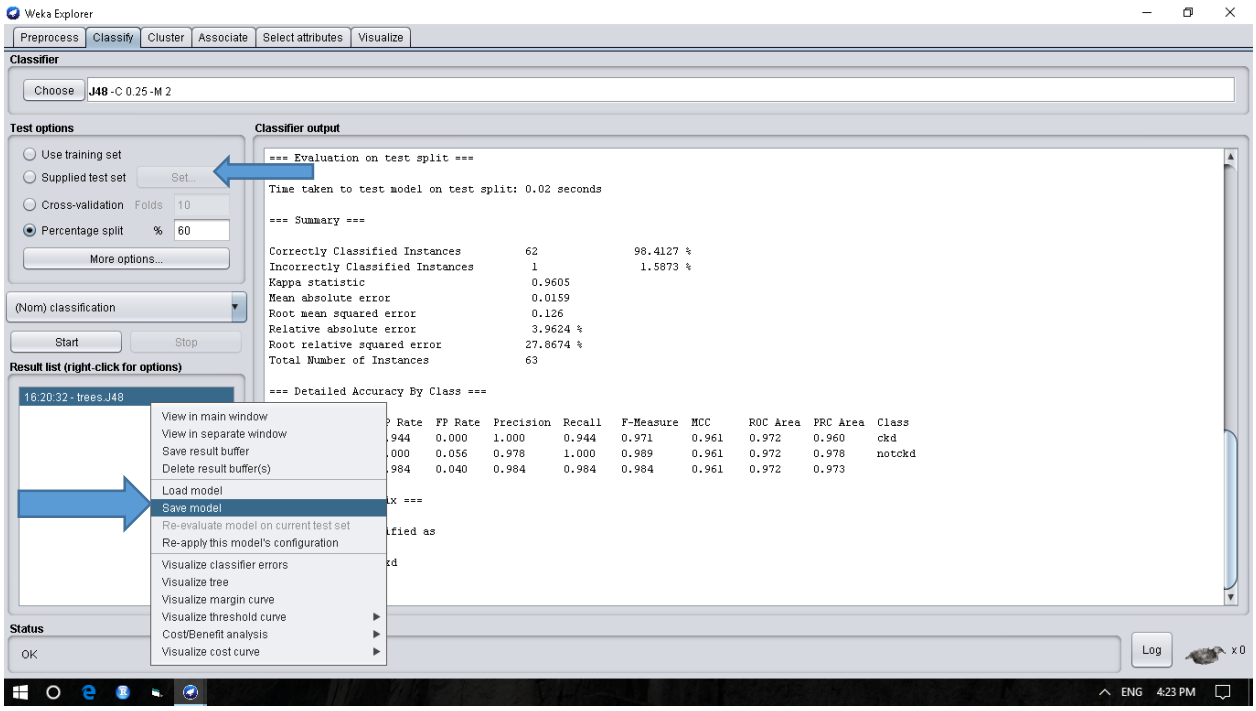


- Once we choose the algorithm we will split the data into 60% of the all the data for training. (the general thumb rule is 60 % for training and 40 % for testing the data).

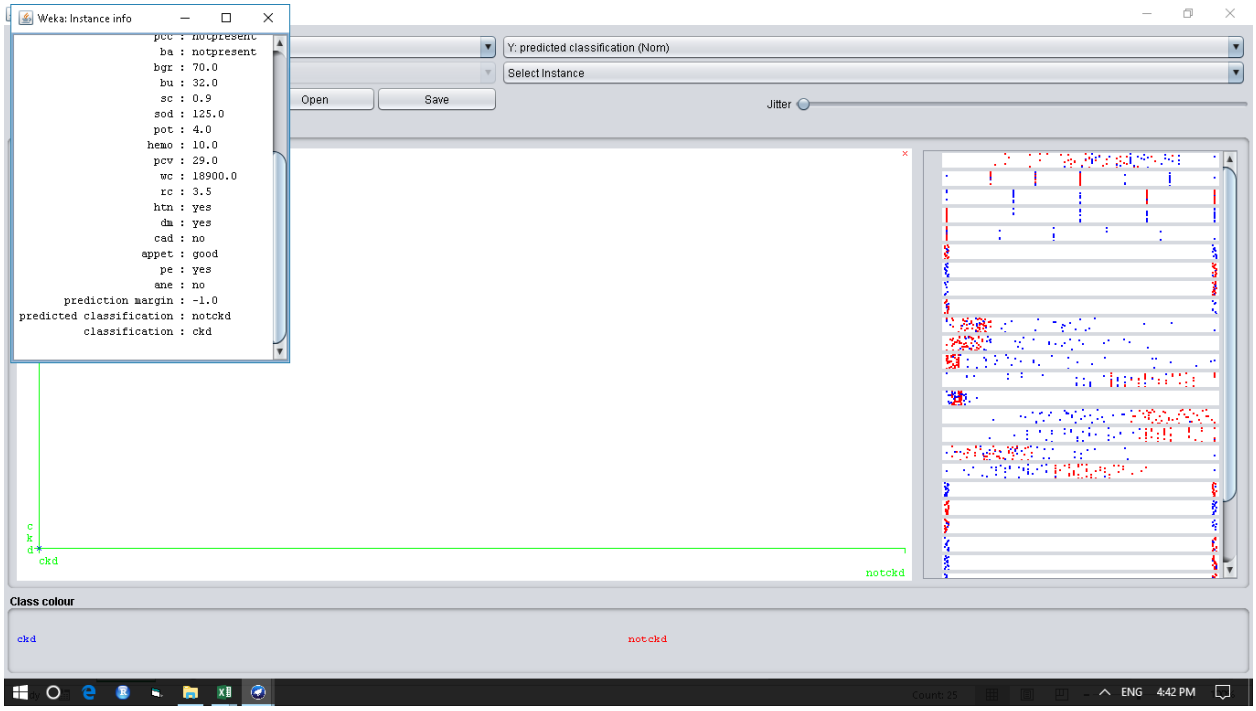
Using this training data we will test the remaining data.



- After performing the algorithm for training data it will give us the output as the above window. One can also see the graphical view of the result through the visualize tree.



- Once the model is formed over the training data save the model. We can also see the classification of corresponding observation and the misclassification of the observation if it is using visualize the classifier errors.



- Once we saved the model we can load the model which contains the unknown observations to be classify.

Terms used :

1. **Use training set.** The classifier is evaluated on how well it predicts the class of the instances it was trained on.

2. **Supplied test set.** The classifier is evaluated on how well it predicts the class of a set of instances loaded from a file. Clicking the Set... button brings up a dialog allowing you to choose the file to test on.

3. **Cross-validation.** The classifier is evaluated by cross-validation, using the number of folds that are entered in the Folds text field.

4. **Percentage split.** The classifier is evaluated on how well it predicts a certain percentage of the data which is held out for testing. The amount of data held out depends on the value entered in the % field.

5. **Output model.** The classification model on the full training set is output so that it can be viewed, visualized, etc. This option is selected by default.

6. **Output per-class stats.** The precision/recall and true/false statistics for each class are output. This option is also selected by default.

7. **Output entropy evaluation measures.** Entropy evaluation measures are included in the output. This option is not selected by default.

8. **Output confusion matrix.** The confusion matrix of the classifier's predictions is included in the output. This option is selected by default.

9. **Store predictions for visualization.** The classifier's predictions are remembered so that they can be visualized. This option is selected by default.

10. **Output predictions.** The predictions on the evaluation data are output. Note that in the case of a cross-validation the instance numbers do not correspond to the location in the data!

Chapter 4 : Data analysis of Chronic Kidney Disease

Chronic Kidney Disease

What is CKD?

Chronic kidney disease (CKD) is a disease which results undying loss of kidney function usually over the course of months or years, Kidneys are responsible for filtering waste from body. The disease not only affect the kidney but also on the other organs to stop functioning properly.

The most common causes of chronic kidney disease are also known as chronic renal disease. Well kidney disease is a disorder in which normal functioning of filtration, reabsorption and secretion etc. is affected.

It has become most important, chronic and mostly no communicable disease epidemics in overall world including India.

Need of analysis

It is estimated that each year in United States more than 100,000 individuals are diagnosed with kidney disease, a condition in which the kidneys fail to remove the body wastes. Similarly, about 175,000 new patients every year in India develop potentially fatal end-stage renal failure. This generates a huge amount of patient's data and requires a proper and efficient way of handling patient recording. a disease is a complicated task in many existing medical expert systems, diagnosing a disease is based on the patient symptoms and other details that are given as input to the system. Several levels of uncertainty are involved in medical diagnosis. However, early identification and detection can help to prevent the headway of kidney disease to kidney failure.

So, *here the aim of analyzing the CKD dataset is to classify the sample dataset as weather it is CKD or NOTCKD and then give a predicted result.* So that one can go through the corresponding treatment as early as possible and simultaneously the consumption of time and money can be achieved.

Data

age	bp	sg	al	su	rbc	pc	pcc	ba	bgr	bu	sc	sod	pot	hemo	pcv	wc	rc	htn	dm	cad	appet	pe	ane	class
48	70	1.005	4	0	normal	abnormal	present	notpresent	117	56	3.8	111	2.5	11.2	32	6700	3.9	yes	no	no	poor	yes	yes	ckd
53	90	1.02	2	0	abnormal	abnormal	present	notpresent	70	107	7.2	114	3.7	9.5	29	12100	3.7	yes	yes	no	poor	no	yes	ckd
63	70	1.01	3	0	abnormal	abnormal	present	notpresent	380	60	2.7	131	4.2	10.8	32	4500	3.8	yes	yes	no	poor	yes	no	ckd
68	80	1.01	3	2	normal	abnormal	present	present	157	90	4.1	130	6.4	5.6	16	11000	2.6	yes	yes	yes	poor	yes	no	ckd
61	80	1.015	2	0	abnormal	abnormal	notpresent	notpresent	173	148	3.9	135	5.2	7.7	24	9200	3.2	yes	yes	yes	poor	yes	yes	ckd
48	80	1.025	4	0	normal	abnormal	notpresent	notpresent	95	163	7.7	136	3.8	9.8	32	6900	3.4	yes	no	no	good	no	yes	ckd
69	70	1.01	3	4	normal	abnormal	notpresent	notpresent	264	87	2.7	130	4	12.5	37	9600	4.1	yes	yes	yes	good	yes	no	ckd
73	70	1.005	0	0	normal	normal	notpresent	notpresent	70	32	0.9	125	4	10	29	18900	3.5	yes	yes	no	good	yes	no	ckd
73	80	1.02	2	0	abnormal	abnormal	notpresent	notpresent	253	142	4.6	138	5.8	10.5	33	7200	4.3	yes	yes	yes	good	no	no	ckd
46	60	1.01	1	0	normal	normal	notpresent	notpresent	163	92	3.3	141	4	9.8	28	14600	3.2	yes	yes	no	good	no	no	ckd
56	90	1.015	2	0	abnormal	abnormal	notpresent	notpresent	129	107	6.7	131	4.8	9.1	29	6400	3.4	yes	no	no	good	no	no	ckd
48	80	1.005	4	0	abnormal	abnormal	notpresent	present	133	139	8.5	132	5.5	10.3	36	6200	4	no	yes	no	good	yes	no	ckd
59	70	1.01	3	0	normal	abnormal	notpresent	notpresent	76	186	15	135	7.6	7.1	22	3800	2.1	yes	no	no	poor	yes	yes	ckd
63	100	1.01	2	2	normal	normal	notpresent	present	280	35	3.2	143	3.5	13	40	9800	4.2	yes	no	yes	good	no	no	ckd
56	70	1.015	4	1	abnormal	normal	notpresent	notpresent	210	26	1.7	136	3.8	16.1	52	12500	5.6	no	no	no	good	no	no	ckd
71	70	1.01	3	0	normal	abnormal	present	present	219	82	3.6	133	4.4	10.4	33	5600	3.6	yes	yes	yes	good	no	no	ckd
73	100	1.01	3	2	abnormal	abnormal	present	notpresent	295	90	5.6	140	2.9	9.2	30	7000	3.2	yes	yes	yes	poor	no	no	ckd
71	60	1.015	4	0	normal	normal	notpresent	notpresent	118	125	5.3	136	4.9	11.4	35	15200	4.3	yes	yes	no	poor	yes	no	ckd
52	90	1.015	4	3	normal	abnormal	notpresent	notpresent	224	166	5.6	133	47	8.1	23	5000	2.9	yes	yes	no	good	no	yes	ckd
50	90	1.01	2	0	normal	abnormal	present	present	128	208	9.2	134	4.8	8.2	22	16300	2.7	no	no	no	poor	yes	yes	ckd
70	100	1.015	4	0	normal	normal	notpresent	notpresent	118	125	5.3	136	4.9	12	37	8400	8	yes	no	no	good	no	no	ckd
60	90	1.01	2	0	abnormal	normal	notpresent	notpresent	105	53	2.3	136	5.2	11.1	33	10500	4.1	no	no	no	good	no	no	ckd
60	60	1.01	3	1	normal	abnormal	present	notpresent	288	36	1.7	130	3	7.9	25	15200	3	yes	no	no	poor	no	yes	ckd

Complete Data set is given in Compact Disk .

Details of Chronic kidney disease dataset:

Data Set	Attribute	Associated	Number of	Number of	Missing
Multivariate	Real	Classification	400	25	Yes

Table1: Details of dataset

Features of Chronic kidney disease dataset

The Chronic Kidney Dataset contains 400 chronic kidney disease patient records with 25 attributes. This dataset contains 250 chronic kidney disease patients records and 150 non chronic kidney disease patient's records.

Number	Attribute	Full form of	Unit	Data type
1	Age	Age	Years	Numerical
2	Bp	Blood pressure	mm/Hg	Numerical
3	Sg	Specific gravity	-	Nominal
4	Al	Albumin	-	Nominal
5	Su	Sugar	-	Nominal
6	Rbc	Red blood cell	-	Nominal
7	Pc	Pus cell	-	Nominal
8	Pcc	Pus cell clumps	-	Nominal
9	Ba	Bacteria	-	Nominal
10	Bgr	Blood glucose	mgs/dl	Numerical
11	Bu	Blood urea	mgs/dl	Numerical
12	SC	Serum creatinine	mgs/dl	Numerical
13	Sod	Sodium	mEq/L	Numerical
14	Pot	Potassium	mEq/L	Numerical
15	Haemo	Heamoglobin	Gms	Numerical
16	Pcv	Packed cell		Numerical
17	Wc	White blood cell	cells/cumm	Numerical
18	Rc	Red blood cell	millions/cmm	Numerical
19	Htn	Hypertension	-	Nominal
20	Dm	Diabeties mellitus	-	Nominal
21	Cad	Coronary artery	-	Nominal
22	Appet	Appetite	-	Nominal
23	Pe	Pedal edema	-	Nominal
24	Ane	Anemia	-	Nominal
25	Class	Classification	-	Nominal

Table 2:data type of each variable

Problem Under Study

In Existing System Chronic Kidney Disease Dataset alone will give the results to the end user. The last field in the dataset is the class label which has two values, *ckd means chronic kidney disease and notckd means non chronic kidney disease*. The Problem with the Existing system is *if unknown sample will came as training data it is difficult to classify the disease.*

Information of parameters

Age: The prevalence of CKD rises dramatically with age.

Blood pressure:

Blood pressure usually ranges between 90 to 250 for the top or maximum number (systolic) and 60 to 140 for the bottom or minimum number (diastolic). A healthy blood pressure is 120/80 or less, but the lower you can get it, the better. When your systolic pressure is between 120 and 129 mm Hg and your diastolic pressure is less than 80 mm Hg, it means you have elevated blood pressure.

Specific gravity(sg) :

Specific gravity is the ratio of the density of a substance to the density of a reference substance; equivalently, it is the ratio of the mass of a substance to the mass of a reference substance for the same given volume. Adults generally have a specific gravity in the range of 1.000 to 1.030 Increases in specific gravity may be associated with dehydration, diarrhea, emesis, excessive sweating, urinary tract/bladder infection, glycosuria, renal artery stenosis.

Albumin :

Albuminuria is a sign of kidney disease and means that you have too much albumin in your urine. Albumin is a protein found in the blood. A healthy kidney doesn't let albumin pass from the blood into the urine. A damaged kidney lets some albumin pass into the urine.

red blood cell :

A red blood cell count is a blood test that your doctor uses to find out how many red blood cells (RBCs) you have. It's also known as an erythrocyte count. The test is important because RBCs contain hemoglobin, which carries oxygen to your body's tissues.

Pus Cell :

The presence of pus cells in urine is called as pyuria and is defined as > 10. Normal no of pus cell are up to 5 in males and may be up to 10 in females.

Pus Cell clumps :

It is usually taken as indicative of infection. increased no of pus cell clumps may revel some healing process in urinary tract anywhere from kidney to bladder.

Bacteria :

Recurrent bacterial infections have more probably impacts on CKD.
The culprit in most urinary tract and kidney infections is uropathogenic E-colli.

Blood glucose random :

A random blood glucose test is used to diagnose diabetes. If your blood glucose level is 200 mg/dL or higher and you have the classic symptoms of high blood sugar (excessive thirst, urination at night, blurred vision and, in some cases, weight loss) your doctor may diagnose you with diabetes.

Serum creatinine

SC is a waste product that comes from muscle activity. A normal serum creatinine range is 0.6-1.1 mg/dL in women and 0.7-1.3 mg/dL in men. As kidney function slows blood levels of creatinine rise below.

Sodium :

A sodium blood test is used to detect abnormal concentrations of sodium, including low sodium and high sodium (hypernatremia). Urine sodium testing is also used for people with abnormal kidney tests to help the healthcare practitioner determine the cause of kidney disease and to help guide treatment.

Potassium :

Potassium is a chemical that is critical to the function of nerve and muscle cells, including those in your heart. Your blood potassium level is normally 3.6 to 5.2 mill moles per liter. When kidneys fail they can no longer remove excess potassium High potassium in the blood is called hyperkalemia, which may occur in people with advanced stages of chronic kidney disease (CKD).

Hemoglobin :

Hemoglobin is a protein in the red blood cells that carries oxygen gives blood its red color. In CKD prefers hb target should be 9-12 gm/dL.

Packed cell volume :

Pcv is the percentage of red blood cells in circulating cells. Increase pcv means generally means dehydration or an abnormal increase in red blood cell production.

White blood cell count:

Wbc are called leucocytes. these are the cells of the human system that are involved in protective the body against both infectious disease and foreign invaders.
Normal range: 4500 – 11500 wbc/ microliter

Red blood cell count:

Are called erythrocytes and these are the most common blood cell deliverying oxygen to the body tissues via blood flow through the circulatory system.
Normal range: Men:4.7-6.1 million cells/ ul
And in Women: 4.2-5.4 million cells/ ul

Appetite:

This decline may be explained by an increase in uremic symptoms, such a nausea and anorexia. Defined as loss of desire to eat or a loss of appetite, develops in 10%-25% of patients with ckd.

Cad:

Leading cause of morbidity and mortality in patients with ckd. The outcomes are poorer in patients with ckd.

Hypertension :

Htn is a major risk factor for the cardiovascular and renal disease. Elevated bp leads to damage of blood vessels within kidney and thought the body.

Diabetes mellitus :

A disease in which the body's ability to produce or respond to the hormone insulin is impaired resulting in abnormal metabolism of carbohydrate and elevated levels of glucose in the blood.

Pedal edema :

Edema is observable swelling from fluid accumulation in body tissues. Edema of the foot is sometimes called pedal edema.

Anemia:

It is the condition in which body has fewer red blood cell than the normal rbc. anemia might begin to develop in early stages of ckd. Most people who have total loss of kidney function have anemia

Data cleaning

As the original data has several missing values we decided to clean the data first by identifying the noisy data and removing them on the basis of outlier.

By using MS-excel:

- we counted the blanks using "count blank"
- sorted them using "count blank"
- Arranged them from smallest to largest.
- Finally took into account the rows which has only zero blank counts.
- At the end we got the data of 158 observations with all parameters under study.

Descriptive statistics:

Variable	Classify	Mean	StDev	Mini	Q1	Median	Q3	Maxi	Range
Age	Ckd	57.28	13.46	6	50	59	64	83	77
	Notckd	46.68	15.29	12	34	46	58	80	68
Bp	ckd	80	14.47	50	70	80	90	110	60
	notckd	71.826	8.744	60	60	70	80	80	20
Sg	ckd	1.0128	0.00504	1.005	1.01	1.01	1.015	1.025	0.02
	notckd	1.0225	0.00251	1.02	1.02	1.025	1.025	1.025	0.005
Al	ckd	2.93	1.033	0	2	3	4	4	4
	notckd	0	0	0	0	0	0	0	0
Su	ckd	0.93	1.352	0	0	0	2	5	5
	notckd	0	0	0	0	0	0	0	0
Bgr	ckd	193.9	96.5	70	117	173	253	490	420
	notckd	107.94	18.71	70	94	108	124	140	70
Bu	ckd	104.93	64.54	26	54	90	148	309	283
	notckd	33	11.77	10	23	34	44	50	40
Sc	ckd	5.712	4.215	0.9	2.6	4.1	7.7	15.2	14.3
	notckd	0.8713	0.2585	0.4	0.6	0.9	1.1	1.2	0.8
Sod	ckd	131.02	7.92	111	125	133	136	143	32
	notckd	141.77	4.74	135	138	141	146	150	15
Pot	ckd	5.51	6.57	2.5	3.8	4.6	5.4	47	44.5
	notckd	4.3113	0.5978	3.3	3.7	4.5	4.9	5	1.7
Hemo	ckd	9.77	2.172	3.1	8.3	9.8	11.1	16.1	13
	notckd	15.152	1.322	13	14	15	16.2	17.8	4.8
Pcv	ckd	29.63	7.17	9	25	30	34	52	43
	notckd	46.513	4.12	40	43	46	50	54	14
Wc	ckd	10553	4671	3800	7000	9800	12800	26400	22600
	notckd	7699	1786	4300	6300	7300	9300	11000	6700
Rc	ckd	3.695	0.956	2.1	3.2	3.7	4.1	8	5.9
	notckd	5.3391	0.5935	4.5	4.8	5.3	5.8	6.5	2

Table:3 Summary of parameter

Chi square test for independent of attribute:

Here we have taken the categorical variables listed below, and checked weather there is any association between these parameters and our classification or we may say response. For this we have used chi-square Tet of independence.

Hypothesis:

H0: There is no association between two variables

H1: There is association between two variables

sr.no	1	2	3	4	5	6	7	8	9	10
Variable	Htn	Dm	Rbc	Pc	Pcc	Ba	Cad	pedal edema	Anemia	Appetite
p-value	0	0	0	0	0	0	0	0	0	0

Table:4 p-values of parameters

Conclusion:

Here all p-values are less than 0.05, hence we reject H0 so, there is greater association between Htn, dm, rbc, pc, pcc, ba, cad, pedal edema, anemia, appetite and Classification.

Correlation(plot)between predictors:

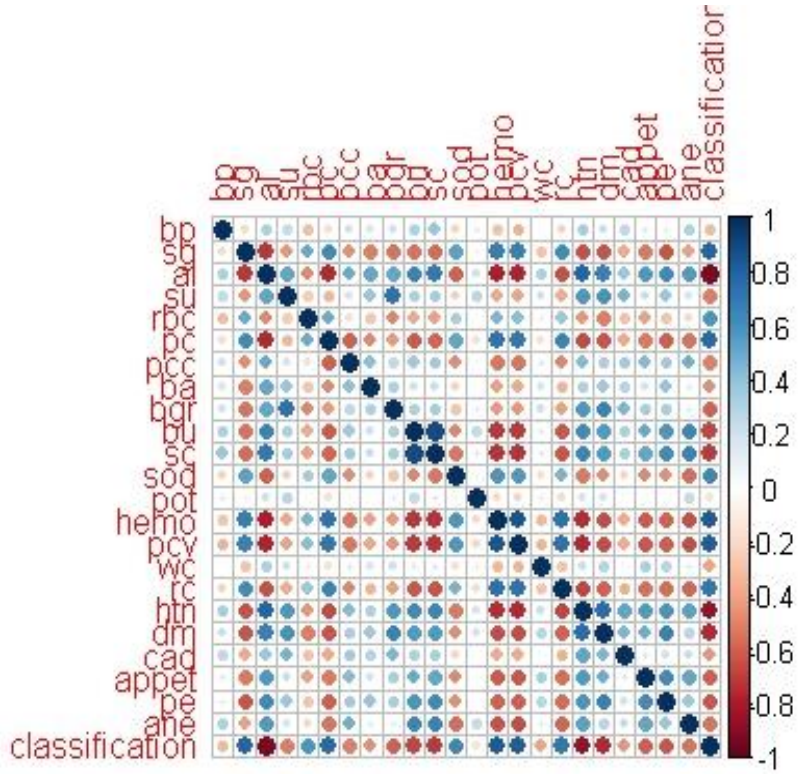


Fig :8 Correlation Plot of Variables

Conclusion:

From the above correlation plot we see that the most of the parameters are highly correlated with each other (since the color red and blue shows us the correlation between parameters are beyond 0.6 and -0.6 respectively).

NEED OF CLASSIFICATION USING DATA MINING TECHNIQUES:

To solve the real datasets or the complicated datasets where the problems like high multicolinearity or complexion such as multidimensional data or unstructured data we need some advance analytics which uses the basic statistical concepts.

Thus we came to such data mining techniques to analyze the data more precisely. Here we have used J48 classification algorithm to classify and to have prediction on sample observations using weka software.

Analysis:

The study aimed diagnosis and prediction of disease using the data set that composed of data of 158 patients with chronic kidney disease. First, the chronic kidney disease data was classified with machine learning algorithms and then training and test results were analyzed

Classification using J48 technique:

Test I)

Scheme: weka. classifiers. trees. J48 -C 0.25 -M 2
Relation: final data
Instances: 158
Attributes: 25
Test mode: split 10.0% train, remainder test

=== Classifier model (full training set) ===

J48 pruned tree

al <= 0: notckd (116.0/1.0)
al > 0: ckd (42.0)

Number of Leaves: 2
Size of the tree: 3
Time taken to build model: 0 seconds

=== Evaluation on test split ===
Time taken to test model on test split: 0.01 seconds

=== Summary ===

Correctly Classified Instances	135	95.0704(%)
Incorrectly Classified	7	4.9296(%)
Kappa statistic	0.8608	
Mean absolute error	0.0493	
Root mean squared error	0.222	
Relative absolute error	10.4305	%
Root relative squared error	46.7347	%
Total Number of Instances	142	

Table 5: : Summary of Test I

=== Detailed Accuracy by Class ===

	FP	Precision	Recall	F-Measure	MCC	ROC	PRC	Class
0.806	0	1	0.806	0.892	0.869	0.903	0.855	ckd
1	0.194	0.938	1	0.968	0.869	0.903	0.938	notckd
0.951	0.145	0.954	0.951	0.949	0.869	0.903	0.917	Wei.avg

=== Confusion Matrix ===

```
  a   b   <-- classified as
 29   7 |  a = ckd
  0 106 |  b = notckd
```

Test II)

```
=== Run information ===
Scheme:       weka.classifiers.trees.J48 -C 0.25 -M 2
Relation:     finaldata
Instances:    158
Attributes:   25
classification
Test mode:    split 66.0% train, remainder test

=== Classifier model (full training set) ===
J48 pruned tree
------------------
al <= 0: notckd (116.0/1.0)
al > 0: ckd (42.0)
Number of Leaves  :  2
Size of the tree :  3
Time taken to build model: 0.02 seconds

=== Evaluation on test split ===
Time taken to test model on test split: 0 seconds
```

=== Summary ===

Correctly Classified Instances	53	98.1481(%)
Incorrectly Classified	1	1.8519(%)
Kappa statistic	0.9548	
Mean absolute error	0.0185	
Root mean squared error	0.1361	
Relative absolute error	4.5848	%
Root relative squared error	29.7284	%
Total Number of Instances	54	

Table 6: Summary of Test II

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC	PRC	Class
0.938	0	1	0.938	0.968	0.956	0.969	0.956	ckd
1	0.063	0.974	1	0.987	0.956	0.969	0.974	Notckd
0.981	0.044	0.982	0.981	0.981	0.956	0.969	0.969	Wei.Avg.

=== Confusion Matrix ===

a	b	<-- classified as
15	1	| a = ckd
0	38	| b = notckd

Algorithm test result:

=== Summary ===	Test II		Test I	
Correctly Classified Instances	53	98.15%	135	95.07%
Incorrectly Classified Instance	1	1.85%	7	4.93%
Kappa statistic	0.9548		0.8608	
Mean absolute error	0.0185	0.0493		
Root mean squared error	0.1361		0.222	
Relative absolute error	4.58%		10.43%	
Root relative squared error	29.73%		46.73%	
Total Number of Instances	54		142	

Table:7 comparisons of two test

Description of Output of j48:

Characteristics required for Classification Algorithm:

In this work, we have focused on the following three measures namely correctly classified instances, incorrectly classified instances, and accuracy.

(i) **Correctly classified instance**: These are the instances which are correctly classified by any classification algorithm. Percentage of correctly classified instances is called as **accuracy**.

(ii) **Incorrectly classified instances**: These instances are not correctly classified by the algorithm. Sometimes it is observed that the data which is incorrectly classified may contain inconsistent data, noisy data or data out of scope.

(iii) **Accuracy**: Accuracy is how a measured value is closed to the true value. The general formula is given below: Accuracy $=\frac{Tp+Tn}{P+N}$ (1) where, Tp indicates True positive, Tn indicates True negative, P indicates total positive, N indicates total negative and $P = Tp + Fn$, $N = Fp + Tn$.

- In classification system, the algorithm with highest accuracy will be selected for the prediction. Accuracy of the algorithm varies according to the dataset used. So before using the algorithms for prediction system, we must check the accuracy of the algorithm. So it will reduce the cost of doing trial and error of using algorithms in the prediction system.
- performance Evaluation 10-fold cross validation technique is used to evaluate the performance classification methods, Data set is randomly sub divided into ten equal sized partitions. Among the partitions nine of them are used as training set and the remaining one is used as a test set. Evaluation of performance is compared using Mean absolute error, mean squared error, Receiver Operating Characteristic (ROC) Area and Kappa statistics.
- Large test sets give a good assessment of the classifier's performance and small training sets which result in a poor classifier.

iv) **Kappa Statistics**: Kappa Statistics measure degree of agreement between two sets of categorized data. Kappa result varies between 0 to 1 intervals. Higher the value of Kappa means stronger the agreement. Kappa is a normalized value of agreement for chance of agreement.

K = $P\ A\ -(E)\ 1-P(E)$ Where

P(A) = percentage of agreement

P(E) = chance of agreement.

If K =1 agreement is perfect between the classifier and ground truth.

If K=0 indicates there is a chance of agreement.

V) Mean Absolute Error (MAE) : The mean absolute error (MAE) is a quantity used to measure predictions of the eventual outcomes. The mean absolute error is given by **MAE** = $1\ n\ |f\ i - yi|n\ i{=}1\ \ = 1\ n\ \ |ei|\ n\ i{=}1$ The mean absolute error is an average of the absolute errors ea. = | fi – yen |, where fi = prediction, yen = true value.

VI) Root Mean Squared Error (RMSE): Root mean squared error is the square root of the mean of the squares of the values. It squares the errors before they are averaged and RMSE gives a relatively high weight to large errors.

INTERPRTATION:

As it is seen from the confusion matrix of algorithms;
J48 classification algorithm was making some mistake in Test-1 as it is classifying the 7 observations as no patient and error is 4.9296%.
However, only 1 patient was classified as non-patient and the error is 1.85% in Test-2.

Chapter 5. Breast Cancer Data Analysis

What is breast cancer?

Breast cancer is a disease in which *malignant (cancer)* cells form in the tissues of the breast. A cancer that forms in the cells of the breasts. Breast cancer starts when cells in the breast begin to divide and grow in an abnormal way. It's caused by a combination of lots of different factors, many of which are beyond our control.

Need of analysis:

Breast cancer is the most common cancer type of cancer among women and rarely in men. This is one of the common cause of death in the world but we are unware of the fact. Every year approximately 124 out of 100,000 women are diagnosed with breast cancer, and the estimation is that 23 out of the 124 women will die of this disease. *This generates a huge amount of patient data and requires a proper and efficient way of handling patient recording.* A diagnosis of disease is a complicated task in many existing medical expert systems, diagnosing a disease is based on the patient's measurement of cell nucleus and other details that are given as input to the system. Several levels of uncertainty are involved in medical diagnosis. However, early identification and detection can help to prevent the headway of breast cancer. *So, here the aim of analyzing the breast cancer dataset is to classify the sample dataset as weather it is malignant or benign and then give a predicted result.*

Breast Cancer Data

Id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	symmetry_mean	fractal_dimension_mean	radius_se	texture_se	perimeter_se	area_se	smoothness_se
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
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
84300903	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
84348301	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
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
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
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
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
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
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
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
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
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
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
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
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
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
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
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
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
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
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
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
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
Data continue (Last 30 observation)																
921092	B	7.729	25.49	47.98	178.8	0.08098	0.04878	0	0	0.187	0.07285	0.3777	1.462	2.492	19.14	0.01266
921362	B	7.691	25.44	48.34	170.4	0.08668	0.1199	0.09252	0.01364	0.2037	0.07751	0.2196	1.479	1.445	11.73	0.01547
921385	B	11.54	14.44	74.65	402.9	0.09984	0.112	0.06737	0.02594	0.1818	0.06782	0.2784	1.768	1.628	20.86	0.01215
921386	B	14.47	24.99	95.81	656.4	0.08837	0.123	0.1009	0.0389	0.1872	0.06341	0.2542	1.079	2.615	23.11	0.007138

compactness_se	concavity_se	concave points_se	symmetry_se	fractal_dimension_se	radius_worst	texture_worst	perimeter_worst	area_worst	smoothness_worst	compactness_worst	concavity_worst	concave points_worst	symmetry_worst	fractal_dimension_worst
0.04653	0.03829	0.01162	0.02068	0.006111	16.22	31.73	113.5	808.9	0.134	0.4202	0.404	0.1205	0.3187	0.1023
0.01172	0.01947	0.01269	0.0187	0.002626	16.51	32.29	107.4	826.4	0.106	0.1376	0.1611	0.1095	0.2722	0.06956
0.01372	0.01498	0.009117	0.01724	0.001343	14.37	37.17	92.48	629.6	0.1072	0.1381	0.1062	0.07958	0.2473	0.06443
0.02172	0.02615	0.009061	0.0149	0.003599	15.05	24.75	99.17	688.6	0.1264	0.2037	0.1377	0.06845	0.2249	0.08492
0.02099	0.02021	0.009064	0.02087	0.002583	15.35	29.09	97.58	729.8	0.1216	0.1517	0.1049	0.07174	0.2642	0.06953
0.007247	0.01012	0.005495	0.0156	0.002606	11.25	21.77	71.12	384.9	0.1285	0.08842	0.04384	0.02381	0.2681	0.07399
0.03084	0.02613	0.01097	0.02277	0.00589	10.83	22.04	71.08	357.4	0.1461	0.2246	0.1783	0.08333	0.2691	0.09479
0.01123	0.02337	0.009615	0.02203	0.004154	10.93	25.59	69.1	364.2	0.1199	0.09546	0.0935	0.03846	0.2552	0.0792
0.0187	0.01277	0.005917	0.02466	0.002977	13.03	31.45	83.9	505.6	0.1204	0.1633	0.06194	0.03264	0.3059	0.07626
0.01104	0	0	0.03004	0.002228	11.66	24.77	74.08	412.3	0.1001	0.07348	0	0	0.2458	0.06592
0.03051	0.03445	0.01024	0.02912	0.004723	12.02	28.26	77.8	436.6	0.1087	0.1782	0.1564	0.06413	0.3169	0.08032
0.01233	0.01328	0.009305	0.01897	0.001726	13.87	36	88.1	594.7	0.1234	0.1064	0.08653	0.06498	0.2407	0.06484
0.01834	0.03996	0.01282	0.03759	0.004623	9.845	25.05	62.86	295.8	0.1103	0.08298	0.07993	0.02564	0.2435	0.07393
0.02153	0.03898	0.00762	0.01695	0.002801	13.89	35.74	88.84	595.7	0.1227	0.162	0.2439	0.06493	0.2372	0.07242
0.02736	0.04804	0.01721	0.01843	0.004938	10.84	34.91	69.57	357.6	0.1384	0.171	0.2	0.09127	0.2226	0.08283
0.02222	0.004174	0.007082	0.02572	0.002278	10.65	22.88	67.88	347.3	0.1265	0.12	0.01005	0.02232	0.2262	0.06742
0.01124	0	0	0.03004	0.003324	10.49	34.24	66.5	330.6	0.1073	0.07158	0	0	0.2475	0.06969
0.04639	0.06578	0.01606	0.01638	0.004406	15.48	27.27	105.9	733.5	0.1026	0.3171	0.3662	0.1105	0.2258	0.08004
0.02982	0.05738	0.01267	0.01488	0.004738	12.48	37.16	82.28	474.2	0.1298	0.2517	0.363	0.09653	0.2112	0.08732
0.02678	0.02071	0.01626	0.0208	0.005304	15.3	33.17	100.2	706.7	0.1241	0.2264	0.1326	0.1048	0.225	0.08321
0.008878	0	0	0.01989	0.001773	11.92	38.3	75.19	439.6	0.09267	0.05494	0	0	0.1566	0.05905
0.04844	0.07359	0.01608	0.02137	0.006142	17.52	42.79	128.7	915	0.1417	0.7917	1.17	0.2356	0.4089	0.1409
0.0431	0.07845	0.02624	0.02057	0.006213	24.29	29.41	179.1	1819	0.1407	0.4186	0.6599	0.2542	0.2929	0.09873
0.02891	0.05198	0.02454	0.01114	0.004239	25.45	26.4	166.1	2027	0.141	0.2113	0.4107	0.2216	0.206	0.07115
0.02423	0.0395	0.01678	0.01898	0.002498	23.69	38.25	155	1731	0.1166	0.1922	0.3215	0.1628	0.2572	0.06637
0.03731	0.0473	0.01557	0.01318	0.003892	18.98	34.12	126.7	1124	0.1139	0.3094	0.3403	0.1418	0.2218	0.0782
0.06158	0.07117	0.01664	0.02324	0.006185	25.74	39.42	184.6	1821	0.165	0.8681	0.9387	0.265	0.4087	0.124
0.00466	0	0	0.02676	0.002783	9.456	30.37	59.16	268.6	0.08996	0.06444	0	0	0.2871	0.07039

Complete Data is given in Compact Disk.

Details of Wisconsin Breast cancer dataset

Data set characteristics	Multivariate
Attribute characteristics	Real
Associated task	PCA,LDA,Classification
Number of instances	569
Number of variables	30
Missing values	None

Table:1 Information of data

Features of Wisconsin Breast cancer dataset:

- The breast cancer Dataset contains 569 patients records with 30 attributes. This dataset contains 212 malignant patients records and 357 benign patient's records.
- The dataset contains fluid samples, taken from patients with solid breast masses.
- The technique used to detect the breast cancer is FNA that is fine needle aspiration and the parameters used are *measurements of cell nucleus. each feature is evaluated on continuous scale.*

Information of parameters:

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

are real-valued features are computed for each cell nucleus

Problem Under Study

In Existing System Breast Cancer Disease Dataset alone will give the results to the end user. The last field in the dataset is the class label which has two values, Malignant means cancerous cells and benign means non-cancerous. The Problem with the Existing system is if unknown sample will came as training data it is difficult to classify the disease.

Aim of the analysis:

To predict weather the cancer is malignant or benign for the available values of the parameters using the predictive model and classification.

Analysis of dataset

1) Analysis using first 10 variables which are measured in terms of mean values

breast_cancer_ana_mean.R

```
wdbc=read.csv(file.choose(),sep=",",header =TRUE)
dim(wdbc)
```

```
## [1] 569  12
```

```
#convert the features of the data: wdbs.data
wdbc.data=as.matrix(wdbc[,c(3:12)])
#set the row names of wdbc.data
row.names(wdbc.data)=wdbc$id
#create diagnosis vector
diagnosis=as.numeric(wdbc$diagnosis=="M")
head(diagnosis)
```

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

```
#summary of data
summary(wdbc.data)
```

```
##   radius_mean      texture_mean   perimeter_mean     area_mean
##  Min.   : 6.981   Min.   : 9.71   Min.   : 43.79   Min.   : 143.5
##  1st Qu.:11.700   1st Qu.:16.17   1st Qu.: 75.17   1st Qu.: 420.3
##  Median :13.370   Median :18.84   Median : 86.24   Median : 551.1
##  Mean   :14.127   Mean   :19.29   Mean   : 91.97   Mean   : 654.9
##  3rd Qu.:15.780   3rd Qu.:21.80   3rd Qu.:104.10   3rd Qu.: 782.7
##  Max.   :28.110   Max.   :39.28   Max.   :188.50   Max.   :2501.0
##  smoothness_mean   compactness_mean  concavity_mean    concave.po
ints_mean
##  Min.   :0.05263   Min.   :0.01938   Min.   :0.00000   Min.   :0.00000
##  1st Qu.:0.08637   1st Qu.:0.06492   1st Qu.:0.02956   1st Qu.:0.02031
##  Median :0.09587   Median :0.09263   Median :0.06154   Median :0.03350
##  Mean   :0.09636   Mean   :0.10434   Mean   :0.08880   Mean   :0.04892
##  3rd Qu.:0.10530   3rd Qu.:0.13040   3rd Qu.:0.13070   3rd Qu.:0.07400
##  Max.   :0.16340   Max.   :0.34540   Max.   :0.42680   Max.   :0.20120
```

```
##  symmetry_mean    fractal_dimension_mean
##  Min.   :0.1060   Min.   :0.04996
##  1st Qu.:0.1619   1st Qu.:0.05770
##  Median :0.1792   Median :0.06154
##  Mean   :0.1812   Mean   :0.06280
##  3rd Qu.:0.1957   3rd Qu.:0.06612
##  Max.   :0.3040   Max.   :0.09744
```

```
str(wdbc.data)
```

```
##  num [1:569, 1:10] 18 20.6 19.7 11.4 20.3 ...
##  - attr(*, "dimnames")=List of 2
##   ..$ : chr [1:569] "842302" "842517" "84300903" "84348301" ...
##   ..$ : chr [1:10] "radius_mean" "texture_mean" "perimeter_mean"
"area_mean" ...
```

```
# total no of observation of malignant diagnosis
table(wdbc$diagnosis)
```

```
##
##   B   M
## 357 212
```

```
# what is mean of each of the columns ?
round(colMeans(wdbc.data),2)
```

```
##            radius_mean           texture_mean         perimeter_mean
##                  14.13                  19.29                  91.97
##              area_mean        smoothness_mean       compactness_mean
##                 654.89                   0.10                   0.10
##         concavity_mean    concave.points_mean          symmetry_mean
##                   0.09                   0.05                   0.18
## fractal_dimension_mean
##                   0.06
```

```
# what is sd of each of the columns ?
roundSD=function(x){
  round(sd(x),2)
}
apply(wdbc.data,2,roundSD)
```

```
##            radius_mean           texture_mean         perimeter_mean
##                   3.52                   4.30                  24.30
##              area_mean        smoothness_mean       compactness_mean
##                 351.91                   0.01                   0.05
##         concavity_mean    concave.points_mean          symmetry_mean
##                   0.08                   0.04                   0.03
## fractal_dimension_mean
##                   0.01
```

```
# how the variables related to each other ?
library(corrplot)

## corrplot 0.84 loaded

corMatrix=wdbc[,c(3:12)]
# rename the columns ?
cNames=c("rad_m","txt_m","per_m","are_m","smt_m","cmp_m","con_m","cc
p_m","sym_m","frd_m")
colnames(corMatrix)=cNames
# create the correlation matrix
M=round(cor(corMatrix),2)
# create corrplot
corrplot(M,diag=FALSE,method="color",order="FPC",tl.srt=90)
# from the corrplot it is evident that there
# are many variable that are highly correlated with each other
```

#**Principle component Analysis**

why PCA ? Due to the number of variables in the model,we can try u #sing a dimensio nality reduction technique to unveil any patterns in the data. As mentioned in the Explo ratory Data Analysis section,there are thirty variables that when combined #can be use d to model each patient's diagnosis .Using PCA we can combine our many variables into different linear combinations that #each explain a part of the variance of model . By pr oceeding with a#PCA we are assuming the linearity of the of our variables within datase t.By choosing only the linear combinations that provide a maj#ority (>=85%)of the cov ariance, we can reduce the complexity of our #model.We can then more easily see how t he model works and provide # meaningful graphs and representations of our complex d ataset.

#The first step in doing a PCA , is to ask ourselves whether or not the data should be sca led to unit variance .That is ,to bring all #the numeric variables to the same scale .In oth er words, we are try#ing to determine whether we should use a correlation matrix or co variance matrix in our calculationns of eigen value and eigen vectors .

#Running PCA using correlationn matrix: when the correlation matrix is used to calcula te the eigen values and eigen vectors, we use the prcomp()funtion.

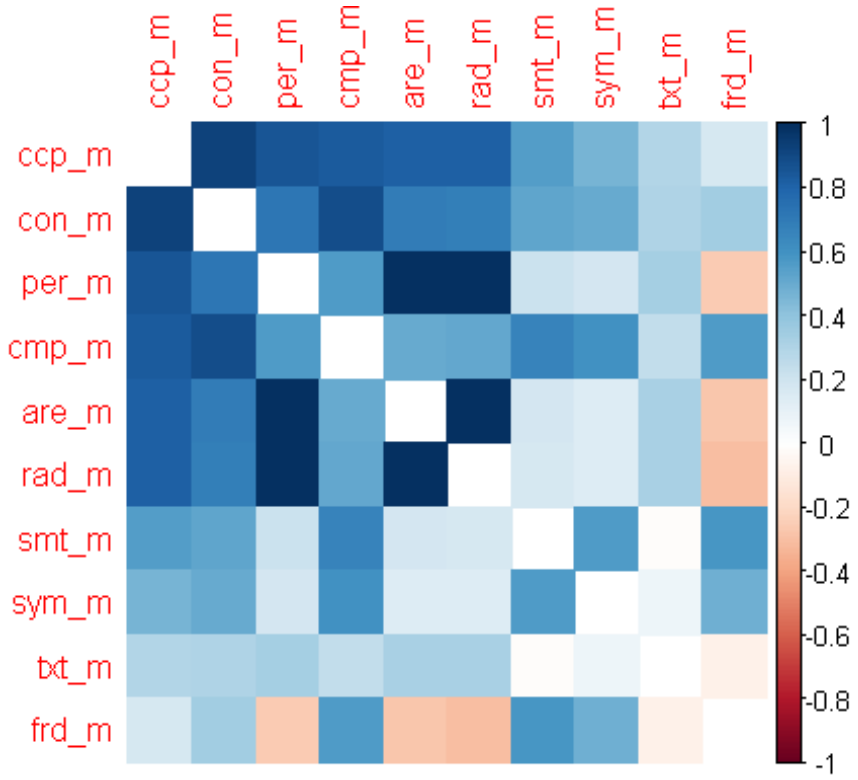
```
wdbc.pr=prcomp(wdbc.data,scale=TRUE,center=TRUE)
attributes(wdbc.pr)

## $names
## [1] "sdev"     "rotation" "center"   "scale"    "x"
##
## $class
## [1] "prcomp"
```

```
summary(wdbc.pr)
```

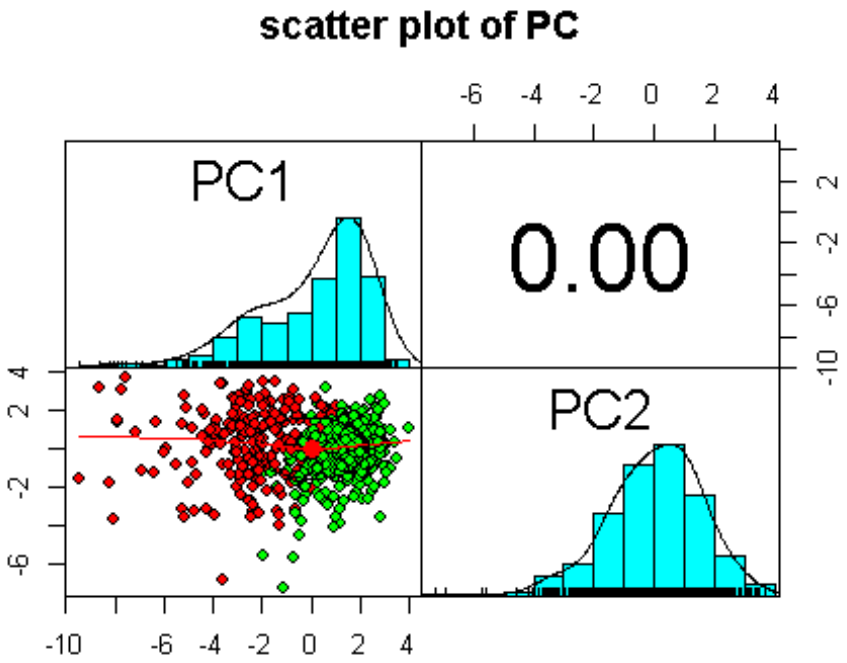
```
## Importance of components:
##                           PC1    PC2     PC3    PC4     PC5     PC6
## Standard deviation     2.3406 1.5870 0.93841 0.7064 0.61036 0.35234
## Proportion of Var      0.5479 0.2519 0.08806 0.0499 0.03725 0.01241
## Cumulative Prop        0.5479 0.7997 0.88779 0.9377 0.97495 0.98736
##                            PC7     PC8     PC9    PC10
## Standard deviation     0.28299 0.18679 0.10552 0.01680
## Proportion of Variance 0.00801 0.00349 0.00111 0.00003
## Cumulative Proportion  0.99537 0.99886 0.99997 1.00000
```

```
library(psych)
```

```
## Warning: package 'psych' was built under R version 3.4.4
```




```
pairs.panels(wdbc.pr$x[,(1:2)],gap=0,bg=c("green","red")[wdbc$diagnosis],pch=21,main="scatter plot of PC")
```




```
#Let's visualize this using a scree plot
#Calculate variablity of each component
pr.var=wdbc.pr$sdev^2
pr.var
```

```
##  [1] 5.4785879917 2.5187135854 0.8806151792 0.4990094357 0.3725391897
##  [6] 0.1241417485 0.0800853104 0.0348897928 0.0111354606 0.0002823059
```

```
# Variance explained by each principal component :pve
pve=pr.var/sum(pr.var)
pve
```

```
##  [1] 5.478588e-01 2.518714e-01 8.806152e-02 4.990094e-02 3.725392e-02
##  [6] 1.241417e-02 8.008531e-03 3.488979e-03 1.113546e-03 2.823059e-05
```

```
#eigen values
round(pr.var,2)
```

```
##  [1] 5.48 2.52 0.88 0.50 0.37 0.12 0.08 0.03 0.01 0.00
```

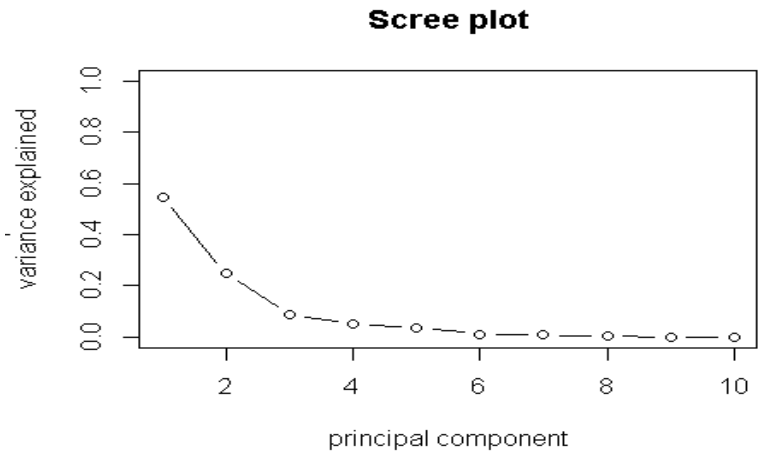
```
#percent variation explained
round(pve,2)
```

```
##  [1] 0.55 0.25 0.09 0.05 0.04 0.01 0.01 0.00 0.00 0.00
```

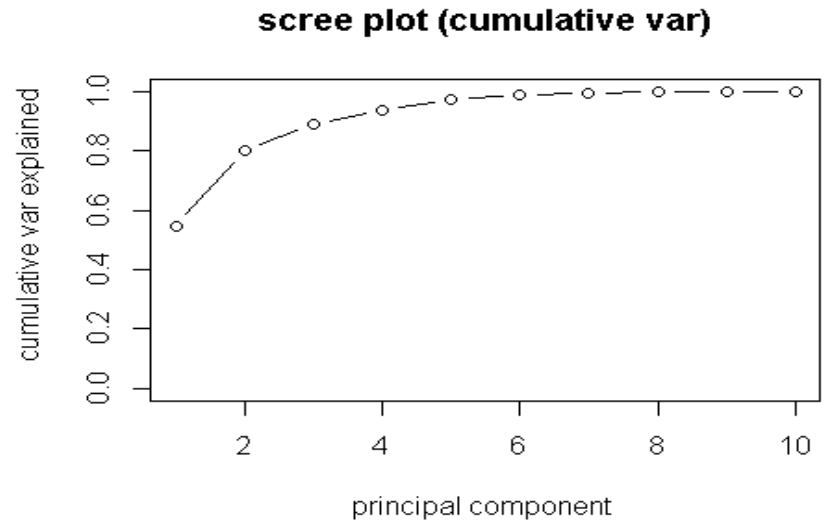
```
#cumulative percent explained
round(cumsum(pve),2)
```

```
##  [1] 0.55 0.80 0.89 0.94 0.97 0.99 1.00 1.00 1.00 1.00
```

```
# create a plot of variance explained for each principal component .
plot(pve,xlab="principal component",ylab="Proportion of
     variance explained ",ylim=c(0,1),type="b",main="Scree plot")
```




```
# plot cumulative proportion of variance explained
plot(cumsum(pve),xlab="principal component ",ylab ="
     cumulative var explained",
     ylim=c(0,1),type="b",main="scree plot (cumulative var)")
```

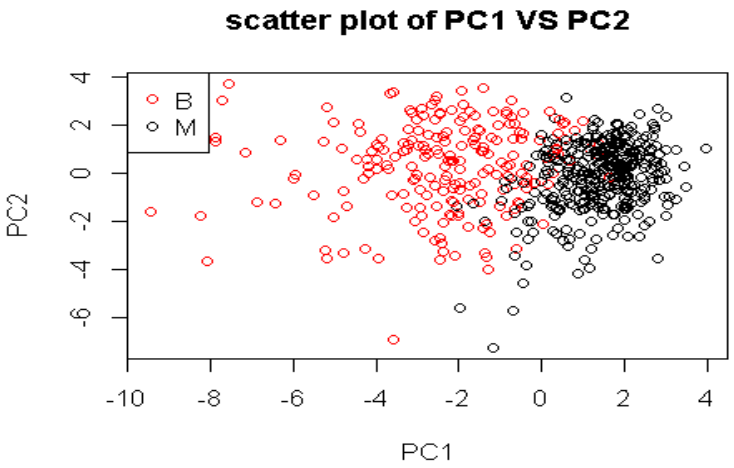


85 % variation is explained by the first 2 PC's.Moreover,the eigen values associated wi th the first 2 PC's are greater than 1 .We will use this criteria to decide on how many PC' s to include in the model building phase .

Next lets create a scatter plot of observations by PC one and two :

```
plot(wdbc.pr$x[,c(1,2)],col=(diagnosis+1),xlab ="PC1",
     ylab="PC2",main="scatter plot of PC1 VS PC2")
```

```
legend(x="topleft",pch=1,col=c("red","black"),
       legend=c("B","M"))
```




```
# There is clear separation of diagnosis(M or B) that is evident in the PC1 VS PC2 plot.
# conclusion: By using PCA we took a complex model of 30 predictors the model down t
o 2 linear combinations of the various predictors.
#Linear Discriminant Analysis(LDA)
#From the principal component's scatter plots it is evident that there is some clustering
of benign and malignant poin. This suggests that we could build a linear discriminant fu
nction using these principal components.Now that we have our chosen principal compo
nents we can perform the linear discriminant analysis.
#       -----Model building and validation-------
#Here's the high level process followed:
#Build the model using training data
#Predict using the test data
#Evaluate model performance using ROC and AUC
#Our next task is to use the first 2 PCs to build a Linear  Discrimainant function using th
e lda() function in R.
#From the wdbc.pr object ,we need to extract the first 2 PCs .To do this lets first check a
vailable for this object .
ls(wdbc.pr)

## [1] "center"   "rotation" "scale"    "sdev"     "x"

# We are interested in the rotation (also called loadings) of the
# first six PCs multiplied by the scaled data,which are called
# scores (basically pc  transformed data)
wdbc.pcs=wdbc.pr$x[,1:2]
head(wdbc.pcs,20)
```

```
##                 PC1         PC2
## 842302   -5.2195619 -3.20161108
## 842517   -1.7265746  2.53860540
## 84300903 -3.9662671  0.54959130
## 84348301 -3.5935507 -6.89899936
## 84358402 -3.1483214  1.35687844
## 843786   -1.3801055 -3.31149767
## 844359   -1.6004484  1.49741225
## 84458202 -1.2557612 -2.49237973
## 844981   -2.3883470 -3.27192935
## 84501001 -2.4425370 -3.62284993
## 845636    0.5730025  2.08052835
## 84610002 -0.8563753  0.30058529
## 846226   -4.6980134 -1.40849997
## 846381   -0.4616555  1.63533213
## 84667401 -2.4610749 -2.72621446
## 84799002 -2.1932452 -1.80947491
## 848406    0.2719287  0.84457062
## 84862001 -3.0524163 -2.00724495
## 849014   -2.4444052  2.46549274
## 8510426   0.6511564  0.07133842
```

```
# here the rownames help us see the how PC transformed data
# looks like.Now ,we need to append the diagnosis column to this PC #transformed data
frame wdbc.pcs.Lets call the new data frame as
# wdbc.pcst.
wdbc.pcst=wdbc.pcs
wdbc.pcst=cbind(wdbc.pcs,diagnosis)
head(wdbc.pcst,25)
```

```
##                 PC1         PC2 diagnosis
## 842302   -5.2195619 -3.20161108         1
## 842517   -1.7265746  2.53860540         1
## 84300903 -3.9662671  0.54959130         1
## 84348301 -3.5935507 -6.89899936         1
## 84358402 -3.1483214  1.35687844         1
## 843786   -1.3801055 -3.31149767         1
## 844359   -1.6004484  1.49741225         1
## 84458202 -1.2557612 -2.49237973         1
## 844981   -2.3883470 -3.27192935         1
## 84501001 -2.4425370 -3.62284993         1
## 845636    0.5730025  2.08052835         1
## 84610002 -0.8563753  0.30058529         1
## 846226   -4.6980134 -1.40849997         1
## 846381   -0.4616555  1.63533213         1
## 84667401 -2.4610749 -2.72621446         1
## 84799002 -2.1932452 -1.80947491         1
## 848406    0.2719287  0.84457062         1
## 84862001 -3.0524163 -2.00724495         1
## 849014   -2.4444052  2.46549274         1
## 8510426   0.6511564  0.07133842         0
## 8510653   0.3638370 -1.41866440         0
## 8510824   2.3104508 -1.74903196         0
## 8511133  -2.8406977 -2.48707350         1
## 851509   -2.6581061  2.83808957         1
## 852552   -2.3447745 -0.31267728         1
```

```
# Here ,diagnosis==1 represents malignant
# and diagnosis==0 represents benign
#  ------split the dataset into training/test  data----
# using the training data we can build the LDA function .
# Next ,we use the test data to make predictions.
# calculate N
N=nrow(wdbc.pcst)
N

## [1] 569

ind=sample(2,nrow(wdbc.pcst),replac=TRUE,prob=c(0.8,0.2))
wdbc.pcst.train=wdbc.pcst[ind==1,]
wdbc.pcst.test

=wdbc.pcst[ind==2,]
nrow(wdbc.pcst.train)

## [1] 447

nrow(wdbc.pcst.test)

## [1] 122

#so 447 observations are in the training dataset and 122 observations are in the test da
taset.We will use the training dataset to calcu#late the linear discriminant function  by p
assing it to the lda(
# fucntion to the MASS PACAKAGE
library(MASS)
wdbc.pcst.train.df=wdbc.pcst.train
# convert matrix to a dataframe
wdbc.pcst.train.df=as.data.frame(wdbc.pcst.train)
wdbc.pcst.test.df=as.data.frame(wdbc.pcst.test)
#       PERFORM LDA ON DIAGNOSIS
wdbc.lda=lda(diagnosis~PC1+PC2,data=wdbc.pcst.train.df)
# lets summarize the LDA OUTPUT
attributes(wdbc.lda)

## $names
##  [1] "prior"   "counts"  "means"   "scaling" "lev"     "svd"
"N"
##  [8] "call"    "terms"   "xlevels"
##
## $class
## [1] "lda"
```

```
head(wdbc.lda$prior)
```

```
##         0         1
## 0.6219239 0.3780761
```

```
wdbc.lda$counts
```

```
##   0   1
## 278 169
```

```
wdbc.lda$scaling
```

```
##            LD1
## PC1 -0.7063751
## PC2  0.2036354
```

```
p=predict(wdbc.lda,wdbc.pcst.train.df)$class
# confusion matrix and accuracy ~ training data
tab=table(predicted=p,actual=wdbc.pcst.train.df$diagnosis)
tab
```

```
##          actual
## predicted   0   1
##         0 276  37
##         1   2 132
```

```
table(wdbc.pcst.train.df$diagnosis)
```

```
##
##   0   1
## 278 169
```

```
# accuracy of training data
sum(diag(tab))/sum(tab)
```

```
## [1] 0.9127517
```

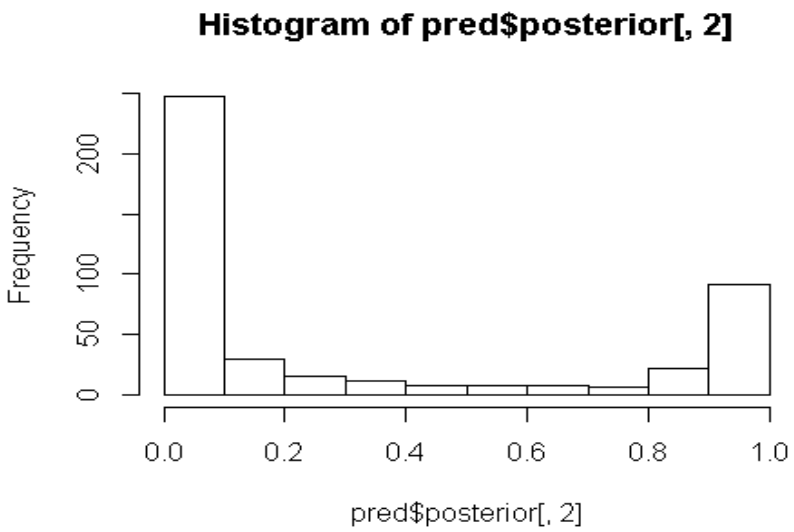
```
# confusion matrix and accuracy ~ testting  data
p1=predict(wdbc.lda,wdbc.pcst.test.df)$class
tab1=table(predicted=p1,actual=wdbc.pcst.test.df$diagnosis)
tab1
```

```
##          actual
## predicted  0  1
##         0 77  6
##         1  2 37
```

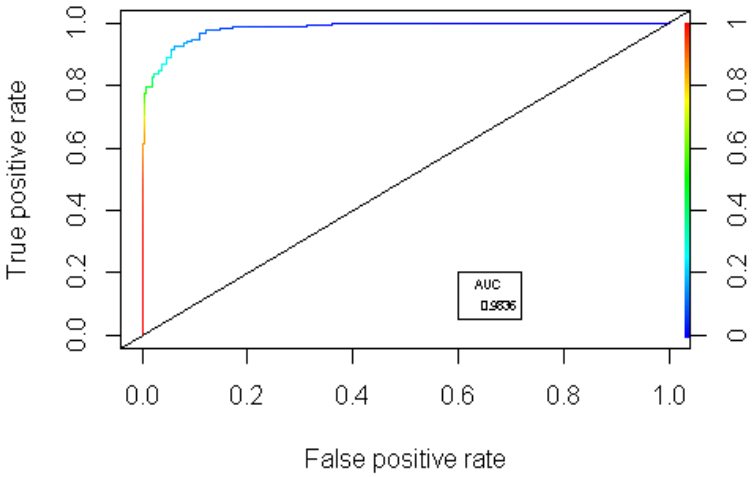
```
table(wdbc.pcst.test.df$diagnosis)
```

```
##
##  0  1
## 79 43
```

```
# accuracy of testing data
sum(diag(tab1))/sum(tab1)

## [1] 0.9344262

# model peroformance evaluation
library(ROCR)

pred=predict(wdbc.lda,wdbc.pcst.train.df,type='prob')
hist(pred$posterior[,2])
```




```
pred=prediction(pred$posterior[,2],wdbc.pcst.train.df$diagnosis)
roc=performance(pred,"tpr","fpr")
plot(roc,colorize=T)
abline(a=0,b=1)
# area under the curve
auc=performance(pred,"auc")
auc=unlist(slot(auc,"y.values"))
auc

## [1] 0.9835895

auc=round(auc,4)
legend(0.6,0.2,auc,title="AUC",cex=0.5)
```



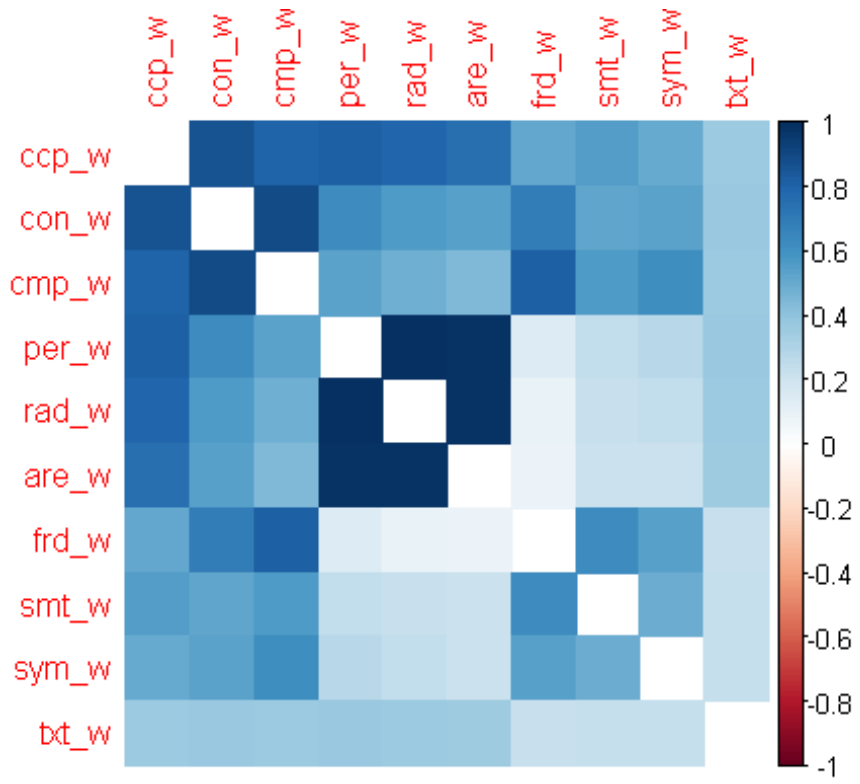
By applying the classification rule we have constructed a diagnostic system that predict malignant tumors at 96.36% if the variables are measured in terms of mean

2) Building a model using variables which are measured in terms of maximum values of parameters.

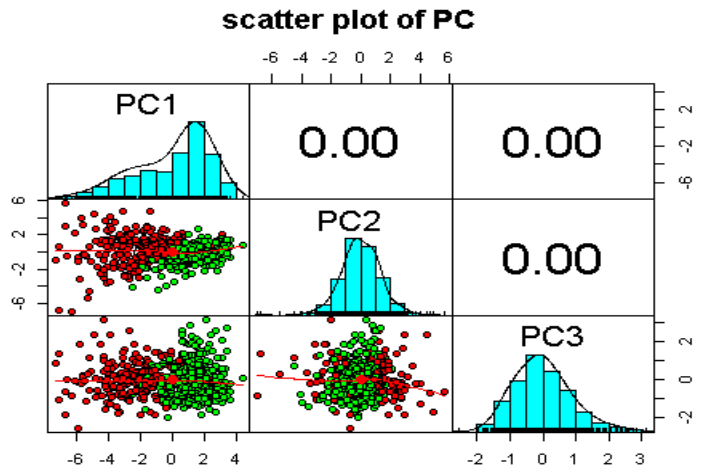
breast_cancer_ana_max.R

```
wdbc=read.csv(file.choose(),sep=",",header =TRUE)
dim(wdbc)

## [1] 569  12

#how the variables related to each other ?
library(corrplot)

## corrplot 0.84 loaded

corMatrix=wdbc[,c(3:12)]
#rename the columns ?
cNames=c( "rad_w","txt_w","per_w","are_w","smt_w","cmp_w", "con_w","ccp_w","sym_
w","frd_w")
colnames(corMatrix)=cNames
#create the correlation matrix
M=round(cor(corMatrix),2)
#create corrplot
```

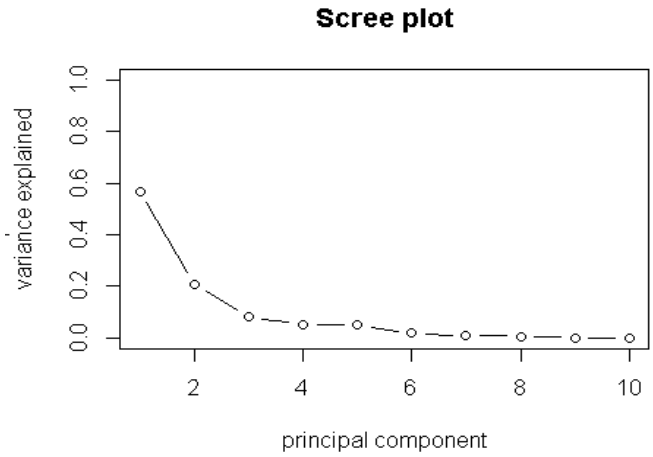
```
corrplot(M,diag=FALSE,method="color",order="FPC",tl.srt=90)
#from the corrplot it is evident that there are many variable that are highly correlated with each other

wdbc.pr=prcomp(wdbc.data,scale=TRUE,center=TRUE)
summary(wdbc.pr)

## Importance of components:
##                          PC1    PC2     PC3     PC4     PC5   PC6
Standard deviation     2.3869 1.4443 0.89597 0.73531 0.71741  0.42862
Proportion of Var      0.5697 0.2086 0.08028 0.05407 0.05147 0.01837 Cumulative Pro
portion 0.5697 0.7783 0.85860 0.91267 0.96413 0.98251
##                             PC7     PC8     PC9    PC10
## Standard deviation     0.28959 0.26802 0.12343 0.06326
## Proportion of Variance 0.00839 0.00718 0.00152 0.00040
## Cumulative Proportion  0.99089 0.99808 0.99960 1.00000

library(psych)
```



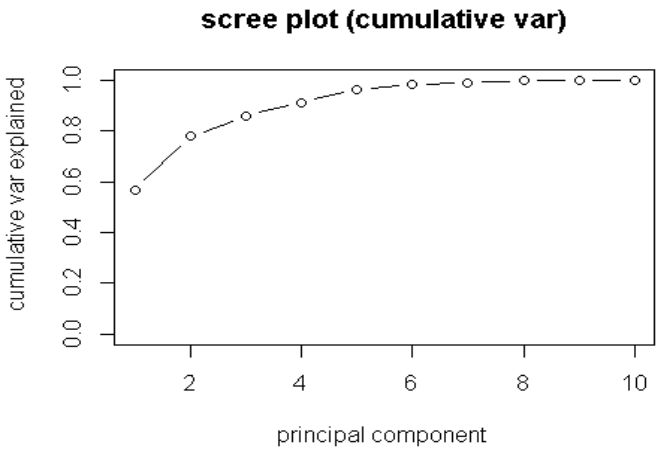

```
pairs.panels(wdbc.pr$x[,(1:3)],gap=0,bg=c("green","red")[wdbc$diagnosis],pch=21,main="scatter plot of PC")
```



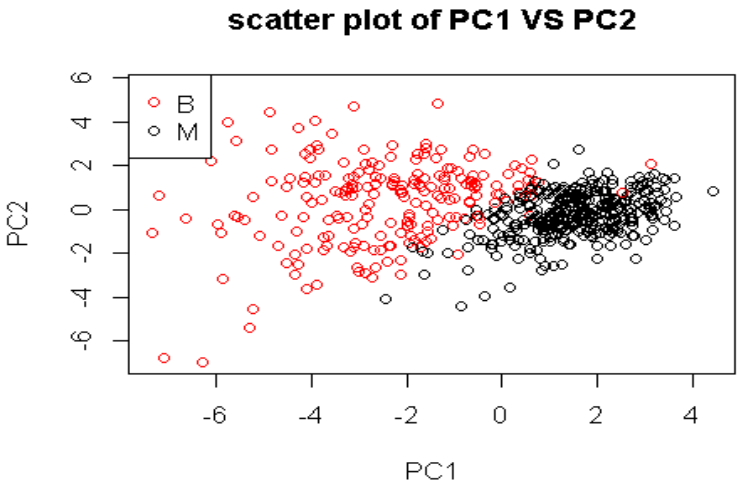

```
# create a plot of variance explained for each principal
# component .
plot(pve,xlab="principal component",ylab="Proportion of
     variance explained ",ylim=c(0,1),type="b",main="Scree plot")
```




```
# plot cumulative proportion of variance explained
plot(cumsum(pve),xlab="principal component ",ylab ="
   cumulative var explained",
   ylim=c(0,1),type="b",main="scree plot (cumulative var)")
```




```
# 88% variation is explained by the first 3 PC's.Moreover,the eigen values associated wit
h the first 3 PC's are greater than 1 .We will use this criteria to decide on how many PC's
to include in the model building phase .
# Next ,lets create a scaater plot observations by principal components one and two :
plot(wdbc.pr$x[,c(1,2)],col=(diagnosis+1),xlab ="PC1",
   ylab="PC2",main="scatter plot of PC1 VS PC2")
legend(x="topleft",pch=1,col=c("red","black"),
    legend=c("B","M"))
```




```
# There is clear separation of diagnosis(M or B) that is evident in the PC1 VS PC2 plot.

 conclusion:By using PCA we took a complex model of 30 predictors the model down to
3 linear combinations of the various predictors.
#   ------split the dataset into training/test  data----
# using the training data we can build the LDA function .
# Next ,we use the test data to make predictions.
# calculate N
```

```
N=nrow(wdbc.pcst)
N
```

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

```
ind=sample(2,nrow(wdbc.pcst),replac=TRUE,prob=c(0.8,0.2))
wdbc.pcst.train=wdbc.pcst[ind==1,]
wdbc.pcst.test=wdbc.pcst[ind==2,]
nrow(wdbc.pcst.train)
```

```
## [1] 462
```

```
nrow(wdbc.pcst.test)
```

```
## [1] 107
```

```
# so 462 observations are in the training datasetand 107observation are in the test data set. We will use the training dataset to calculate the linear discriminant function  by pas sing it to the lda()
# fucntion to the MASS PACAKAGE
library(MASS)
wdbc.pcst.train.df=wdbc.pcst.train
# convert matrix to a dataframe
wdbc.pcst.train.df=as.data.frame(wdbc.pcst.train)
wdbc.pcst.test.df=as.data.frame(wdbc.pcst.test)
#     PERFORMANCE LDA ON DIAGNOSIS
wdbc.lda=lda(diagnosis~PC1+PC2+PC3,data=wdbc.pcst.train.df)
# lets summarize the LDA OUTPUT
p=predict(wdbc.lda,wdbc.pcst.train.df)$class
#confusion matrix and accuracy ~ training data
tab=table(predicted=p,actual=wdbc.pcst.train.df$diagnosis)
tab
```

```
##           actual
## predicted   0   1
##         0 290  22
##         1   1 149
```

```
table(wdbc.pcst.train.df$diagnosis)
```

```
##
##   0   1
## 291 171
```

```
# accuracy of training data
sum(diag(tab))/sum(tab)
```

```
## [1] 0.9502165
```

```
#confusion matrix and accuracy ~ testting  data
p1=predict(wdbc.lda,wdbc.pcst.test.df)$class
tab1=table(predicted=p1,actual=wdbc.pcst.test.df$diagnosis)
tab1
```

```
##          actual
## predicted  0  1
##         0 65  3
##         1  1 38
```

```
table(wdbc.pcst.test.df$diagnosis)
```

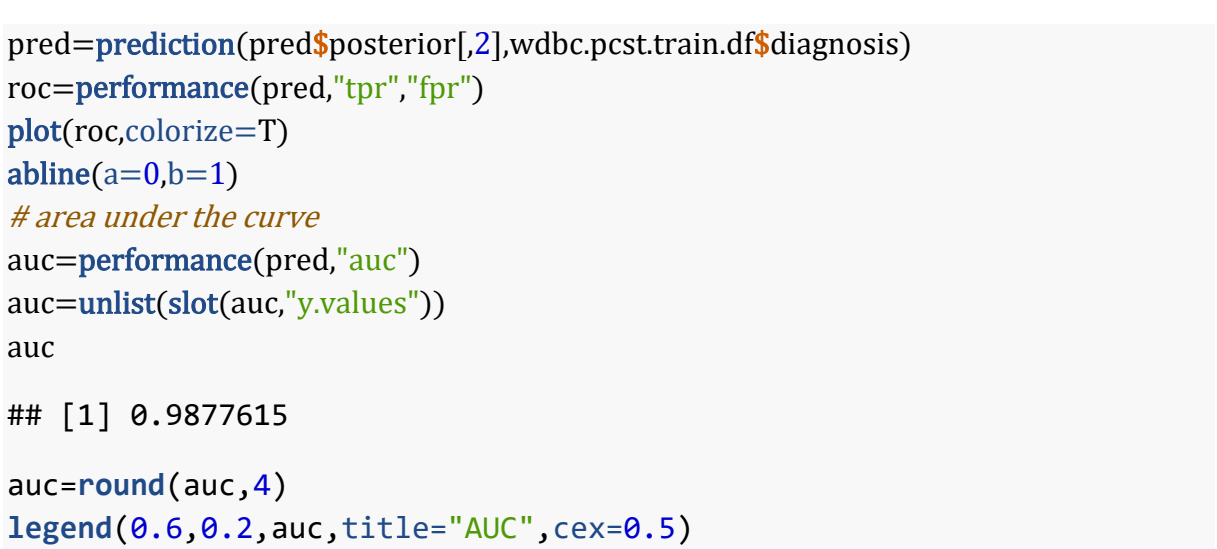
```
##
##  0  1
## 66 41
```

```
# accuracy of testing data
sum(diag(tab1))/sum(tab1)
```

```
## [1] 0.9626168
```

```
# model peroformance evaluation
library(ROCR)
```

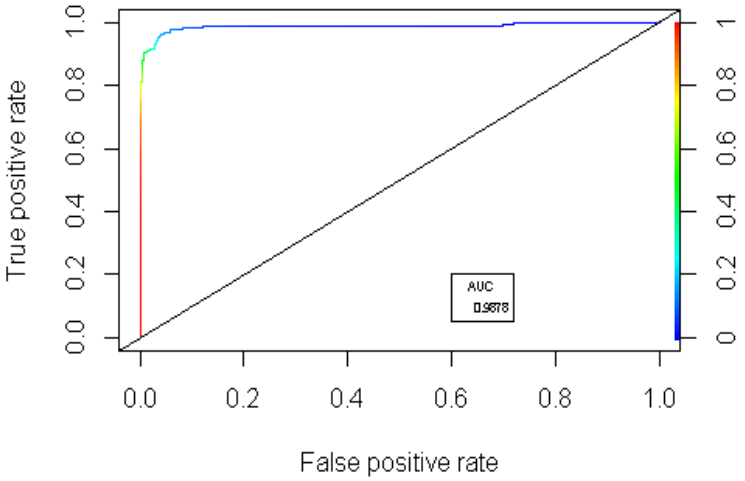
```
pred=predict(wdbc.lda,wdbc.pcst.train.df,type='prob')
hist(pred$posterior[,2])
```



```
pred=prediction(pred$posterior[,2],wdbc.pcst.train.df$diagnosis)
roc=performance(pred,"tpr","fpr")
plot(roc,colorize=T)
abline(a=0,b=1)
# area under the curve
auc=performance(pred,"auc")
auc=unlist(slot(auc,"y.values"))
auc
```

```
## [1] 0.9877615
```

```
auc=round(auc,4)
legend(0.6,0.2,auc,title="AUC",cex=0.5)
```



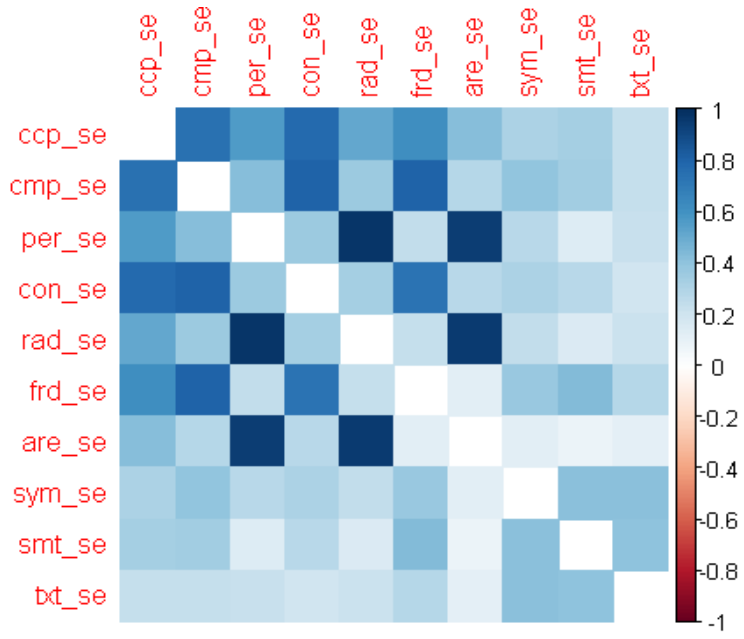
By applaying the classification rule we have constructed a diagnostisystem that predicts malignant tumors at 96.78% when the variables are measured in terms of maximum.

3) Building a model using variables which are measured in terms of standard error values of parameters.

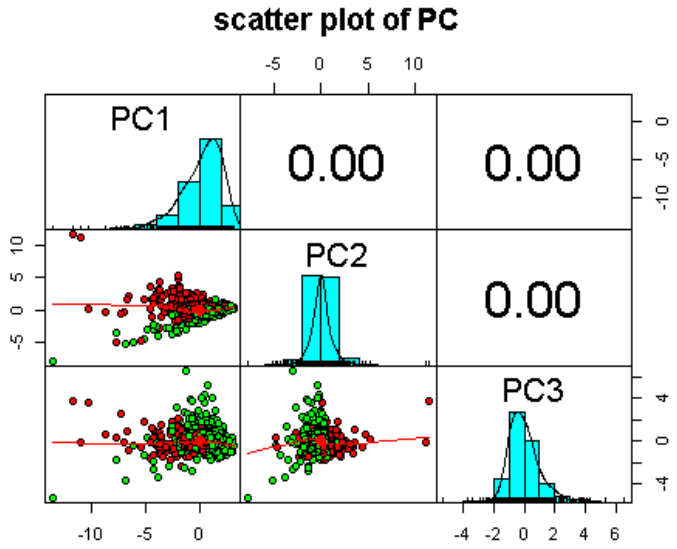
breast_cancer_ana_se.R

```
wdbc=read.csv(file.choose(),sep=",",header =TRUE)
dim(wdbc)

## [1] 569  12

library(corrplot)

## corrplot 0.84 loaded

corMatrix=wdbc[,c(3:12)]
#rename the columns ?
cNames=c( "rad_se","txt_se","per_se","are_se","smt_se","cmp_se", "con_se","ccp_se","sym_se","frd_se")
colnames(corMatrix)=cNames
#create the correlation matrix
M=round(cor(corMatrix),2)

# create corrplot
```

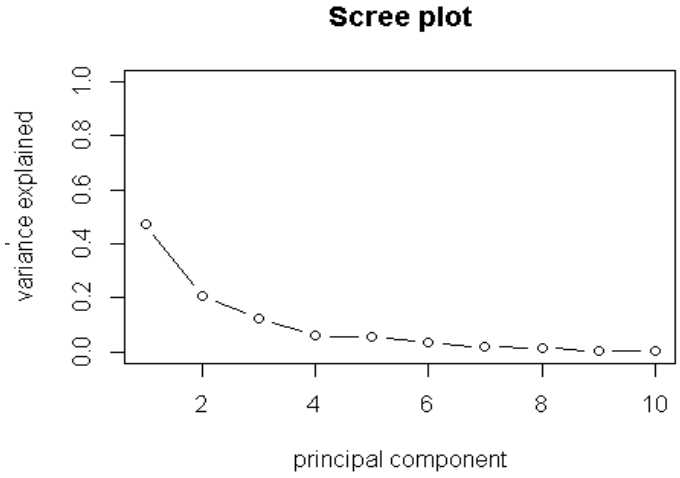
```
corrplot(M,diag=FALSE,method="color",order="FPC",tl.srt=90)
# from the corrplot it is evident that there are many variable that are highly correlated with each other
#Principle component Analysis
attributes(wdbc.pr)

summary(wdbc.pr)

## Importance of components:
##                           PC1    PC2    PC3     PC4     PC5     PC6
## Standard deviation     2.1779 1.4406 1.1245 0.77095 0.75991 0.57939
## Proportion of Variance 0.4743 0.2075 0.1264 0.05944 0.05775 0.03357
## Cumulative Proportion  0.4743 0.6819 0.8083 0.86774 0.92548 0.95905
##                           PC7    PC8     PC9    PC10
## Standard deviation     0.43512 0.3962 0.20436 0.14635
## Proportion of Variance 0.01893 0.0157 0.00418 0.00214
## Cumulative Proportion  0.97798 0.9937 0.99786 1.00000

library(psych)
```



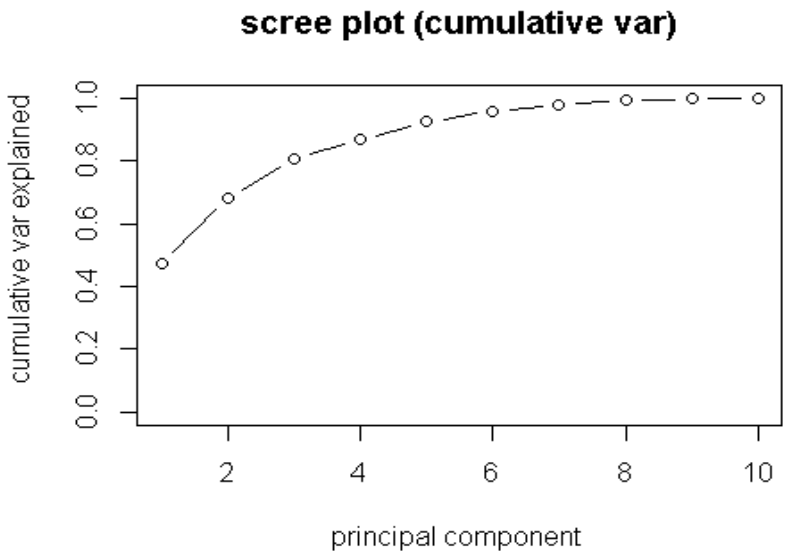

```
pairs.panels(wdbc.pr$x[,(1:3)],gap=0,bg=c("green","red")[wdbc$diagnosis],pch=21,main="scatter plot of PC")
```



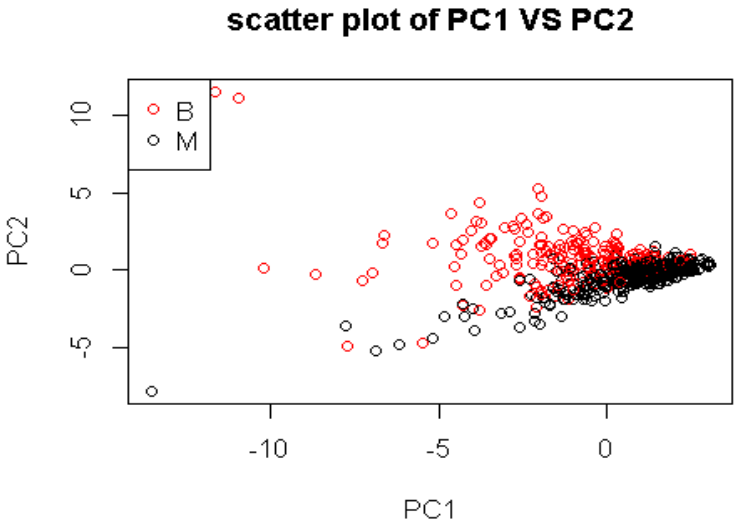

```
# create a plot of variance explained for each principal
# component .
plot(pve,xlab="principal component",ylab="Proportion of
   variance explained ",ylim=c(0,1),type="b",main="Scree plot")
```




```
# plot cumulative proportion of variance explained
plot(cumsum(pve),xlab="principal component ",ylab ="
     cumulative var explained",
     ylim=c(0,1),type="b",main="scree plot (cumulative var)")
```




```
# 88% variation is explained by the first 3 PC's.Moreover, the eigen values associated wi
th the first 3 PC's are greater than 1 .We will use this criteria to decide on how many PC'
s to include in the model building phase.Next ,lets create a scaater plot observations by
principal components one and two :
plot(wdbc.pr$x[,c(1,2)],col=(diagnosis+1),xlab ="PC1",
   ylab="PC2",main="scatter plot of PC1 VS PC2")
legend(x="topleft",pch=1,col=c("red","black"),
    legend=c("B","M"))
```




```
# There is clear separation of diagnosis(M or B) that is evident in the PC1 VS PC2 plot.
# conclusion:By using PCA we took a complex model of 30 predictors the modeldown to
six linear combinations of the various predictors.
```

```
N=nrow(wdbc.pcst)
N
```

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

```
ind=sample(2,nrow(wdbc.pcst),replac=TRUE,prob=c(0.8,0.2))
wdbc.pcst.train=wdbc.pcst[ind==1,]
wdbc.pcst.test=wdbc.pcst[ind==2,]
nrow(wdbc.pcst.train)
```

```
## [1] 446
```

```
nrow(wdbc.pcst.test)
```

```
## [1] 123
```

```
# so 446 observations are in the training dataset and 123 observations are in the test da
taset. We will use the training dataset to calculate the.linear discriminant function  by pa
ssing it to the lda()
# fucntion to the MASS PACAKAGE
library(MASS)
wdbc.pcst.train.df=wdbc.pcst.train
# convert matrix to a dataframe
wdbc.pcst.train.df=as.data.frame(wdbc.pcst.train)
wdbc.pcst.test.df=as.data.frame(wdbc.pcst.test)
#     PERFORMANCE LDA ON DIAGNOSIS
wdbc.lda=lda(diagnosis~PC1+PC2+PC3,data=wdbc.pcst.train.df)
# lets summarize the LDA OUTPUT
head(wdbc.lda$prior)
```

```
##        0        1
## 0.632287 0.367713
```

```
wdbc.lda$counts
```

```
##   0   1
## 282 164
```

```
wdbc.lda$scaling
```

```
##            LD1
## PC1 -0.4243748
## PC2  0.6313954
## PC3 -0.3877121
```

```
p=predict(wdbc.lda,wdbc.pcst.train.df)$class
#confusion matrix and accuracy ~ training data
```

```
tab=table(predicted=p,actual=wdbc.pcst.train.df$diagnosis)
tab
```

```
##           actual
## predicted   0   1
##         0 275  72
##         1   7  92
```

```
table(wdbc.pcst.train.df$diagnosis)
```

```
##
##   0   1
## 282 164
```

```
#accuracy of training data
sum(diag(tab))/sum(tab)
```

```
## [1] 0.82287
```

```
#confusion matrix and accuracy ~ testting  data
p1=predict(wdbc.lda,wdbc.pcst.test.df)$class
tab1=table(predicted=p1,actual=wdbc.pcst.test.df$diagnosis)
tab1
```

```
##           actual
## predicted  0  1
##         0 74 21
##         1  1 27
```

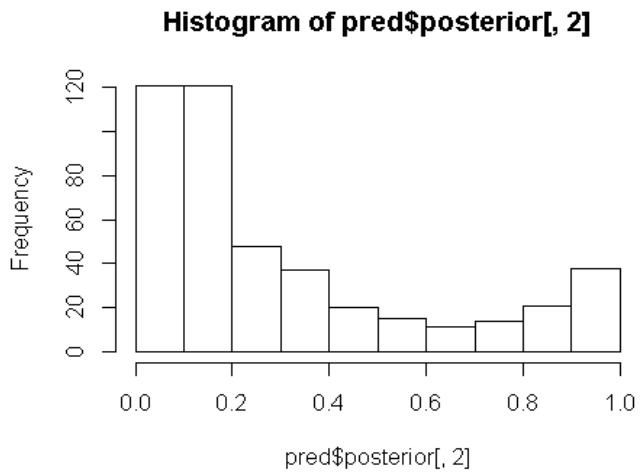
```
table(wdbc.pcst.test.df$diagnosis)
```

```
##
##  0  1
## 75 48
```

```
# accuracy of testing data
sum(diag(tab1))/sum(tab1)
```

```
## [1] 0.8211382
```

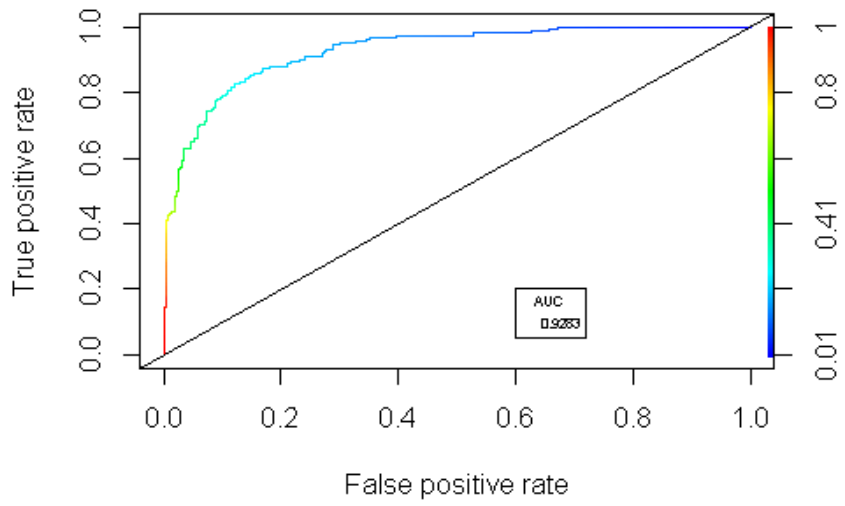
```
# model peroformance evaluation
library(ROCR)

pred=predict(wdbc.lda,wdbc.pcst.train.df,type='prob')

hist(pred$posterior[,2])
```




```
pred=prediction(pred$posterior[,2],wdbc.pcst.train.df$diagnosis)
roc=performance(pred,"tpr","fpr")
plot(roc,colorize=T)
abline(a=0,b=1)
# area under the curve
auc=performance(pred,"auc")
auc=unlist(slot(auc,"y.values"))
auc
```

```
## [1] 0.9282996
```

```
auc=round(auc,4)
legend(0.6,0.2,auc,title="AUC",cex=0.5)
```



By applying the classification rule we have constructed a diagnostic system that predicts malignant tumors at 92.83% when the variables are measured in terms of std.error.

4) Model building when we consider all the given 30 variables

```
wdbc=read.csv(file.choose(),sep=",",header =TRUE)
dim(wdbc)
```

```
## [1] 569  32
```

```
#convert the features of the data: wdbs.data
wdbc.data=as.matrix(wdbc[,c(3:32)])
#set the row names of wdbc.data
row.names(wdbc.data)=wdbc$id
#create diagnosis vector
diagnosis=as.numeric(wdbc$diagnosis=="M")
head(diagnosis)
```

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

```
#summary of data
summary(wdbc.data)
```

```
##   radius_mean      texture_mean   perimeter_mean     area_mean
##  Min.   : 6.981   Min.   : 9.71   Min.   : 43.79   Min.   : 143.5
##  1st Qu.:11.700   1st Qu.:16.17   1st Qu.: 75.17   1st Qu.: 420.3
##  Median :13.370   Median :18.84   Median : 86.24   Median : 551.1
##  Mean   :14.127   Mean   :19.29   Mean   : 91.97   Mean   : 654.9
##  3rd Qu.:15.780   3rd Qu.:21.80   3rd Qu.:104.10   3rd Qu.: 782.7
##  Max.   :28.110   Max.   :39.28   Max.   :188.50   Max.   :2501.0
##  smoothness_mean   compactness_mean  concavity_mean    concave.points_mean
##  Min.   :0.05263   Min.   :0.01938   Min.   :0.00000   Min.   :0.00000
##  1st Qu.:0.08637   1st Qu.:0.06492   1st Qu.:0.02956   1st Qu.:0.02031
##  Median :0.09587   Median :0.09263   Median :0.06154   Median :0.03350
##  Mean   :0.09636   Mean   :0.10434   Mean   :0.08880   Mean   :0.04892
##  3rd Qu.:0.10530   3rd Qu.:0.13040   3rd Qu.:0.13070   3rd Qu.:0.07400
##  Max.   :0.16340   Max.   :0.34540   Max.   :0.42680   Max.   :0.20120
##  symmetry_mean    fractal_dimension_mean   radius_se        texture_se
##  Min.   :0.1060   Min.   :0.04996        Min.   :0.1115   Min.   :0.3602
##  1st Qu.:0.1619   1st Qu.:0.05770        1st Qu.:0.2324   1st Qu.:0.8339
##  Median :0.1792   Median :0.06154        Median :0.3242   Median :1.1080
##  Mean   :0.1812   Mean   :0.06280        Mean   :0.4052   Mean   :1.2169
##  3rd Qu.:0.1957   3rd Qu.:0.06612        3rd Qu.:0.4789   3rd Qu.:1.4740
##  Max.   :0.3040   Max.   :0.09744        Max.   :2.8730   Max.   :4.8850
```

```
##   perimeter_se       area_se         smoothness_se      compactness_se
##  Min.   : 0.757   Min.   :  6.802   Min.   :0.001713   Min.   :0.002252
```

```
##  1st Qu.: 1.606   1st Qu.: 17.850   1st Qu.:0.005169   1st Qu.:0.013080
##  Median : 2.287   Median : 24.530   Median :0.006380   Median :0.020450
##  Mean   : 2.866   Mean   : 40.337   Mean   :0.007041   Mean   :0.025478
##  3rd Qu.: 3.357   3rd Qu.: 45.190   3rd Qu.:0.008146   3rd Qu.:0.032450
##  Max.   :21.980   Max.   :542.200   Max.   :0.031130   Max.   :0.135400
##
```

```
  concavity_se     concave.points_se   symmetry_se
##  Min.   :0.00000   Min.   :0.000000   Min.   :0.007882
##  1st Qu.:0.01509   1st Qu.:0.007638   1st Qu.:0.015160
##  Median :0.02589   Median :0.010930   Median :0.018730
##  Mean   :0.03189   Mean   :0.011796   Mean   :0.020542
##  3rd Qu.:0.04205   3rd Qu.:0.014710   3rd Qu.:0.023480
##  Max.   :0.39600   Max.   :0.052790   Max.   :0.078950
##  fractal_dimension_se  radius_worst   texture_worst   perimeter_worst
##  Min.   :0.0008948    Min.   : 7.93   Min.   :12.02   Min.   : 50.41
##  1st Qu.:0.0022480    1st Qu.:13.01   1st Qu.:21.08   1st Qu.: 84.11
##  Median :0.0031870    Median :14.97   Median :25.41   Median : 97.66
##  Mean   :0.0037949    Mean   :16.27   Mean   :25.68   Mean   :107.26
##  3rd Qu.:0.0045580    3rd Qu.:18.79   3rd Qu.:29.72   3rd Qu.:125.40
##  Max.   :0.0298400    Max.   :36.04   Max.   :49.54   Max.   :251.20
##    area_worst     smoothness_worst  compactness_worst concavity_worst
##  Min.   : 185.2   Min.   :0.07117   Min.   :0.02729   Min.   :0.0000
##  1st Qu.: 515.3   1st Qu.:0.11660   1st Qu.:0.14720   1st Qu.:0.1145
##  Median : 686.5   Median :0.13130   Median :0.21190   Median :0.2267
##  Mean   : 880.6   Mean   :0.13237   Mean   :0.25427   Mean   :0.2722
##  3rd Qu.:1084.0   3rd Qu.:0.14600   3rd Qu.:0.33910   3rd Qu.:0.3829
##  Max.   :4254.0   Max.   :0.22260   Max.   :1.05800   Max.   :1.2520
##  concave.points_worst symmetry_worst   fractal_dimension_worst
##  Min.   :0.00000      Min.   :0.1565   Min.   :0.05504
##  1st Qu.:0.06493      1st Qu.:0.2504   1st Qu.:0.07146
##  Median :0.09993      Median :0.2822   Median :0.08004
##  Mean   :0.11461      Mean   :0.2901   Mean   :0.08395
##  3rd Qu.:0.16140      3rd Qu.:0.3179   3rd Qu.:0.09208
##  Max.   :0.29100      Max.   :0.6638   Max.   :0.20750
```

how the variables related to each other ?

```
library(corrplot)
```

```
## corrplot 0.84 loaded
```

```
corMatrix=wdbc[,c(3:32)]
# rename the columns ?
cNames=c("rad_m","txt_m","per_m","are_m","smt_m","cmp_m","con_m","ccp_m","sym_
m","frd_m","rad_se","txt_se","per_se","are_se","smt_se","cmp_se","con_se","ccp_se","sym_
se","frd_se", "rad_w","txt_w","per_w","are_w","smt_w","cmp_w", "con_w","ccp_w","sym_
w","frd_w")
colnames(corMatrix)=cNames
# create the correlation matrix
M=round(cor(corMatrix),2)

# create corrplot
corrplot(M,diag=FALSE,method="color",order="FPC",tl.srt=90)
```

```
 # from the corrplot it is evident that there are many variable that are highly correlated with each other
  wdbc.pr=prcomp(wdbc.data,scale=TRUE,center=TRUE)
  summary(wdbc.pr)
```

```
## 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
```

```
  library(psych)
```

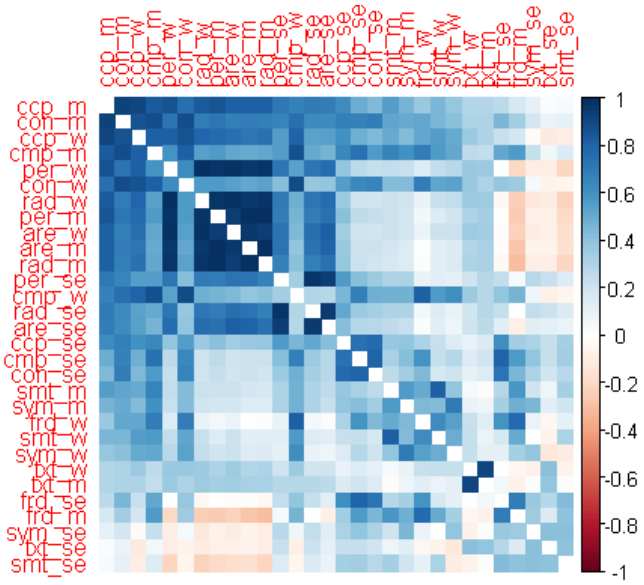


Fig 1. Correlation plot of varaibles

```
  pairs.panels(wdbc.pr$x[,(1:6)],gap=0,bg=c("green","red")[wdbc$diagnosis],pch=21,
main="scatter plot of PC")
```

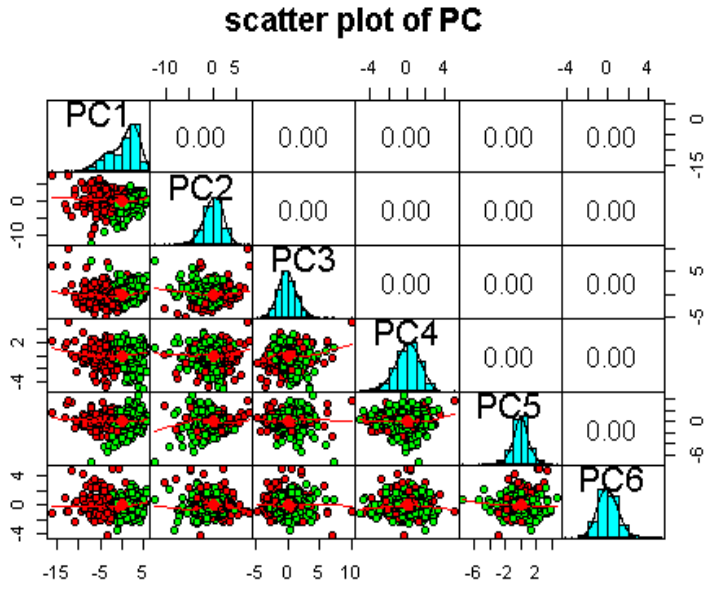


Fig 2.Scatter plot of Principal component

```
 # Let's visualize this using a scree plot
    #Calculate variability of each component
   pr.var=wdbc.pr$sdev^2
   pr.var
```

```
##  [1] 1.328161e+01 5.691355e+00 2.817949e+00 1.980640e+00 1.648731e+00
##  [6] 1.207357e+00 6.752201e-01 4.766171e-01 4.168948e-01 3.506935e-01
## [11] 2.939157e-01 2.611614e-01 2.413575e-01 1.570097e-01 9.413497e-02
## [16] 7.986280e-02 5.939904e-02 5.261878e-02 4.947759e-02 3.115940e-02
## [21] 2.997289e-02 2.743940e-02 2.434084e-02 1.805501e-02 1.548127e-02
## [26] 8.177640e-03 6.900464e-03 1.589338e-03 7.488031e-04 1.330448e-04
```

```
 # Variance explained by each principal component :pve
   pve=pr.var/sum(pr.var)
   pve
```

```
##  [1] 4.427203e-01 1.897118e-01 9.393163e-02 6.602135e-02 5.495768e-02
##  [6] 4.024522e-02 2.250734e-02 1.588724e-02 1.389649e-02 1.168978e-02
## [11] 9.797190e-03 8.705379e-03 8.045250e-03 5.233657e-03 3.137832e-03
## [16] 2.662093e-03 1.979968e-03 1.753959e-03 1.649253e-03 1.038647e-03
## [21] 9.990965e-04 9.146468e-04 8.113613e-04 6.018336e-04 5.160424e-04
## [26] 2.725880e-04 2.300155e-04 5.297793e-05 2.496010e-05 4.434827e-06
```

```
 # eigen values
   round(pr.var,2)
```

```
##  [1] 13.28  5.69  2.82  1.98  1.65  1.21  0.68  0.48  0.42  0.35  0.29
## [12]  0.26  0.24  0.16  0.09  0.08  0.06  0.05  0.05  0.03  0.03  0.03
## [23]  0.02  0.02  0.02  0.01  0.01  0.00  0.00  0.00
```

```
# percent variation explained
  round(pve,2)
```

```
## [1] 0.44  0.19  0.09 0.07 0.05 0.04 0.02 0.02 0.01 0.01 0.01 0.01 0.01 0.01
## [15] 0.00 0.00  0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
## [29] 0.00  0.00
```

```
# cumulative percent explained
  round(cumsum(pve),2)
```

```
## [1]  0.44 0.63 0.73 0.79 0.85 0.89 0.91 0.93 0.94 0.95 0.96 0.97 0.98 0.98
## [15] 0.99 0.99 0.99 0.99 0.99 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00  1.00
## [29] 1.00 1.00
```

```
# create a plot of variance explained for each principal component .
  plot(pve,xlab="principal component",ylab="Proportion of
      variance explained ",ylim=c(0,1),type="b",main="Scree plot")
```

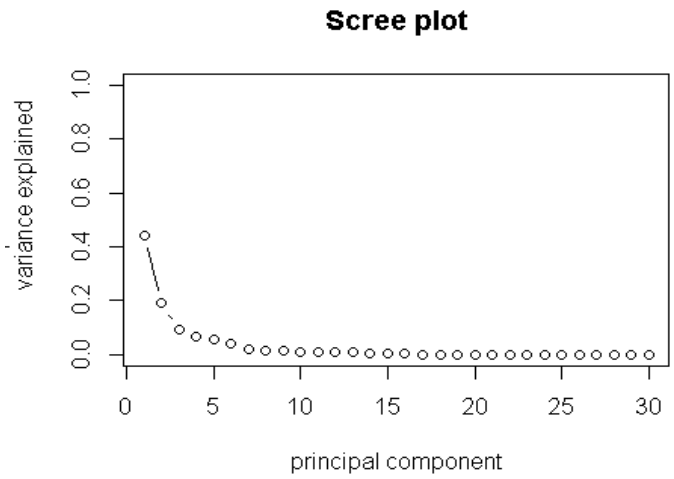


Fig :3

```
# plot cumulative proportion of variance explained
  plot(cumsum(pve),xlab="principal component ",ylab ="
      cumulative var explained",
      ylim=c(0,1),type="b",main="scree plot (cumulative var)")
```

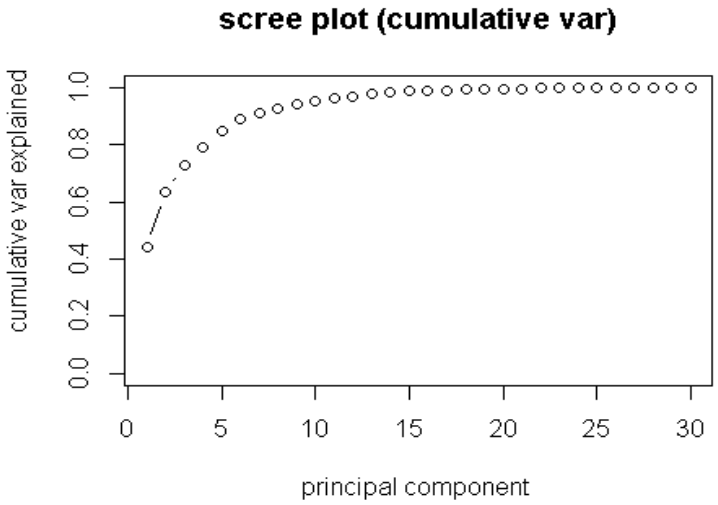


Fig :4

```
 # 89 % variation is explained by the first 6 PC's.Moreover, the Eigen values associated with the first 6 PC's are greater than 1 .We will use this criteria to decide on how many

PC's to include in the model building phase .Next ,lets create a scatter plot observations by principal components one and two :
   plot(wdbc.pr$x[,c(1,2)],col=(diagnosis+1),xlab ="PC1",
      ylab="PC2",main="scatter plot of PC1 VS PC2")
   legend(x="topleft",pch=1,col=c("red","black"),
        legend=c("B","M"))
```

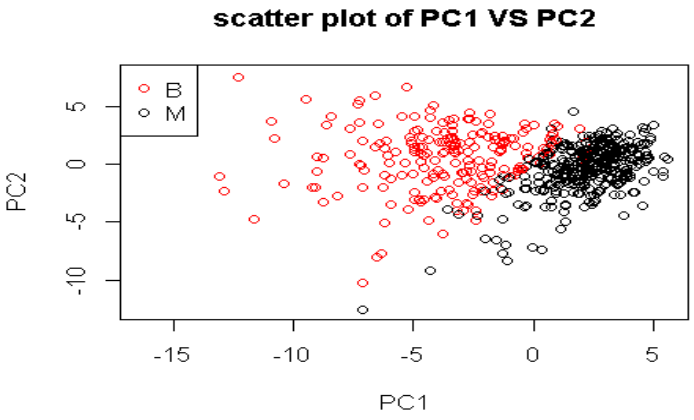


Fig no 5.Scatter plot PC 1 VS PC 2

```
# There is clear separation of diagnosis(M or B) that is evident in the PC1 VS PC2 plot.
# conclusion: By using PCA we took a complex model of 30 predictors the model down t
o six linear combinations of the various predictors.
```

```
wdbc.pcst=wdbc.pcs
wdbc.pcst=cbind(wdbc.pcs,diagnosis)
head(wdbc.pcst,25)
```

```
##                PC1          PC2        PC3         PC4         PC5
## 842302   -9.1847552  -1.94687003 -1.1221788  3.63053641  1.19405948
## 842517   -2.3857026   3.76485906 -0.5288274  1.11728077 -0.62122836
## 84300903 -5.7288555   1.07422859 -0.5512625  0.91128084  0.17693022
## 84348301 -7.1166913 -10.26655564 -3.2299475  0.15241292  2.95827543
## 84358402 -3.9318425   1.94635898  1.3885450  2.93805417 -0.54626674
## 843786   -2.3781546  -3.94645643 -2.9322967  0.94020959  1.05511354
## 844359   -2.2369151   2.68766641 -1.6384712  0.14920860 -0.04032404
## 84458202 -2.1414143  -2.33818665 -0.8711807 -0.12693117  1.42618178
## 844981   -3.1721332  -3.38883114 -3.1172431 -0.60076844  1.52095211
## 84501001 -6.3461628  -7.72038095 -4.3380987 -3.37223437 -1.70875961
## 845636    0.8097013   2.65693767 -0.4884001 -1.67109618 -0.27556910
## 84610002 -2.6487698  -0.06650941 -1.5251134  0.05121650 -0.33165929
## 846226   -8.1778388  -2.69860201  5.7251932 -1.11127875 -1.04255311
## 846381   -0.3418251   0.96742803  1.7156620 -0.59447987 -0.46759907
## 84667401 -4.3385617  -4.85680983 -2.8136398 -1.45327830 -1.28892873
## 84799002 -4.0720732  -2.97444398 -3.1225267 -2.45590991  0.40798314
## 848406   -0.2298528   1.56338211 -0.8018136 -0.65001097  0.49427614
## 84862001 -4.4141269  -1.41742315 -2.2683231 -0.18610866  1.42260945
## 849014   -4.9443530   4.11071653 -0.3144724 -0.08812897  0.05666532
## 8510426   1.2359758   0.18804949 -0.5927619  1.59494272  0.44176553
## 8510653   1.5767738  -0.57230462 -1.7998630  1.12428647  0.39492224
## 8510824   3.5542090  -1.66148797  0.4507908  2.07194159  0.49031507
## 8511133  -4.7290497  -3.30205827 -1.4652474  2.03935567  0.02619707
## 851509   -4.2048244   5.12385806 -0.7517406 -0.86195184  0.47055388
## 852552   -4.9452807   1.54239514 -1.7116878  0.04671806  1.73770121
##                PC6          diagnosis
## 842302    1.41018364         1
## 842517    0.02863116         1
## 84300903  0.54097615         1
## 84348301  3.05073750         1
## 84358402 -1.22541641         1
## 843786   -0.45064213         1
## 844359   -0.12883507         1
## 84458202 -1.25593410         1
## 844981    0.55905282         1
## 84501001 -0.72327234         1
## 845636    0.12721990         1
## 84610002  0.76419423         1
## 846226    2.59229030         1
## 846381    1.00677426         1
## 84667401 -0.34940880         1
## 84799002  0.49534213         1
## 848406   -0.76152096         1
## 84862001 -0.75182778         1
## 849014   -1.13668869         1
## 8510426  -0.04859402         0
## 8510653   0.43046249         0
## 8510824  -0.76939136         0
## 8511133   3.02038624         1
## 851509   -0.59531662         1
## 852552   -0.81216500         1
```

```
   N=nrow(wdbc.pcst)
   N
```

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

```
   a=matrix(0,1000,2)
   for (I in 1:1000){
   for(j in 1:2){
   ind=sample(2,nrow(wdbc.pcst),replac=TRUE,prob=c(0.8,0.2))
   wdbc.pcst.train=wdbc.pcst[ind==1,]
   wdbc.pcst.test=wdbc.pcst[ind==2,]
   nrow(wdbc.pcst.train)
   nrow(wdbc.pcst.test)


# so 442 observations are in the training dataset
#  and 127 observations are in the test dataset.
#  We will use the training dataset to calculate the
# linear discriminant function  by passing it to the lda()
# function to the MASS PACAKAGE
  library(MASS)

 wdbc.pcst.train.df=wdbc.pcst.train
# convert matrix to a dataframe
   wdbc.pcst.train.df=as.data.frame(wdbc.pcst.train)
   wdbc.pcst.test.df=as.data.frame(wdbc.pcst.test)


#     PERFORM LDA ON DIAGNOSIS
   wdbc.lda=lda(diagnosis~PC1+PC2+PC3+PC4+PC5+PC6,data=wdbc.pcst.train.df)
# lets summarize the LDA OUTPUT
   attributes(wdbc.lda)
   head(wdbc.lda$prior)
   wdbc.lda$counts
   wdbc.lda$scaling
   p=predict(wdbc.lda,wdbc.pcst.train.df)$class
#confusion matrix and accuracy ~ training data
   tab=table(predicted=p,actual=wdbc.pcst.train.df$diagnosis)
   tab
   table(wdbc.pcst.train.df$diagnosis)
```

```
# accuracy of training data
   a[i,1]=sum(diag(tab))/sum(tab)
# confusion matrix and accuracy ~ testting  data
   p1=predict(wdbc.lda,wdbc.pcst.test.df)$class
   tab1=table(predicted=p1,actual=wdbc.pcst.test.df$diagnosis)
   tab1
   table(wdbc.pcst.test.df$diagnosis)
    # accuracy of testing data
   a[i,2]=sum(diag(tab1))/sum(tab1)}}
# model peroformance evaluation
   library(ROCR)
```

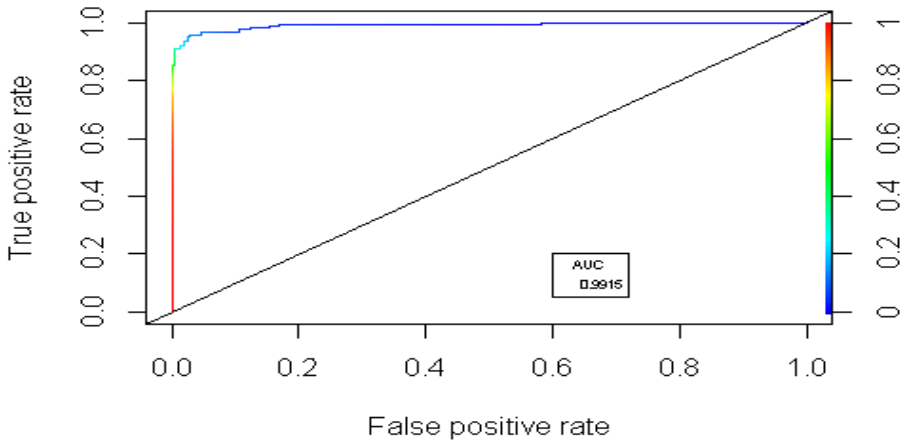
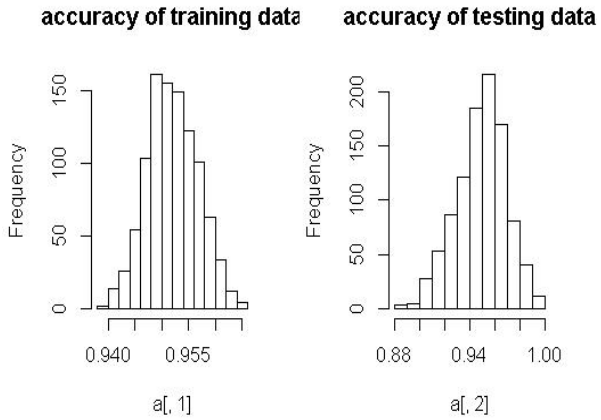


Fig no:7 Roc curve

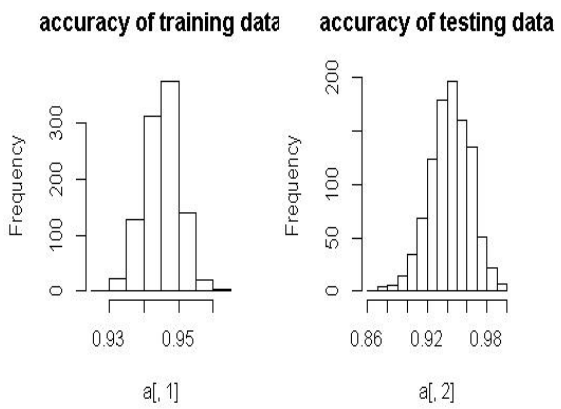
By applying the classification rule we have constructed a diagnostic system that predicts malignant tumors at 99.17% if all the variables are taken into consideration

```
# checking distribution of accuracy of testing and training data
head(a)

par(mfrow=c(1,2))
hist(a[,1],main="accuracy of training data")
hist(a[,2],main="accuracy of testing data")
```

After this we have run the program 1000 times and stored accuracy of training and testing model at each time and then we plotted the histogram of the same which shows normality.We have also tested accuracy of model by changing the percentage of training and test data



Histogram based on 30 variables



Histogram based on 20 variables

Conclusion

We have shown how dimensionality reduction technique like principal component analysis can be used to reduce a large number of highly correlated predictors to small set of linear combinations of those predictors. In doing so, we unveiled patterns in the data which led us to build a classification rule using linear discriminant analysis.

If we use 20 variables the AUC is 0.9894 and 30 variables it will be 0.9917.

Chapter 6. Overall Conclusion:

1. Chronic Kidney Disease Data

- From the CKD dataset analysis, we can conclude that the J48 algorithm gives us better accuracy of classifying the data.
- We can see that if we increase the training data the accuracy of the model also increases.

  J48 classification algorithm was making some mistake in Test-1 as it is classifying the 7 observations as no patient and error is 4.9296%.

  However, only 1 patient was classified as non-patient and the error is 1.85% in Test-2.
- From the decision tree we conclude that albumin factor is most important while classifying the data as CKD or NOTCKD.

3. Breast Cancer Data

We have shown that how PCA can be used for dimensionality reduction.

- We conclude that how precisely discriminant analysis can be used for classification.
- If we use 10 variables separately then the performance of model is
  - If we use variables in terms of mean the AUC is 96.36%
  - If we use variables in terms of maximum(worst) the AUC is 96.78%
  - If we use variables in terms of standard error the AUC is 92.83%
  - If we use all 30 variables the AUC is 99.17%
- If we use 20 (mean and maximum) variables the AUC is 0.9894 and 30 variables it will be 0.9917
- The accuracy of model for various testing and training Data sets.

Data (train ,test)	AUC	Training Data (Mean,SD)	Testing Data (Mean, SD)
(80,20)	0.9917	(0.9517,0.00462)	(0.9498,0.019321)
(60,40)	0.9931	(0.9524,0.0072)	(0.9485,0.01386)
(70,30)	0.9914	(0.9522,0.0057)	(0.9484,0.01603)
(50,50)	0.9937	(0.9525,0.008327)	(0.9479,0.012318)

- By applying the classification rule we have constructed a diagnostic system that predict malignant tumor.

Chapter 7.Scope and Limitations of the project

Scope:

- Here we have used J48 technique for classification but data mining provides us various strategies, one may use random forest, naïve bayse, SVM for better accuracy and further analysis.
- For breast cancer dataset one can also go with neural network instead of linear discriminant analysis.

Limitation:

- We were unaware of some medical terms regarding the datasets and as the datasets were secondary we had some limitation over our analysis.

References:

- Springer Series in Statistics **Data Mining, Inference, and Prediction**
  Trevor Hastie ,Robert Tibshirani ,Jerome Friedman (second edition)

- Analytics mantra weka tutorials
  https://www.youtube.com/playlist?list=PLJbE6j2EG1pZnBhOg3Rb63WLCprtyJag

- Lecture notes of "Multivariate Analysis" of M.sc (industrial statistics) of
  Prof. K.K.Kamalja ,
  North Maharashtra University, Jalgaon.
- Software : Weka ,R-Studio ,Minitab 17 , MS-Excel.

Datasets :

- Chronic kidney disease dataset is downloaded from UCI machine learning repository,USA.
  https://archive.ics.uci.edu/ml/datasets/Chronic_Kidney_Disease

- Breast Cancer datset is downloaded from
  https://archive.ics.uci.edu/ml/datasets/Brest+Cancer+Wisconsin+%28Diagnostic%29