Project 4.1

Disease Prediction Analysis

Introduction

Breast cancer (BC) is one of the most common cancers among women worldwide, representing most new cancer cases and cancer-related deaths according to global statistics, making it a significant public health problem today. Detecting breast (or any other type of) cancer before noticing symptoms is a key first step in fighting the disease. The process involves examining breast tissue for lumps or masses. Fine needle aspirate (FNA) biopsy is performed if such irregularity is found. The extracted tissue is then examined under a microscope by a clinician.

Can a machine help the clinician do a better job? Can the doctor focus more on treating the disease rather than detecting it? Recently, Deep Learning (DL) has seen major advances in the area of computer vision. Naturally, some scientists tried to apply it to breast cancer detection - and did so with great success!

The early diagnosis of BC can improve the prognosis and chance of survival significantly, as it can promote timely clinical treatment to patients. Further accurate classification of benign tumours can prevent patients undergoing unnecessary treatments. Thus, the correct diagnosis of BC and classification of patients into malignant or benign groups is the subject of much research. Because of its unique advantages in critical features detection from complex BC datasets, machine learning (ML) is widely recognized as the methodology of choice in BC pattern classification and forecast modelling.

Classification and data mining methods are an effective way to classify data. Especially in medical field, where those methods are widely used in diagnosis and analysis to make decisions.

Recommended Screening Guidelines:

Mammography. The most important screening test for breast cancer is the mammogram. A mammogram is an X-ray of the breast. It can detect breast cancer up to two years before the tumour can be felt by you or your doctor.

Women age 40–45 or older who are at average risk of breast cancer should have a mammogram once a year.

Women at high risk should have yearly mammograms along with an MRI starting at age 30.

Some Risk Factors for Breast Cancer

The following are some of the known risk factors for breast cancer. However, most cases of breast cancer cannot be linked to a specific cause. Talk to your doctor about your specific risk.

Age. The chance of getting breast cancer increases as women age. Nearly 80 percent of breast cancers are found in women over the age of 50.

Personal history of breast cancer. A woman who has had breast cancer in one breast is at an increased risk of developing cancer in her other breast.

Family history of breast cancer. A woman has a higher risk of breast cancer if her mother, sister or daughter had breast cancer, especially at a young age (before 40). Having other relatives with breast cancer may also raise the risk.

Genetic factors. Women with certain genetic mutations, including changes to the BRCA1 and BRCA2 genes, are at higher risk of developing breast cancer during their lifetime. Other gene changes may raise breast cancer risk as well.

Childbearing and menstrual history. The older a woman is when she has her first child, the greater her risk of breast cancer and at higher risk are:

Women who menstruate for the first time at an early age (before 12)

Women who go through menopause late (after age 55)

Women who've never had children

Data Preparation

The program uses a curve-fitting algorithm, to compute ten features from each one of the cells in the sample, then it calculates the mean value, extreme value and standard error of each feature for the image, returning a 30 real-valuated vector

Attribute Information:

ID number 2) Diagnosis (M = malignant, B = benign) 3–32)

Ten real-valued features are computed for each cell nucleus:

radius (mean of distances from centre to points on the perimeter)

texture (standard deviation of grey-scale values)

perimeter

area

smoothness (local variation in radius lengths)

compactness (perimeter^2 / area — 1.0)

concavity (severity of concave portions of the contour)

concave points (number of concave portions of the contour)

symmetry

fractal dimension ("coastline approximation" — 1)

The mean, standard error and "worst" or largest (mean of the three largest values) of these features were computed for each image, resulting in 30 features. For instance, field 3 is Mean Radius, field 13 is Radius SE, field 23 is Worst Radius.

Objectives

This analysis aims to observe which features are most helpful in predicting malignant or benign cancer and to see general trends that may aid us in model selection and hyper parameter selection. The goal is to classify whether the breast cancer is benign or malignant. To achieve this machine learning classification methods to fit a function that can predict the discrete class is used.

Build Machine Learning Models to predict the type of Breast Cancer (Malignant or Benign) as well as identify the drivers of cancer.

Apply the concepts like Logistic Regression and Random Forest.

3. Approach

• Exploring features and Data Preparation which includes missing value treatment and Outlier Detection

• Visualizing relationships among features

• Split the data into train and test data and build sophisticated Machine Learning models

• Evaluating Model performance on test data using Precision, Recall, Accuracy and ROC curve metrics

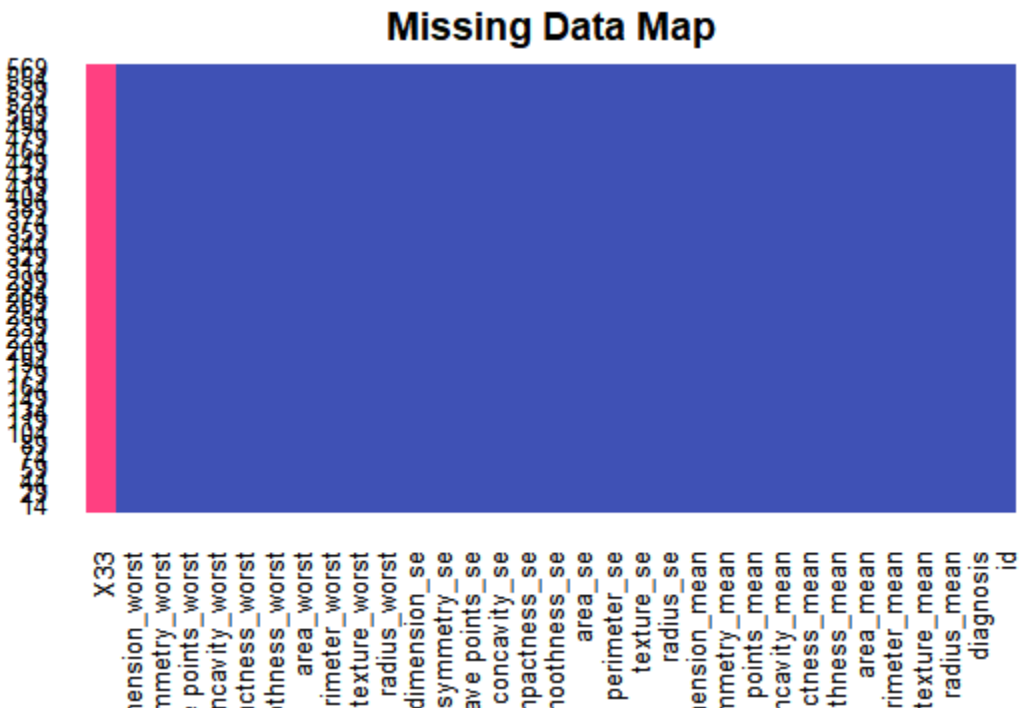
• Determining the factors driving the cancer.

• Choosing best model based on the accuracy and other measures.

This analysis is on a dataset containing information on over 500 incidences of breast cancer. Each instance is classified as either benign or malicious and has various characteristics that can be used in determining the threat of the cancerous region. Various machine learning techniques were used to model the breast cancer dataset, Random Forest, Logistic Regression, Naive Bayes, Support Vector Machines and Decision Trees in this project.

Exploratory Phase

```
miss map(Cancer Data, main="Missing Data Map", col=c("#FF4081", "#3F51B5"),
    legend=FALSE)
```



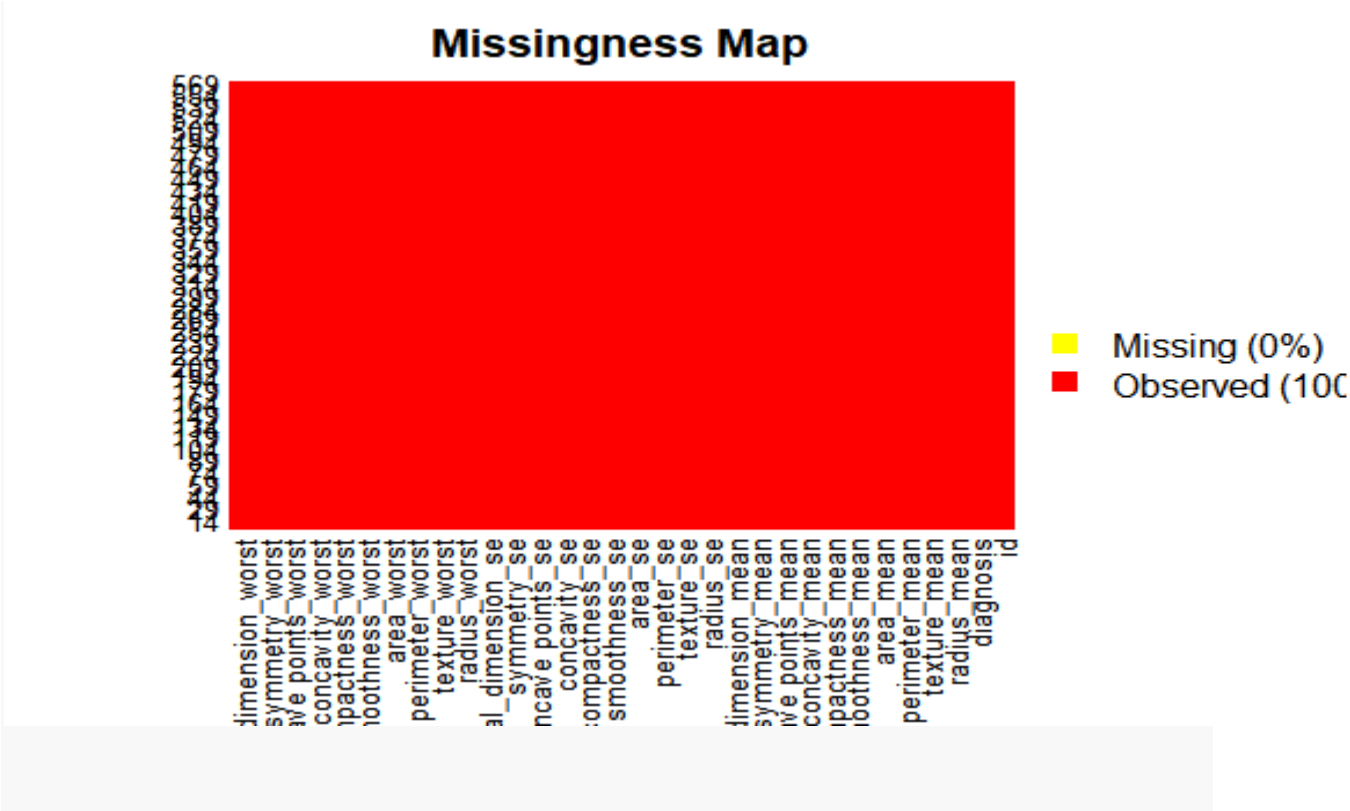
There is a 3% missing value in the data and the column 33 provided has all NA values and the same is r emoved

```
dim(Cancer Data)
```

```
[1] 569  33
```

```
data[,33]<-NULL
```

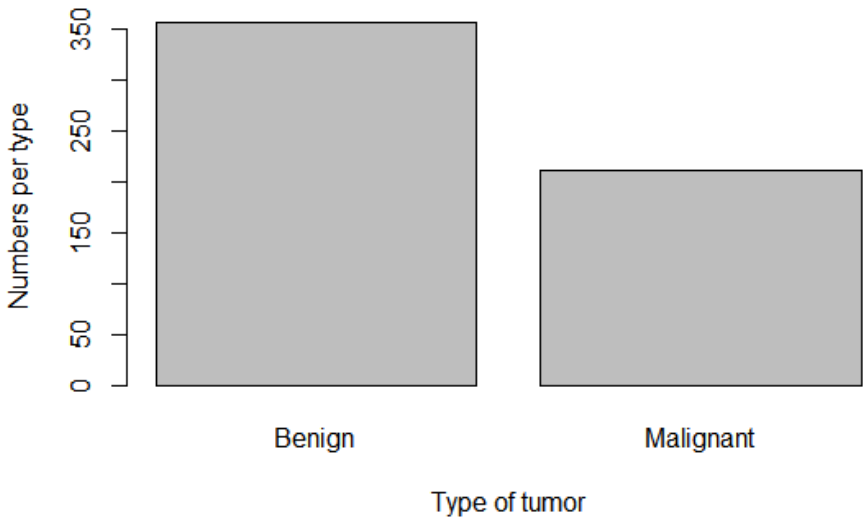
```
# visualize the missing values using the missing map after removal of colu
mn 33 from the Amelia package
miss map(data,col=c("yellow","red"))
```



There are two main classifications **of** tumors. One is known as **benign** and the other as **malignant**. A **benign** tumor is a tumor that does not invade its surrounding tissue or spread around the body. A **malignant** tumor is a tumor that may invade its surrounding tissue or spread around the body. **Malignant tumors** are **cancerous tumors** that can potentially result in death. Unlike benign **tumors**, **malignant** ones grow quickly, and can spread to new territory in a process known as metastasis.

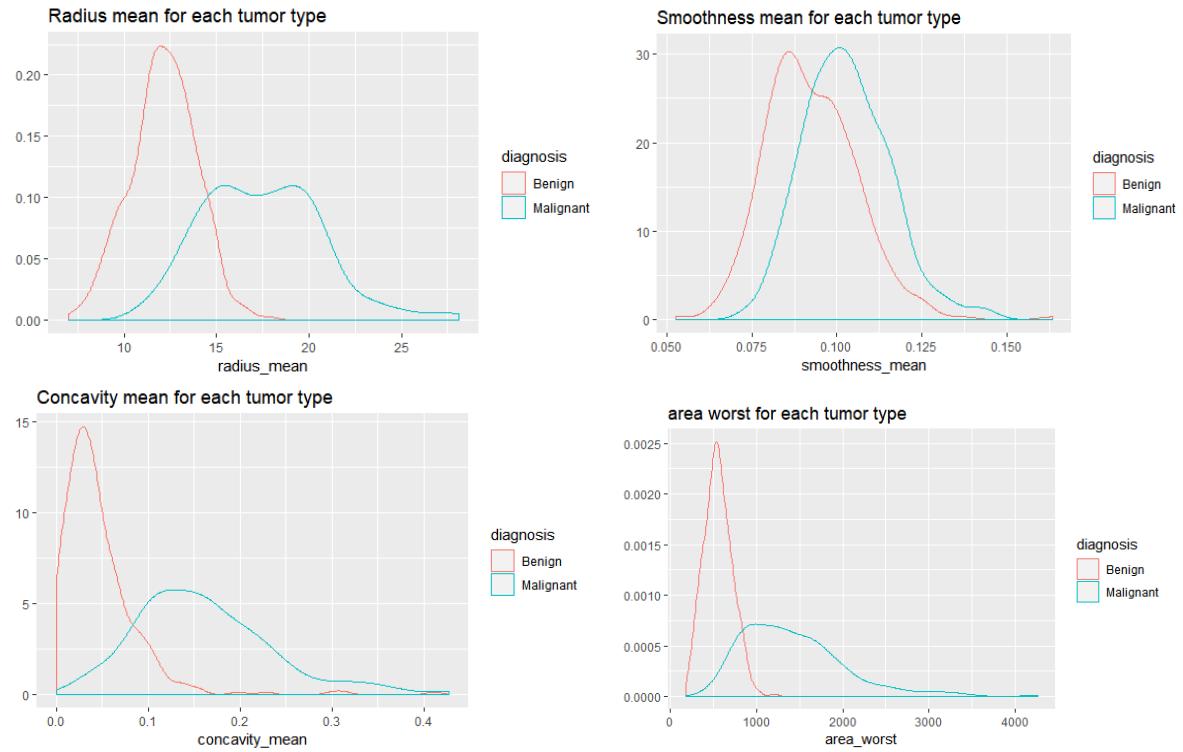
The abnormal cells that form a **malignant tumor** multiply at a faster rate. Often, benign tumors need no treatment, but they can become dangerous if they grow large enough to press on vital organs, blood vessels or nerves. In such cases they are generally removed through surgery, which also allows pathologists to confirm that they are not malignant.

barplot(table(data$diagnosis), xlab = "Type of tumor", ylab="Numbers per type")

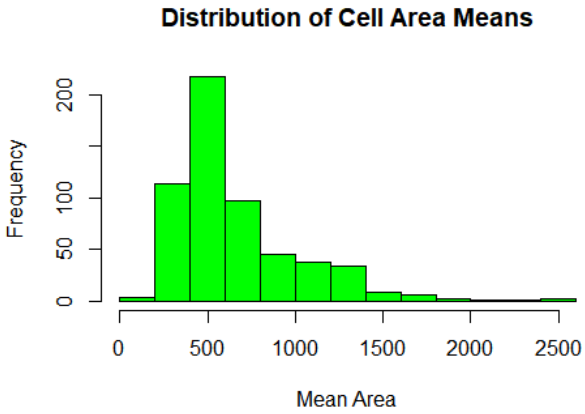


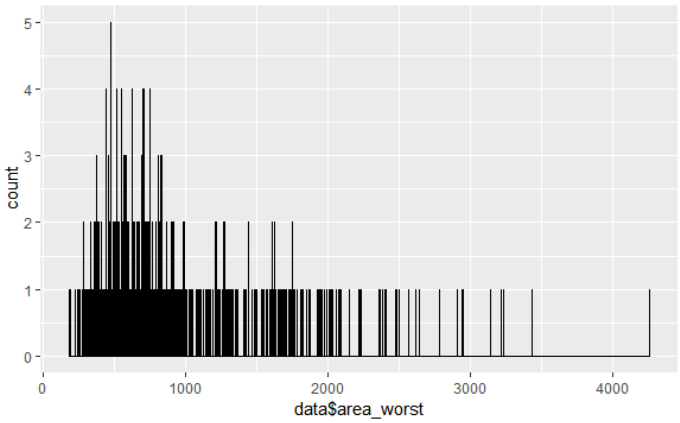
Let's see if we can differentiate between tumor types using some features (randomly chosen?):

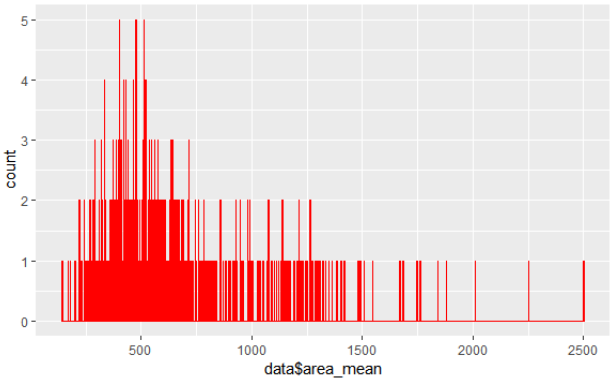
```
qplot(radius_mean, data=data, colour=diagnosis, geom="density",
      main="Radius mean for each tumor type")

qplot(smoothness_mean, data=data, colour=diagnosis, geom="density",
      main="Smoothness mean for each tumor type")

qplot(concavity_mean, data=data, colour=diagnosis, geom="density",
      main="Concavity mean for each tumor type")

qplot(area_worst , data=data, colour=diagnosis, geom="density",
      main="area worst for each tumor type")
```

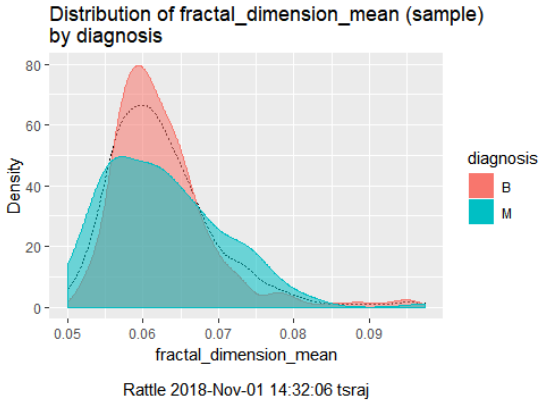


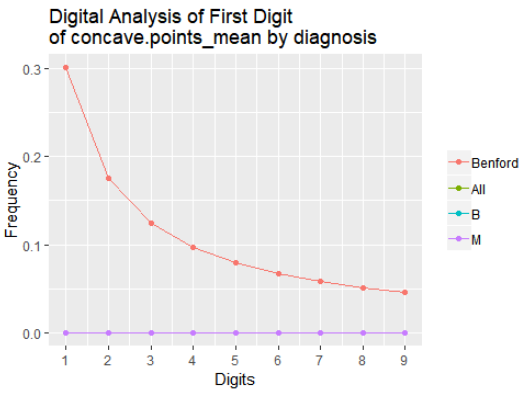

```r
# Looking at distribution for area.mean variable
plot.new()
hist(CancerData$area_mean,
     main = 'Distribution of Cell Area Means',
     xlab = 'Mean Area',
     col = 'green')
```

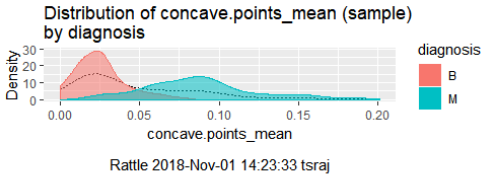

Distribution of Cell Area Means
Frequency
0
50
100
200
0
500
1000
1500
2000
2500
Mean Area

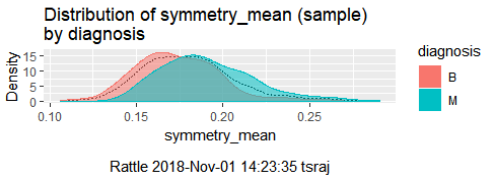

count
0
1
2
3
4
5
0
1000
2000
3000
4000
data$area_worst


count
0
1
2
3
4
5
500
1000
1500
2000
2500
data$area_mean

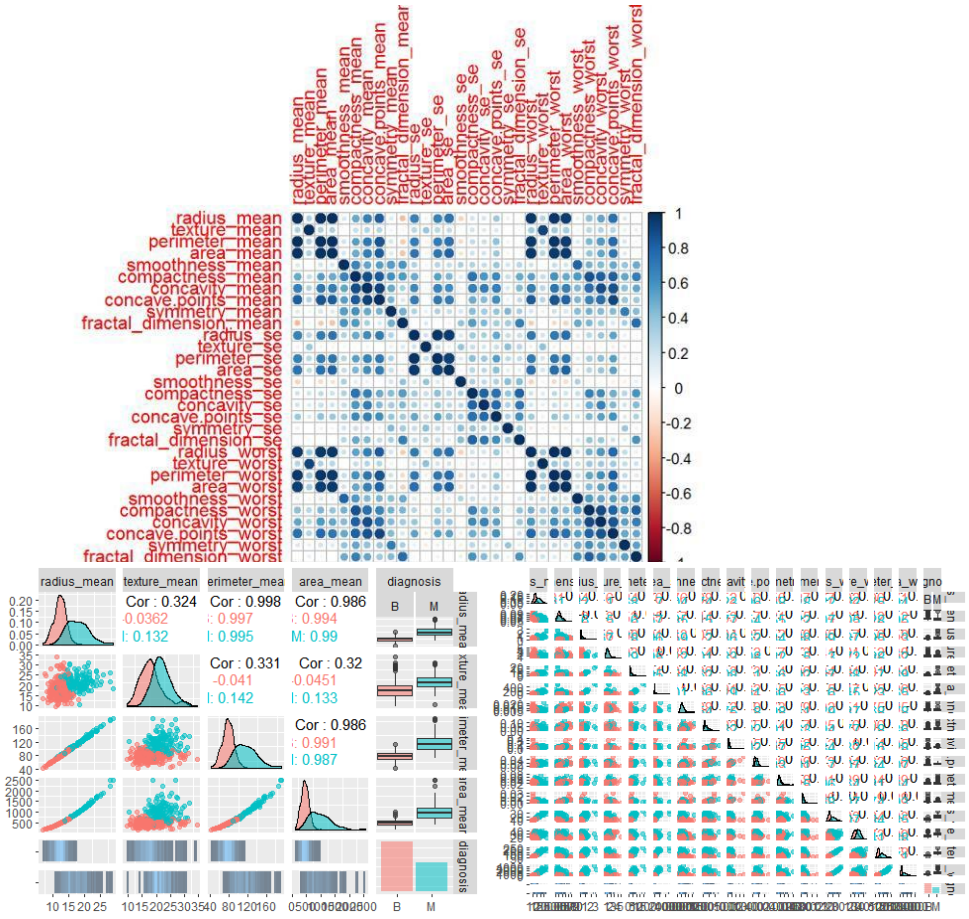









```
prop.table(table(data$diagnosis))

##
##         B         M
## 0.6274165 0.3725835

## we then show some correlation
corr_mat<-cor(data[,3:ncol(data)])
corrplot(corr_mat)
```




```
#Modelling
#We are going to get a training and a testing set to use when building
some models:
set.seed(1234)
data_index<-createDataPartition(data$diagnosis,p=0.75,list = FALSE)
train_data<-data[data_index,-1]
test_data<-data[data_index,-1]

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

```
#Model1: Random Forest
#Building the model on the training data
## random forest
model_rf <- train(diagnosis~.,
                  train_data,
                  method="ranger",
                  metric="ROC",
                  #tuneLength=10,
```

```
                  #tuneGrid = expand.grid(mtry = c(2, 3, 6)),
                  preProcess = c('center', 'scale'),
                  trControl=fitControl)

#Testing on the testing data
## testing for random forets
pred_rf <- predict(model_rf, test_data)
cm_rf <- confusionMatrix(pred_rf, test_data$diagnosis, positive = "M")
cm_rf
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   B   M
##          B 268   0
##          M   0 159
##
##                Accuracy : 1
##                  95% CI : (0.9914, 1)
##     No Information Rate : 0.6276
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 1
##  Mcnemar's Test P-Value : NA
##
##             Sensitivity : 1.0000
##             Specificity : 1.0000
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 1.0000
##              Prevalence : 0.3724
##          Detection Rate : 0.3724
##    Detection Prevalence : 0.3724
##       Balanced Accuracy : 1.0000
##
##        'Positive' Class : M
##
```

```
# We find the accuracy of the model is 100%
```

```
#Model2: Naive Bayes
#Building and testing the model
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   B   M
##          B 259  17
##          M   9 142
##
##                Accuracy : 0.9391
##                  95% CI : (0.9121, 0.9598)
##     No Information Rate : 0.6276
##     P-Value [Acc > NIR] : <2e-16
```

```
##
##                    Kappa : 0.8684
##  Mcnemar's Test P-Value : 0.1698
##
##             Sensitivity : 0.8931
##             Specificity : 0.9664
##          Pos Pred Value : 0.9404
##          Neg Pred Value : 0.9384
##              Prevalence : 0.3724
##          Detection Rate : 0.3326
##    Detection Prevalence : 0.3536
##       Balanced Accuracy : 0.9297
##
##        'Positive' Class : M
##
```

```
#Accuracy of the model is 93.9%
```

```
#Model3: glm
#Building and testing the model
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   B   M
##          B 265   4
##          M   3 155
##
##                Accuracy : 0.9836
##                  95% CI : (0.9665, 0.9934)
##     No Information Rate : 0.6276
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.9649
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 0.9748
##             Specificity : 0.9888
##          Pos Pred Value : 0.9810
##          Neg Pred Value : 0.9851
##              Prevalence : 0.3724
##          Detection Rate : 0.3630
##    Detection Prevalence : 0.3700
##       Balanced Accuracy : 0.9818
##
##        'Positive' Class : M
##
```

```
#Accuracy of the model is 98.3%
```

```
## Evaluation on training data (569 cases):
##
##      Decision Tree
```

```
##     ----------------
##     Size       Errors
##
##       11     7( 1.2%)   <<
##
##
##      (a)   (b)    <-classified as
##     ----  ----
##      356     1    (a): class 1
##        6   206    (b): class 2
##
##
##   Attribute usage:
##
##   100.00% area_worst
##    67.84% concave points_worst
##    63.44% area_se
##    32.16% concavity_mean
##     8.61% texture_worst
##     3.34% texture_mean
##     3.16% symmetry_worst
##     2.11% perimeter_se

## Evaluation on training data (569 cases):
##
##           Rules
##     ----------------
##       No       Errors
##
##        6   13( 2.3%)   <<
##
##
##      (a)   (b)    <-classified as
##     ----  ----
##      357          (a): class 1
##       13   199    (b): class 2
##
##
##   Attribute usage:
##
##    98.42% area_worst
##    68.01% concavity_mean
##    61.34% texture_mean
##    26.89% concave points_worst
##    20.04% texture_worst
##


## Root node error: 159/427 = 0.37237
##
## n= 427
```

```
##
##         CP nsplit rel error  xerror     xstd
## 1 0.811321      0   1.00000 1.00000 0.062828
## 2 0.069182      1   0.18868 0.26415 0.038703
## 3 0.031447      2   0.11950 0.22013 0.035651
## 4 0.010000      3   0.08805 0.19497 0.033722
```

```
summary(fit1)
```

```
## Call:
## rpart(formula = diagnosis ~ ., data = train_data)
##   n= 427
##
##           CP nsplit  rel error    xerror       xstd
## 1 0.81132075      0 1.00000000 1.0000000 0.06282824
## 2 0.06918239      1 0.18867925 0.2201258 0.03565053
## 3 0.03144654      2 0.11949686 0.1635220 0.03107762
## 4 0.01000000      3 0.08805031 0.1823899 0.03269862
##
## Variable importance
##         radius_worst            area_worst      perimeter_worst
##                   16                    16                   15
##            area_mean          radius_mean       perimeter_mean
##                   14                    14                   14
## concave points_worst      concavity_worst       concavity_mean
##                    3                     2                    1
##    compactness_worst  concave points_mean     compactness_mean
##                    1                     1                    1
##        texture_worst
##                    1
```

```
data_classifier
```

```
## Support Vector Machine object of class "ksvm"
##
## SV type: C-svc  (classification)
##  parameter : cost C = 1
##
## Linear (vanilla) kernel function.
##
## Number of Support Vectors : 28
##
## Objective Function Value : -13.7674
## Training error : 0.007026
```

```
table(data_predictions, test_data$diagnosis)
```

```
##
## data_predictions   B   M
##                B 267   2
##                M   1 157
```

```
agreement<-data_predictions == test_data$diagnosis
table(agreement)
```

```
## agreement
## FALSE  TRUE
##     3   424
```

```
prop.table(table(agreement))
```

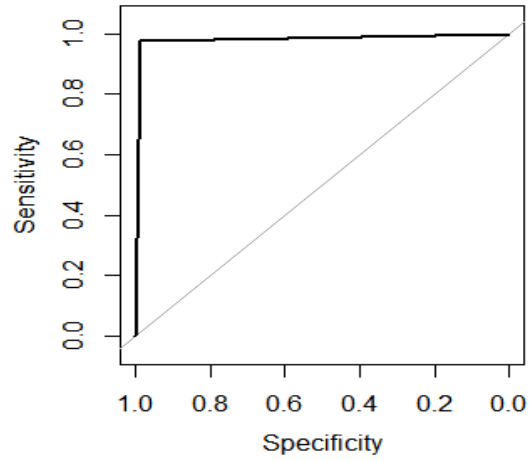
```
## agreement
##       FALSE        TRUE
## 0.007025761 0.992974239
```

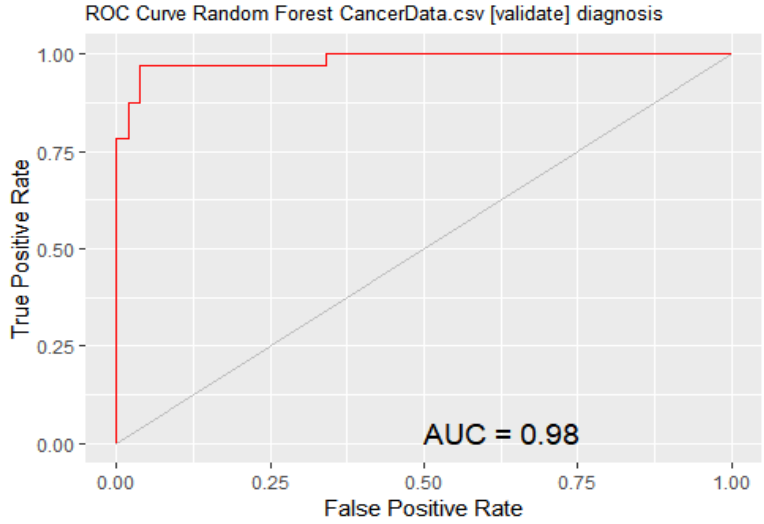
```
## agreement_rbf
## FALSE  TRUE
##     2   425
```

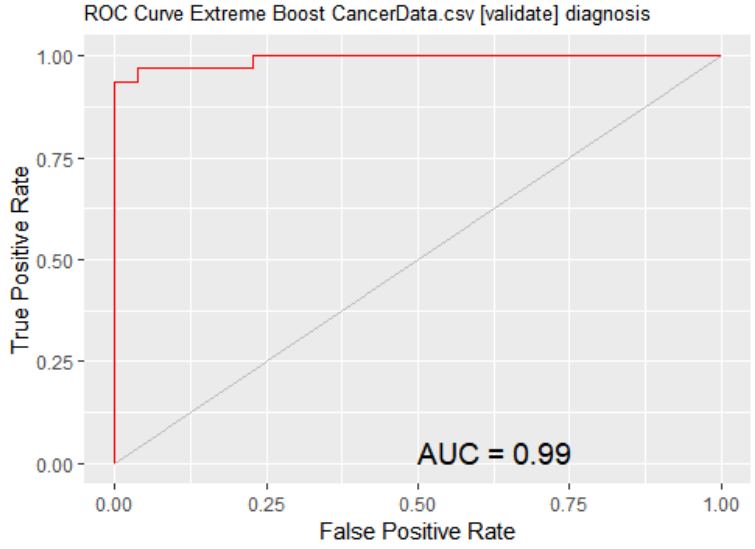
```
prop.table(table(agreement_rbf))
```

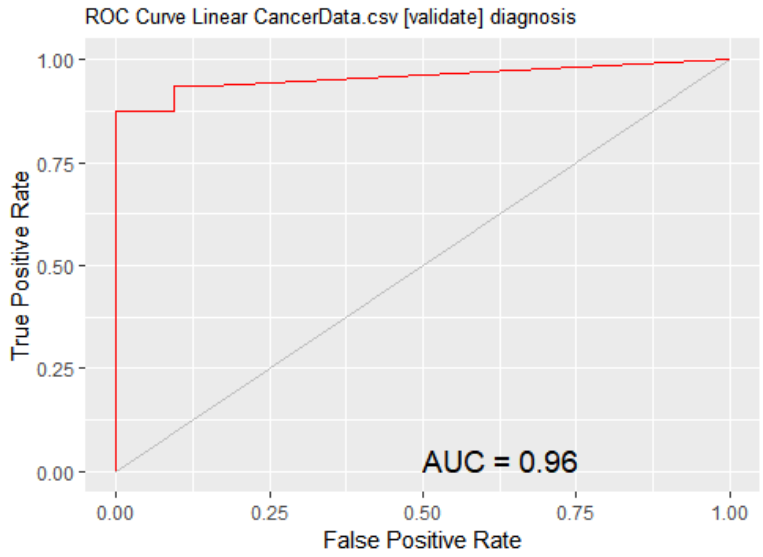
```
## agreement_rbf
##       FALSE        TRUE
## 0.004683841 0.995316159
```

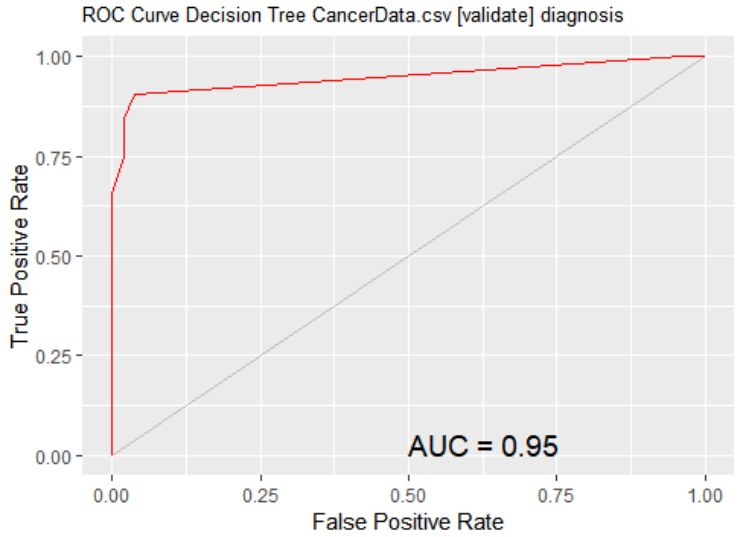
Various ROC curves. Based on the various classification models all classifications show more than 90% accuracy and we performed 10-fold validation check in Logistic regression through Vif(step fit) and step default. Further variable importance findings in Random forest, Logistic regression through iv.plot, 10-fold repeated 3 times var important(step fit) with Library(caret),Relative variable importance with *Mean Decrease Accuracy, Mean Decrease Gini. Important variables through* ***library****(Boruta) identified more than 26 variables. Finally, identification of variable importance performed through Mars(earth package) and the same identified around 9 important variables. These variables are mostly identified by models like Random Forest,vif(stepfit) etc.,*

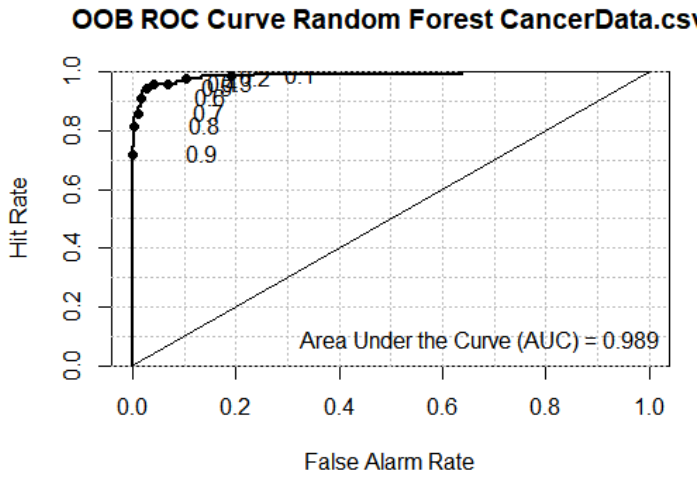

Sensitivity
0.0
0.2
0.4
0.6
0.8
1.0
1.0
0.8
0.6
0.4
0.2
0.0
Specificity


ROC Curve Random Forest CancerData.csv [validate] diagnosis
True Positive Rate
1.00
0.75
0.50
0.25
0.00
AUC = 0.98
0.00
0.25
0.50
0.75
1.00
False Positive Rate


ROC Curve Extreme Boost CancerData.csv [validate] diagnosis
True Positive Rate
1.00
0.75
0.50
0.25
0.00
AUC = 0.99
0.00
0.25
0.50
0.75
1.00
False Positive Rate


ROC Curve Linear CancerData.csv [validate] diagnosis
True Positive Rate
False Positive Rate
AUC = 0.96
0.00
0.25
0.50
0.75
1.00


ROC Curve Decision Tree CancerData.csv [validate] diagnosis
True Positive Rate
False Positive Rate
AUC = 0.95
0.00
0.25
0.50
0.75
1.00


OOB ROC Curve Random Forest CancerData.csv
Hit Rate
False Alarm Rate
Area Under the Curve (AUC) = 0.989
0.0
0.2
0.4
0.6
0.8
1.0

```
# logistic regression model:
## Coefficients:
##                            Estimate Std. Error   z value Pr(>|z|)
## (Intercept)              -5.487e+15  1.418e+08 -38703923   <2e-16 ***
## radius_mean              -1.401e+13  5.949e+07   -235423   <2e-16 ***
## texture_mean             -5.783e+13  2.594e+06 -22293459   <2e-16 ***
## perimeter_mean           -1.954e+14  8.518e+06 -22935779   <2e-16 ***
## area_mean                 7.231e+12  1.723e+05  41962794   <2e-16 ***
## smoothness_mean           1.141e+16  6.970e+08  16374586   <2e-16 ***
## compactness_mean         -1.560e+16  4.601e+08 -33898361   <2e-16 ***
## concavity_mean            3.612e+15  3.663e+08   9859481   <2e-16 ***
## `concave points_mean`     3.368e+16  6.496e+08  51839897   <2e-16 ***
## symmetry_mean             7.166e+14  2.485e+08   2883416   <2e-16 ***
## fractal_dimension_mean   -1.875e+16  1.853e+09 -10119625   <2e-16 ***
## radius_se                -1.780e+14  1.147e+08  -1552350   <2e-16 ***
## texture_se               -5.141e+14  1.143e+07 -44982769   <2e-16 ***
## perimeter_se             -1.506e+14  1.516e+07  -9929607   <2e-16 ***
## area_se                   3.909e+12  4.713e+05   8294154   <2e-16 ***
## smoothness_se             6.741e+16  2.230e+09  30224242   <2e-16 ***
## compactness_se           -1.263e+16  7.957e+08 -15868906   <2e-16 ***
## concavity_se             -6.112e+15  4.465e+08 -13688233   <2e-16 ***
## `concave points_se`       2.479e+16  1.882e+09  13170418   <2e-16 ***
## symmetry_se               3.309e+16  8.953e+08  36963236   <2e-16 ***
## fractal_dimension_se      2.482e+16  4.032e+09   6155984   <2e-16 ***
## radius_worst              7.751e+14  2.067e+07  37495454   <2e-16 ***
## texture_worst             1.151e+14  2.192e+06  52500738   <2e-16 ***
## perimeter_worst           7.806e+13  2.049e+06  38088467   <2e-16 ***
## area_worst               -5.352e+12  1.108e+05 -48313624   <2e-16 ***
## smoothness_worst         -4.364e+15  4.930e+08  -8850467   <2e-16 ***
## compactness_worst         1.527e+15  1.306e+08  11684310   <2e-16 ***
## concavity_worst           2.629e+15  9.403e+07  27964084   <2e-16 ***
## `concave points_worst`   -5.585e+15  3.231e+08 -17282850   <2e-16 ***
## symmetry_worst           -1.380e+15  1.615e+08  -8543749   <2e-16 ***
## fractal_dimension_worst   8.968e+15  7.758e+08  11560246   <2e-16 ***
```

```
#ANOVA on base model
anova(fit,test = 'Chisq')
```

```
## Model: binomial, link: logit
##
## Response: diagnosis
##
## Terms added sequentially (first to last)
##
##
##                          Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
```

```
## NULL                                    426     563.81
## radius_mean              1   312.35       425     251.46 < 2.2e-16 ***
## texture_mean             1    22.22       424     229.24 2.431e-06 ***
## perimeter_mean           1    60.59       423     168.65 7.016e-15 ***
## area_mean                1     7.82       422     160.83 0.0051568 **
## smoothness_mean          1    34.03       421     126.79 5.416e-09 ***
## compactness_mean         1     0.02       420     126.77 0.8900612
## concavity_mean           1    11.89       419     114.88 0.0005637 ***
## `concave points_mean`    1     2.64       418     112.24 0.1041743
## symmetry_mean            1     3.55       417     108.69 0.0595695 .
## fractal_dimension_mean   1     0.48       416     108.21 0.4872629
## radius_se                1     4.78       415     103.42 0.0287116 *
## texture_se               1     9.47       414      93.95 0.0020869 **
## perimeter_se             1     0.05       413      93.90 0.8153014
## area_se                  1    12.15       412      81.75 0.0004913 ***
## smoothness_se            1     1.73       411      80.02 0.1883121
## compactness_se           1    20.73       410      59.29 5.295e-06 ***
## concavity_se             1     6.22       409      53.07 0.0126083 *
## `concave points_se`      1     1.12       408      51.94 0.2891473
## symmetry_se              1     1.00       407      50.94 0.3161479
## fractal_dimension_se     1     1.34       406      49.59 0.2461846
## radius_worst             1     0.00       405     648.79 1.0000000
## texture_worst            1   648.79       404       0.00 < 2.2e-16 ***
## perimeter_worst          1     0.00       403       0.00 0.9999778
## area_worst               1     0.00       402       0.00 0.9998569
## smoothness_worst         1     0.00       401       0.00 0.9998323
## compactness_worst        1     0.00       400       0.00 0.9998844
## concavity_worst          1     0.00       399       0.00 1.0000000
## `concave points_worst`   1     0.00       398       0.00 0.9999370
## symmetry_worst           1     0.00       397       0.00 1.0000000
## fractal_dimension_worst  1     0.00       396     504.61 1.0000000
## ---
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: diagnosis
##
## Terms added sequentially (first to last)
##
##
##                          Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                                     426     563.81
## concavity_mean            1  290.218       425     273.60 < 2.2e-16 ***
## `concave points_mean`     1   76.300       424     197.30 < 2.2e-16 ***
## symmetry_mean             1    4.970       423     192.32   0.02578 *
## smoothness_se             1    6.224       422     186.10   0.01260 *
## fractal_dimension_se      1   33.111       421     152.99 8.706e-09 ***
## texture_worst             1   46.144       420     106.85 1.099e-11 ***
```

```
## perimeter_worst           1   59.618        419      47.23 1.152e-14 ***
## compactness_worst         1    3.765        418      43.46   0.05234 .
## fractal_dimension_worst   1   43.464        417       0.00 4.319e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
vif(step_fit)
```

```
##          concavity_mean    `concave points_mean`           symmetry_mean
##                244.05337                99.94645                317.05513
##           smoothness_se    fractal_dimension_se           texture_worst
##             4608.37740             6335.09066             1093.86196
##         perimeter_worst       compactness_worst fractal_dimension_worst
##             1517.71228             5118.72975             6430.41696
```

```
print(fit_default)
```

```
## Generalized Linear Model
##
## 427 samples
##  30 predictor
##   2 classes: 'B', 'M'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 3 times)
## Summary of sample sizes: 384, 384, 385, 384, 385, 384, ...
## Resampling results:
##
##   Accuracy   Kappa
##   0.9516242  0.8968547
```

```
library(caret)
varImp(step_fit)
```

```
##                            Overall
## concavity_mean          0.04016248
## `concave points_mean`   0.04060020
## symmetry_mean           0.04004251
## smoothness_se           0.04107363
## fractal_dimension_se    0.04113828
## texture_worst           0.04104256
## perimeter_worst         0.04095488
## compactness_worst       0.04099049
## fractal_dimension_worst 0.04099415
```

```
varImp(fit_default)
```

```
## glm variable importance
##
##   only 20 most important variables shown (out of 30)
##
##                               Overall
## texture_worst                  100.00
## `\\`concave points_mean\\``     98.74
```

```
## area_worst                      91.99
## texture_se                      85.62
## area_mean                       79.84
## perimeter_worst                 72.42
## radius_worst                    71.29
## symmetry_se                     70.27
## compactness_mean                64.41
## smoothness_se                   57.38
## concavity_worst                 53.05
## perimeter_mean                  43.43
## texture_mean                    42.20
## `\\`concave points_worst\\``    32.62
## smoothness_mean                 30.88
## compactness_se                  29.91
## concavity_se                    25.74
## `\\`concave points_se\\``       24.75
## compactness_worst               21.91
## fractal_dimension_worst         21.67
```

```
library(woe)

library(riv)

train_data<-as.data.frame(train_data)

iv_df <- iv.mult(train_data, y="diagnosis", summary=TRUE, verbose=TRUE)

iv_df

iv <- iv.mult(train_data, y="diagnosis", summary=FALSE, verbose=TRUE)

iv_df
```

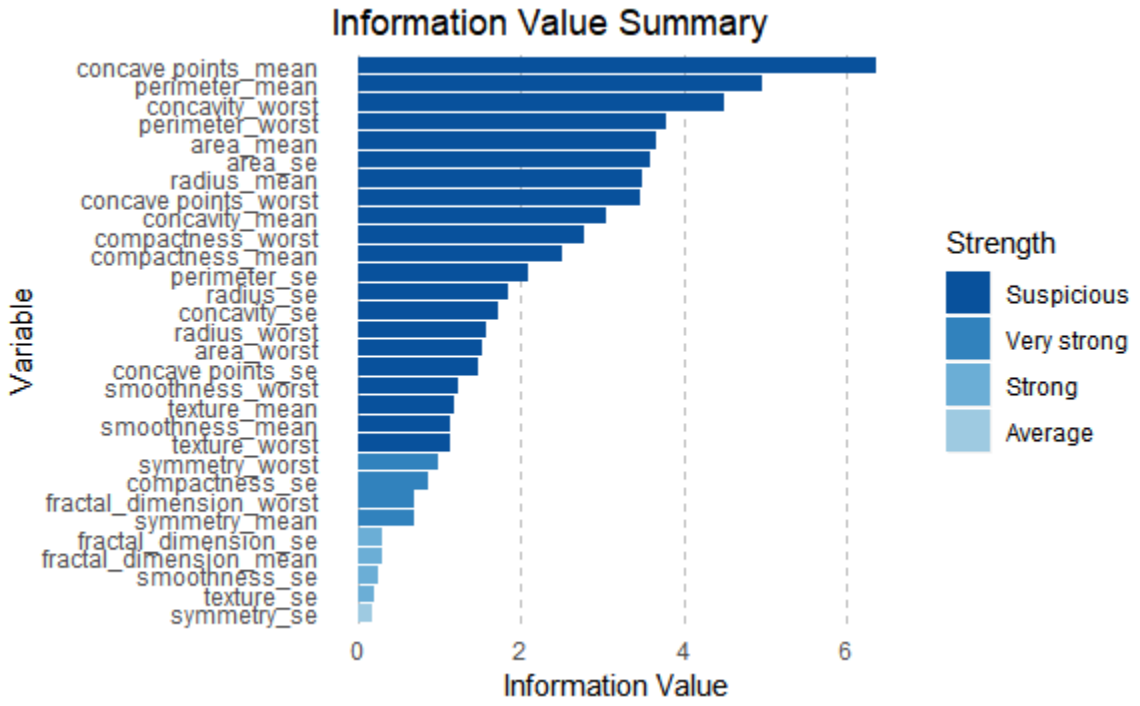
	Variable	Information	Value	Bins	Zero Bins	Strength
1	concave points_mean		6.3541081	5	0	Suspicious
2	perimeter_mean		4.9638289	4	0	Suspicious
3	concavity_worst		4.4909270	4	0	Suspicious
4	perimeter worst		3.7922674	5	1	Suspicious
5	area_mean		3.6702849	4	1	Suspicious
6	area_se		3.5749979	4	0	Suspicious
7	radius mean		3.4772020	5	1	Suspicious
8	concave points worst		3.4756344	5	1	Suspicious
9	concavity mean		3.0356262	6	1	Suspicious
10	compactness worst		2.7665883	5	0	Suspicious
11	compactness mean		2.5078805	5	0	Suspicious
12	perimeters		2.0849968	6	1	Suspicious
13	radius_se		1.8363325	5	1	Suspicious
14	concavity_se		1.7134338	5	0	Suspicious
15	radius worst		1.5670693	5	2	Suspicious
16	area_worst		1.5115545	5	2	Suspicious
17	concave points_se		1.4623521	5	0	Suspicious
18	smoothness_worst		1.2334093	5	0	Suspicious
19	texture_mean		1.1714620	6	0	Suspicious
20	smoothness_mean		1.1352591	6	0	Suspicious
21	texture_worst		1.1186736	5	0	Suspicious

22	symmetry_worst	0.9764180	5	0	Very strong
23	compactness_se	0.8494686	6	0	Very strong
24	fractal_dimension_worst	0.6992234	5	0	Very strong
25	symmetry_mean	0.6878786	6	0	Very strong
26	fractal_dimension_se	0.3035412	5	0	Strong
27	fractal_dimension_mean	0.2839318	6	0	Strong
28	smoothness_se	0.2490128	6	0	Strong
29	texture_se	0.2015776	6	0	Strong
30	symmetry_se	0.1679877	6	0	Average

Plot information value summary

iv.plot.summary(iv_df)

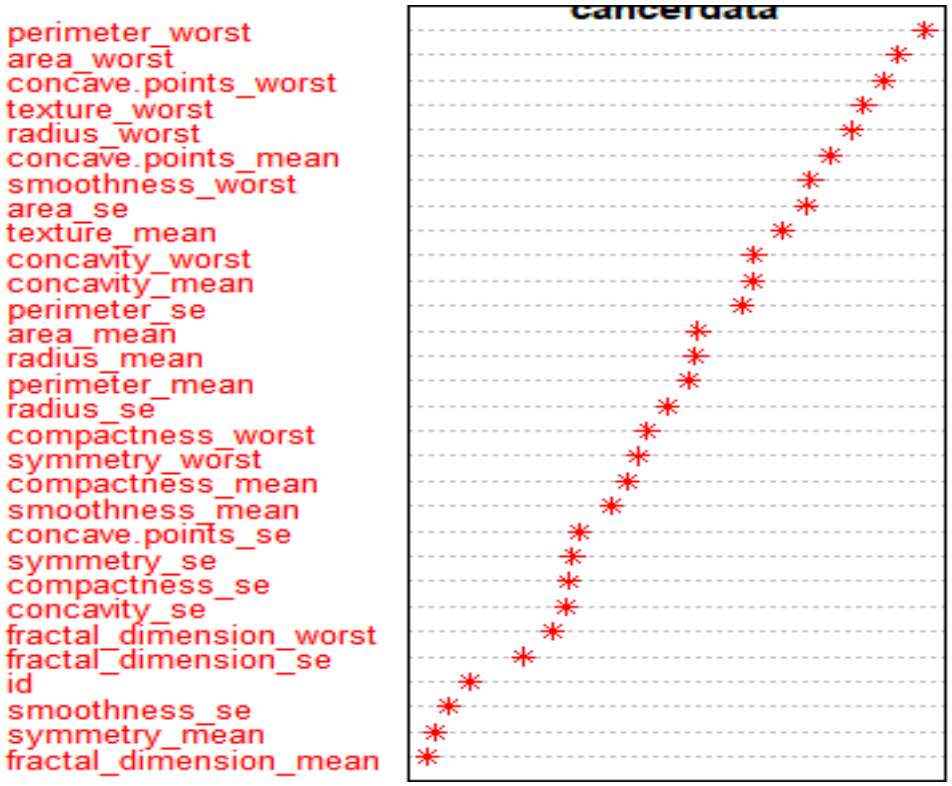
Information Value (IV) is frequently used to compare predictive power among variables. When developing new scorecards using logistic regression, variables are often binned and recoded using WoE concept. Package riv will help you to assess predicive power of variables, assess WoE patterns and recode raw variables to WoE.

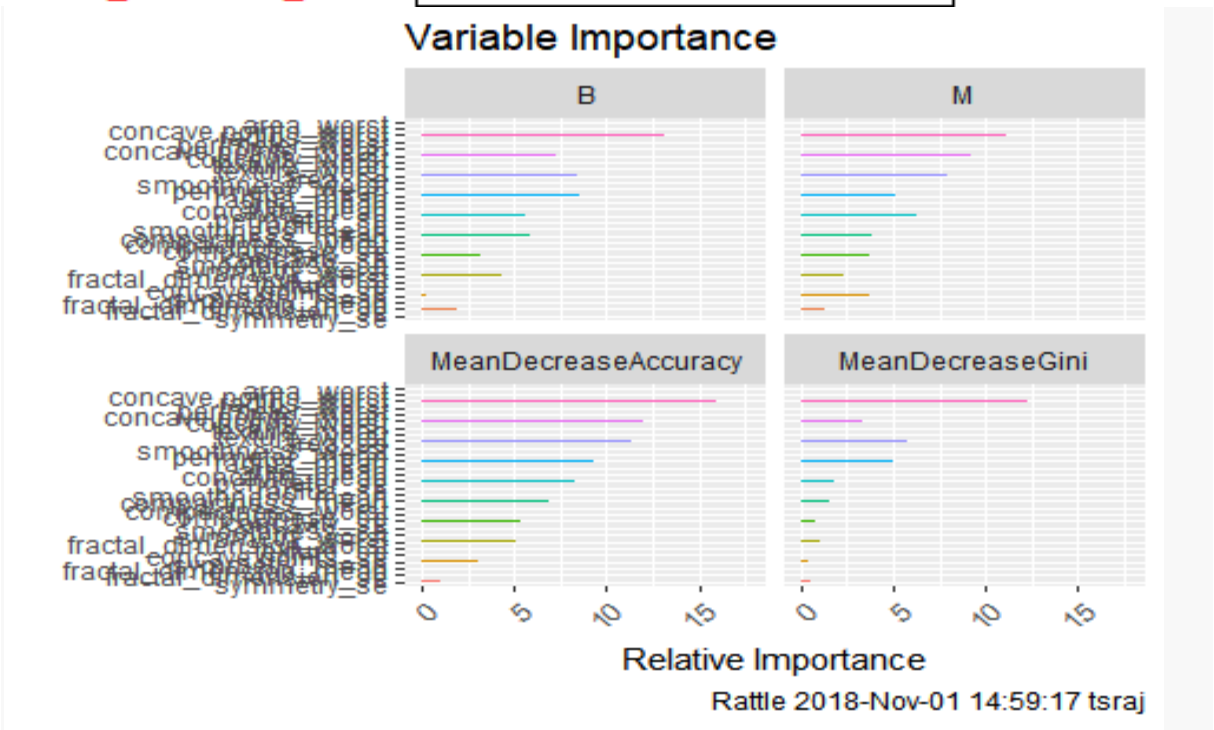



```
#Random forest model- takes decision trees and averages them
normalize<-function(x){return((x-min(x))/(max(x)-min(x)))}
```

```
data$diagnosis<-as.numeric(data$diagnosis)
data_n<-as.data.frame(lapply(data,normalize))
traindata_n<--data_n[1:426,]
testdata_n<-data_n[427:569,]
rf <- randomForest(diagnosis ~., data= traindata_n, ntree =300, mtry = 5,
importance = TRUE)

## Warning in randomForest.default(m, y, ...): The response has five or fe
wer
## unique values. Are you sure you want to do regression?

print(rf)

##
## Call:
##  randomForest(formula = diagnosis ~ ., data = traindata_n, ntree = 300,
mtry = 5, importance = TRUE)
##                Type of random forest: regression
##                      Number of trees: 300
## No. of variables tried at each split: 5
##
##           Mean of squared residuals: 0.03693862
##                     % Var explained: 84.79

plot.new()

varImpPlot(rf, type = 1, pch =8, col = 2, cex =0.8, main = "cancerdata")
abline(v= 45, col= "red")
```

Mean Decrease Accuracy (%IncMSE) and Mean Decrease Gini (IncNodePurity) (sorted decreasingly from top to bottom) of attributes as assigned by the random forest.


cancerdata
perimeter_worst
area_worst
concave.points_worst
texture_worst
radius_worst
concave.points_mean
smoothness_worst
area_se
texture_mean
concavity_worst
concavity_mean
perimeter_se
area_mean
radius_mean
perimeter_mean
radius_se
compactness_worst
symmetry_worst
compactness_mean
smoothness_mean
concave.points_se
symmetry_se
compactness_se
concavity_se
fractal_dimension_worst
fractal_dimension_se
id
smoothness_se
symmetry_mean
fractal_dimension_mean


Variable Importance
B
M
MeanDecreaseAccuracy
MeanDecreaseGini
0
5
10
15
Relative Importance
Rattle 2018-Nov-01 14:59:17 tsraj

		MeanDecreaseAccuracyB	MeanDecreaseGiniM
area_worst	15.13 10.84	17.79	13.78
concave.points_worst	13.84 11.08	17.58	12.86
radius_worst	13.19 11.08	15.99	12.32
perimeter_worst	13.16 10.67	15.65	14.85
concave.points_mean	9.53 10.94	13.77	13.81
concavity_worst	7.32 9.27	11.99	3.33
texture_mean	8.28 9.79	11.95	2.1
texture_worst	8.63 10.24	11.74	2.3
area_se	8.40 7.98	11.33	5.83
smoothness_worst	6.42 8.05	10.23	1.57
perimeter_mean	8.58 5.62	9.6	7.04
radius_mean	8.55 5.14	9.37	4.99
area_mean	8.50 5.28	9.3	4.07
concavity_mean	5.31 6.54	9.03	3.9
perimeter_se	5.63 6.26	8.33	1.88
radius_se	5.66 4.59	7.6	1.23
smoothness_	4.07 6.30	7.34	0.92
compactness_mean	5.84 3.89	6.92	1.51
compactness_worst	4.29 4.11	6.37	1.44
compactness_se	4.34 2.83	5.35	0.59
concavity_se	3.20 3.77	5.33	0.76
smoothness_se	3.65 3.47	5.3	0.58
symmetry_worst		5.15	1.17
fractal_dimension_worst	4.31 2.39	5.05	1.06
texture_se	3.97 1.92	4.44	0.55
concave.points_se	3.70 2.72	4.39	0.51
symmetry_mean	0.22 3.69	3.03	0.45

fractal_dimension_mean	2.10 1.25	2.57	0.43
fractal_dimension_se	1.96 1.34	2.56	0.64
symmetry_se	0.96 0.48	1.03	0.55

No regression model technique is best for all situations.

MARS models are more flexible than linear regression models.

- MARS (like recursive partitioning) does *automatic variable selection* (meaning it includes important variables in the model and excludes unimportant ones).
  - MARS models tend to have a good bias-variance trade-off. The models are flexible enough to model non-linearity and variable interactions (thus MARS models have fairly low bias), yet the constrained form of MARS basis functions prevents too much flexibility (thus MARS models have fairly low variance).
- MARS models do not give as good fits as boosted trees but can be built much more quickly and are more interpretable. (An 'interpretable' model is in a form that makes it clear what the effect of each predictor is.)

The more the **accuracy** of the random forest **decreases** due to the exclusion (or permutation) of a single variable, the more important that variable is deemed, and therefore variables with a large **mean decrease** in **accuracy** are more important for classification of the data.

Mean Decrease Accuracy (%IncMSE) and Mean Decrease Gini (IncNodePurity) (sorted decreasingly from top to bottom) of attributes as assigned by the random forest.

MARS

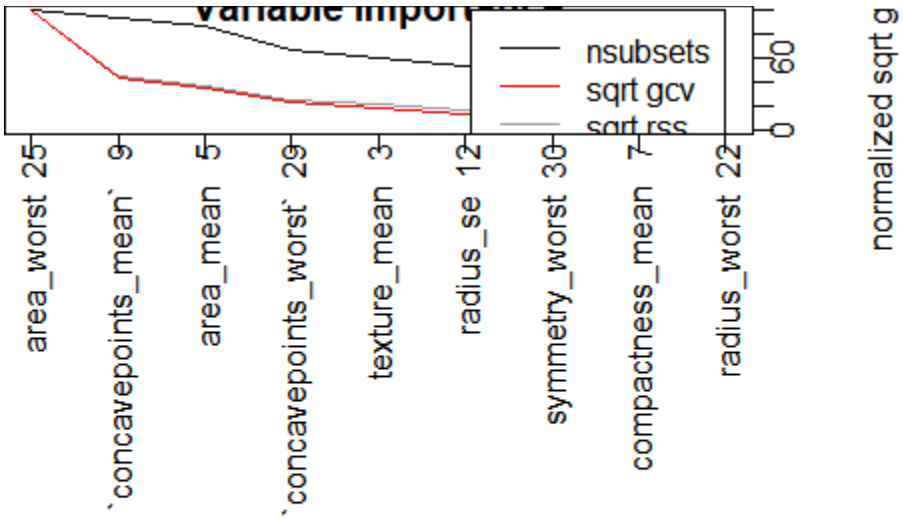
```
##                         nsubsets   gcv    rss
## area_worst                    15 100.0  100.0
## `concavepoints_mean`          14  43.1   44.5
## area_mean                     13  34.5   36.2
## `concavepoints_worst`         10  22.9   24.9
## texture_mean                   9  18.2   20.5
## radius_se                      8  13.3   16.2
## symmetry_worst                 7   9.6   13.0
## compactness_mean               6   7.6   11.1
## radius_worst                   2   1.5    5.1
```

Both the predictions(Mars&RF) on importance are comparable

```
#4. MARS (earth package)
#The earth package implements variable importance based on Generalized cro
ss validation (GCV),
#number of subset models the variable occurs (nsubsets) and residual sum o
f squares (RSS).
library(earth)

marsModel<-earth(diagnosis~ ., data=data) # build model
ev <- evimp (marsModel) # estimate variable importance
ev
```

```
##                         nsubsets   gcv    rss
## area_worst                    15 100.0  100.0
## `concavepoints_mean`          14  43.1   44.5
## area_mean                     13  34.5   36.2
## `concavepoints_worst`         10  22.9   24.9
## texture_mean                   9  18.2   20.5
## radius_se                      8  13.3   16.2
## symmetry_worst                 7   9.6   13.0
## compactness_mean               6   7.6   11.1
## radius_worst                   2   1.5    5.1
```

Plot(ev)



Conclusions

In this project, we applied various prediction models like Random Forest, Naive Bayes, SVM, Decision trees and Logistic Regression models for breast cancer survivability on two parameters: benign and malignant cancer patients. We acquired a dataset (569). We applied data selection, pre-processing, exploratory phase and transformation to develop the prediction models. In this project, we used a binary categorical survival variable, which was calculated from the variables in the dataset, to represent the survivability where malignant is represented with a value of ‘‘B’’ and benign is represented with ‘‘M’’. In order to measure the unbiased prediction accuracy of the various methods, we used a 10-fold cross-validation procedure, that is we divided the dataset into 10 mutually exclusive partitions. This provided us with a less biased prediction performance measures. The obtained results indicated that all the models performed a classification accuracy of >90%. Random Forest 100%, Logistic Regression 98.3%,SVM -Linear kernel vanilla – 99.3% and SVM Rbf-99.6%, Naive Bayes – 93.5% and Decision trees 95% accuracy.

IV-df plot provides information value summary like suspicious, very strong, strong and Average important variables and Var imp (step fit) identify around 9 variables as important.

Variable importance of Random forest identified in the decreasing order of importance, Relative importance through Mean Decrease Accuracy (%IncMSE) and Mean Decrease Gini (IncNodePurity) (sorted decreasingly from top to bottom) of attributes as assigned by the random forest. This important prediction and Mars(Earth package) variable important prediction and other models variable important predictions are comparable.

Hence, the variables like area_worst, concave points mean, area mean, concave points worst, texture mean etc., and which are identified on the first order by other classification models can be concluded as the factors driving the cancer identification.

The early diagnosis of BC can improve the prognosis and chance of survival significantly, as it can promote timely clinical treatment to patients. Further accurate classification of benign tumors can prevent patients undergoing unnecessary treatments. Thus, the correct diagnosis of BC and classification of patients into malignant or benign groups is the subject of much research

Further, the note about the risk of breast cancer is well explained in the introduction part under "Recommended Screening Guidelines" gender, age group and other significant symptoms etc. for Mammography to be followed to avoid the risk of Breast cancer. Early diagnosis and regular Mammography screening with the help of Machine learning through proper classification models can be predicted and guided properly for further testing and treatment to avoid

early death of the patients. Further the right identification through ML classification could avoid unnecessary treatment because of wrong identification.

Acknowledgement:-

This is a quite interesting project and I have gained a lot of knowledge about breast cancer and the identification of tumors through Machine Learning classification Model. I thank the institute Acadgild and the Mentors, Mr. Mohit & Mr. Gaurav, who taught us the R and related subjects to understand the Analytics.

Thank you Acadgild!

.