BREAST CANCER DETECTION AND DIAGNOSIS USING CLASSIFICATION TECHNIQUES

INTRODUCTION

Breast cancer is one of the most common cancers among women worldwide, representing the majority of new cancer cases and cancer related deaths according to global statistics, making it a significant public health problem in today's society.

Early diagnosis of cancer is critical for its successful treatment. Thus, there is a high demand for accurate and cheap diagnostic methods. In this project, we explored the applicability of decision tree machine learning techniques like CART, Random forest, Naïve Bayes, Logistic regression and K nearest neighbor models. The most accurate traditional method for diagnostic is a rather invasive technique, called breast biopsy, where a small piece of breast tissue is surgically removed, and then the tissue sample has to be examined by specialist. However, a much less invasive technique can be used, where the samples can be obtained by a minimally invasive fine needle aspirate method. The sample obtained by this method can be easily digitized and used for computationally based diagnostic. Using machine learning methods for diagnostic can significantly increase processing speed and on a big scale can make the diagnostic significantly cheaper.

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.

ABOUT THE DATASET

We will use the Breast cancer Wisconsin (Diagnostic) data set from Kaggle for the Breast cancer data

https://www.kaggle.com/lbronchal/breast-cancer-dataset-analysis/data

This data set can also be found on UCI Machine learning repository:

https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Diagnostic%29

To create the dataset, Dr. Wolberg used fluid samples, taken from patients with solid breast masses and an easy-to-easy graphical computer program called Xcyt, which is capable of performing the analysis of cytological features based on a digital scan. 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 for each feature of the image, returning a 30 real-valuated vector.

Attribute information

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

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

I. Radius (mean of distance from center to point s on the perimeter)
II. Texture (standard deviation of gray scale values)
III. Perimeter
IV. Area
V. Smoothness (local variation in radius length)
VI. Compactness (perimeter^2 or area)
VII. Concavity (severity of concave portions of the contour)

VIII. Concave points (number of concave portions of the contour)
IX. Symmetry
X. Fractal dimension (coastline approximation)

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, we have used machine learning classification methods to fit a function that can predict the discrete class of new input.

ANALYSIS AND INTERPRETATION

Before starting with the analysis, the required packages are called using the library function. The libraries like ggplot2, corrplot, caret reshape2 etc. is being loaded into the R environment.

On installing all the packages, we can now import the dataset R for further analysis.

Head function shows the first 5-6 entries of each columns in the dataset.

```
cancer<-read.csv("C:/Users/RENJITA/Downloads/data.csv")
head(cancer)
```

```
##       id diagnosis radius_mean texture_mean perimeter_mean area_mean
## 1 842302         M       17.99        10.38         122.80    1001.0
## 2 842517         M       20.57        17.77         132.90    1326.0
## 3 84300903       M       19.69        21.25         130.00    1203.0
## 4 84348301       M       11.42        20.38          77.58     386.1
## 5 84358402       M       20.29        14.34         135.10    1297.0
```

In order to see the structure of the dataset we can use the str () function available in R to visualize the type of data. We can observe the number of variables, number of observations, data type of the variables, factors involve etc.

Summary statistics of the dataset gives the basic description of the dataset i.e., min/max values of the variables,1st quartile,2nd quartile and Median or 2nd quartile of the dataset.

We can observe from the above summary statistics that there exists a column without any variables. Therefore, we don't have any use of it. So we are removing the last column from the dataset and also the first column 'as we don't have any use with the ID of the patients.

```
#Remove the first column
c_data <- cancer[,-c(0:1)]
#Remove the last column
c_data <- c_data[,-32]
```

Now in order to check for presence of any null values in the dataset, we can use the below snippet of code.

```
colSums(is.na(c_data))
```

```
##             diagnosis             radius_mean            texture_mean
##                     0                       0                       0
##          perimeter_mean               area_mean         smoothness_mean
##                     0                       0                       0
##        compactness_mean          concavity_mean      concave.points_mean
##                     0                       0                       0
##          symmetry_mean  fractal_dimension_mean                radius_se
##                     0                       0                       0
##             texture_se           perimeter_se                area_se
##                     0                       0                       0
##          smoothness_se         compactness_se           concavity_se
##                     0                       0                       0
##       concave.points_se            symmetry_se      fractal_dimension_se
##                     0                       0                       0
##           radius_worst          texture_worst         perimeter_worst
##                     0                       0                       0
##             area_worst        smoothness_worst       compactness_worst
##                     0                       0                       0
##         concavity_worst    concave.points_worst          symmetry_worst
##                     0                       0                       0
## fractal_dimension_worst
##                     0
```

Here we can observe that there exists no presence of null values in any of the columns. Therefore, we can proceed with the rest of data analysis as it is a cleaned data.

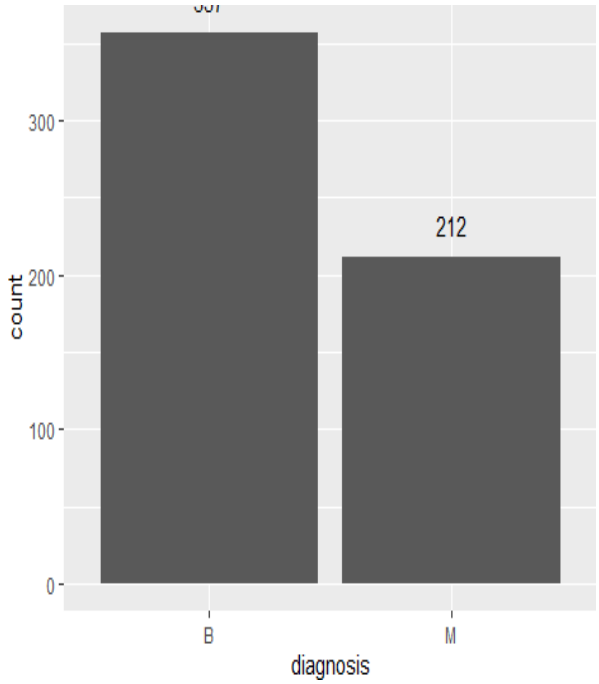
Now let us analysis the diagnosis column of the dataset which gives us the information about how many people are cancer free and how many are not.

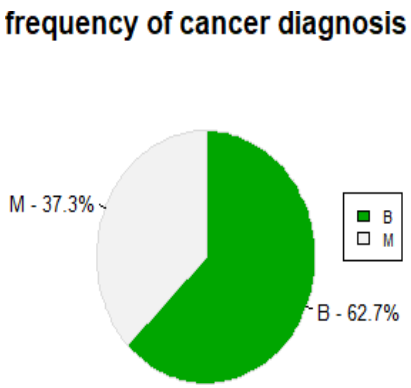
```
summary(cancer$diagnosis)
```

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

Here M= Malignant (indicates presence of cancer cells); B= Benign (indicates absence). The number of patients with Benign cancer cells are 357 and the number of patients with Malignant Cancer cells are 212. We can visualize the above calculation graphically using histograms and pie charts.

From the above histogram we can cee that the patients with Benign cancer cells are more in number than Malignant cancer cell patients.





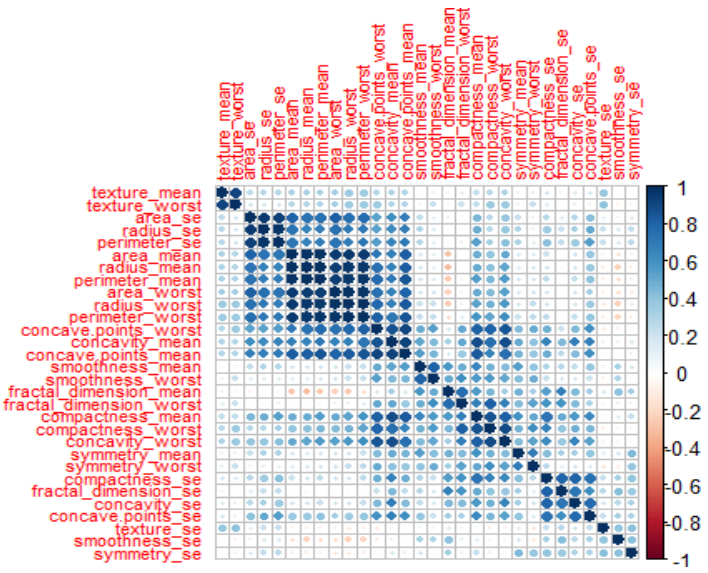
As we can see from the above pie chart, around 62.7% of the patients under study doesn't have any presence of cancer cells which is good and around 37.3% of the sample population has malignant cancer cells which is a cause of concern.

IMPROVING ACCURACY USING FEATURE SELECTION

Feature Selection is one of the core concepts in machine learning which hugely impacts the performance of your model. Any irrelevant or partially relevant features can negatively impact model performance. Feature Selection is the process where you automatically or manually select those features which contribute most to your prediction variable or output in which you are interested in. Mostly it is done using the concept of correlation between variables. It is to be noted that less misleading data improves the model accuracy and feature selection is best way to do that.

Now in our dataset we can implement feature selection by first creating a correlation matrix using heat map for all the variable.

```
corMatMy <- cor(c_data[,2:31])
corrplot(corMatMy, order = "hclust", tl.cex = 0.7)
```



From the above correlation matrix and correlation table we can observe that the variable "compactness_mean","concavity_mean","texture_worst","fractal_dimension_se","texture_mea","perimeter_worst", "diagnosis", "texture_se","perimeter_se" and "radius_mean" are the variables with correlation values more than 0.9. Often, we have features that are highly correlated and those provide redundant information. By eliminating highly correlated features we can avoid a predictive bias for the information contained in these features. This also shows us, that when we want to make statements about the biological/ medical importance of specific features, we need to keep in mind that just because they are suitable to predicting an outcome, they are not necessarily causal - they could simply be correlated with causal factors.

We are removing all features with a correlation higher than 0.9, keeping the feature with the lower mean. Now, we are shorter of 10 variables and we have 21 variables in total of analysis. Excluding these variables are useful in increasing the accuracy of models.

SPLITING THE DATSET

The simplest method to evaluate the performance of a machine learning algorithm is to use different training and testing datasets. The available data is split into a training set and a testing set. (70% training, 30% test)

```
set.seed(1234)

df <- cbind(diagnosis = c_data$diagnosis, c_data_cor)
```

```
train_indx <- createDataPartition(df$diagnosis, p = 0.7, list = FALSE)

train_set <- df[train_indx,]

test_set <- df[-train_indx,]

nrow(train_set)
```

399

```
nrow(test_set)
```

170

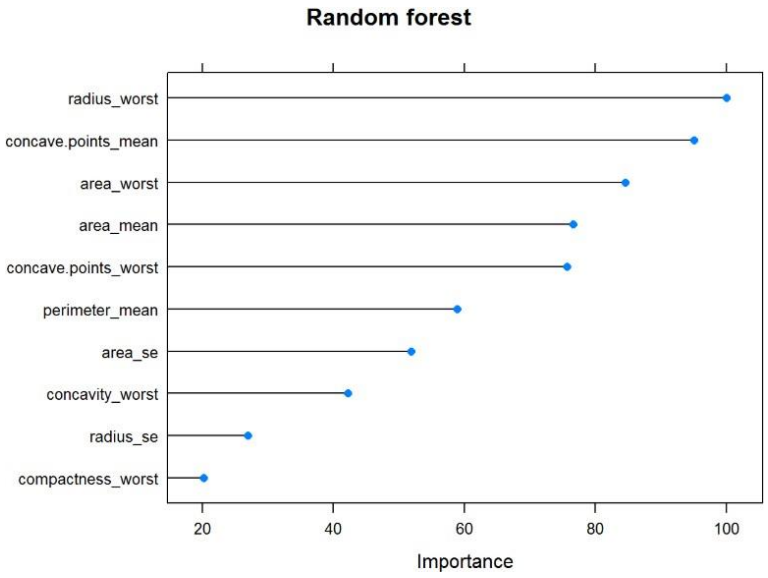
MODEL SELCTION.

- **RANDOM FOREST**

Random Forest is one such very powerful assembling machine learning algorithm which works by creating multiple decision trees and then combining the output generated by each of the decision trees.

```
fitControl <- trainControl(method="cv", number = 5,preProcOptions = list(thresh = 0.99),

 classProbs = TRUE, summaryFunction = twoClassSummary)

model_rf <- train (diagnosis~., train_set, method="ranger", metric="ROC", preProcess = c('center', 'scale'), trControl = fitControl)

pred_rf <- predict(model_rf, test_set)

cm_rf <- confusionMatrix(pred_rf, test_set$diagnosis, positive = "M")

cm_rf
```

OUTPUT:



We observe that radius_worst, concave.points_mean, area_worst, area_mean, concave.points_worst, perimeter_mean, area_se and concavity_worst are the most important features. Based on these values we are building the model below.

```
Confusion Matrix and Statistics

          Reference
Prediction B  M
         B 106   5
         M   1  58

               Accuracy : 0.9647
                 95% CI : (0.9248, 0.9869)
    No Information Rate : 0.6294
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.9233

 Mcnemar's Test P-Value : 0.2207

            Sensitivity : 0.9206
            Specificity : 0.9907
         Pos Pred Value : 0.9831
         Neg Pred Value : 0.9550
             Prevalence : 0.3706
         Detection Rate : 0.3412
   Detection Prevalence : 0.3471
      Balanced Accuracy : 0.9556

       'Positive' Class : M
```

Here accuracy is 96% and the p value (2e-16)is given to be less than 0.05. Therefore, the model Random forest is an apt model.

- **NAÏVE BAYES**

Naive Bayes classifier is a simple classifier that has its foundation on the well-known Bayes's theorem.

```
library(klaR)

model_nb <- train(diagnosis~.train_set,method="nb",metric="ROC",preProcess=c('center',
'scale'),trace=FALSE,trControl=fitControl)

pred_nb <- predict(model_nb, test_set)

cm_nb <- confusionMatrix(pred_nb, test_set$diagnosis, positive = "M")

cm_nb
```

OUTPUT:

```
Confusion Matrix and Statistics

          Reference
Prediction B  M
         B 103  10
         M   4  53
```

```
               Accuracy : 0.9176
                 95% CI : (0.8657, 0.9542)
    No Information Rate : 0.6294
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.8199

 Mcnemar's Test P-Value : 0.1814

            Sensitivity : 0.8413
            Specificity : 0.9626
         Pos Pred Value : 0.9298
         Neg Pred Value : 0.9115
             Prevalence : 0.3706
         Detection Rate : 0.3118
   Detection Prevalence : 0.3353
      Balanced Accuracy : 0.9019

       'Positive' Class : M
```

Here accuracy is 91% and the p value (2e-16)is given to be less than 0.05. Therefore, the model Naïve Bayes is an apt model.

- **CART MODEL**

The decision tree method is a powerful and popular predictive machine learning technique that is used for both classification *and* regression. So, it is also known as Classification and Regression Trees **(**CART**).**

```
library("rpart")

library(rattle)

set.seed(1)

cart_model <- train(diagnosis~., train_set, method = "rpart")

cart_model
```

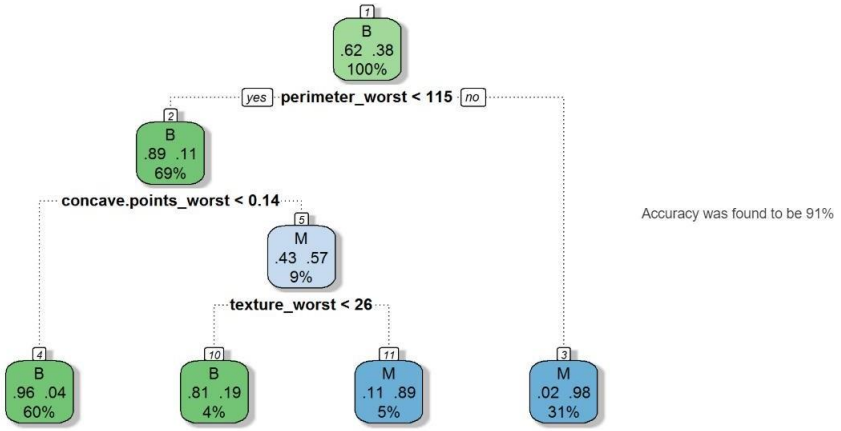
OUTPUT:

```
CART

399 samples
 21 predictor
  2 classes: 'B', 'M'

No pre-processing
Resampling: Bootstrapped (25 reps)
Summary of sample sizes: 399, 399, 399, 399, 399, 399, ...
Resampling results across tuning parameters:

  cp          Accuracy  Kappa
```

```
 0.01677852  0.9192599  0.8262348
 0.05369128  0.9091931  0.8046549
 0.79194631  0.8707228  0.6935816

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was cp = 0.01677852.
```

Here accuracy is 92%. Therefore, the model CART is successful in detecting the breast cancer in patients.

```
fancyRpartPlot(cart_model$finalModel, sub="")
```



It can be observed from the above graph that for regression trees the feature that is selected is perimeter_worst which is taken for further dividing. It shows that more than other features perimeter_worst,concave_points_worst and texture_worst plays an important role in detecting breast cancer in patients. And it is seen that the accuracy of the model is 91% which is pretty good for breast cancer detection.

- **KNN**

  K nearest neighbours is a simple algorithm that stores all available cases and classifies new cases by a majority vote of its k neighbours. This algorithm segregates unlabelled data points into well-defined groups.

```
model_knn <- train(diagnosis~.,data = train_set, method="knn", metric="ROC",
preProcess = c('center', 'scale'), tuneLength=10, trControl=fitControl)

pred_knn <- predict(model_knn, test_set)

cm_knn <- confusionMatrix(pred_knn, test_set$diagnosis, positive = "M")

cm_knn
```

```
Confusion Matrix and Statistics

          Reference
Prediction   B   M
         B 107   9
         M   0  54

               Accuracy : 0.9471
                 95% CI : (0.9019, 0.9755)
    No Information Rate : 0.6294
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.8831

 Mcnemar's Test P-Value : 0.007661

            Sensitivity : 0.8571
            Specificity : 1.0000
         Pos Pred Value : 1.0000
         Neg Pred Value : 0.9224
             Prevalence : 0.3706
         Detection Rate : 0.3176
   Detection Prevalence : 0.3176
      Balanced Accuracy : 0.9286

       'Positive' Class : M
```

Here accuracy is 94% and the p value (2.2e-16) is given to be less than 0.05. Therefore, the model KNN is an apt model.

- **LOGISTIC REGRESSION**

Logistic regression is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables.

```
breast_cancer_glm <- glm(diagnosis ~ area_mean, data = train_set, family = "binomial")

summary(breast_cancer_glm)
```

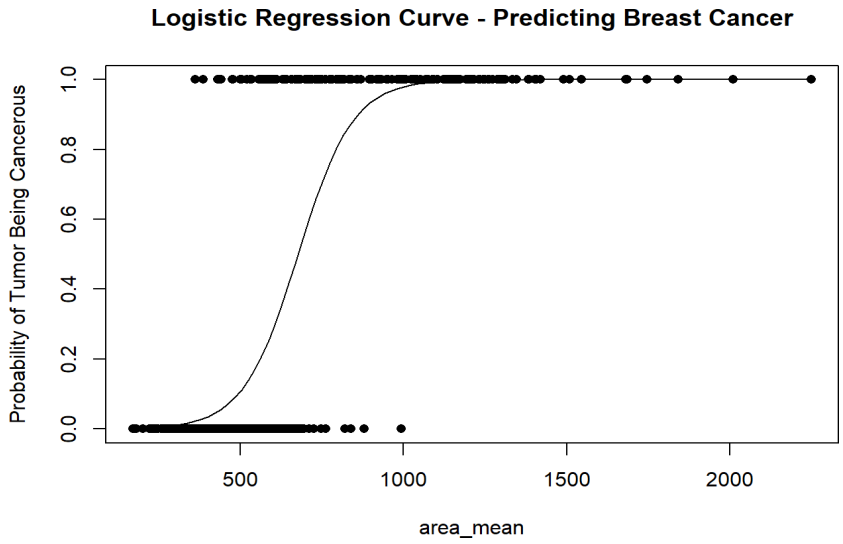
OUTPUT:

```
Call:
glm(formula = diagnosis ~ area_mean, family = "binomial", data = train_set)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.6122  -0.4813  -0.2293   0.1070   2.6114

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -7.249898   0.736314  -9.846  <2e-16 ***
area_mean    0.010713   0.001188   9.021  <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 527.28  on 398  degrees of freedom
Residual deviance: 244.48  on 397  degrees of freedom
AIC: 248.48
```

```
Number of Fisher Scoring iterations: 7
```

```
curve(
  exp(breast_cancer_glm$coef[1]+breast_cancer_glm$coef[2]*x)/
    (1+exp(breast_cancer_glm$coef[1]+breast_cancer_glm$coef[2]*x)), add = TRUE)
```

OUTPUT:




```
#Making predictions
train$pred_prob <- predict(breast_cancer_glm, data=train, type='response')
train$pred <- ifelse(train$pred_prob >= .5, 1, 0)
#Testing our predictions against our training set
results <- train %>%
   mutate(correct = (diagnosis_0_1 == pred)) %>%
   group_by(correct) %>%
   tally()
accuracy <- round(results$n[2]/(nrow(train)),3) #Calculating accuracy
print(paste0("We made ", results$n[2], " correct predictions,"))
```

"We made 406 correct predictions,"

```
print(paste0(results$n[1], " incorrect predictions, "))
```

"50 incorrect predictions, "

```
print(paste0("thus giving us an accuracy rating of: ", accuracy*100, "%"))
```

"Thus giving us an accuracy rating of: 89%"

Here the p-value is less that0.05 so the model is a perfect fit.

INTERPRETATION

Classification techniques have shown recently their usefulness for complex process diagnosis. Various classification techniques have been used for classifying masses as malignant or benign based on the features.

The models considered are given below:

- Random forest: 96%
- CART model: 91%
- Logistic regression: 89%
- KNN: 94%
- Naïve Bayes: 92%

The best results for sensitivity (detection of breast cases) is Random Forest which is giving 96%of accuracy. The order of models based on the accuracy is: Random Forest, KNN, Naïve Bayes, CART and Logistic Regression.

CONCLUSION

The automatic diagnosis of Breast cancer is an important real-world medical problem. Detection of breast cancer in its early stages is the key for treatment. This paper shows how different models can be used to predict actual diagnosis of Breast cancer for local and systematic treatment. Experimental results show the effectiveness of the proposed model. The performance of classification and regression techniques were investigated for the Breast cancer diagnosis problem.

REFERENCES

[1] [Online]. Available: https://www.kaggle.com/buddhiniw/breast-cancer-prediction/data.

[2] " VikasChaurasia, BB Tiwari and Saurabh Pal – "Prediction of benign and malignant breast cancer using data mining".

[3] *D.Dubey ,S.Kharya, S.Soni and –"Predictive Machine Learning techniques for Breast Cancer Detection", International Journal of Computer Science and Information Technologies,Vol.4(6),2013,1023-1028 .*

[4] *Chao-Ying ,Joanne, PengKukLida Lee, Gary M. Ingersoll –"An Introduction to Logistic Regression Analy.*

[5] [Online]. Available: https://www.kaggle.com/buddhiniw/breast-cancer-prediction/data.

[7] [Online]. Available: https://towardsdatascience.com/building-a-simple-machine-learning-model-on-breast-cancer-data-eca4b3b99fa3.