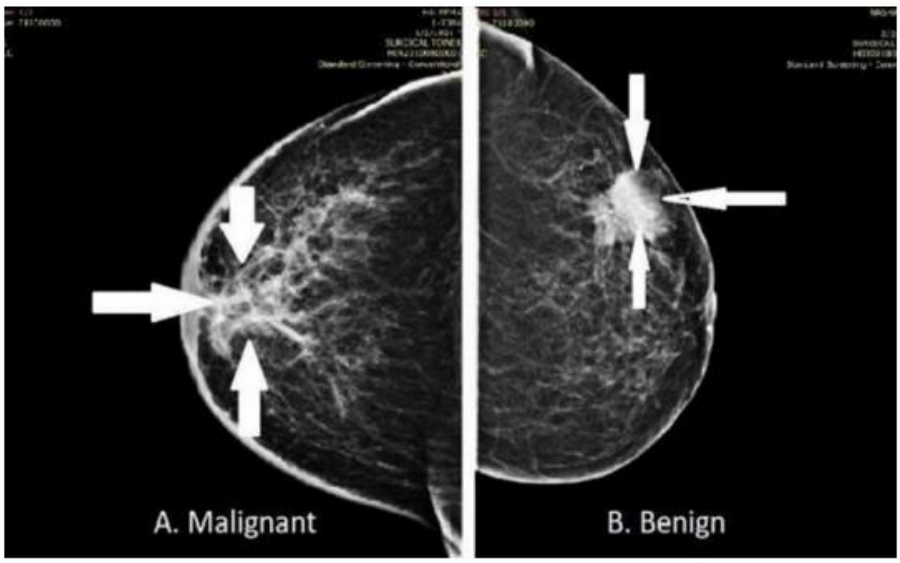
Final Project Submission

MACHINE LEARNING APPROACHES – TO CLASSIFY BREAST CANCER (TO PREDICT BREAST CANCER)

Figure 1. Images of mammography for breast cancer types [4]



Fitchburg State University
SP23_Intro to Data Science-52
Instructor: Prof Ricky Sethi
Student Name: Yasaswini Nallapula
Student ID:( )

Table of Contents

1. ABSTRACT

Breast cancer remains one of the most common diseases, killing thousands of women each year. Accurate diagnosis of breast tumors can be done early and quickly by using Artificial Intelligence. The aim of this project is to review recent research on the classification of these tumors. To classify medical images into malignant and benign, machine learning algorithms such as support vector machines (SVM), Logistic Regression, decision trees, and Gradient Boosting Machines will be used. As a result, we found that Gradient Boosting Machine achieves a high accuracy of about 99-100% Therefore, researchers have explored and used various features of this algorithm and added features such as bagging and boosting to improve its effectiveness.

Breast cancer is undoubtedly a serious disease if it is not recognized and treated for a long period of time. It is one of the most common cancers among women worldwide, accounting for many new cancer cases and cancer-related deaths as per global statistics, making it a serious public health problem in present society.

It is caused by uncontrolled proliferation of some cells within the mammary gland that transform into malignant cells. In this way, they could detach from the tissue that created them and invade surrounding tissues and eventually organs on the opposite side of the body. Cancers can arise from any type of breast tissue, but most commonly arise from the glandular cells or those that line the walls of the milk ducts. The goal in this context is to discover each of many benign or malignant classes.

2. INTRODUCTION

Many developed and non-developed countries around the world suffer from deadly cancer-related diseases. In particular, the incidence of breast cancer in women is increasing day by day, partly due to ignorance and undiagnosed in its early stages. Adequate first-line treatment of breast cancer can only be achieved through proper detection of the cancer very early in its development.

The aim is to discover each of numerous benign or malignant classes. To do this, we can use records retrieved from the UCI repository of machine learning databases and such DNA samples arrive at this repository on a regular basis. So, the database displays this chronological grouping of records. This dataset was created by Dr. William H. Wolberg, physician at the University of Wisconsin Hospital at Madison, Wisconsin, USA. For this he used fluid samples taken from patients with solid breast masses and a computer program called Xcyt, which computes ten features from each one of the cells in the sample.Each variable, besides the first, changed into transformed into 32 primitive numerical attributes with real-valued input features. There are no missing values. The data frames comprise 569 observations on 32 variables—1 being a character variable, and 31 are ordinal variables.

Early diagnosis of breast cancer (BC) can facilitate timely clinical management of patients, thus significantly improving prognosis and survival chances. Further accurate classification of benign tumors can prevent patients undergoing unnecessary treatments. Therefore, the correct diagnosis of BC and the classification of patients into malignant or benign groups has been the subject of many studies.

Risk factors for BC include age, personal history, family history, genetic factors, childbearing, and menstrual history. I would like to build and test some machine learning models to classify breast cancer based on available sample data. Machine learning (ML) is widely recognized as the method of choice for classification and predictive modeling of BC patterns due to its unique advantages in detecting important features from complex BC datasets. Our goal is to create a classification model that can classify biopsy data points as benign (noncancerous) or malignant (cancerous). Classification and data mining techniques are effective ways to classify data. Especially in the medical field, these methods are widely used for diagnosis and analysis and decision making.

3. RELATED WORK

1. Breast Cancer Classification Using Machine Learning Techniques: A Review [September 2021] Turkish Journal of Computer and Mathematics Education (TURCOMAT) 12(14):1970-1979 Authors: Srwa Hasan, Ali M Sagheer, Hadi Veisi
   https://www.researchgate.net/publication/356844442_Breast_Cancer_Classification_Using_Machine_Learning_Techniques_A_Review
   It is one of the research papers used for reference where I got to know different approached and possibilities while developing ML models

2. Siyabend Turgut et al., "Microarray Breast Cancer Data Classification Using Machine Learning Methods" [IEEE 2018]
   https://ieeexplore.ieee.org/abstract/document/8391468
   The paper uses microarray breast cancer data for classification of the patients using machine learning methods

4. RESEARCH QUESTION

Question is: How can we classify/predict breast cancer from the dataset by using Machine Learning approaches and the target variable is diagnosis attribute that classifies the patient record as benign or malignant ((by referring biopsy data points like cell size etc., for prediction).

Each instance from the dataset represents one patient record and it comprises of patient's cancer condition whether it is benign or malignant. A total of 569 cases in the dataset with 30 attributes (Excluding ID Number) represents independent variables and one attribute, i.e., diagnosis represents the output or dependent variable. This dependent variable can take only two values (two categories) and this ML model will try to predict that an observation with a particular characteristics will fall into a specific one of the categories either 'M' or 'B'.

Our goal is to build a classification model that can classify biopsy data points as either Benign (non-cancerous) or Malignant (cancerous). I will measure the performance of each model, compare accuracy rates, and find the best model among them. The platform used for analysis was R Studio.

Below are the classification metrics:

Decision trees: CART (Classification and regression tree) MODEL - confusion matrix and statistics. Logistic Regression (Binary Classification): confusion matrix and statistics. Support Vector machines (SVM): confusion matrix and statistics. The Confusion Matrix for Predictive Analysis is a 2-by-2 table showing the percentage of false positive, false negative, true positive, and true negative results for a test or predictor. We can make a confusion matrix if we know both the predicted values and the true values for a sample set.

Classification and data mining techniques are effective ways to classify data. Especially in the medical field, these methods are often used for diagnosis and analysis, and decision making.
I am planning to run the gradient boosting model through a grid search to find the optimized combination of hyperparameters. And planning to evaluate all four classification models by comparing their area under the curve (AUC/ROC) when fitted into the training, test, and the entire datasets.

5. DATA APPRAISAL

The data has been split into two groups: training set and test set. The training set should be used to build machine learning models. For the training set, we provide the outcome also known as the "ground truth" or "target variable" which is "diagnosis" in our case for each Id Number. Our model will be based on "features" like radius_mean, area_mean, of cell size etc.
Dataset Source: https://archive.ics.uci.edu/ml/datasets/breast+cancer+wisconsin+(diagnostic)

Description of Attributes:
ID number
Diagnosis (M = malignant, B = benign)
Features that are computed for each cell nucleus:
->radius (mean of distances from points on the perimeter to center)
->texture (gray-scale values)
->area
->perimeter
->compactness (perimeter^2 / area - 1.0)
->smoothness (local variation in radius lengths)
->concave points (number of concave portions of the contour)
->concavity (severity of concave portions of the contour)
->fractal dimension (approximation to - 1)
->symmetry

Target feature: Diagnosis (M = malignant, B = benign)
The WDBC dataset contains 569 instances and 32 attributes in which 357 were benign and 212 were malignant cases. In the WDBC data, missing attribute values are none. In the dataset ID number describes the patient record and diagnosis as the target/dependent variable and remaining 30 attributes are independent variables.

6. EXPLORATORY DATA ANALYSIS (EDA)

LIBRARIES—

```
install.packages("RCurl")
install.packages("GGally")
install.packages("xgboost")

library(RCurl)
library(dplyr)
library(caret)
library(PerformanceAnalytics)
library(gridExtra)
library(ggplot2)
library(GGally)
library(xgboost)
library(pROC)
```

Read the Data—

```
# UCI_data_URL2 <- getURL('https://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/wdbc.data')
UCI_data_URL <- "https://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/wdbc.data"
BC_wiscosin <- read.csv(UCI_data_URL, header = FALSE)
str(BC_wiscosin)
colnames(BC_wiscosin) <- c('id_number', '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', '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')
```

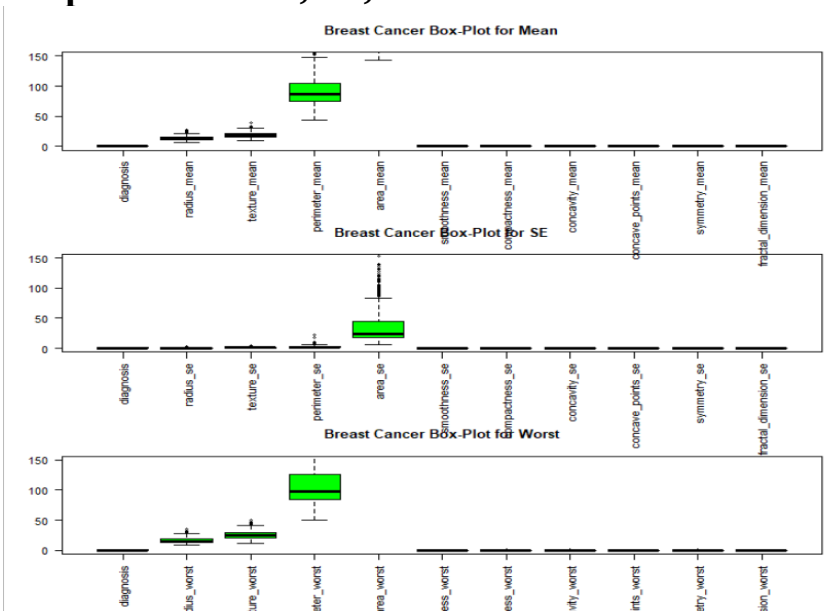
Structure of the Data—

```
R 4.2.3 · D:/Aang/YASHU_FSU HUB/Yashu/SEM2_Spring23/Intro Data Science/Module 5/Milestone/
> str(BC_wiscosin)
'data.frame':   569 obs. of  32 variables:
 $ id_number              : int  842302 842517 84300903 84348301 84358402 843786 844359 84458202 844981 84501001 ...
 $ diagnosis              : chr  "M" "M" "M" "M" ...
 $ radius_mean            : num  18 20.6 19.7 11.4 20.3 ...
 $ texture_mean           : num  10.4 17.8 21.2 20.4 14.3 ...
 $ perimeter_mean         : num  122.8 132.9 130 77.6 135.1 ...
 $ area_mean              : num  1001 1326 1203 386 1297 ...
 $ smoothness_mean        : num  0.1184 0.0847 0.1096 0.1425 0.1003 ...
 $ compactness_mean       : num  0.2776 0.0786 0.1599 0.2839 0.1328 ...
 $ concavity_mean         : num  0.3001 0.0869 0.1974 0.2414 0.198 ...
 $ concave_points_mean    : num  0.1471 0.0702 0.1279 0.1052 0.1043 ...
 $ symmetry_mean          : num  0.242 0.181 0.207 0.26 0.181 ...
 $ fractal_dimension_mean : num  0.0787 0.0567 0.06 0.0974 0.0588 ...
 $ radius_se              : num  1.095 0.543 0.746 0.496 0.757 ...
 $ texture_se             : num  0.905 0.734 0.787 1.156 0.781 ...
 $ perimeter_se           : num  8.59 3.4 4.58 3.44 5.44 ...
 $ area_se                : num  153.4 74.1 94 27.2 94.4 ...
 $ smoothness_se          : num  0.0064 0.00522 0.00615 0.00911 0.01149 ...
 $ compactness_se         : num  0.049 0.0131 0.0401 0.0746 0.0246 ...
 $ concavity_se           : num  0.0537 0.0186 0.0383 0.0566 0.0569 ...
 $ concave_points_se      : num  0.0159 0.0134 0.0206 0.0187 0.0188 ...
 $ symmetry_se            : num  0.03 0.0139 0.0225 0.0596 0.0176 ...
 $ fractal_dimension_se   : num  0.00619 0.00353 0.00457 0.00921 0.00511 ...
 $ radius_worst           : num  25.4 25 23.6 14.9 22.5 ...
 $ texture_worst          : num  17.3 23.4 25.5 26.5 16.7 ...
 $ perimeter_worst        : num  184.6 158.8 152.5 98.9 152.2 ...
 $ area_worst             : num  2019 1956 1709 568 1575 ...
 $ smoothness_worst       : num  0.162 0.124 0.144 0.21 0.137 ...
 $ compactness_worst      : num  0.666 0.187 0.424 0.866 0.205 ...
 $ concavity_worst        : num  0.712 0.242 0.45 0.687 0.4 ...
 $ concave_points_worst   : num  0.265 0.186 0.243 0.258 0.163 ...
 $ symmetry_worst         : num  0.46 0.275 0.361 0.664 0.236 ...
 $ fractal_dimension_worst: num  0.1189 0.089 0.0876 0.173 0.0768 ...
> dim(BC_wiscosin)
[1] 569  32
>
```

Adjust data set for analysis. Remove ID column---

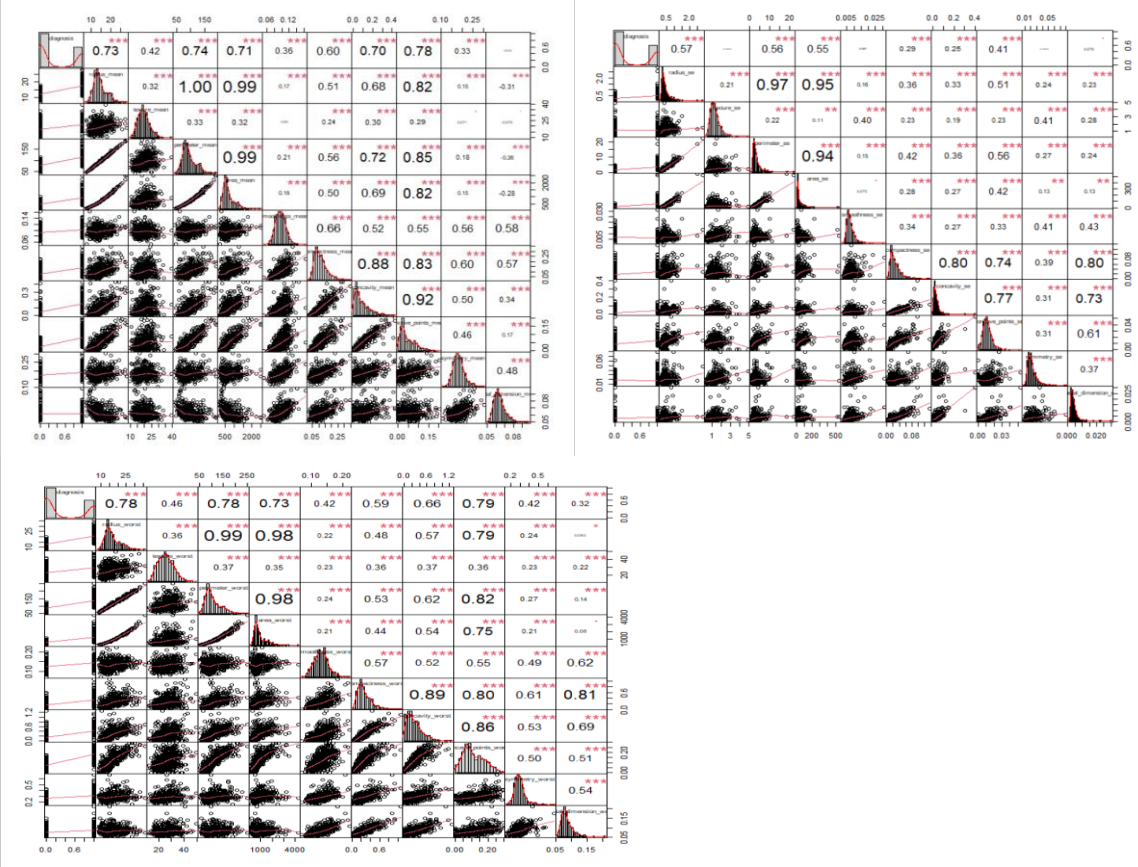
```
Console  Terminal  Jobs
R 4.2.3 · D:/Aang/YASHU_FSU HUB/Yashu/SEM2_Spring23/Intro Data Science/Module 5/Milestone/
> BC_wiscosin <- subset(BC_wiscosin, select=-c(id_number)) # removes the id attribute from dataset
> BC_wiscosin_eda <- BC_wiscosin
> BC_wiscosin_eda$diagnosis<-ifelse(BC_wiscosin_eda$diagnosis == "B", 0, 1)
>
```

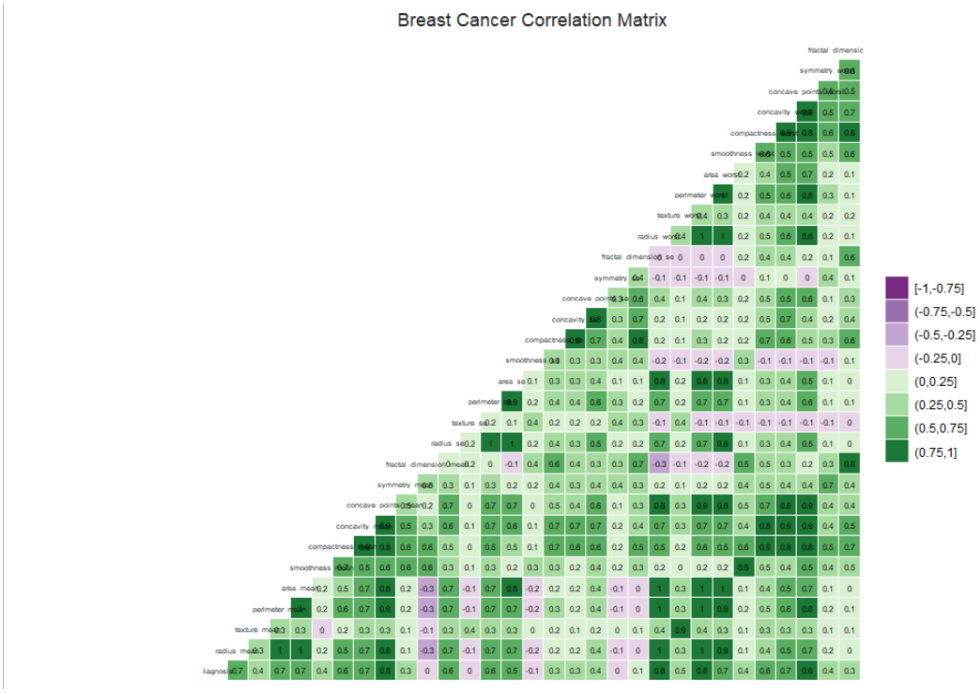
Boxplot for Means, SE, and Worst records of dataset:



From the plot, Nuclei mean of the perimeter and area is higher. Standard Error of the Area is higher. In the worst nuclei scenario, area has extremely high values.

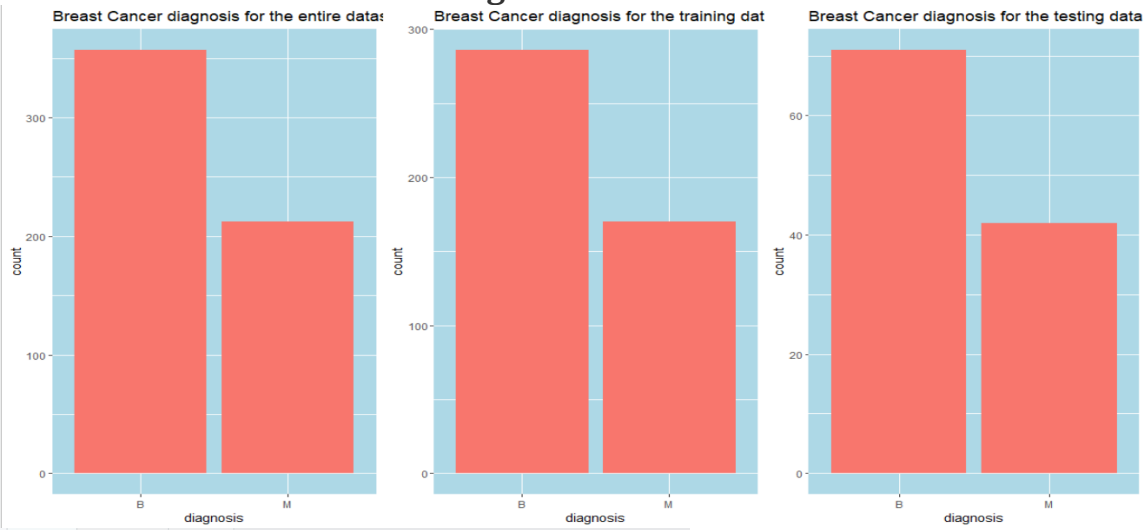
Create Correlation Charts for Means, SE, Worst and for Entire dataset:





The correlation values range from -1 to 1. When the correlation between variables is near to 1 or greater than or equal to 0.75, then the variables are positively highly correlated, which is represented by dark green color. For example, diagnosis is highly related to concavity_means, fractional dimension_se, texture_worst and concavity_worst.

Bar Plots of Breast Cancer diagnosis for the dataset:

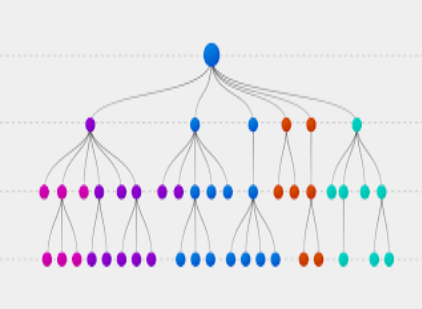


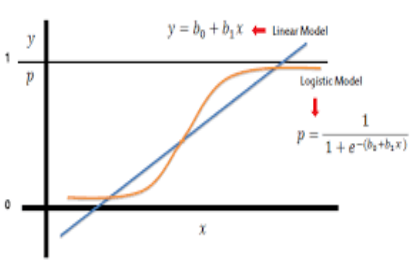

```
R 4.2.3 · D:/Aang/YASHU_FSU HUB/Yashu/SEM2_Spring23/Intro Data Science/Module 5/Milestone/
> #data analysis
> table(BC_wiscosin$diagnosis)

  B   M
357 212
> round(prop.table(table(BC_wiscosin$diagnosis)) * 100, digits = 1)

   B    M
62.7 37.3
>
```

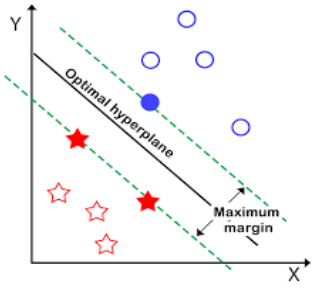
In the entire dataset, 37.3% of observations were positively diagnosed with breast cancer and those are 212 in number.

7. TECHNIQUES/METHODS

We train and test our data using Decision trees - CART (Classification and regression tree) MODEL, Logistic Regression (Binary Classification) and Support Vector machines (SVM) models. Let us look at the definition of four machine learning models we intend to use to solve this problem. These are three machine learning methods that I am planning to apply to classification models:







- Decision trees: A decision tree is a type of supervised learning algorithm that repeatedly splits a sample based on specific questions about the sample. These are very useful for classification problems. They are relatively easy to understand and very effective. A decision tree represents multiple decisions followed by different probabilities of occurrence. This technique helps define the most important variables and the relationships between two or more variables. Decision tree regression observes object features, trains a model on the structure of the tree to predict future dates, and produces meaningful and continuous output. Continuous output means that the output/result is not discrete, i.e., it is not represented just by a discrete, known set of numbers or values.

- Logistic Regression: This type of statistical model (also called logit model) is commonly used for classification and predictive analysis. Logistic regression estimates the probability of an event occurring, such as voted or didn't vote based on a given set of independent variables. The dependent variable is constrained between 0 and 1 because the outcome is a probability. Logistic regression applies a logit transformation to the odds, i.e., probability of success divided by probability of failure. It is also called log odds or natural logarithm of odds.

- Support vector Machines (SVM): SVM is supervised machine learning algorithm used for both classification and regression. It is best suited for classification. The goal of the SVM algorithm is to find a hyperplane in the N-dimensional space that uniquely classifies the data points. The dimension of the hyperplane depends on the number of features. If the number of input features is 2, the hyperplane is just a line. If the number of input features is 3, the hyperplane will be a 2D plane. It becomes difficult to imagine when the number of features exceeds three.

Performance of each model and compare the accuracy rate to find the best model among them. The platform used for analysis is R studio.

A confusion matrix, for Predictive Analysis is a 2-by-2 table showing the percentage of false positive, false negative, true positive, and true negative results for a test or predictor. We can make a confusion matrix if we know both the predicted values and the true values for a sample set. In

machine learning and statistical classification, a confusion matrix is a table in which predictions are represented in columns and actual status is represented by rows.

		Predicted	
		Negative	Positive
Actual	Negative	a	b
	Positive	c	d

8. EVALUATION

```
R 4.2.3 · D:/Aang/YASHU_FSU HUB/Yashu/SEM2_Spring23/Intro Data Science/Module 5/Milestone/
> # set.seed(1123)
> ## splitting the data
> set.seed(1123)
> trainIndex <- createDataPartition(BC_wiscosin$diagnosis, p = .8, list = FALSE, times = 1)
> training_set <- BC_wiscosin[ trainIndex, ]
> test_set <- BC_wiscosin[ -trainIndex, ]
> ####techniques
>
> fitControl <- trainControl(## 10-fold CV
+   method = "repeatedcv",number = 3, repeats = 10) ## repeated ten times
>
```

Decision trees - CART (Classification and regression tree) MODEL

```
#decision trees

dtree_model <- train(as.factor(diagnosis) ~ .,
                     data = training_set,
                     method = "rpart",
                     metric = "Accuracy",
                     trControl = fitControl)

feature_importance <- varImp(dtree_model, scale = FALSE)
feature_importance_scores <- data.frame(feature_importance$importance)

feature_importance_scores <- data.frame(names = row.names(feature_importance_scores),
                                        var_imp_scores = feature_importance_scores$Overall)

ggplot(feature_importance_scores,
       aes(reorder(names, var_imp_scores), var_imp_scores)) +
  geom_bar(stat='identity',
           fill = '#875FDB') +
  theme(panel.background = element_rect(fill = '#fafafa')) +
  coord_flip() +
  labs(x = 'Feature', y = 'Importance') +
  ggtitle('Importance of specific feature for decision trees')

predict_values <- predict(dtree_model,newdata = test_set)
confusionMatrix(as.factor(test_set$diagnosis),predict_values)

predict_values <- predict(dtree_model, newdata = training_set)
confusionMatrix(as.factor(training_set$diagnosis),predict_values)
```

Logistic Regression (Binary Classification):

```r
#logistic regression

LR_model <- train(diagnosis ~ .,
                  data = training_set,
                  method = "glmnet",
                  metric = "Accuracy",
                  family="binomial",
                  trControl = fitControl)

feature_importance1 <- varImp(LR_model, scale = FALSE)
feature_importance_scores1 <- data.frame(feature_importance1$importance)

feature_importance_scores1 <- data.frame(names = row.names(feature_importance_scores1),
                                         var_imp_scores1 = feature_importance_scores1$Overall)

ggplot(feature_importance_scores1,
       aes(reorder(names, var_imp_scores1), var_imp_scores1)) +
  geom_bar(stat='identity',
           fill = '#875FDB') +
  theme(panel.background = element_rect(fill = '#fafafa')) +
  coord_flip() +
  labs(x = 'Feature', y = 'Importance') +
  ggtitle('Importance of specific feature for logistic regression')

predict_values <- predict(LR_model,newdata = test_set)
confusionMatrix(as.factor(test_set$diagnosis),predict_values)
predict_values <- predict(LR_model, newdata = training_set)
confusionMatrix(as.factor(training_set$diagnosis),predict_values)
```

Support Vector machines (SVM):

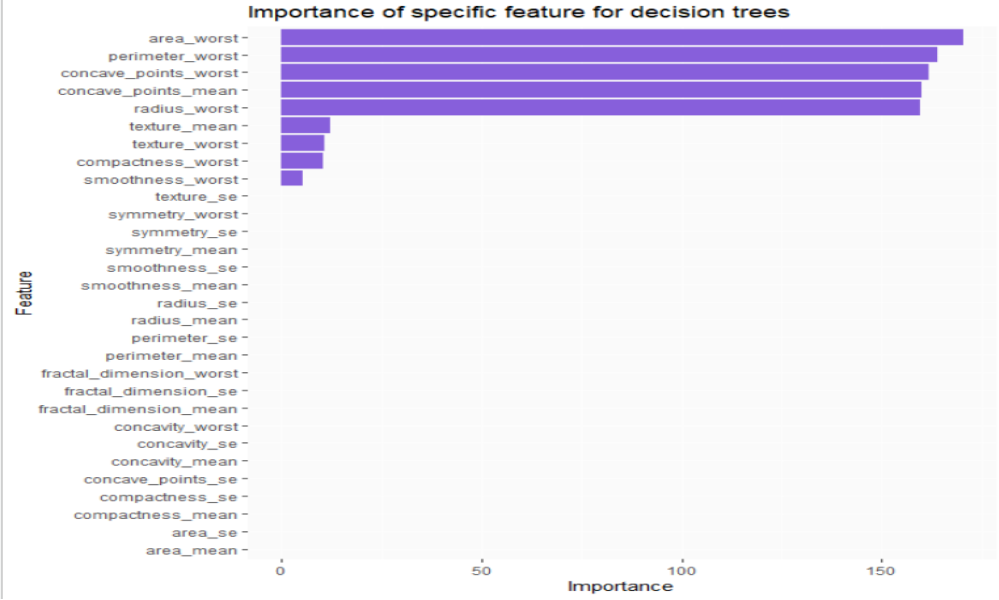
```r
#Support Vector Machine
training_set_svm <- training_set
training_set_svm$diagnosis <- as.factor(training_set_svm$diagnosis)
char_columns <- sapply(training_set_svm, is.character)
training_set_svm[ , char_columns] <- as.data.frame(apply(training_set_svm[ , char_columns], 2, as.numeric))


svm_model <- train(diagnosis ~ .,
                   data = training_set_svm,
                   method = "svmLinear",
                   metric = "Accuracy",
                   trControl = fitControl)

feature_importance2 <- varImp(svm_model, scale = FALSE)

# plot(feature_importance2)
ggplot(feature_importance2,
       aes(reorder(names, Importance), Importance)) +
  geom_bar(stat='identity',
           fill = '#875FDB') +
  theme(panel.background = element_rect(fill = '#fafafa')) +
  coord_flip() +
  labs(x = 'Feature', y = 'Importance') +
  ggtitle('Feature Importance for support vector machines')

predict_values <- predict(svm_model, newdata = test_set)
confusionMatrix(as.factor(test_set$diagnosis),predict_values)
predict_values <- predict(svm_model, newdata = training_set_svm)
confusionMatrix(as.factor(training_set_svm$diagnosis),predict_values)
```

Below are the preliminary observations which we have made from the data visualization done as part of the Data Understanding process. Most of the models will be given below accuracy.

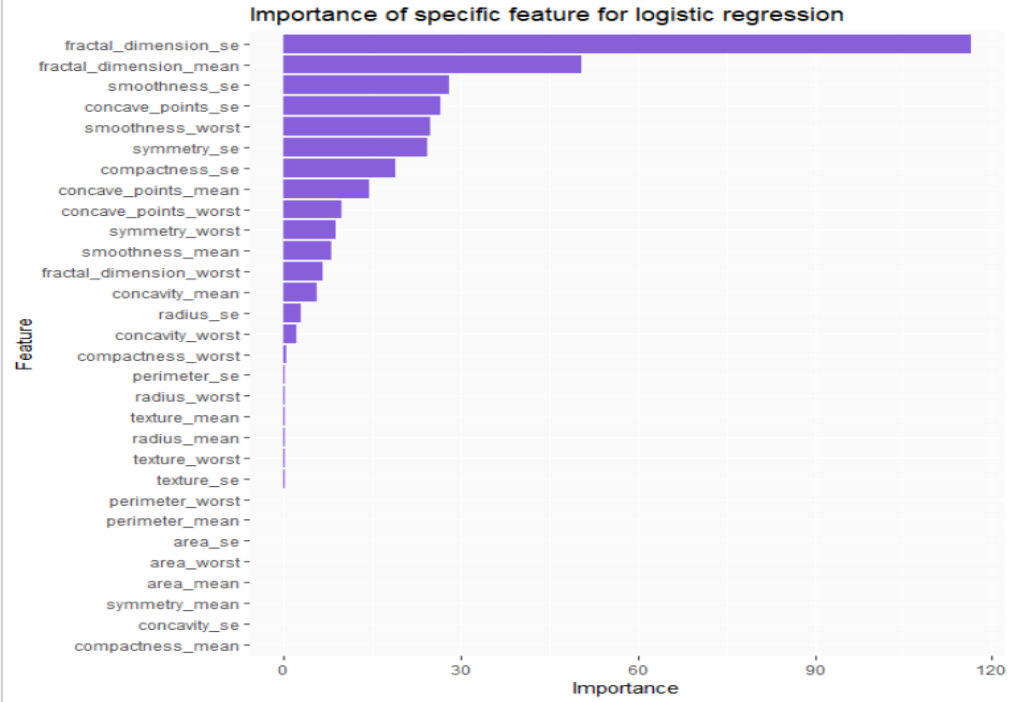
Classifier	Train Accuracy	Test Accuracy
Decision Tree	96.27%	92.4%
Logistic Regression	98.6%	97.35%
SVM	98.9%	96.46%

Variable/Feature Importance for models:

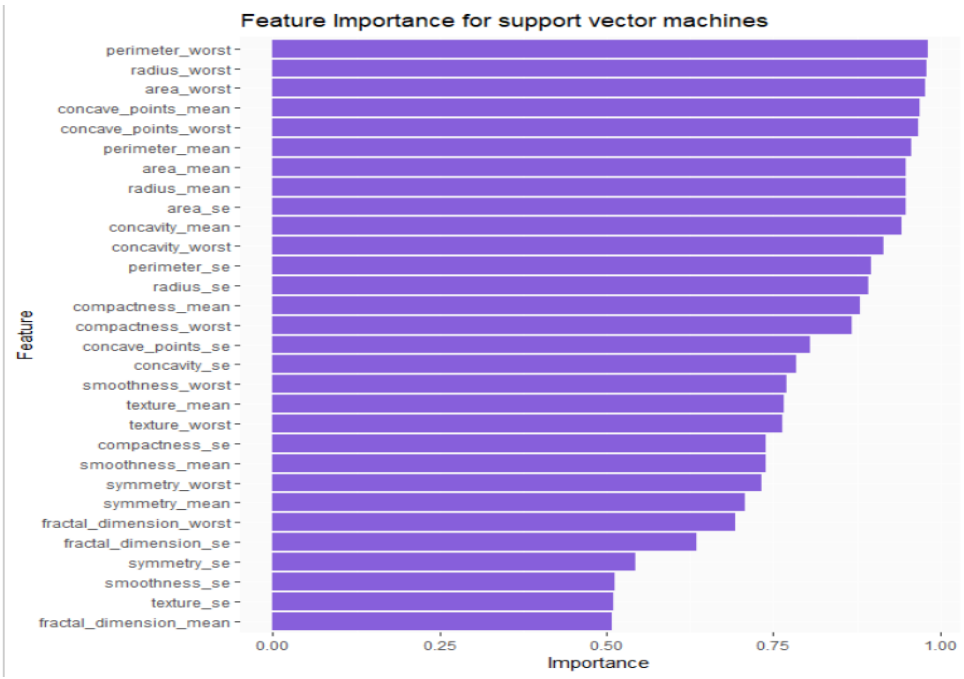
Decision trees - CART (Classification and regression tree) MODEL



Logistic Regression (Binary Classification):



Support Vector machines (SVM):



9. MODEL REVISION AND OPTIMIZATION

Gradient Boosting Machine: As a revision to the model, we will run the gradient boosting model to find the optimized combination of hyperparameters. The hyperparameters used are n.trees (the number of decision trees), shrinkage (learning rate), interaction.depth (the depth of each tree) bag.fraction (the sample size of each tree as a fraction of the dataset), and n.minobsinnode (the minimum number of observations in the terminal nodes).

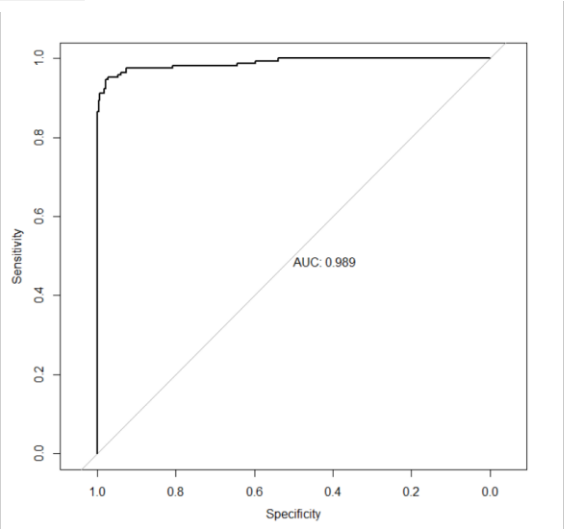
To develop an accurate binary classification model, we first split our dataset randomly into a training and a test set.

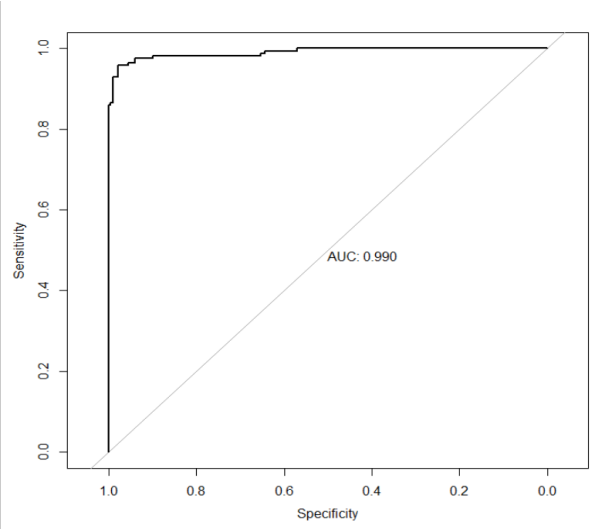
```
189 ## Gradient Boosting Machine model
190
191 # set.seed(1123)
192 set.seed(3011)
193 trainIndex1 <- createDataPartition(BC_wiscosin$diagnosis, p = .8, list = FALSE, times = 1)
194 train_all <- BC_wiscosin[ trainIndex1, ]
195 test_all <- BC_wiscosin[ -trainIndex1, ]
```

```r

## Creating the independent variable and label matricies of train/test data
train_all_data  <- as.matrix(train_all[-1])
train_all_label <- train_all$diagnosis
## Converting labels to 0,1 where "M" is coded at 1
train_all_label <- as.numeric(c("M" = "1", "B" = "0")[train_all_label])
train_all$diagnosis[1:5]; train_all_label[1:5]

## Repeat for test dataset
test_all_data   <- as.matrix(test_all[-1])
test_all_label  <- test_all$diagnosis
test_all_label <- as.numeric(c("M" = "1", "B" = "0")[test_all_label])
test_all$diagnosis[1:5]; test_all_label[1:5]


train_all_data <- as.data.frame(apply(train_all_data, 2, as.numeric))
test_all_data <- as.data.frame(apply(test_all_data, 2, as.numeric))

## Formatting data for XGBoost matricies
all_dtrain = xgb.DMatrix(data = as.matrix(sapply(train_all_data,as.numeric)), label=as.matrix(train_all_label))
all_dtest = xgb.DMatrix(data = as.matrix(sapply(test_all_data,as.numeric)), label=as.matrix(test_all_label))


### parameters: max_depth, eta, subsample, colsample_bytree, and min_child_weight
all_low_err_list <- list()
all_parameters_list <- list()
set.seed(99)
for(i in 1:100){
  params <- list(booster = "gbtree",
                 objective = "binary:logistic",
                 max_depth = sample(3:25, 1),
                 eta = runif(1, 0.01, 0.3),
                 subsample = runif(1, 0.5, 1),
                 colsample_bytree = runif(1, 0.5, 1),
                 min_child_weight = sample(0:10, 1)
  )

  parameters <- as.data.frame(params)
  all_parameters_list[[i]] <- parameters
}

all_parameters_df <- do.call(rbind, all_parameters_list) #df containing random search params

### Fitting xgboost models based on search parameters
for (row in 1:nrow(all_parameters_df)){
  set.seed(99)
  all_tmp_mdl <- xgb.cv(data = all_dtrain,
                        booster = "gbtree",
                        objective = "binary:logistic",
                        nfold = 5,
                        prediction = TRUE,
                        max_depth = all_parameters_df$max_depth[row],
                        eta = all_parameters_df$eta[row],
                        subsample = all_parameters_df$subsample[row],
                        colsample_bytree = all_parameters_df$colsample_bytree[row],
                        min_child_weight = all_parameters_df$min_child_weight[row],
                        nrounds = 200,
                        eval_metric = "error",
                        early_stopping_rounds = 20,
                        print_every_n = 500,
                        verbose = 0
  )

  #this is the lowest error for the iteration
  all_low_err <- as.data.frame(1 - min(all_tmp_mdl$evaluation_log$test_error_mean))
  all_low_err_list[[row]] <- all_low_err
}

all_low_err_df <- do.call(rbind, all_low_err_list) #accuracies
all_randsearch <- cbind(all_low_err_df, all_parameters_df) #data frame with everything

###Reformatting the dataframe
all_randsearch <- all_randsearch %>%
  dplyr::rename(val_acc = '1 - min(all_tmp_mdl$evaluation_log$test_error_mean)') %>%
  dplyr::arrange(-val_acc)

###Grabbing just the top model
all_randsearch_best <- all_randsearch[1,]

###Storing best parameters in list
all_best_params <- list(booster = all_randsearch_best$booster,
                        objective = all_randsearch_best$objective,
                        max_depth = all_randsearch_best$max_depth,
                        eta = all_randsearch_best$eta,
                        subsample = all_randsearch_best$subsample,
                        colsample_bytree = all_randsearch_best$colsample_bytree,
                        min_child_weight = all_randsearch_best$min_child_weight)
```

Finding the best nround parameter for the model using 5-fold cross validation. Model training using the best hyperparameters and then model testing and predictions.

```r
### Finding the best nround parameter for the model using 5-fold cross validation
set.seed(99)
all_xgbcv <- xgb.cv(params = all_best_params,
                    data = all_dtrain,
                    nrounds = 500,
                    nfold = 5,
                    prediction = TRUE,
                    print_every_n = 50,
                    early_stopping_rounds = 25,
                    eval_metric = "error",
                    verbose = 0
)
all_xgbcv$best_iteration


## Model training using best hyperparameters
set.seed(99)
all_best_xgb <- xgb.train(params = all_best_params,
                          data = all_dtrain,
                          nrounds = all_xgbcv$best_iteration,
                          eval_metric = "error",
)

xgb.save(all_best_xgb, 'final_xgb_cancerall')


## Model testing and visualizations
cancer_all.pred <- predict(all_best_xgb, all_dtest)
cancer_all.pred <- factor(ifelse(cancer_all.pred > 0.5, 1, 0),
                          labels = c("B", "M"))
test_all$diagnosis <- as.factor(test_all$diagnosis)
confusionMatrix(cancer_all.pred, test_all$diagnosis,
                mode = 'everything',
                positive = 'M')

cancer_all_train.pred <- predict(all_best_xgb, all_dtrain)
cancer_all_train.pred <- factor(ifelse(cancer_all_train.pred > 0.5, 1, 0),
                          labels = c("B", "M"))
train_all$diagnosis <- as.factor(train_all$diagnosis)
confusionMatrix(cancer_all_train.pred, train_all$diagnosis,
                mode = 'everything',
                positive = 'M')


### ROC curve for 5-fold CV random parameter search
all_randsearch_roc <- roc(response = train_all_label,
                          predictor = all_tmp_mdl$pred,
                          print.auc = TRUE,
                          plot = TRUE)

### ROC curve for 5-fold CV nround parameter search
all_nround_roc <- roc(response = train_all_label,
                      predictor = all_xgbcv$pred,
                      print.auc = TRUE,
                      plot = TRUE)
```

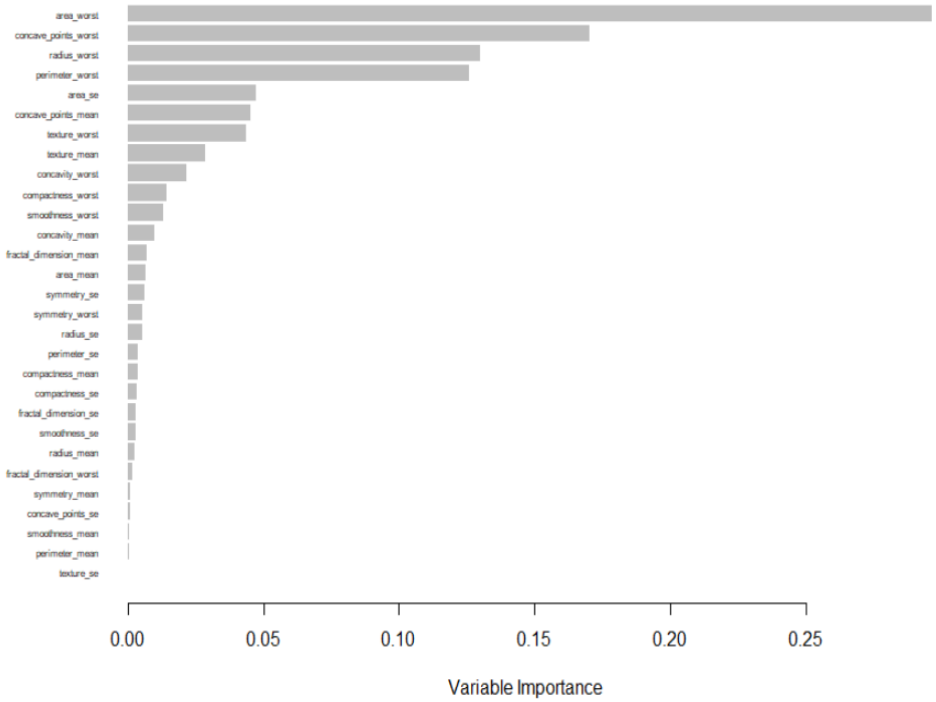




Classifier	Train Accuracy	Test Accuracy	AUC for random parameters	AUC for best parameters
Gradient Boosting Machine	100%	97.35%	98.9%	99%

ROC curves of the 5-fold cross validated hyperparameter searches had high AUC values (AUC = 98.9%, AUC = 99%) indicated that the training model performed well at classification. All four models perform well, probably because the cytological features of benign and malignant tumors are extremely different. In this case, I`d choose the gradient boosting model for their exceptionally high and consistent performance across all datasets. Let's take a closer look at the other key performance measures for this prediction model.

```
327
328 ## Visualizations
329 all_impt_mtx <- xgb.importance(feature_names = colnames(test_all_data), model = all_best_xgb)
330 xgb.plot.importance(importance_matrix = all_impt_mtx,
331                     xlab = "Variable Importance")
332
```



Radii of the cell, cell area, the number of concave points on the cell perimeters, and the perimeters themselves are the most influencing parameters and this is the same in the correlation matrix results as well.

10. RESULTS

Decision trees - CART (Classification and regression tree) MODEL
Test data Accuracy Vs Train data Accuracy:

```
> confusionMatrix(as.factor(test_set$diagnosis),predict_values)
Confusion Matrix and Statistics

          Reference
Prediction  B  M
         B 68  3
         M  6 36

               Accuracy : 0.9204
                 95% CI : (0.8542, 0.9629)
    No Information Rate : 0.6549
    P-Value [Acc > NIR] : 3.741e-11

                  Kappa : 0.827

 Mcnemar's Test P-Value : 0.505

            Sensitivity : 0.9189
            Specificity : 0.9231
         Pos Pred Value : 0.9577
         Neg Pred Value : 0.8571
             Prevalence : 0.6549
         Detection Rate : 0.6018
   Detection Prevalence : 0.6283
      Balanced Accuracy : 0.9210

       'Positive' Class : B
```

```
> confusionMatrix(as.factor(training_set$diagnosis),predict_values)
Confusion Matrix and Statistics

          Reference
Prediction   B   M
         B 278   8
         M   9 161

               Accuracy : 0.9627
                 95% CI : (0.941, 0.9781)
    No Information Rate : 0.6294
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.9202

 Mcnemar's Test P-Value : 1

            Sensitivity : 0.9686
            Specificity : 0.9527
         Pos Pred Value : 0.9720
         Neg Pred Value : 0.9471
             Prevalence : 0.6294
         Detection Rate : 0.6096
   Detection Prevalence : 0.6272
      Balanced Accuracy : 0.9607

       'Positive' Class : B
```

Logistic Regression (Binary Classification):
Train data Accuracy Vs Test data Accuracy:

```
> confusionMatrix(as.factor(test_set$diagnosis),predict_values)
Confusion Matrix and Statistics

          Reference
Prediction  B  M
         B 71  0
         M  3 39

               Accuracy : 0.9735
                 95% CI : (0.9244, 0.9945)
    No Information Rate : 0.6549
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.9423

 Mcnemar's Test P-Value : 0.2482

            Sensitivity : 0.9595
            Specificity : 1.0000
         Pos Pred Value : 1.0000
         Neg Pred Value : 0.9286
             Prevalence : 0.6549
         Detection Rate : 0.6283
   Detection Prevalence : 0.6283
      Balanced Accuracy : 0.9797

       'Positive' Class : B
```

```
> confusionMatrix(as.factor(training_set$diagnosis),predict_values)
Confusion Matrix and Statistics

          Reference
Prediction   B   M
         B 285   1
         M   5 165

               Accuracy : 0.9868
                 95% CI : (0.9716, 0.9952)
    No Information Rate : 0.636
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.9717

 Mcnemar's Test P-Value : 0.2207

            Sensitivity : 0.9828
            Specificity : 0.9940
         Pos Pred Value : 0.9965
         Neg Pred Value : 0.9706
             Prevalence : 0.6360
         Detection Rate : 0.6250
   Detection Prevalence : 0.6272
      Balanced Accuracy : 0.9884

       'Positive' Class : B
```

Support Vector machines (SVM):
Test data Accuracy Vs Train data Accuracy:

```
> confusionMatrix(as.factor(test_set$diagnosis),predict_values)
Confusion Matrix and Statistics

          Reference
Prediction  B  M
         B 70  1
         M  3 39

               Accuracy : 0.9646
                 95% CI : (0.9118, 0.9903)
    No Information Rate : 0.646
    P-Value [Acc > NIR] : 2.242e-16

                  Kappa : 0.9235

 Mcnemar's Test P-Value : 0.6171

            Sensitivity : 0.9589
            Specificity : 0.9750
         Pos Pred Value : 0.9859
         Neg Pred Value : 0.9286
             Prevalence : 0.6460
         Detection Rate : 0.6195
   Detection Prevalence : 0.6283
      Balanced Accuracy : 0.9670

       'Positive' Class : B
```

```
> confusionMatrix(as.factor(training_set_svm$diagnosis),predict_values)
Confusion Matrix and Statistics

          Reference
Prediction   B   M
         B 286   0
         M   5 165

               Accuracy : 0.989
                 95% CI : (0.9746, 0.9964)
    No Information Rate : 0.6382
    P-Value [Acc > NIR] : < 2e-16

                  Kappa : 0.9764

 Mcnemar's Test P-Value : 0.07364

            Sensitivity : 0.9828
            Specificity : 1.0000
         Pos Pred Value : 1.0000
         Neg Pred Value : 0.9706
             Prevalence : 0.6382
         Detection Rate : 0.6272
   Detection Prevalence : 0.6272
      Balanced Accuracy : 0.9914

       'Positive' Class : B
```

Gradient Boosting Machine:
Test data Accuracy Vs Train data Accuracy:

```
+                 positive = 'M')
Confusion Matrix and Statistics

          Reference
Prediction  B  M
         B 69  1
         M  2 41

               Accuracy : 0.9735
                 95% CI : (0.9244, 0.9945)
    No Information Rate : 0.6283
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.9434

 Mcnemar's Test P-Value : 1

            Sensitivity : 0.9762
            Specificity : 0.9718
         Pos Pred Value : 0.9535
         Neg Pred Value : 0.9857
              Precision : 0.9535
                 Recall : 0.9762
                     F1 : 0.9647
             Prevalence : 0.3717
         Detection Rate : 0.3628
   Detection Prevalence : 0.3805
      Balanced Accuracy : 0.9740

       'Positive' Class : M
```

```
+                 positive = 'M')
Confusion Matrix and Statistics

          Reference
Prediction   B   M
         B 286   0
         M   0 170

               Accuracy : 1
                 95% CI : (0.9919, 1)
    No Information Rate : 0.6272
    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
              Precision : 1.0000
                 Recall : 1.0000
                     F1 : 1.0000
             Prevalence : 0.3728
         Detection Rate : 0.3728
   Detection Prevalence : 0.3728
      Balanced Accuracy : 1.0000

       'Positive' Class : M
```

Classifier	Test Accuracy	Train Accuracy
Decision Tree	92.04%	96.27%
Logistic Regression	97.35%	98.6%
SVM	96.46%	98.9%
Gradient Boosting Model	97.35%	100%

11. LIMITATIONS

Further Analysis and Drawbacks –
Medical history needs to be collected. Demographic details need to be included.
For practical reasons, if the number of observations is small, I should know how to sample or collect the data.
To identify the limitations of current research on pathophysiology, detection, treatment, prevention, and psychosocial aspects of breast cancer. The purpose of this analysis is to:
To identify gaps in our knowledge of breast cancer that could benefit patients if addressed; To encourage breast cancer researchers and funding agencies around the world to focus their resources on highlighted areas of research to have a greater impact on patients; To make priority action recommendations.
Although there are several previous studies on breast tumors using different types of models, with some caveats, studies on breast cancer are limited due to the lack of published benchmark datasets. This study is the first to compare three common datasets and suggest the use of customized transfer learning algorithms for breast cancer classification and detection on multiple datasets.

12. CONCLUSION

The gradient boosting model correctly predicted 110 out of 113 diagnoses with an accuracy of 97.35%. However, this power measurement can be misleading, especially when you have an imbalanced data set as in this case. Fortunately, we got a more balanced accuracy of 100% for the training dataset. This means that the classifier performs equally well on both classes, rather than utilizing a skewed dataset. Based on this gradient boosting model, the top covariates that influenced the training model performance were cell radii, cell area, the number of concave points on the cell perimeters, and the perimeters themselves.

In this project, we have created four classification machine learning models that can predict if a person has breast cancer based on digitized image readings of patients' fine-needle aspirates. The best performing model, the gradient boosting, correctly classifies patients with and without breast cancer 97.35% of the time. ROC curves of the 5-fold cross validated hyperparameter searches had high AUC values (AUC = 99%, AUC = 98.9%) indicated that the training model performed well at classification and indicates a great ability to distinguish between a benign lump and a malignant tumor. The top cytological characteristics in identifying breast cancer are the cell radii, cell area, the number of concave points on the cell perimeters. And Logistic regression has distinguished

results with 97.35% and 98.6% accuracy with test data and train data subsequently prior to developing Gradient boosting model.

By recommending ways to fill these gaps in future research, long-term benefits for patients include - better prediction of drug response and patient prognosis; Improved tailoring of treatment to patient subgroups and developing new therapeutic approaches; Early initiation of treatment; more effective use of population screening resources; Improving the experience of people with breast cancer or at risk of breast cancer and their families. The challenge for funding agencies and researchers in all fields is to address these gaps and translate advances in knowledge into improved patient care.

13. REFERENCES

- **https://archive.ics.uci.edu/ml/datasets/breast+cancer+wisconsin+(diagnostic)**
- **https://towardsdatascience.com/building-a-simple-machine-learning-model-on-breast-cancer-data-eca4b3b99fa3**
- **https://rstudio-pubs-static.s3.amazonaws.com/411600_3185f5d17d104cc5beb4587094b905e9.html**
- **https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7351679/**
- **https://ieeexplore.ieee.org/abstract/document/8391468**

14. APPENDICES

Here is the link to the source code for reference.

Project on Breast Cancer dataset_Yasas