BREAST CANCER DIAGNOSIS

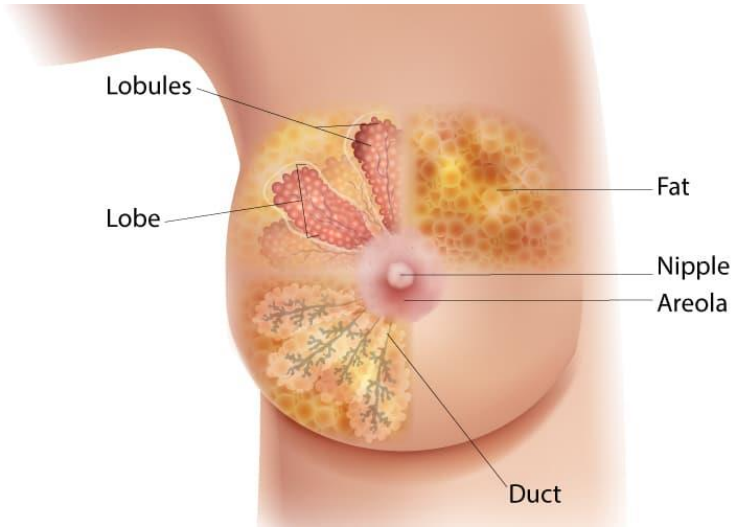
MICHAEL NGALA

MA 5790

DECEMBER 2023

Abstract

The Wisconsin Diagnostic Breast Cancer (WDBC) dataset, with 569 instances, is pivotal for developing predictive models to distinguish between benign and malignant breast cancer diagnoses. Each instance in the dataset corresponds to a fine needle aspirate (FNA) of a breast mass, digitally imaged and quantitatively described by 30 features that characterize the cell nuclei present. This study presents a robust predictive model based on the Wisconsin Diagnostic Breast Cancer (WDBC) dataset provided by the University of California, Irvine. The dataset contains 569 samples, each labeled as benign (B) or malignant (M), with an unbalanced distribution of 63% to 37%, respectively. The features, computed from digitized images of FNA of breast masses, include 30 numerical predictors describing cellular characteristics such as radius, texture, perimeter, area, smoothness, compactness, concavity, concave points, symmetry, and fractal dimension. Relationship between predictors will be explored to look for those with high correlation to ensure that an effective machine learning model can be built; both linear and nonlinear classification models will be built based on training and testing data set, then we will choose the best model for each and their variables and finally the ultimate model will be chosen from both the linear and nonlinear classification model.



Contents

1 Background

Breast cancer is a disease characterized by the growth of cells in the breast that multiply uncontrollably. It is the most prevalent cancer among women globally, aside from skin cancer, and it is the second leading cause of cancer death among women overall and the leading cause of cancer death among Hispanic women in the United States [1].

Many breast cancers originate in the milk ducts or the milk-producing lobules. While the earliest form of breast cancer, known as in situ, is not life-threatening, if left unchecked, cancerous tumors can become fatal by spreading throughout the body. The abnormal cells that characterize breast cancer divide more rapidly than healthy cells and may form a lump or mass. These cells can metastasize, meaning they spread from the breast to the lymph nodes or other parts of the body [2].

Signs of breast cancer can include a lump in the breast, changes in breast shape or size, dimpling of the skin, fluid discharge from the nipple, a newly inverted nipple, or a red or scaly patch of skin. When the disease has spread to other parts of the body, symptoms may include bone pain, swollen lymph nodes, shortness of breath, or yellowing of the skin.

Globally, breast cancer is one of the most diagnosed cancers, with an estimated 2.3 million new cases as per GLOBOCAN 2020 data. The mortality rate due to breast cancer is significantly higher in transitioning countries, highlighting the disparity in healthcare access and quality between different regions [3]. It's essential to recognize the risk factors associated with breast cancer, which can include genetic predispositions, lifestyle factors, certain types of hormonal therapy, and more. Understanding these risks can help in developing strategies for prevention, early detection, and treatment.

The accurate differentiation between malignant and benign breast tumors is vital for determining the appropriate course of treatment, impacting patient prognosis, quality of life, and healthcare resource allocation. Malignant tumors, which are life-threatening and can spread to other body parts, necessitate prompt and often aggressive treatment strategies, while benign tumors typically require less intensive intervention and carry a better prognosis. This distinction also plays a key role in managing patient anxiety, guiding research and epidemiological studies, and informing preventive measures for those at increased risk of breast cancer development in the future. Therefore, precise diagnosis is a critical component in the overall management of breast health, affecting not just individual patient care but also broader public health outcomes.

2 Variable Introductions and Definitions

The Breast Cancer Wisconsin (Diagnostic) dataset, curated by the University of California, Irvine, comprises 569 digitized images of fine needle aspirates (FNAs) from breast masses, out of which features describing the characteristics of the cell nuclei are computed. This dataset, which includes 32 variables such as sample ID and diagnostic outcome, uses 30 numerical predictors to model and classify each sample as either benign or malignant. The response variable 'Diagnosis' is categorical, with an unbalanced distribution of 63% benign to 37%

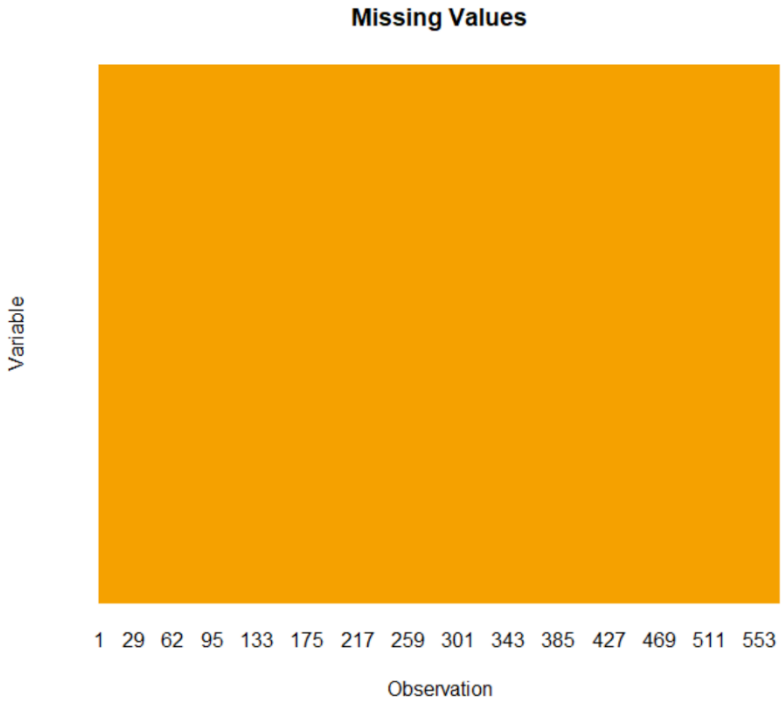
malignant cases. The selection of relevant features from these images was conducted using the Multisurface Method-Tree (MSM-T), applying an exhaustive search within the space of 1-4 features and 1-3 separating planes to optimize the model's predictive accuracy.

```
'data.frame':   569 obs. of  33 variables:
 $ id                     : 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 ...
```

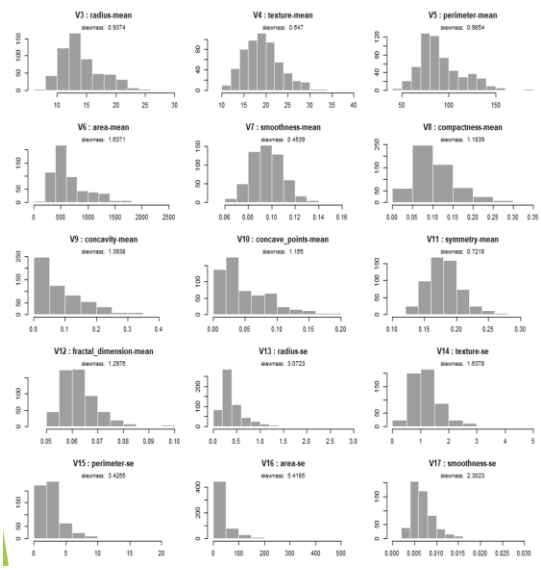
3 Preprocessing of the data

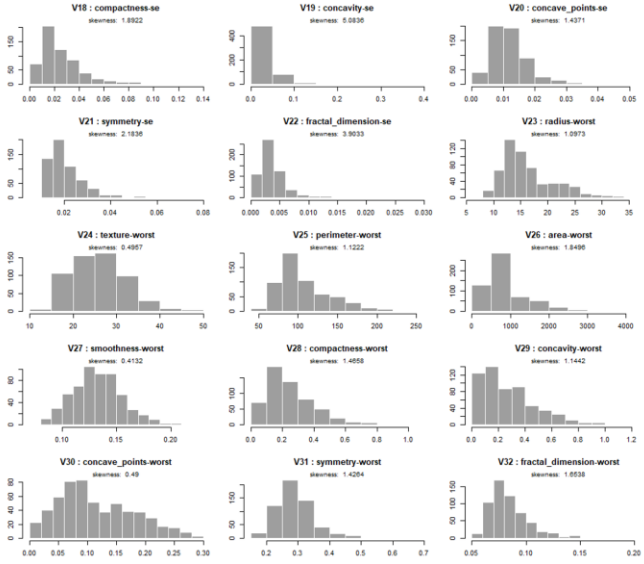
For data preprocessing, missing values were checked, and it was found that there were no missing values. Then the predictors were also checked for skewness and outliers by plotting box plots and bar graphs for visualization of the data. The predictors were also investigated for levels of correlation. The predictors were checked for degenerate predictors, and they were found to have no degenerate predictors.

a) Missing Values



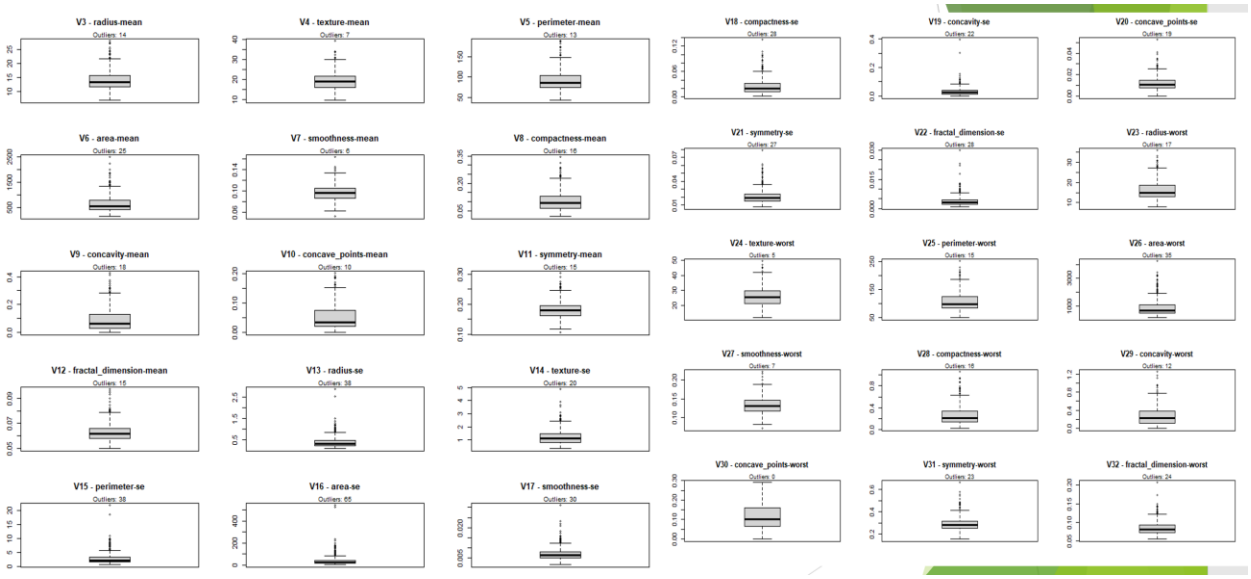
b) Visualizations of Predictors



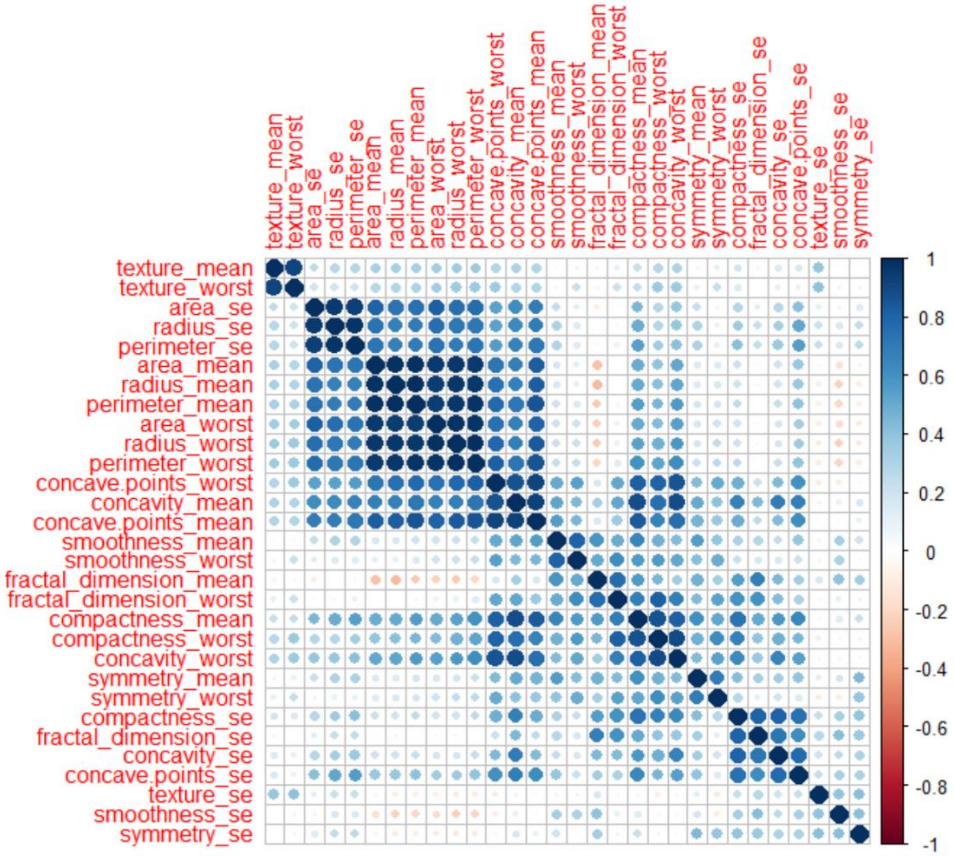


Predictor	Skewness	Range	Interpretation
V3-radius-mean	0.9374168	Between +0.5 and +1	Moderately right skewed
V4-texture-mean	0.6470241	Between +0.5 and +1	Moderately right skewed
V5-perimeter-mean	0.9854334	Between +0.5 and +1	Moderately right skewed
V6-area-mean	1.6370654	Greater than +1	Heavily right skewed
V7-smoothness-mean	0.4539207	Between -0.5 and +0.5	Approximately symmetric
V8-compactness-mean	1.1838556	Greater than +1	Heavily right skewed
V9-concavity-mean	1.3938008	Greater than +1	Heavily right skewed
V10-concave_points-mean	1.1650124	Greater than +1	Heavily right skewed
V11-symmetry-mean	0.7217877	Between +0.5 and +1	Moderately right skewed
V12-fractal_dimension-mean	1.2976191	Greater than +1	Heavily right skewed
V13-radius-se	3.072347	Greater than +1	Heavily right skewed
V14-texture-se	1.637773	Greater than +1	Heavily right skewed
V15-perimeter-se	3.425480	Greater than +1	Heavily right skewed
V16-area-se	5.418500	Greater than +1	Heavily right skewed
V17-smoothness-se	2.302262	Greater than +1	Heavily right skewed
V18-compactness-se	1.892203	Greater than +1	Heavily right skewed
V19-concavity-se	5.083550	Greater than +1	Heavily right skewed
V20-concave_points-se	1.437070	Greater than +1	Heavily right skewed
V21-symmetry-se	2.183573	Greater than +1	Heavily right skewed
V22-fractal_dimension-se	3.903304	Greater than +1	Heavily right skewed
V23-radius-worst	1.0973059	Greater than +1	Heavily right skewed
V24-texture-worst	0.4956970	Between -0.5 and +0.5	Approximately symmetric
V25-perimeter-worst	1.1222227	Greater than +1	Heavily right skewed
V26-area-worst	1.8495814	Greater than +1	Heavily right skewed
V27-smoothness-worst	0.4132383	Between -0.5 and +0.5	Approximately symmetric
V28-compactness-worst	1.4657948	Greater than +1	Heavily right skewed
V29-concavity-worst	1.1441794	Greater than +1	Heavily right skewed
V30-concave_points-worst	0.4900213	Between -0.5 and +0.5	Approximately symmetric
V31-symmetry-worst	1.4263764	Greater than +1	Heavily right skewed
V32-fractal_dimension-worst	1.6538237	Greater than +1	Heavily right skewed

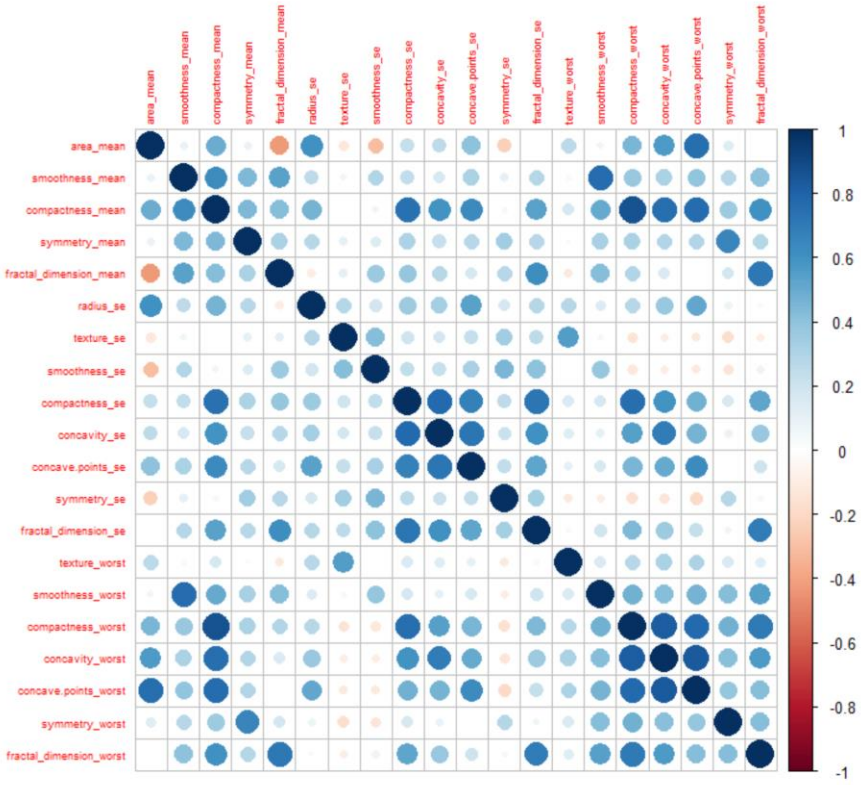
From the skewness values in the table above, 4 predictors are approximately symmetric, 4 predictors are moderately right skewed, and remaining 22 of the 30 predictors are heavily skewed to the right.



c) Visualizations of Correlation

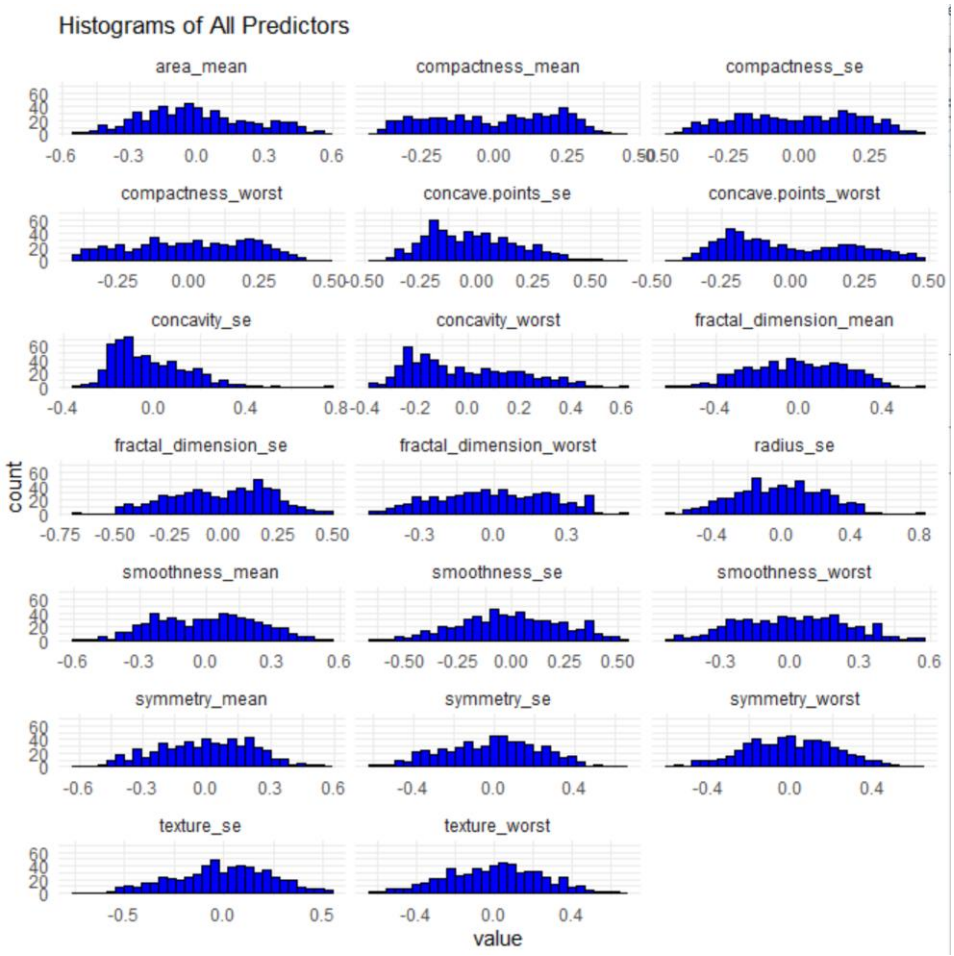


There seems to be a high correlation among the predictors above. The cut off was set at 90% and out of the 30 predictors, 10 were removed that were found to have high correlation.

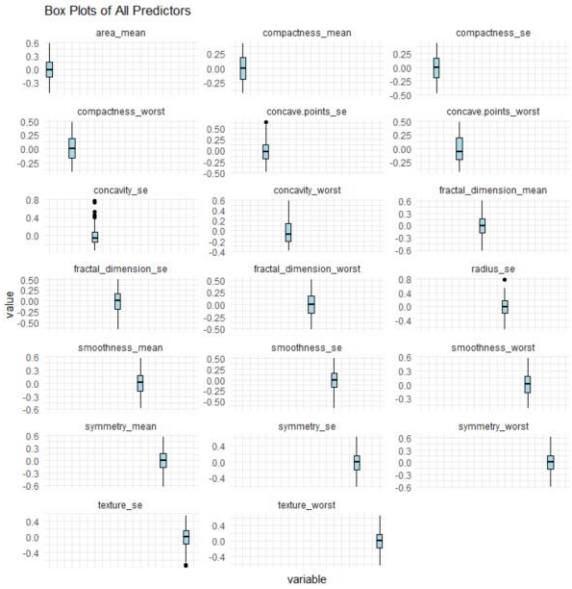


d) Visualizations of Transformed Predictors

The predictors were transformed by boxcox to remove skewness and spatialsign to remove outliers and finally center and scale.



From the above image we can see that the predictors have been successfully transformed and skewness has been reduced.



The above image shows a reduction in the number of outliers showing successful spatial sign transformation.

4 Splitting of the Data

The data was split into 80% training and 20% for testing and validation of the models where random stratified sampling was used to provide a good distribution of data that can be used to create and test the model.

5 Model Fitting

Linear and Nonlinear classification models were trained on the data and the figures and tables are displayed for the confusion matrix, the training of the model, variable of importance and the testing results. Leave Group Out Cross Validation was used as the data contains groups. This method ensures that all observations in a group are either in the training set or the test set, but not both. This is important to prevent information about the group from "leaking" into the model training process, which could result in overly optimistic performance estimates. The metrics used to determine efficacy of the model were Accuracy, Kappa, Sensitivity and Specificity since the response variable is categorical, either Malignant [M] or Benign [B].

For the Linear Models, the Logistic regression model and Linear Discriminant Analysis had no tuning parameters, while the PLSDA, Penalized model and Nearest Shrunken Centroids were tuned to fit the models. For the Non Linear models, the Naïve Bayes model had no tuning parameters while MDA, RDA, Neural Network, FDA, SVM and KNN were tuned to fit the models.

6 Summary

a) LINEAR MODELS

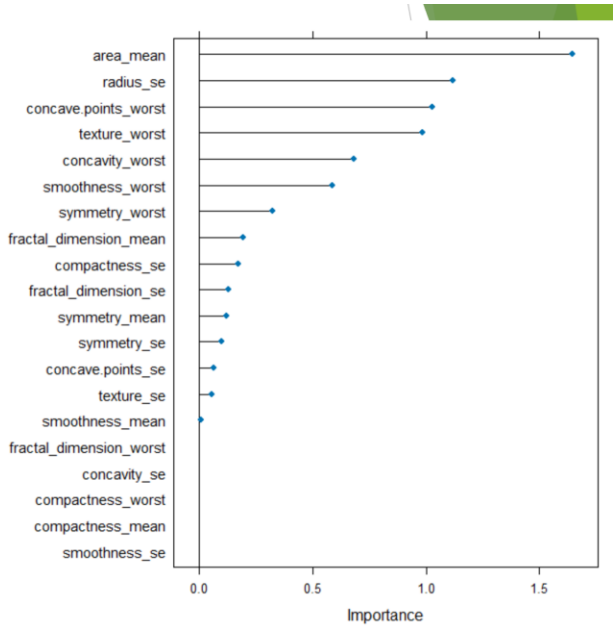
MODEL	TUNING PARAMETER	TRAINING	TESTING
LOGISTIC REGRESSION MODEL	No tuning parameters	ROC = 0.9615493 Sens = 0.9466667 Spec = 0.9549296	Sens = 0.9286 Spec = 0.9718 Accuracy = 0.9558 Kappa = 0.9048
LINEAR DISCRIMIANT ANALYSIS	No tuning parameters	ROC = 0.9930114 Sens = 0.9409524 Spec = 0.976338	Sens = 0.9286 Spec = 0.9718 Accuracy=0.9558 Kappa = 0.9048
PLSDA	Ncomp = 5	ROC = 0.9927431 Sens = 0.9400000 Spec = 0.9780282	Sens = 0.9286 Spec = 0.9859 Accuracy=0.9646 Kappa = 0.9235
PENALIZED MODEL	Alpha = 0.4 Lambda = 0.01	ROC = 0.9939638 Sens = 0.9314286 Spec = 0.9791549	Sens = 0.9524 Spec = 0.9859 Accuracy=0.9735 Kappa = 0.9429
NEAREST SHRUNKEN CENTROIDS	Threshold = 4	ROC = 0.9745137 Sens = 0.8352381 Spec = 0.9538028	Sens = 0.9286 Spec = 0.9155 Accuracy = 0.9204 Kappa = 0.8319

The Penalized Model is considered the best Linear model because it consistently shows high performance across all these metrics with the highest accuracy values of 0.9735 and Kappa = 0.9429 followed by PLSDA model with accuracy values of 0.9646 and Kappa = 0.9235. It suggests that the model balances the trade-off between sensitivity and specificity well while maintaining high overall accuracy and a strong agreement between predictions and actual outcomes. This balance is crucial for a robust model, especially in practice where both false positives and false negatives carry a cost.

Below are the variables of importance for the Penalized Model.

glmnet variable importance

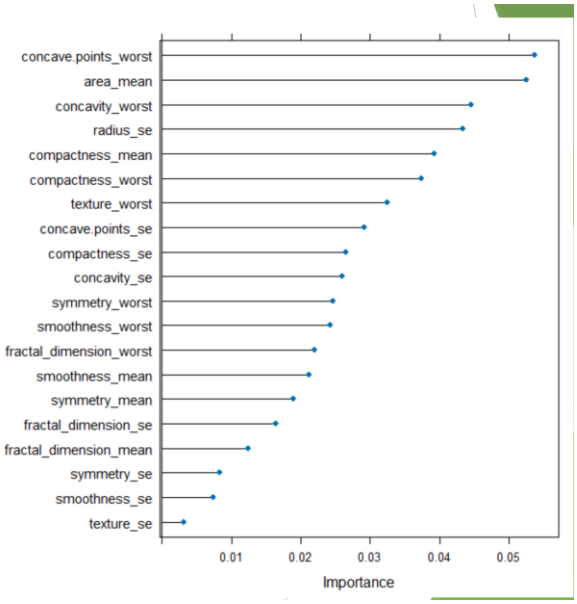
	Overall
area_mean	1.643700
radius_se	1.119219
concave.points_worst	1.028896
texture_worst	0.981978
concavity_worst	0.683181
smoothness_worst	0.585932
symmetry_worst	0.323118
fractal_dimension_mean	0.195431
compactness_se	0.173473
fractal_dimension_se	0.127598
symmetry_mean	0.120175
symmetry_se	0.096728
concave.points_se	0.065023
texture_se	0.054681
smoothness_mean	0.007555
compactness_mean	0.000000
smoothness_se	0.000000
fractal_dimension_worst	0.000000
compactness_worst	0.000000
concavity_se	0.000000



Below are the variables of importance for the PLSDA model

pls variable importance

	Overall
concave.points_worst	0.053701
area_mean	0.052589
concavity_worst	0.044515
radius_se	0.043428
compactness_mean	0.039251
compactness_worst	0.037351
texture_worst	0.032486
concave.points_se	0.029181
compactness_se	0.026518
concavity_se	0.025975
symmetry_worst	0.024663
smoothness_worst	0.024302
fractal_dimension_worst	0.021997
smoothness_mean	0.021133
symmetry_mean	0.018970
fractal_dimension_se	0.016469
fractal_dimension_mean	0.012479
symmetry_se	0.008255
smoothness_se	0.007357
texture_se	0.003157



b) NON-LINEAR MODELS

MODELS	TUNING PARAMETER	TRAINING	TESTING
NON LINEAR[MIXTURE DISCRIMINANT ANALYSIS]	Subclasses = 1	ROC = 0.9948893 Sens = 0.9371429 Spec = 0.9814085	Sens = 0.9286 Spec = 0.9718 Accuracy = 0.9558 Kappa = 0.9048
REGULARIZED DISCRIMINANT ANALYSIS	Gamma = 0.2 Lambda = 0.1	ROC = 0.9956942 Sens = 0.9580952 Spec = 0.9701408	Sens = 0.9286 Spec = 0.9437 Accuracy = 0.9381 Kappa = 0.868
NEURAL NETWORK	Size = 4 Decay = 0.1	ROC = 0.9965258 Sens = 0.9590476 Spec = 0.9684507	Sens = 0.9524 Spec = 0.9718 Accuracy = 0.9646 Kappa = 0.9242
FLEXIBLE DISCRIMINANT ANALYSIS	Degree = 2 Nprune = 24	ROC = 0.9861368 Sens = 0.9057143 Spec = 0.9678873	Sens = 0.9286 Spec = 0.9859 Accuracy = 0.9646 Kappa = 0.9235
SUPPORT VECTOR MACHINE	Sigma = 0.003001256 C = 32	Accuracy = 0.4936759 Kappa = 0.076998753	Sens = 0.9524 Spec = 0.9718 Accuracy = 0.9646 Kappa = 0.9242
K NEAREST NEIGHBOR	K = 7	ROC = 0.9861234 Sens = 0.9342857 Spec = 0.9594366	Sens = 0.9048 Spec = 0.9296 Accuracy = 0.9204 Kappa = 0.8303
NAÏVE BAYES	No tuning parameters	ROC = 0.9776123 Sens = 0.912381 Spec = 0.9092958	Sens = 0.8333 Spec = 0.8592 Accuracy = 0.8496 Kappa = 0.6825

The Neural Network model and Support Vector Machine model had the best Accuracy and Kappa values compared to all the models. All their values were similar for the testing.

Sens = 0.9524

Spec = 0.9718

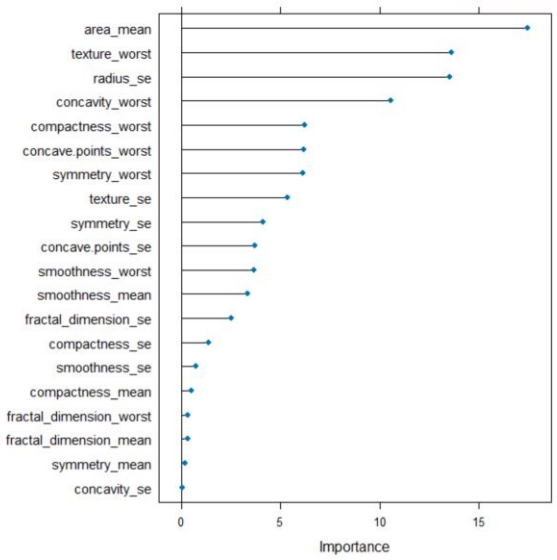
Accuracy = 0.9646

Kappa = 0.9242

The variable of importance for Neural Network

nnet variable importance

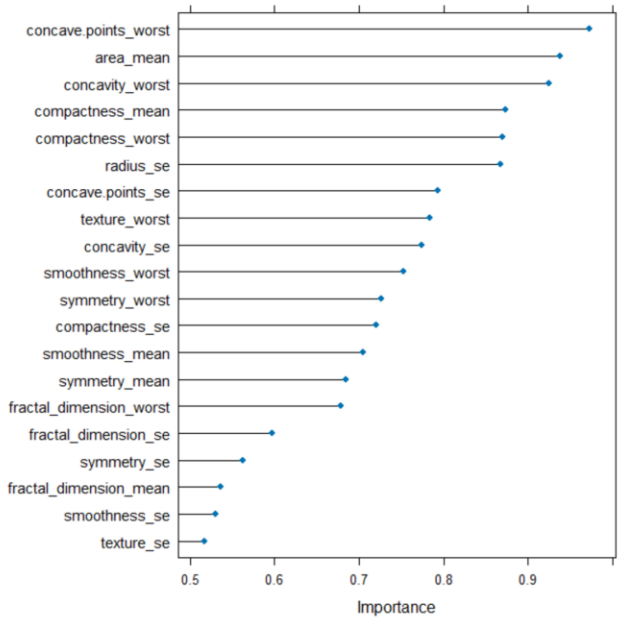
	Overall
area_mean	17.4555
texture_worst	13.6372
radius_se	13.5286
concavity_worst	10.5796
compactness_worst	6.2039
concave.points_worst	6.1958
symmetry_worst	6.1129
texture_se	5.3557
symmetry_se	4.1278
concave.points_se	3.7292
smoothness_worst	3.6517
smoothness_mean	3.3333
fractal_dimension_se	2.5153
compactness_se	1.3634
smoothness_se	0.7296
compactness_mean	0.5247
fractal_dimension_worst	0.3540
fractal_dimension_mean	0.3269
symmetry_mean	0.2057
concavity_se	0.0692



The carriable of importance of SVM

ROC curve variable importance

	Importance
concave.points_worst	0.9723
area_mean	0.9384
concavity_worst	0.9254
compactness_mean	0.8737
compactness_worst	0.8694
radius_se	0.8672
concave.points_se	0.7928
texture_worst	0.7834
concavity_se	0.7741
smoothness_worst	0.7525
symmetry_worst	0.7260
compactness_se	0.7202
smoothness_mean	0.7045
symmetry_mean	0.6847
fractal_dimension_worst	0.6782
fractal_dimension_se	0.5971
symmetry_se	0.5630
fractal_dimension_mean	0.5366
smoothness_se	0.5301
texture_se	0.5175



For the best overall model between both Linear and Non Linear model, The Penalized Linear model is the best model.

OVERALL BEST MODEL

- Considering the best overall model between the best Linear and Non Linear model, The Penalized model has higher testing accuracy and testing Kappa values as compared to Neural Network model. Hence the best overall model is the Linear Penalized model.

Penalized Model:

- ROC: 0.9939638
- Training Sensitivity: 0.9314286
- Training Specificity: 0.9791549
- Testing Sensitivity: 0.9524
- Testing Specificity: 0.9859
- Testing Accuracy: 0.9735
- Testing Kappa: 0.9429

Neural Network:

- ROC: 0.9965258
- Training Sensitivity: 0.9590476
- Training Specificity: 0.9684507
- Testing Sensitivity: 0.9524
- Testing Specificity: 0.9718
- Testing Accuracy: 0.9646
- Testing Kappa: 0.9242

Appendix 1: LINEAR CLASSIFICATION MODELS

a) LOGISTIC REGRESSION MODEL

```
Generalized Linear Model

456 samples
 20 predictor
  2 classes: 'M', 'B'

Pre-processing: centered (20), scaled (20)
Resampling: Repeated Train/Test Splits Estimated (25 reps, 75%)
Summary of sample sizes: 343, 343, 343, 343, 343, 343, ...
Resampling results:

  ROC        Sens       Spec
  0.9615493  0.9466667  0.9549296
```

TRAINING VALUES

ROC = 0.9615493

Sens = 0.9466667

Spec = 0.9549296

```
Confusion Matrix and Statistics

          Reference
Prediction  M  B
         M 39  2
         B  3 69

               Accuracy : 0.9558
                 95% CI : (0.8998, 0.9855)
    No Information Rate : 0.6283
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.9048

 Mcnemar's Test P-Value : 1

            Sensitivity : 0.9286
            Specificity : 0.9718
         Pos Pred Value : 0.9512
         Neg Pred Value : 0.9583
             Prevalence : 0.3717
         Detection Rate : 0.3451
   Detection Prevalence : 0.3628
      Balanced Accuracy : 0.9502

       'Positive' Class : M
```

TESTING VALUES

Sens = 0.9286

Spec = 0.9718

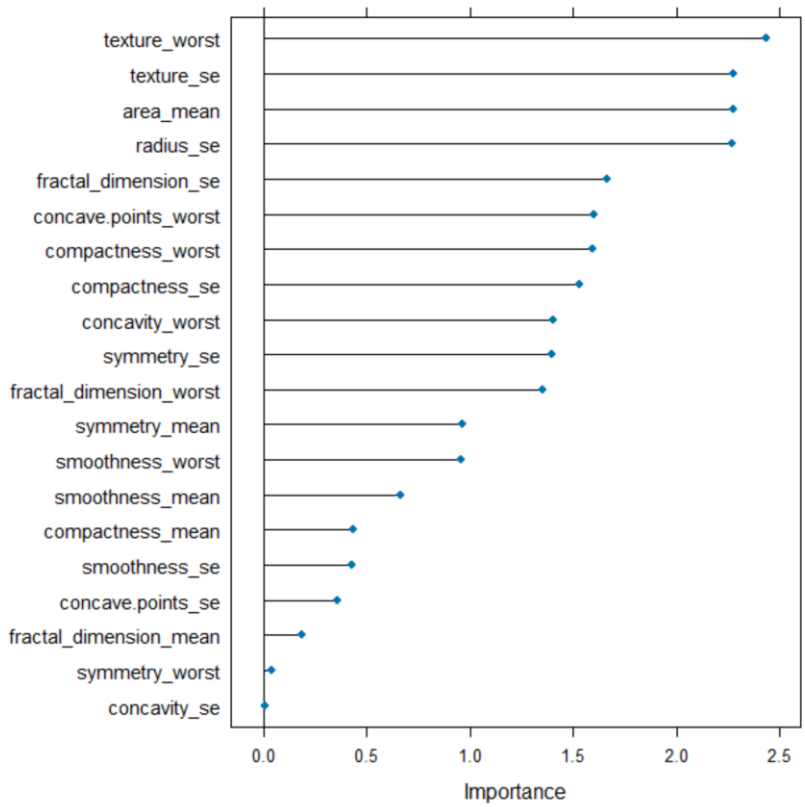
Accuracy = 0.9558

Kappa = 0.9048

There are no tuning parameters to plot a graph.

glm variable importance

	Overall
texture_worst	2.434490
texture_se	2.276107
area_mean	2.275365
radius_se	2.268644
fractal_dimension_se	1.663980
concave.points_worst	1.598233
compactness_worst	1.591903
compactness_se	1.529479
concavity_worst	1.405242
symmetry_se	1.394408
fractal_dimension_worst	1.350119
symmetry_mean	0.962021
smoothness_worst	0.960155
smoothness_mean	0.661709
compactness_mean	0.432326
smoothness_se	0.430617
concave.points_se	0.356920
fractal_dimension_mean	0.182954
symmetry_worst	0.042331
concavity_se	0.008945



b) LINEAR DISCRIMINANT ANALYSIS

```
Linear Discriminant Analysis

456 samples
 20 predictor
  2 classes: 'M', 'B'

Pre-processing: centered (20), scaled (20)
Resampling: Repeated Train/Test Splits Estimated (25 reps, 75%)
Summary of sample sizes: 343, 343, 343, 343, 343, 343, ...
Resampling results:

  ROC        Sens       Spec
  0.9930114  0.9409524  0.976338
```

TRAINING VALUES

ROC = 0.9930114
Sens = 0.9409524
Spec = 0.976338

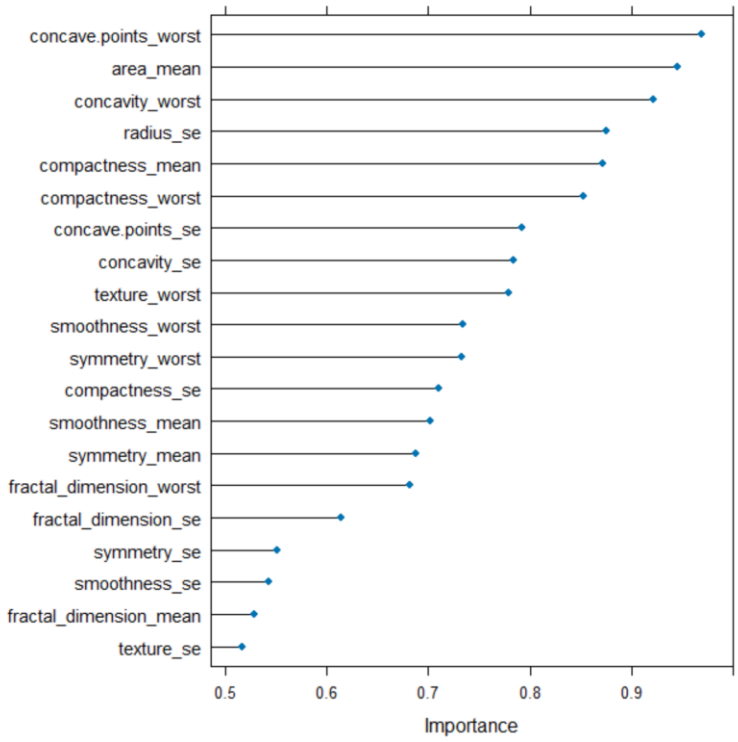
```
Confusion Matrix and Statistics

          Reference
Prediction  M  B
         M 39  2
         B  3 69

               Accuracy : 0.9558
                 95% CI : (0.8998, 0.9855)
    No Information Rate : 0.6283
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.9048

 Mcnemar's Test P-Value : 1

            Sensitivity : 0.9286
            Specificity : 0.9718
         Pos Pred Value : 0.9512
         Neg Pred Value : 0.9583
             Prevalence : 0.3717
         Detection Rate : 0.3451
   Detection Prevalence : 0.3628
      Balanced Accuracy : 0.9502

       'Positive' Class : M
```

Sens = 0.9286
Spec = 0.9718
Accuracy = 0.9558
Kappa = 0.9048

There are no tuning parameters to plot

VARIABLE OF IMPORTANCE

```
ROC curve variable importance

                         Importance
concave.points_worst         0.9691
area_mean                    0.9455
concavity_worst              0.9215
radius_se                    0.8753
compactness_mean             0.8713
compactness_worst            0.8525
concave.points_se            0.7928
concavity_se                 0.7835
texture_worst                0.7789
smoothness_worst             0.7336
symmetry_worst               0.7327
compactness_se               0.7105
smoothness_mean              0.7019
symmetry_mean                0.6878
fractal_dimension_worst      0.6815
fractal_dimension_se         0.6138
symmetry_se                  0.5513
smoothness_se                0.5423
fractal_dimension_mean       0.5287
texture_se                   0.5166
```



c) PARTIAL LEAST SQAURE DISCRIMINANT ANALYSIS (PLSDA)

```
Partial Least Squares

456 samples
 20 predictor
  2 classes: 'M', 'B'

Pre-processing: centered (20), scaled (20)
Resampling: Repeated Train/Test Splits Estimated (25 reps, 75%)
Summary of sample sizes: 343, 343, 343, 343, 343, 343, ...
Resampling results across tuning parameters:

  ncomp  ROC        Sens       Spec
   1     0.9565526  0.8647619  0.8907042
   2     0.9908518  0.9323810  0.9543662
   3     0.9921261  0.9342857  0.9673239
   4     0.9922334  0.9314286  0.9723944
   5     0.9927431  0.9400000  0.9780282
   6     0.9922200  0.9409524  0.9814085
   7     0.9924346  0.9361905  0.9808451
   8     0.9920456  0.9304762  0.9808451
   9     0.9918310  0.9304762  0.9825352
  10     0.9918176  0.9285714  0.9808451
  11     0.9916968  0.9295238  0.9797183
  12     0.9915761  0.9304762  0.9797183
  13     0.9915359  0.9314286  0.9802817
  14     0.9916566  0.9314286  0.9808451

ROC was used to select the optimal model using the largest value.
The final value used for the model was ncomp = 5.
```

TRAINING VALUES

Tuning parameter

Ncomp = 5

ROC = 0.9927431

Sens = 0.9400000

Spec = 0.9780282

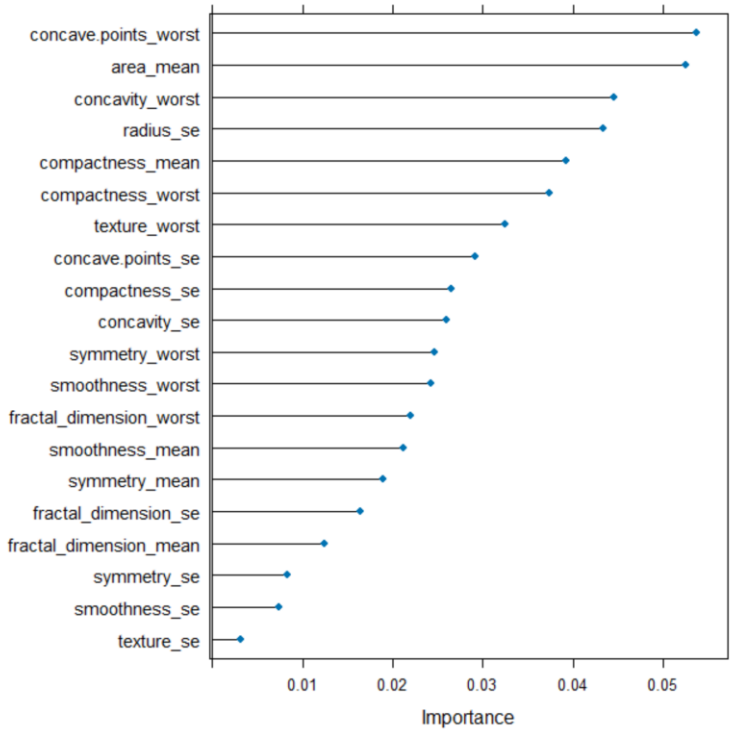
```
Confusion Matrix and Statistics

          Reference
Prediction  M  B
         M 39  1
         B  3 70

               Accuracy : 0.9646
                 95% CI : (0.9118, 0.9903)
    No Information Rate : 0.6283
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.9235

 Mcnemar's Test P-Value : 0.6171

            Sensitivity : 0.9286
            Specificity : 0.9859
         Pos Pred Value : 0.9750
         Neg Pred Value : 0.9589
             Prevalence : 0.3717
         Detection Rate : 0.3451
   Detection Prevalence : 0.3540
      Balanced Accuracy : 0.9572

       'Positive' Class : M
```

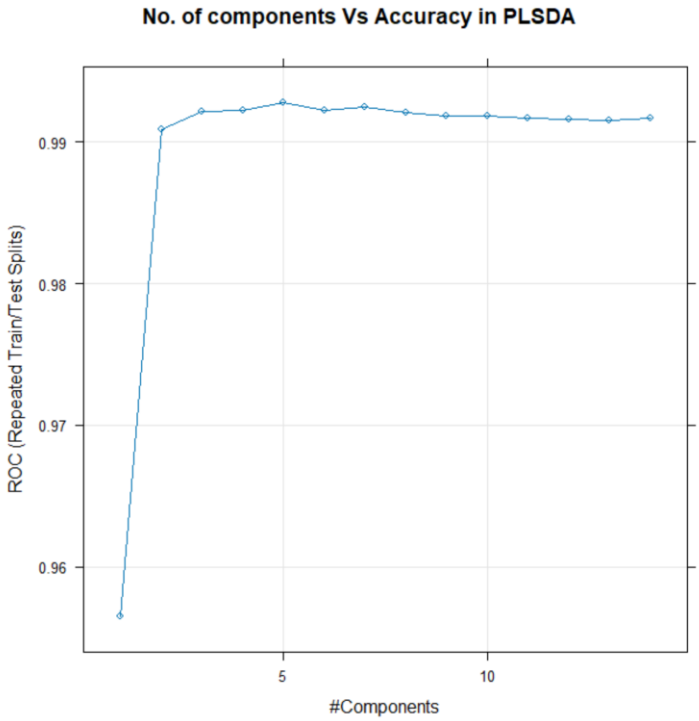
TESTING VALUES

Sens = 0.9286
Spec = 0.9859
Accuracy = 0.9646
Kappa = 0.9235

pls variable importance

	Overall
concave.points_worst	0.053701
area_mean	0.052589
concavity_worst	0.044515
radius_se	0.043428
compactness_mean	0.039251
compactness_worst	0.037351
texture_worst	0.032486
concave.points_se	0.029181
compactness_se	0.026518
concavity_se	0.025975
symmetry_worst	0.024663
smoothness_worst	0.024302
fractal_dimension_worst	0.021997
smoothness_mean	0.021133
symmetry_mean	0.018970
fractal_dimension_se	0.016469
fractal_dimension_mean	0.012479
symmetry_se	0.008255
smoothness_se	0.007357
texture_se	0.003157





The optimum number of components was found to be 5 with the highest ROC value.

d) PENALIZED MODEL

```
glmnet

456 samples
 20 predictor
  2 classes: 'M', 'B'

Pre-processing: centered (20), scaled (20)
Resampling: Repeated Train/Test Splits Estimated (25 reps, 75%)
Summary of sample sizes: 343, 343, 343, 343, 343, 343, ...
Resampling results across tuning parameters:
```

alpha	lambda	ROC	Sens	Spec
0.0	0.01000000	0.9915627	0.9219048	0.9752113
0.0	0.03111111	0.9915627	0.9219048	0.9752113
0.0	0.05222222	0.9910262	0.9200000	0.9735211
0.0	0.07333333	0.9901006	0.9123810	0.9740845
0.0	0.09444444	0.9893628	0.9076190	0.9707042
0.0	0.11555556	0.9885848	0.9057143	0.9701408
0.0	0.13666667	0.9879812	0.9019048	0.9695775
0.0	0.15777778	0.9873910	0.9000000	0.9684507
0.0	0.17888889	0.9868947	0.8990476	0.9678873
0.0	0.20000000	0.9864520	0.8961905	0.9673239
0.1	0.01000000	0.9938967	0.9333333	0.9746479
0.1	0.03111111	0.9922334	0.9247619	0.9791549
0.1	0.05222222	0.9910262	0.9190476	0.9785915
0.1	0.07333333	0.9901274	0.9133333	0.9752113
0.1	0.09444444	0.9894165	0.9066667	0.9740845
0.1	0.11555556	0.9886519	0.9009524	0.9735211
0.1	0.13666667	0.9879678	0.8980952	0.9729577
0.1	0.15777778	0.9873910	0.8961905	0.9718310
0.1	0.17888889	0.9869752	0.8942857	0.9707042
0.1	0.20000000	0.9864386	0.8876190	0.9707042
0.2	0.01000000	0.9938833	0.9304762	0.9763380
0.2	0.03111111	0.9922200	0.9266667	0.9802817
0.2	0.05222222	0.9910530	0.9171429	0.9780282
0.2	0.07333333	0.9902213	0.9066667	0.9785915
0.2	0.09444444	0.9895506	0.9000000	0.9785915
0.2	0.11555556	0.9888934	0.8980952	0.9774648
0.2	0.13666667	0.9882897	0.8942857	0.9774648
0.2	0.15777778	0.9877934	0.8914286	0.9763380
0.2	0.17888889	0.9871362	0.8819048	0.9757746
0.2	0.20000000	0.9865191	0.8742857	0.9746479
0.4	0.01000000	0.9939638	0.9314286	0.9791549
0.4	0.03111111	0.9925553	0.9228571	0.9847887
0.4	0.05222222	0.9917103	0.9133333	0.9825352
0.4	0.07333333	0.9909859	0.9009524	0.9808451
0.4	0.09444444	0.9902616	0.8980952	0.9802817
0.4	0.11555556	0.9896848	0.8933333	0.9797183
0.4	0.13666667	0.9891348	0.8885714	0.9797183
0.4	0.15777778	0.9883166	0.8857143	0.9780282
0.4	0.17888889	0.9877934	0.8800000	0.9769014
0.4	0.20000000	0.9870557	0.8761905	0.9780282
0.6	0.01000000	0.9939370	0.9314286	0.9808451
0.6	0.03111111	0.9924748	0.9180952	0.9847887
0.6	0.05222222	0.9917237	0.9133333	0.9830986
0.6	0.07333333	0.9907579	0.9095238	0.9808451
0.6	0.09444444	0.9899799	0.8971429	0.9785915
0.6	0.11555556	0.9890409	0.8895238	0.9769014
0.6	0.13666667	0.9880349	0.8838095	0.9780282
0.6	0.15777778	0.9871496	0.8742857	0.9791549
0.6	0.17888889	0.9861435	0.8590476	0.9791549
0.6	0.20000000	0.9848826	0.8457143	0.9808451
0.8	0.01000000	0.9938565	0.9285714	0.9814085
0.8	0.03111111	0.9922871	0.9171429	0.9853521
0.8	0.05222222	0.9910530	0.9095238	0.9819718
0.8	0.07333333	0.9892421	0.9019048	0.9729577
0.8	0.09444444	0.9880885	0.8866667	0.9740845
0.8	0.11555556	0.9863045	0.8752381	0.9769014
0.8	0.13666667	0.9843461	0.8609524	0.9791549
0.8	0.15777778	0.9830852	0.8485714	0.9814085
0.8	0.17888889	0.9822938	0.8352381	0.9864789

```
  0.8    0.20000000  0.9813414  0.8247619  0.9870423
  1.0    0.01000000  0.9932797  0.9323810  0.9791549
  1.0    0.03111111  0.9915493  0.9171429  0.9780282
  1.0    0.05222222  0.9887860  0.8933333  0.9752113
  1.0    0.07333333  0.9866667  0.8790476  0.9746479
  1.0    0.09444444  0.9836620  0.8638095  0.9729577
  1.0    0.11555556  0.9805634  0.8552381  0.9695775
  1.0    0.13666667  0.9789537  0.8447619  0.9712676
  1.0    0.15777778  0.9773172  0.8333333  0.9774648
  1.0    0.17888889  0.9757612  0.8104762  0.9830986
  1.0    0.20000000  0.9739370  0.7885714  0.9830986

ROC was used to select the optimal model using the largest value.
The final values used for the model were alpha = 0.4 and lambda = 0.01.
```

TRAINING VALUES

Tuning parameters

Alpha = 0.4

Lambda = 0.01

ROC = 0.9939638

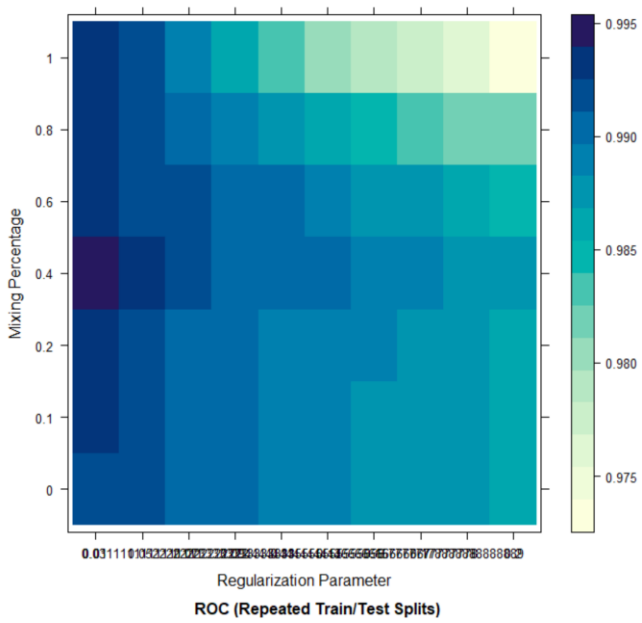
Sens = 0.9314286

Spec = 0.9791549

```
Confusion Matrix and Statistics

          Reference
Prediction  M  B
         M 40  1
         B  2 70

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

                  Kappa : 0.9429

 Mcnemar's Test P-Value : 1

            Sensitivity : 0.9524
            Specificity : 0.9859
         Pos Pred Value : 0.9756
         Neg Pred Value : 0.9722
             Prevalence : 0.3717
         Detection Rate : 0.3540
   Detection Prevalence : 0.3628
      Balanced Accuracy : 0.9691

       'Positive' Class : M
```

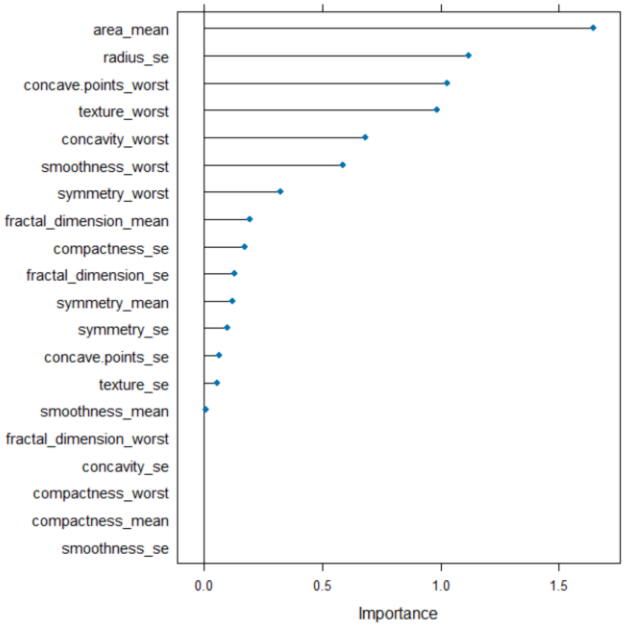
TESTING VALUES

```
Sens = 0.9524
Spec = 0.9859
Accuracy = 0.9735
Kappa = 0.9429
```



VARIABLE OF IMPORTANCE

```
glmnet variable importance

                          overall
area_mean                1.643700
radius_se                1.119219
concave.points_worst     1.028896
texture_worst            0.981978
concavity_worst          0.683181
smoothness_worst         0.585932
symmetry_worst           0.323118
fractal_dimension_mean   0.195431
compactness_se           0.173473
fractal_dimension_se     0.127598
symmetry_mean            0.120175
symmetry_se              0.096728
concave.points_se        0.065023
texture_se               0.054681
smoothness_mean          0.007555
compactness_mean         0.000000
smoothness_se            0.000000
fractal_dimension_worst  0.000000
compactness_worst        0.000000
concavity_se             0.000000
```



e) NEAREST SHRUNKEN CENTROIDS

```
Pre-processing: centered (20), scaled (20)
Resampling: Repeated Train/Test Splits Estimated (25 reps, 75%)
Summary of sample sizes: 343, 343, 343, 343, 343, 343, ...
Resampling results across tuning parameters:

  threshold  ROC        Sens       Spec
  0.0        0.9592220  0.8457143  0.9070423
  0.1        0.9593293  0.8457143  0.9064789
  0.2        0.9595171  0.8457143  0.9064789
  0.3        0.9597720  0.8447619  0.9064789
  0.4        0.9598793  0.8447619  0.9059155
  0.5        0.9599598  0.8447619  0.9064789
  0.6        0.9602146  0.8447619  0.9076056
  0.7        0.9603756  0.8447619  0.9076056
  0.8        0.9607512  0.8457143  0.9076056
  0.9        0.9610597  0.8457143  0.9076056
  1.0        0.9615962  0.8466667  0.9076056
  1.1        0.9620121  0.8466667  0.9076056
  1.2        0.9624279  0.8466667  0.9087324
  1.3        0.9630584  0.8466667  0.9104225
  1.4        0.9635278  0.8466667  0.9132394
  1.5        0.9639705  0.8457143  0.9154930
  1.6        0.9645473  0.8457143  0.9171831
  1.7        0.9652180  0.8457143  0.9200000
  1.8        0.9656338  0.8457143  0.9222535
  1.9        0.9660496  0.8438095  0.9245070
  2.0        0.9665057  0.8438095  0.9250704
  2.1        0.9671093  0.8438095  0.9261972
  2.2        0.9676861  0.8457143  0.9278873
  2.3        0.9683836  0.8438095  0.9290141
  2.4        0.9690275  0.8438095  0.9307042
  2.5        0.9695775  0.8428571  0.9307042
  2.6        0.9701274  0.8419048  0.9323944
  2.7        0.9707445  0.8409524  0.9357746
  2.8        0.9711603  0.8409524  0.9374648

  2.9        0.9715895  0.8419048  0.9397183
  3.0        0.9719517  0.8428571  0.9419718
  3.1        0.9724346  0.8419048  0.9436620
  3.2        0.9727968  0.8419048  0.9453521
  3.3        0.9731992  0.8419048  0.9464789
  3.4        0.9735882  0.8409524  0.9481690
  3.5        0.9740040  0.8400000  0.9481690
  3.6        0.9741784  0.8400000  0.9498592
  3.7        0.9742723  0.8380952  0.9509859
  3.8        0.9743930  0.8371429  0.9521127
  3.9        0.9744869  0.8352381  0.9532394
  4.0        0.9745137  0.8352381  0.9538028

ROC was used to select the optimal model using the largest value.
The final value used for the model was threshold = 4.
```

TRAINING VALUES

Tuning parameters

Threshold = 4

ROC = 0.9745137

Sens = 0.8352381

Spec = 0.9538028

```
Confusion Matrix and Statistics

          Reference
Prediction  M  B
         M 39  6
         B  3 65

               Accuracy : 0.9204
                 95% CI : (0.8542, 0.9629)
    No Information Rate : 0.6283
    P-Value [Acc > NIR] : 9.656e-13

                  Kappa : 0.8319

 Mcnemar's Test P-Value : 0.505

            Sensitivity : 0.9286
            Specificity : 0.9155
         Pos Pred Value : 0.8667
         Neg Pred Value : 0.9559
             Prevalence : 0.3717
         Detection Rate : 0.3451
   Detection Prevalence : 0.3982
      Balanced Accuracy : 0.9220

       'Positive' Class : M
```

Sens = 0.9286

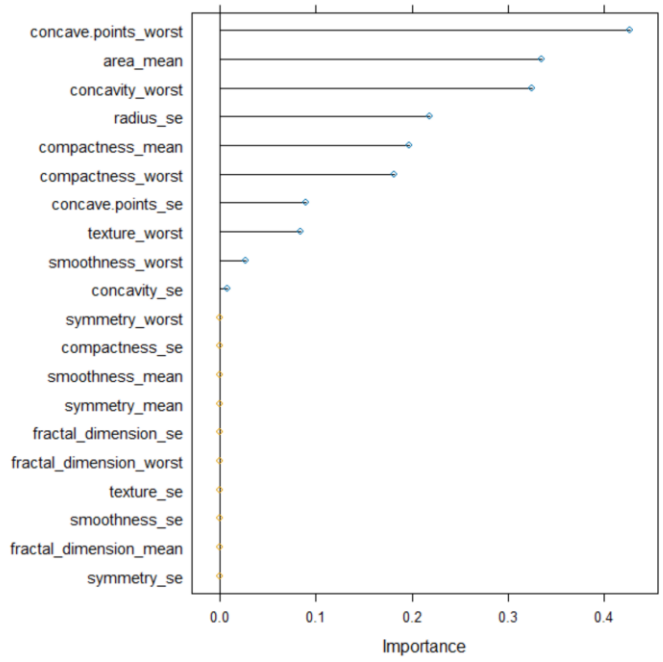
Spec = 0.9155

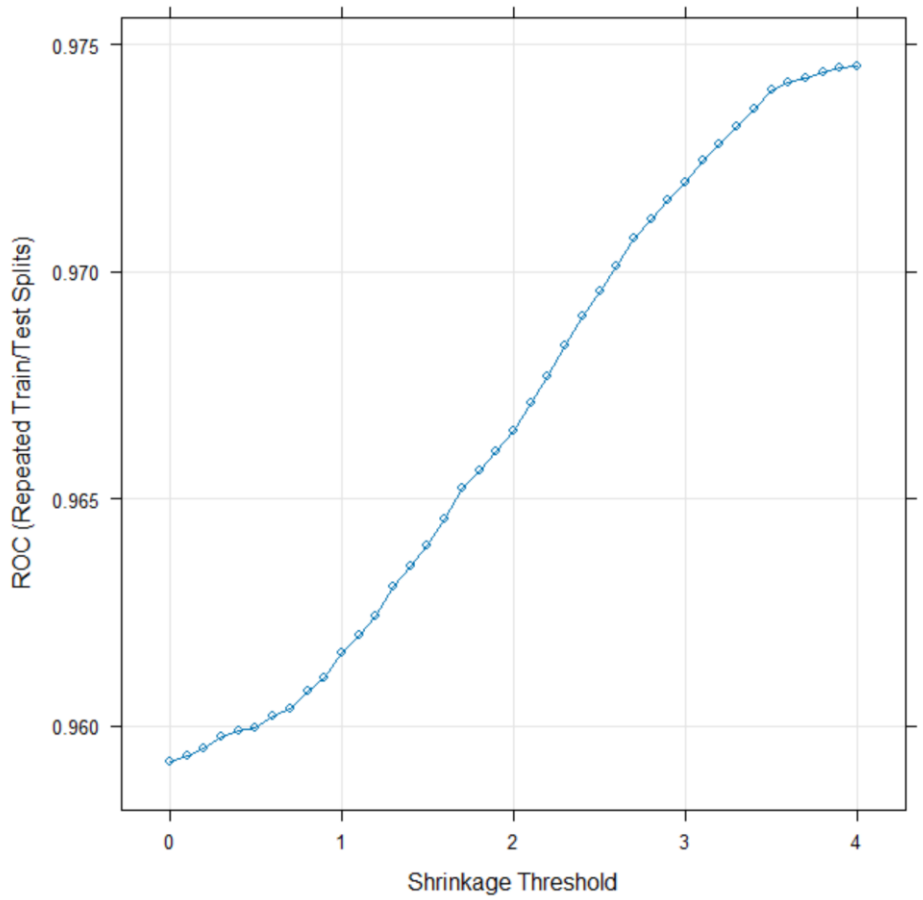
Accuracy = 0.9204

Kappa = 0.8319

pam variable importance

	Importance
concave.points_worst	0.42683
area_mean	0.33559
concavity_worst	0.32459
radius_se	0.21857
compactness_mean	0.19689
compactness_worst	0.18137
concave.points_se	0.08891
texture_worst	0.08367
smoothness_worst	0.02642
concavity_se	0.00750
compactness_se	0.00000
symmetry_se	0.00000
smoothness_se	0.00000
fractal_dimension_se	0.00000
fractal_dimension_worst	0.00000
texture_se	0.00000
fractal_dimension_mean	0.00000
symmetry_worst	0.00000
smoothness_mean	0.00000
symmetry_mean	0.00000





The maximum value of threshold was found to be 4 with the highest ROC value.

APPENDIX 2: NON-LINEAR CLASSIFICATION MODELS

a) NON LINEAR DISCRIMINANT ANALYSIS

1. MIXED DISCRIMINANT ANALYSIS

```
Mixture Discriminant Analysis

456 samples
 20 predictor
  2 classes: 'M', 'B'

No pre-processing
Resampling: Repeated Train/Test Splits Estimated (25 reps, 75%)
Summary of sample sizes: 343, 343, 343, 343, 343, 343, ...
Resampling results across tuning parameters:

  subclasses  ROC        Sens       Spec
   1          0.9948893  0.9371429  0.9814085
   2          0.9931858  0.9323810  0.9802817
   3          0.9927431  0.9361905  0.9723944
   4          0.9931053  0.9457143  0.9701408
   5          0.9911737  0.9466667  0.9667606
   6          0.9903421  0.9428571  0.9684507
   7          0.9918712  0.9447619  0.9650704
   8          0.9905433  0.9400000  0.9656338
   9          0.9909054  0.9428571  0.9667606
  10          0.9900872  0.9400000  0.9611268

ROC was used to select the optimal model using the largest value.
The final value used for the model was subclasses = 1.
```

TRAINING VALUES

Tuning parameter

Subclasses = 1

ROC = 0.9948893
Sens = 0.9371429
Spec = 0.9814085

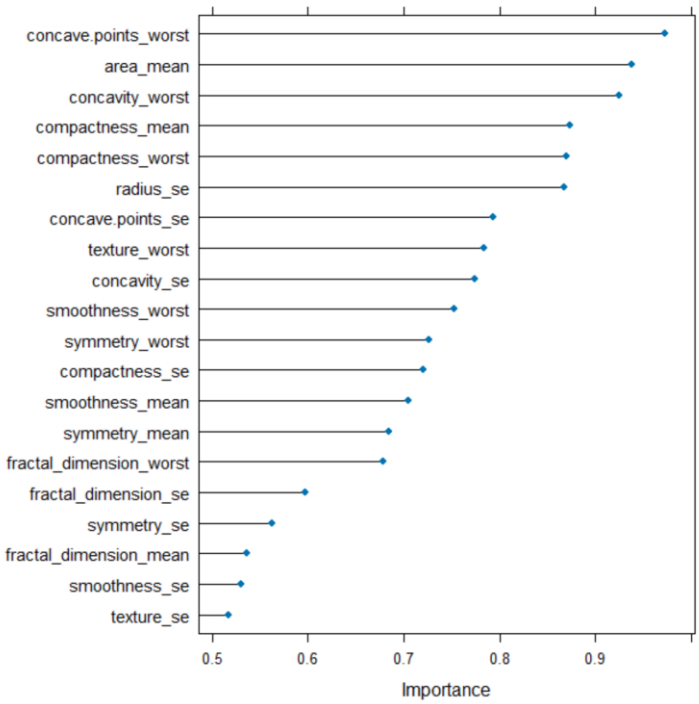
```
Confusion Matrix and Statistics

          Reference
Prediction  M  B
         M 39  2
         B  3 69

               Accuracy : 0.9558
                 95% CI : (0.8998, 0.9855)
    No Information Rate : 0.6283
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.9048

 Mcnemar's Test P-Value : 1

            Sensitivity : 0.9286
            Specificity : 0.9718
         Pos Pred Value : 0.9512
         Neg Pred Value : 0.9583
             Prevalence : 0.3717
         Detection Rate : 0.3451
   Detection Prevalence : 0.3628
      Balanced Accuracy : 0.9502

       'Positive' Class : M
```

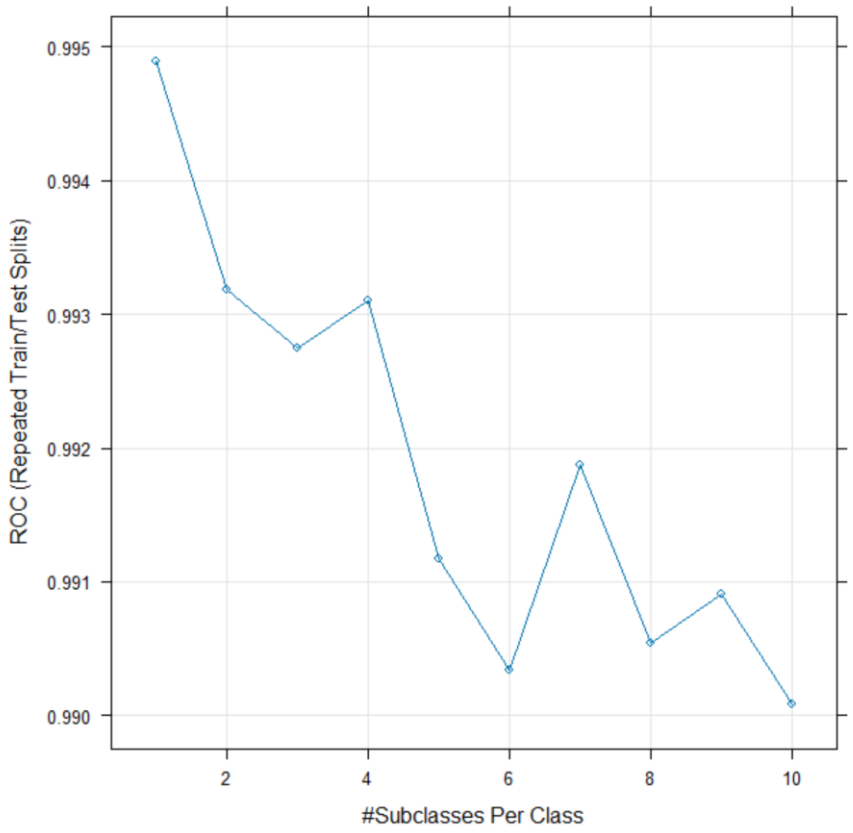
TESTING VALUES

Sens = 0.9286
Spec = 0.9718
Accuracy = 0.9558
Kappa = 0.9048

ROC curve variable importance

	Importance
concave.points_worst	0.9723
area_mean	0.9384
concavity_worst	0.9254
compactness_mean	0.8737
compactness_worst	0.8694
radius_se	0.8672
concave.points_se	0.7928
texture_worst	0.7834
concavity_se	0.7741
smoothness_worst	0.7525
symmetry_worst	0.7260
compactness_se	0.7202
smoothness_mean	0.7045
symmetry_mean	0.6847
fractal_dimension_worst	0.6782
fractal_dimension_se	0.5971
symmetry_se	0.5630
fractal_dimension_mean	0.5366
smoothness_se	0.5301
texture_se	0.5175





The highest subclass was 1 with the highest ROC values.

2. REGULARIZED DISCRIMINANT ANALYSIS

```
Regularized Discriminant Analysis

456 samples
 20 predictor
  2 classes: 'M', 'B'

No pre-processing
Resampling: Repeated Train/Test Splits Estimated (25 reps, 75%)
Summary of sample sizes: 343, 343, 343, 343, 343, 343, ...
Resampling results across tuning parameters:

  gamma  lambda  ROC        Sens       Spec
  0.1    0.01    0.9944467  0.9361905  0.9769014
  0.1    0.02    0.9945272  0.9361905  0.9769014
  0.1    0.03    0.9945942  0.9361905  0.9769014
  0.1    0.04    0.9947015  0.9371429  0.9769014
  0.1    0.05    0.9947686  0.9371429  0.9780282
  0.1    0.06    0.9948893  0.9371429  0.9780282
  0.1    0.07    0.9948893  0.9380952  0.9780282
  0.1    0.08    0.9949162  0.9380952  0.9780282
  0.1    0.09    0.9949430  0.9380952  0.9785915
  0.1    0.10    0.9950235  0.9380952  0.9785915
  0.2    0.01    0.9954259  0.9495238  0.9690141
  0.2    0.02    0.9954527  0.9504762  0.9690141
  0.2    0.03    0.9955064  0.9504762  0.9695775
  0.2    0.04    0.9954930  0.9514286  0.9695775
  0.2    0.05    0.9955198  0.9523810  0.9695775
  0.2    0.06    0.9955198  0.9533333  0.9695775
  0.2    0.07    0.9956003  0.9533333  0.9695775
  0.2    0.08    0.9956405  0.9552381  0.9695775
  0.2    0.09    0.9956942  0.9561905  0.9701408
  0.2    0.10    0.9956942  0.9580952  0.9701408
  0.3    0.01    0.9953186  0.9580952  0.9656338
  0.3    0.02    0.9953588  0.9571429  0.9656338
  0.3    0.03    0.9953454  0.9571429  0.9656338
  0.3    0.04    0.9953588  0.9580952  0.9661972
```

0.3	0.05	0.9953588	0.9580952	0.9661972
0.3	0.06	0.9953856	0.9580952	0.9661972
0.3	0.07	0.9953856	0.9580952	0.9656338
0.3	0.08	0.9954259	0.9580952	0.9650704
0.3	0.09	0.9954527	0.9580952	0.9650704
0.3	0.10	0.9954393	0.9580952	0.9645070
0.4	0.01	0.9948625	0.9609524	0.9611268
0.4	0.02	0.9948625	0.9609524	0.9611268
0.4	0.03	0.9948759	0.9609524	0.9605634
0.4	0.04	0.9948893	0.9609524	0.9611268
0.4	0.05	0.9948893	0.9609524	0.9611268
0.4	0.06	0.9949027	0.9609524	0.9611268
0.4	0.07	0.9949430	0.9619048	0.9611268
0.4	0.08	0.9949564	0.9619048	0.9611268
0.4	0.09	0.9949966	0.9619048	0.9611268
0.4	0.10	0.9950101	0.9619048	0.9611268
0.5	0.01	0.9941918	0.9580952	0.9588732
0.5	0.02	0.9941918	0.9580952	0.9588732
0.5	0.03	0.9942186	0.9580952	0.9588732
0.5	0.04	0.9942052	0.9580952	0.9588732
0.5	0.05	0.9941918	0.9580952	0.9588732
0.5	0.06	0.9942052	0.9580952	0.9588732
0.5	0.07	0.9942052	0.9571429	0.9588732
0.5	0.08	0.9942052	0.9571429	0.9588732
0.5	0.09	0.9942052	0.9571429	0.9588732
0.5	0.10	0.9942321	0.9571429	0.9588732
0.6	0.01	0.9931992	0.9523810	0.9566197
0.6	0.02	0.9931590	0.9523810	0.9566197
0.6	0.03	0.9931724	0.9523810	0.9566197
0.6	0.04	0.9931724	0.9523810	0.9566197
0.6	0.05	0.9931590	0.9514286	0.9566197
0.6	0.06	0.9931455	0.9514286	0.9566197
0.6	0.07	0.9931455	0.9514286	0.9566197
0.6	0.08	0.9931724	0.9514286	0.9566197
0.6	0.09	0.9931858	0.9514286	0.9560563
0.6	0.10	0.9931992	0.9514286	0.9566197
0.7	0.01	0.9918042	0.9419048	0.9526761
0.7	0.02	0.9917907	0.9419048	0.9526761
0.7	0.03	0.9917773	0.9419048	0.9526761
0.7	0.04	0.9917639	0.9419048	0.9526761
0.7	0.05	0.9917773	0.9419048	0.9526761
0.7	0.06	0.9917773	0.9419048	0.9526761
0.7	0.07	0.9917773	0.9409524	0.9526761
0.7	0.08	0.9917639	0.9409524	0.9526761
0.7	0.09	0.9917639	0.9409524	0.9521127
0.7	0.10	0.9917505	0.9390476	0.9521127
0.8	0.01	0.9892019	0.9323810	0.9453521
0.8	0.02	0.9892153	0.9323810	0.9453521
0.8	0.03	0.9892153	0.9323810	0.9453521
0.8	0.04	0.9892153	0.9323810	0.9453521
0.8	0.05	0.9892019	0.9323810	0.9453521
0.8	0.06	0.9891885	0.9323810	0.9453521
0.8	0.07	0.9891885	0.9333333	0.9453521
0.8	0.08	0.9891616	0.9333333	0.9453521
0.8	0.09	0.9891616	0.9333333	0.9453521
0.8	0.10	0.9891348	0.9333333	0.9453521
0.9	0.01	0.9836754	0.9047619	0.9301408
0.9	0.02	0.9836620	0.9047619	0.9301408
0.9	0.03	0.9836620	0.9038095	0.9301408
0.9	0.04	0.9836217	0.9038095	0.9295775
0.9	0.05	0.9835949	0.9038095	0.9295775
0.9	0.06	0.9836083	0.9038095	0.9290141
0.9	0.07	0.9835681	0.9038095	0.9284507
0.9	0.08	0.9835412	0.9038095	0.9278873
0.9	0.09	0.9835144	0.9038095	0.9278873
0.9	0.10	0.9835010	0.9038095	0.9278873
1.0	0.01	0.9692019	0.8761905	0.9064789
1.0	0.02	0.9692019	0.8761905	0.9059155
1.0	0.03	0.9692019	0.8771429	0.9059155
1.0	0.04	0.9692153	0.8771429	0.9059155
1.0	0.05	0.9692019	0.8780952	0.9059155
1.0	0.06	0.9692019	0.8780952	0.9053521

```
  1.0    0.07    0.9691885  0.8780952  0.9047887
  1.0    0.08    0.9692019  0.8780952  0.9047887
  1.0    0.09    0.9692019  0.8800000  0.9047887
  1.0    0.10    0.9692019  0.8809524  0.9042254

ROC was used to select the optimal model using the largest value.
The final values used for the model were gamma = 0.2 and lambda = 0.1.
```

Tuning Parameters

Gamma = 0.2

Lambda = 0.1

ROC = 0.9956942

Sens = 0.9580952

Spec = 0.9701408

```
Confusion Matrix and Statistics

          Reference
Prediction  M  B
         M 39  4
         B  3 67

               Accuracy : 0.9381
                 95% CI : (0.8765, 0.9747)
    No Information Rate : 0.6283
    P-Value [Acc > NIR] : 1.718e-14

                  Kappa : 0.868

 Mcnemar's Test P-Value : 1

            Sensitivity : 0.9286
            Specificity : 0.9437
         Pos Pred Value : 0.9070
         Neg Pred Value : 0.9571
             Prevalence : 0.3717
         Detection Rate : 0.3451
   Detection Prevalence : 0.3805
      Balanced Accuracy : 0.9361

       'Positive' Class : M
```

TESTING VALUES

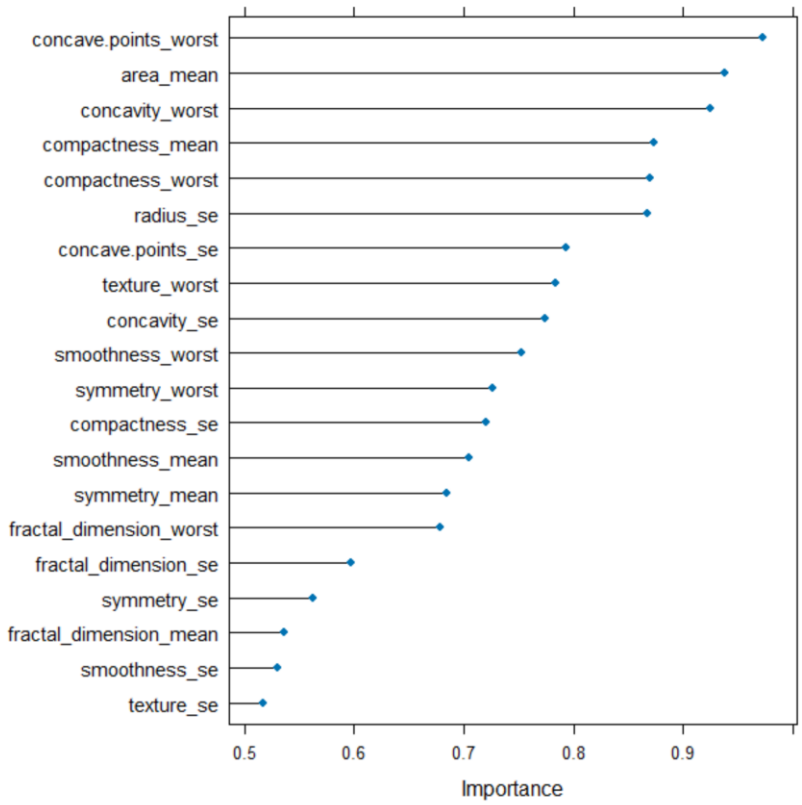
Sens = 0.9286

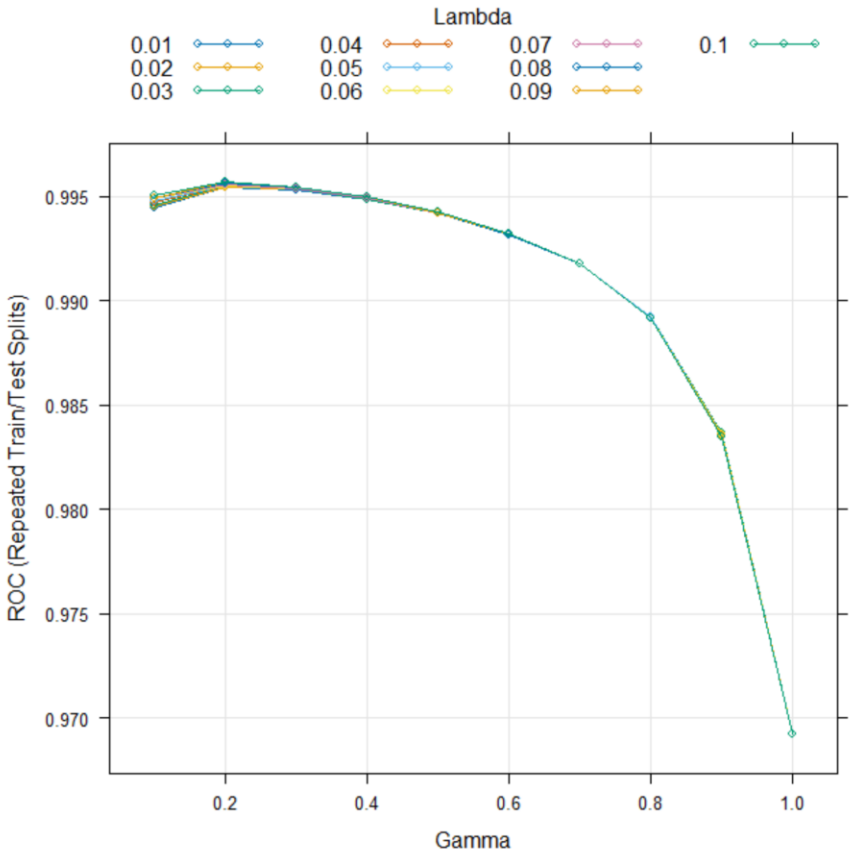
Spec = 0.9437

Accuracy = 0.9381

Kappa = 0.868

```
ROC curve variable importance

                         Importance
concave.points_worst         0.9723
area_mean                    0.9384
concavity_worst              0.9254
compactness_mean             0.8737
compactness_worst            0.8694
radius_se                    0.8672
concave.points_se            0.7928
texture_worst                0.7834
concavity_se                 0.7741
smoothness_worst             0.7525
symmetry_worst               0.7260
compactness_se               0.7202
smoothness_mean              0.7045
symmetry_mean                0.6847
fractal_dimension_worst      0.6782
fractal_dimension_se         0.5971
symmetry_se                  0.5630
fractal_dimension_mean       0.5366
smoothness_se                0.5301
texture_se                   0.5175
```




Lambda
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08
0.09
0.1
ROC (Repeated Train/Test Splits)
0.995
0.990
0.985
0.980
0.975
0.970
0.2
0.4
0.6
0.8
1.0
Gamma

b) NEURAL NETWORKS

```
Neural Network

456 samples
 20 predictor
  2 classes: 'M', 'B'

Pre-processing: centered (20), scaled (20), spatial sign transformation (20)
Resampling: Repeated Train/Test Splits Estimated (25 reps, 75%)
Summary of sample sizes: 343, 343, 343, 343, 343, 343, ...
Resampling results across tuning parameters:

  size  decay  ROC        Sens       Spec
  1     0.0    0.9675788  0.9533333  0.9554930
  1     0.1    0.9963917  0.9590476  0.9678873
  1     1.0    0.9942991  0.9295238  0.9752113
  1     2.0    0.9915627  0.9028571  0.9802817
  2     0.0    0.9657814  0.9523810  0.9633803
  2     0.1    0.9964588  0.9590476  0.9678873
  2     1.0    0.9941382  0.9304762  0.9752113
  2     2.0    0.9914152  0.9076190  0.9774648
  3     0.0    0.9649095  0.9457143  0.9588732
  3     0.1    0.9964856  0.9590476  0.9684507
  3     1.0    0.9942186  0.9304762  0.9746479
  3     2.0    0.9914420  0.9085714  0.9752113
  4     0.0    0.9742924  0.9580952  0.9639437
  4     0.1    0.9965258  0.9590476  0.9684507
  4     1.0    0.9941382  0.9323810  0.9746479
  4     2.0    0.9914688  0.9085714  0.9752113
  5     0.0    0.9607378  0.9390476  0.9577465
  5     0.1    0.9964856  0.9590476  0.9684507
  5     1.0    0.9940577  0.9314286  0.9752113
  5     2.0    0.9913749  0.9085714  0.9735211
  6     0.0    0.9722803  0.9523810  0.9605634
  6     0.1    0.9965124  0.9590476  0.9684507

  6     1.0    0.9940309  0.9304762  0.9752113
  6     2.0    0.9913481  0.9085714  0.9729577
  7     0.0    0.9743394  0.9523810  0.9583099
  7     0.1    0.9964588  0.9600000  0.9684507
  7     1.0    0.9940309  0.9295238  0.9746479
  7     2.0    0.9912542  0.9085714  0.9723944
  8     0.0    0.9725822  0.9571429  0.9600000
  8     0.1    0.9965258  0.9590476  0.9684507
  8     1.0    0.9940040  0.9304762  0.9746479
  8     2.0    0.9912274  0.9085714  0.9723944

ROC was used to select the optimal model using the largest value.
The final values used for the model were size = 4 and decay = 0.1.
```

Tuning Parameters

Size = 4

Decay = 0.1

ROC = 0.9965258

Sens = 0.9590476

Spec = 0.9684507

```
Confusion Matrix and Statistics

          Reference
Prediction  M  B
         M 40  2
         B  2 69

               Accuracy : 0.9646
                 95% CI : (0.9118, 0.9903)
    No Information Rate : 0.6283
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.9242

 Mcnemar's Test P-Value : 1

            Sensitivity : 0.9524
            Specificity : 0.9718
         Pos Pred Value : 0.9524
         Neg Pred Value : 0.9718
             Prevalence : 0.3717
         Detection Rate : 0.3540
   Detection Prevalence : 0.3717
      Balanced Accuracy : 0.9621

       'Positive' Class : M
```

TESTING VALUES

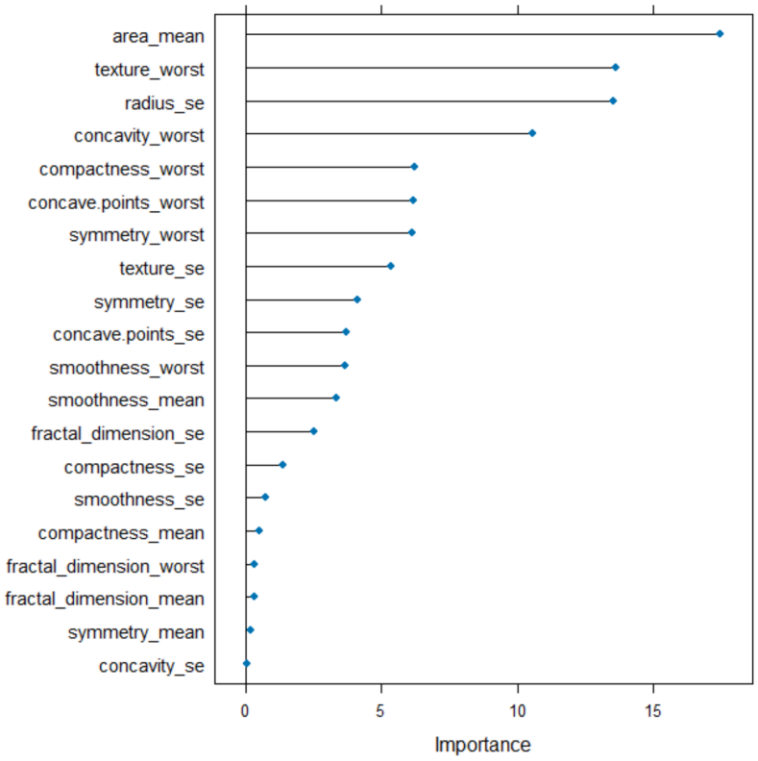
Sens = 0.9524

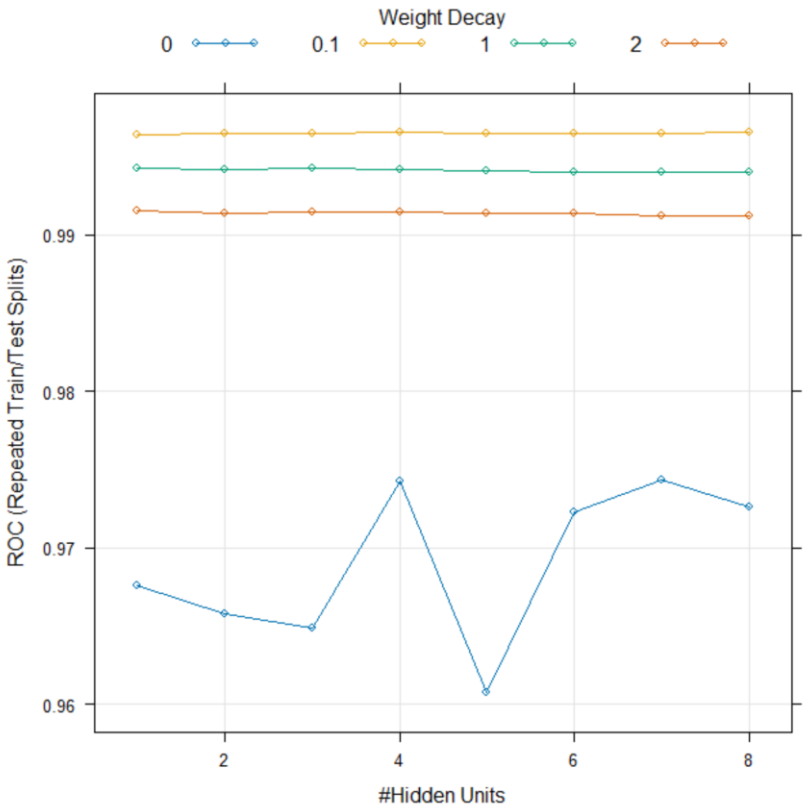
Spec = 0.9718

Accuracy = 0.9646

Kappa = 0.9242

```
nnet variable importance

                        Overall
area_mean               17.4555
texture_worst           13.6372
radius_se               13.5286
concavity_worst         10.5796
compactness_worst        6.2039
concave.points_worst     6.1958
symmetry_worst           6.1129
texture_se               5.3557
symmetry_se              4.1278
concave.points_se        3.7292
smoothness_worst         3.6517
smoothness_mean          3.3333
fractal_dimension_se     2.5153
compactness_se           1.3634
smoothness_se            0.7296
compactness_mean         0.5247
fractal_dimension_worst  0.3540
fractal_dimension_mean   0.3269
symmetry_mean            0.2057
concavity_se             0.0692
```





Where size is 4 and decay is 0.1 had the highest values for ROC.

c) FLEXIBLE DISCRIMINANT ANALYSIS

```
Flexible Discriminant Analysis

456 samples
 20 predictor
  2 classes: 'M', 'B'

Pre-processing: centered (20), scaled (20), spatial sign transformation (20)
Resampling: Repeated Train/Test Splits Estimated (25 reps, 75%)
Summary of sample sizes: 343, 343, 343, 343, 343, 343, ...
Resampling results across tuning parameters:

  degree  nprune  ROC        Sens       Spec
  1        2      0.9628437  0.8914286  0.9419718
  1        3      0.9640241  0.8819048  0.9425352
  1        4      0.9753722  0.8980952  0.9487324
  1        5      0.9817706  0.9095238  0.9560563
  1        6      0.9809524  0.9047619  0.9549296
  1        7      0.9814085  0.9047619  0.9543662
  1        8      0.9821999  0.9000000  0.9560563
  1        9      0.9827632  0.9028571  0.9538028
  1       10      0.9832730  0.9085714  0.9554930
  1       11      0.9833937  0.9095238  0.9549296
  1       12      0.9839973  0.9066667  0.9571831
  1       13      0.9843327  0.9104762  0.9554930
  1       14      0.9840912  0.9104762  0.9577465
  1       15      0.9844534  0.9114286  0.9554930
  1       16      0.9842119  0.9085714  0.9571831
  1       17      0.9842119  0.9104762  0.9566197
  1       18      0.9843327  0.9095238  0.9571831
  1       19      0.9845875  0.9085714  0.9571831
  1       20      0.9847485  0.9076190  0.9571831
  1       21      0.9847619  0.9095238  0.9571831
  1       22      0.9847619  0.9095238  0.9571831
  1       23      0.9847619  0.9076190  0.9571831

  1       24      0.9847619  0.9076190  0.9571831
  1       25      0.9847619  0.9076190  0.9571831
  1       26      0.9847619  0.9076190  0.9571831
  1       27      0.9847619  0.9076190  0.9571831
  1       28      0.9847619  0.9076190  0.9571831
  1       29      0.9847619  0.9076190  0.9571831
  1       30      0.9847619  0.9076190  0.9571831
  1       31      0.9847619  0.9076190  0.9571831
  1       32      0.9847619  0.9076190  0.9571831
  1       33      0.9847619  0.9076190  0.9571831
  1       34      0.9847619  0.9076190  0.9571831
  1       35      0.9847619  0.9076190  0.9571831
  1       36      0.9847619  0.9076190  0.9571831
  1       37      0.9847619  0.9076190  0.9571831
  1       38      0.9847619  0.9076190  0.9571831
  2        2      0.9628437  0.8914286  0.9419718
  2        3      0.9705164  0.8733333  0.9532394
  2        4      0.9714286  0.8933333  0.9481690
  2        5      0.9763380  0.9104762  0.9509859
  2        6      0.9800134  0.9057143  0.9600000
  2        7      0.9813816  0.9095238  0.9628169
  2        8      0.9830986  0.9076190  0.9639437
  2        9      0.9834742  0.9095238  0.9616901
  2       10      0.9839168  0.9047619  0.9656338
  2       11      0.9839839  0.9028571  0.9695775
  2       12      0.9832596  0.9009524  0.9678873
  2       13      0.9841180  0.9028571  0.9678873
  2       14      0.9832998  0.9028571  0.9667606
  2       15      0.9831120  0.9038095  0.9667606
  2       16      0.9835949  0.9028571  0.9667606
  2       17      0.9834608  0.9019048  0.9656338
  2       18      0.9832797  0.9028571  0.9684507
  2       19      0.9843528  0.9019048  0.9678873
  2       20      0.9843796  0.9009524  0.9684507
```

```
  2       21      0.9849162  0.9028571  0.9684507
  2       22      0.9850905  0.9038095  0.9678873
  2       23      0.9858015  0.9047619  0.9678873
  2       24      0.9861368  0.9057143  0.9678873
  2       25      0.9861100  0.9057143  0.9678873
  2       26      0.9861100  0.9057143  0.9678873
  2       27      0.9861100  0.9057143  0.9678873
  2       28      0.9861100  0.9057143  0.9678873
  2       29      0.9861100  0.9057143  0.9678873
  2       30      0.9861100  0.9057143  0.9678873
  2       31      0.9861100  0.9057143  0.9678873
  2       32      0.9861100  0.9057143  0.9678873
  2       33      0.9861100  0.9057143  0.9678873
  2       34      0.9861100  0.9057143  0.9678873
  2       35      0.9861100  0.9057143  0.9678873
  2       36      0.9861100  0.9057143  0.9678873
  2       37      0.9861100  0.9057143  0.9678873
  2       38      0.9861100  0.9057143  0.9678873

ROC was used to select the optimal model using the largest value.
The final values used for the model were degree = 2 and nprune = 24.
```

Tuning Parameters

Degree = 2

Nprune = 24

ROC = 0.9861368

Sens = 0.9057143

Spec = 0.9678873

```
Confusion Matrix and Statistics

          Reference
Prediction  M  B
         M 39  1
         B  3 70

               Accuracy : 0.9646
                 95% CI : (0.9118, 0.9903)
    No Information Rate : 0.6283
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.9235

 Mcnemar's Test P-Value : 0.6171

            Sensitivity : 0.9286
            Specificity : 0.9859
         Pos Pred Value : 0.9750
         Neg Pred Value : 0.9589
             Prevalence : 0.3717
         Detection Rate : 0.3451
   Detection Prevalence : 0.3540
      Balanced Accuracy : 0.9572

       'Positive' Class : M
```

TESTING VALUES

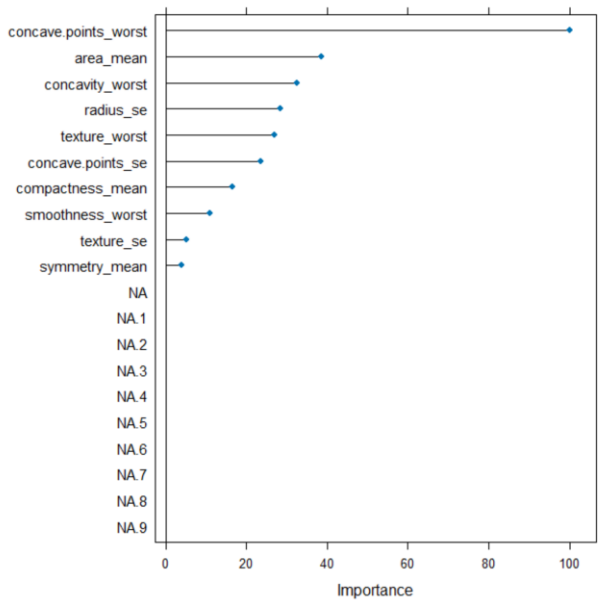
Sens = 0.9286

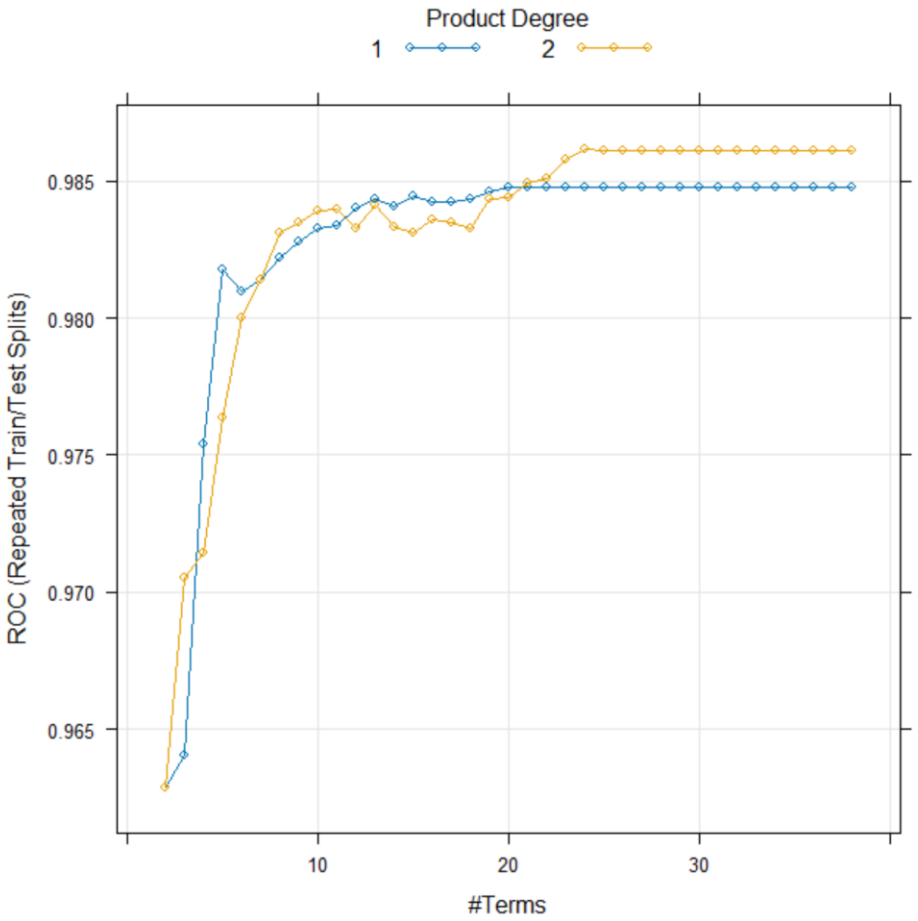
Spec = 0.9859

Accuracy = 0.9646

Kappa = 0.9235

```
fda variable importance

                       Overall
concave.points_worst 100.000
area_mean             38.465
concavity_worst       32.518
radius_se             28.345
texture_worst         26.918
concave.points_se     23.551
compactness_mean      16.523
smoothness_worst      10.859
texture_se             5.239
symmetry_mean          4.022
```





Where the degree =2 had the highest values for ROC.

d) SUPPORT VECTOR MACHINE

```
Support Vector Machines with Radial Basis Function Kernel

456 samples
 20 predictor
  2 classes: 'M', 'B'

Pre-processing: centered (20), scaled (20)
Resampling: Repeated Train/Test Splits Estimated (25 reps, 75%)
Summary of sample sizes: 343, 343, 343, 343, 343, 343, ...
Resampling results across tuning parameters:

  C        ROC        Sens       Spec
   0.0625  0.9816499  0.9552381  0.8602817
   0.1250  0.9816499  0.9552381  0.8614085
   0.2500  0.9877666  0.8980952  0.9470423
   0.5000  0.9901006  0.9142857  0.9532394
   1.0000  0.9918712  0.9276190  0.9695775
   2.0000  0.9939772  0.9333333  0.9774648
   4.0000  0.9954527  0.9447619  0.9763380
   8.0000  0.9963917  0.9552381  0.9763380
  16.0000  0.9964051  0.9590476  0.9791549
  32.0000  0.9965795  0.9600000  0.9746479
  64.0000  0.9960966  0.9609524  0.9752113

Tuning parameter 'sigma' was held constant at a value of 0.003001256
ROC was used to select the optimal model using the largest value.
The final values used for the model were sigma = 0.003001256 and C = 32.
```

TRAINING VALUES

Tuning parameters

Sigma = 0.003001256
C = 32

ROC = 0.9965795

Sens = 0.96

Spec = 0.9746479

```
Confusion Matrix and Statistics

          Reference
Prediction  M  B
         M 40  2
         B  2 69

               Accuracy : 0.9646
                 95% CI : (0.9118, 0.9903)
    No Information Rate : 0.6283
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.9242

 Mcnemar's Test P-Value : 1

            Sensitivity : 0.9524
            Specificity : 0.9718
         Pos Pred Value : 0.9524
         Neg Pred Value : 0.9718
             Prevalence : 0.3717
         Detection Rate : 0.3540
   Detection Prevalence : 0.3717
      Balanced Accuracy : 0.9621

       'Positive' Class : M
```

TESTING VALUES

Sens = 0.9524

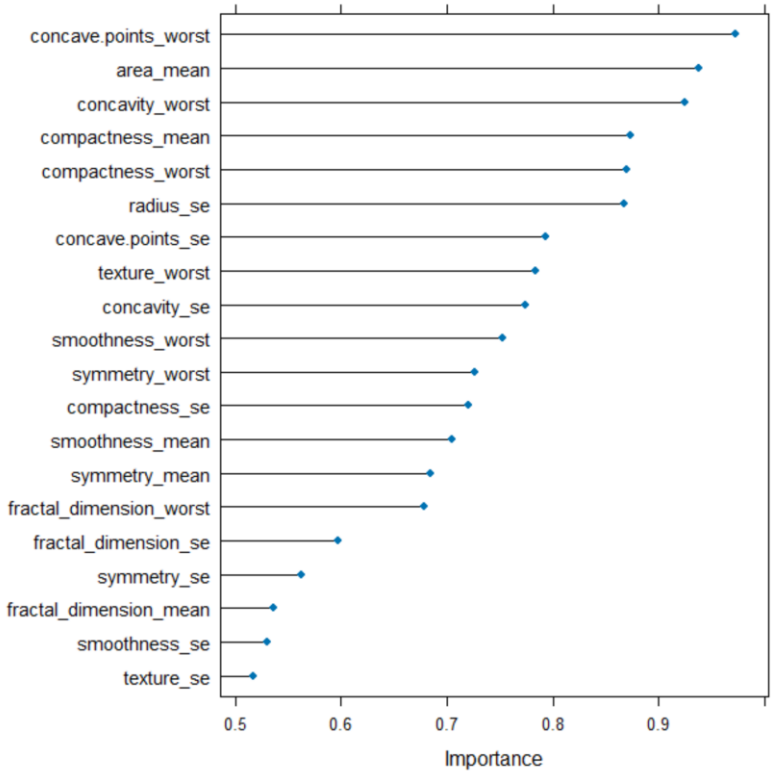
Spec = 0.9718

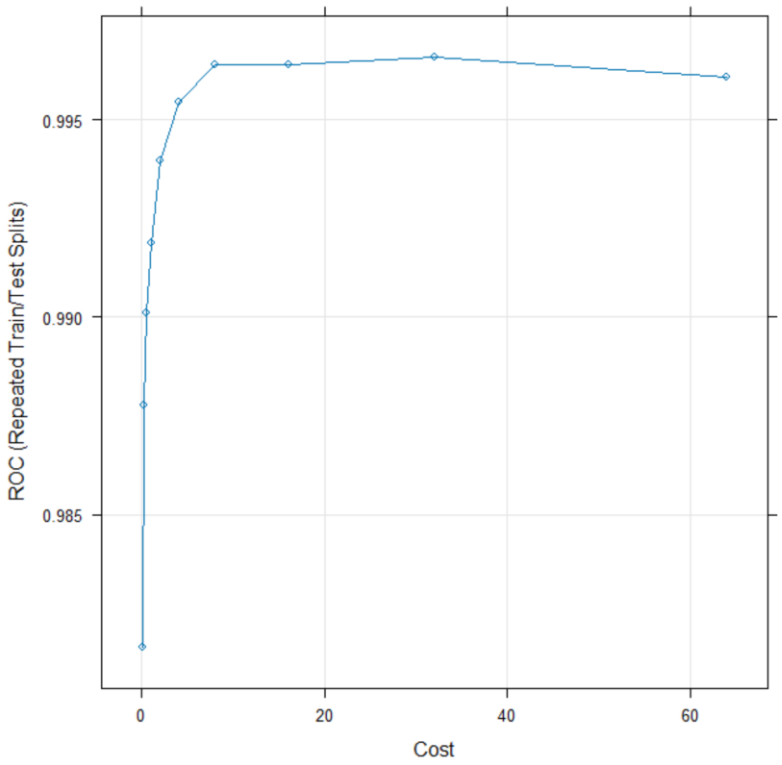
Accuracy = 0.9646

Kappa = 0.9242

ROC curve variable importance

	Importance
concave.points_worst	0.9723
area_mean	0.9384
concavity_worst	0.9254
compactness_mean	0.8737
compactness_worst	0.8694
radius_se	0.8672
concave.points_se	0.7928
texture_worst	0.7834
concavity_se	0.7741
smoothness_worst	0.7525
symmetry_worst	0.7260
compactness_se	0.7202
smoothness_mean	0.7045
symmetry_mean	0.6847
fractal_dimension_worst	0.6782
fractal_dimension_se	0.5971
symmetry_se	0.5630
fractal_dimension_mean	0.5366
smoothness_se	0.5301
texture_se	0.5175





Where Cost is 32 had, the highest values of ROC.

e) K NEAREST NEIGHBORS

```
k-Nearest Neighbors

456 samples
 20 predictor
  2 classes: 'M', 'B'

Pre-processing: centered (20), scaled (20)
Resampling: Repeated Train/Test Splits Estimated (25 reps, 75%)
Summary of sample sizes: 343, 343, 343, 343, 343, 343, ...
Resampling results across tuning parameters:

  k   ROC        Sens       Spec
   1  0.9398189  0.9342857  0.9453521
   2  0.9712274  0.9314286  0.9470423
   3  0.9765594  0.9466667  0.9600000
   4  0.9819920  0.9428571  0.9509859
   5  0.9844131  0.9380952  0.9600000
   6  0.9855600  0.9409524  0.9594366
   7  0.9861234  0.9342857  0.9594366
   8  0.9857814  0.9342857  0.9549296
   9  0.9853454  0.9361905  0.9549296
  10  0.9852314  0.9352381  0.9549296
  11  0.9853856  0.9333333  0.9538028
  12  0.9852918  0.9342857  0.9504225
  13  0.9858015  0.9352381  0.9521127
  14  0.9855332  0.9333333  0.9487324
  15  0.9857612  0.9342857  0.9487324
  16  0.9851979  0.9295238  0.9459155
  17  0.9846948  0.9276190  0.9453521
  18  0.9846546  0.9276190  0.9436620
  19  0.9846479  0.9228571  0.9408451
  20  0.9842254  0.9247619  0.9380282
  21  0.9842119  0.9209524  0.9380282
  22  0.9843192  0.9219048  0.9385915
  23  0.9840979  0.9219048  0.9363380
  24  0.9839772  0.9190476  0.9374648
  25  0.9836620  0.9209524  0.9369014
  26  0.9833199  0.9200000  0.9363380
  27  0.9831858  0.9200000  0.9352113
  28  0.9833535  0.9219048  0.9329577
  29  0.9830449  0.9219048  0.9318310
  30  0.9830047  0.9209524  0.9346479
  31  0.9827834  0.9209524  0.9323944
  32  0.9827096  0.9200000  0.9329577
  33  0.9831388  0.9190476  0.9301408
  34  0.9832327  0.9200000  0.9301408
  35  0.9832529  0.9200000  0.9295775
  36  0.9833266  0.9190476  0.9290141
  37  0.9832529  0.9180952  0.9278873
  38  0.9831120  0.9190476  0.9267606
  39  0.9829309  0.9152381  0.9273239
  40  0.9828102  0.9161905  0.9261972
  41  0.9825553  0.9180952  0.9267606
  42  0.9823474  0.9180952  0.9245070
  43  0.9821462  0.9171429  0.9250704
  44  0.9819048  0.9171429  0.9256338
  45  0.9817103  0.9152381  0.9239437
  46  0.9816298  0.9171429  0.9228169
  47  0.9811670  0.9152381  0.9216901
  48  0.9810329  0.9161905  0.9200000
  49  0.9810262  0.9161905  0.9200000
  50  0.9810127  0.9161905  0.9216901

ROC was used to select the optimal model using the largest value.
The final value used for the model was k = 7.
```

Tuning parameter

K = 7

ROC = 0.9861234

Sens = 0.9342857

Spec = 0.9594366

```
Confusion Matrix and Statistics

          Reference
Prediction  M  B
         M 38  5
         B  4 66

               Accuracy : 0.9204
                 95% CI : (0.8542, 0.9629)
    No Information Rate : 0.6283
    P-Value [Acc > NIR] : 9.656e-13

                  Kappa : 0.8303

 Mcnemar's Test P-Value : 1

            Sensitivity : 0.9048
            Specificity : 0.9296
         Pos Pred Value : 0.8837
         Neg Pred Value : 0.9429
             Prevalence : 0.3717
         Detection Rate : 0.3363
   Detection Prevalence : 0.3805
      Balanced Accuracy : 0.9172

       'Positive' Class : M
```

TESTING VALUES

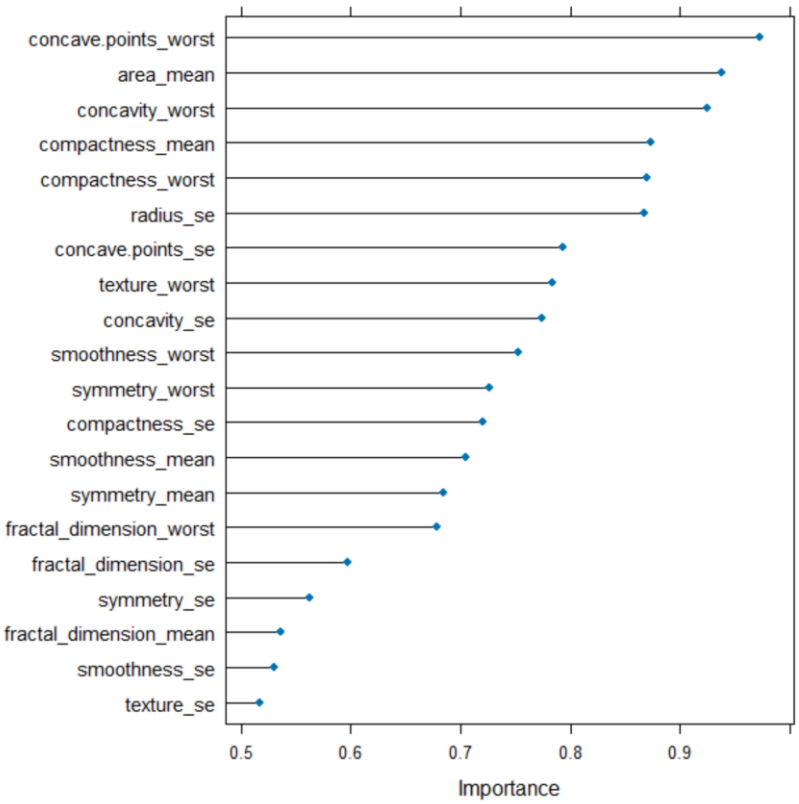
Sens = 0.9048

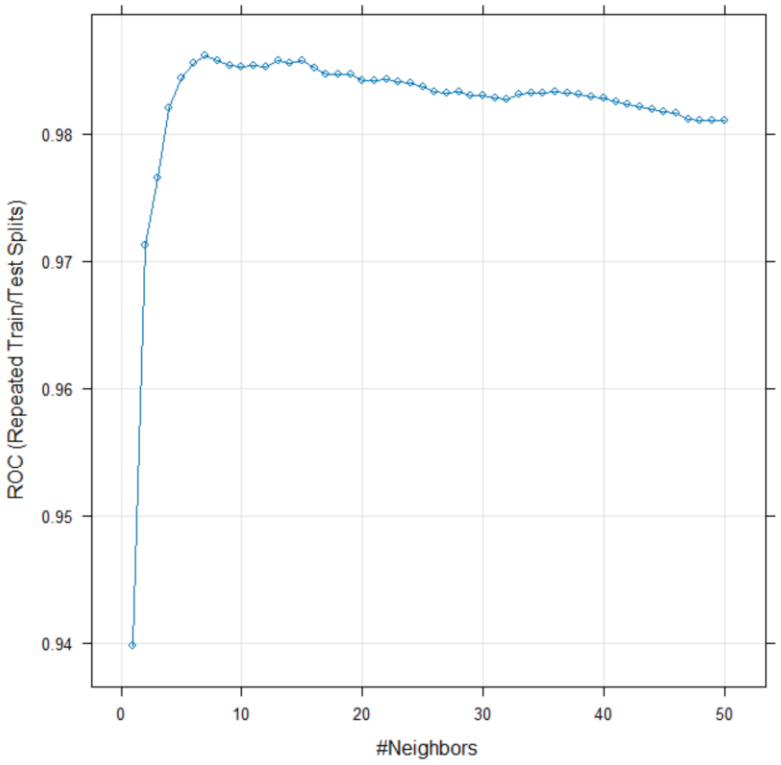
Spec = 0.9296

Accuracy = 0.9204

Kappa = 0.8303

```
ROC curve variable importance

                          Importance
concave.points_worst          0.9723
area_mean                     0.9384
concavity_worst               0.9254
compactness_mean              0.8737
compactness_worst             0.8694
radius_se                     0.8672
concave.points_se             0.7928
texture_worst                 0.7834
concavity_se                  0.7741
smoothness_worst              0.7525
symmetry_worst                0.7260
compactness_se                0.7202
smoothness_mean               0.7045
symmetry_mean                 0.6847
fractal_dimension_worst       0.6782
fractal_dimension_se          0.5971
symmetry_se                   0.5630
fractal_dimension_mean        0.5366
smoothness_se                 0.5301
texture_se                    0.5175
```





Where k=7 had the highest values of ROC.

f) NAÏVE BAYES

```
Naive Bayes

456 samples
 20 predictor
  2 classes: 'M', 'B'

Pre-processing: centered (20), scaled (20)
Resampling: Repeated Train/Test Splits Estimated (25 reps, 75%)
Summary of sample sizes: 343, 343, 343, 343, 343, 343, ...
Resampling results:

  ROC        Sens      Spec
  0.9776123  0.912381  0.9092958

Tuning parameter 'fL' was held constant at a value of 2
Tuning parameter 'usekernel' was held
 constant at a value of TRUE
Tuning parameter 'adjust' was held constant at a value of TRUE
```

Tuning Parameters

No tuning parameters

```
ROC = 0.9776123


Sens = 0.912381


Spec = 0.9092958
```

There are no tuning parameters.

```
Confusion Matrix and Statistics

          Reference
Prediction  M  B
         M 35 10
         B  7 61

               Accuracy : 0.8496
                 95% CI : (0.7701, 0.9099)
    No Information Rate : 0.6283
    P-Value [Acc > NIR] : 1.849e-07

                  Kappa : 0.6825

 Mcnemar's Test P-Value : 0.6276

            Sensitivity : 0.8333
            Specificity : 0.8592
         Pos Pred Value : 0.7778
         Neg Pred Value : 0.8971
             Prevalence : 0.3717
         Detection Rate : 0.3097
   Detection Prevalence : 0.3982
      Balanced Accuracy : 0.8462

       'Positive' Class : M
```

Sens = 0.8333

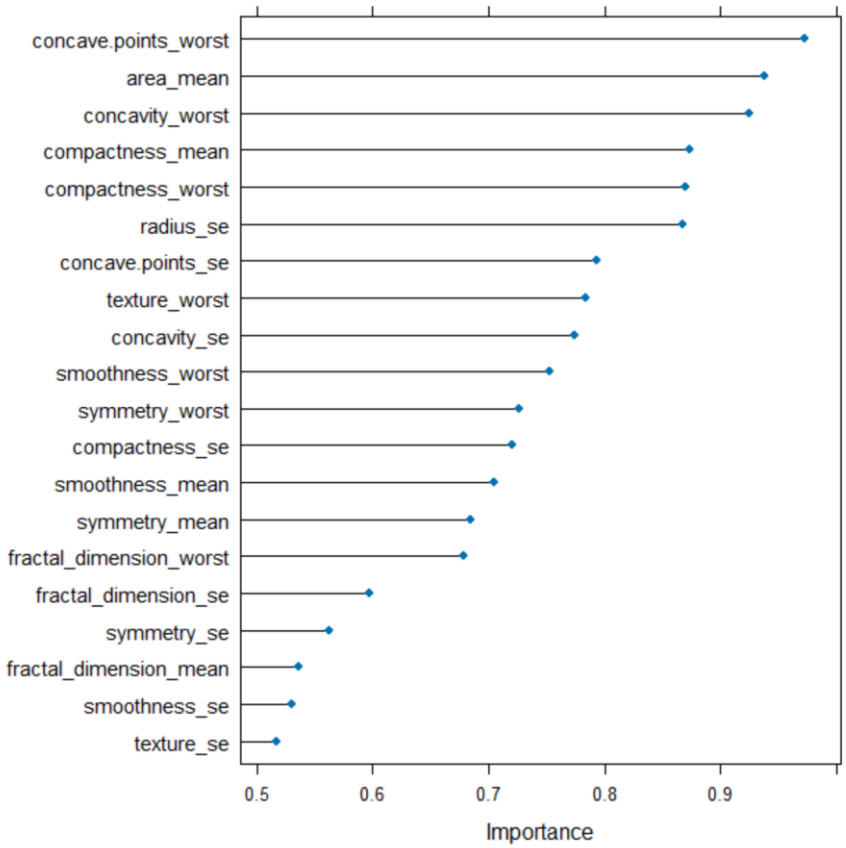
Spec = 0.8592

Accuracy = 0.8496

Kappa = 0.6825

ROC curve variable importance

	Importance
concave.points_worst	0.9723
area_mean	0.9384
concavity_worst	0.9254
compactness_mean	0.8737
compactness_worst	0.8694
radius_se	0.8672
concave.points_se	0.7928
texture_worst	0.7834
concavity_se	0.7741
smoothness_worst	0.7525
symmetry_worst	0.7260
compactness_se	0.7202
smoothness_mean	0.7045
symmetry_mean	0.6847
fractal_dimension_worst	0.6782
fractal_dimension_se	0.5971
symmetry_se	0.5630
fractal_dimension_mean	0.5366
smoothness_se	0.5301
texture_se	0.5175



REFERENCES

[1] https://www.cdc.gov/cancer/breast/basic_info/index.htm

[2] https://www.mayoclinic.org/diseases-conditions/breast-cancer/symptoms-causes/syc-20352470#:~:text=Causes,other%20parts%20of%20your%20body

[3] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8428369/#:~:text=Breast%20cancer%20is%20currently%20one,in%20transitioning%20countries

RCODE

```
#install.packages(c("glmnet", "pamr", "rms", "sparseLDA", "subselect"))
library(mda)

#install.packages(c("earth", "Formula", "plotmo", "plotrix", "TeachingDemos"))
library(earth)

#install.packages(c("glmnet", "pamr", "rms", "sparseLDA", "subselect"))
#install.packages("caret", dependencies = TRUE)
#install.packages("pROC")
library(pROC)
library(AppliedPredictiveModeling)
library(MASS)
library(sparseLDA)
library(caret)
library(nnet) # for multinom function



df=read.csv("C:/Users/Mykey/OneDrive/Desktop/MTU/SEM 3/MA5790/GRP
PROJECT/data.csv")
df
```

```
str(df)

###removed x column with NA values


df$X <- NULL

df

############################################################plot histogram distribution



#Create a layout for the histograms

# Adjust the number of rows and columns as needed

layout(matrix(1:(ncol(df)-2), nrow = 5, ncol = 6))


#Create histograms for each column excluding the first two columns

for (i in 3:ncol(df)) {

  col_name <- colnames(df)[i]  # Get the column name

  hist(df[, i], main = paste("Histogram for", col_name))

}


## ##########################################calculate the correlations between predictor variables:



# Check data types of selected columns

selected_cols <- df[, 2:32]


# Select only numeric columns

numeric_cols <- selected_cols[sapply(selected_cols, is.numeric)]
```

```
# Calculate correlations for numeric columns
correlations <- cor(numeric_cols)
correlations

library(corrplot)
#par(mfrow = c(1,1), pin=c(4,4))
options(repr.plot.width = 10, repr.plot.height = 10)
corrplot(correlations, order = "hclust")



##############################################################################
##

#Outliers Box plots##

par(mfrow=c(3, 3))
lapply(names(numeric_cols), function(x) {boxplot(df[, x], main=x, ylab=x)})

##############################################################################
###

#skewness

library(e1071)

skew = sapply(numeric_cols, skewness)
skew
```

```
################################################################################
####

#missing values

missingdata <- sum(is.na(numeric_cols)==TRUE)
missingdata
image(is.na(numeric_cols), main = "Missing Values", xlab = "Observation",
    ylab = "Variable", xaxt = "n", yaxt = "n", bty = "n")
axis(1, seq(0, 1, length.out = nrow(numeric_cols)), 1:nrow(numeric_cols), col = "white")


################################################################################
####
#response
diagnosis <- df['diagnosis']
diagnosis
dim(diagnosis)

#predictors
predictorz <- df[, !(names(df) %in% c('id', 'diagnosis'))]
predictorz
dim(predictorz)


########################################################## Data Preprocessing
```

```
# Removing near-zero variance predictors

# Correct way to subset the dataframe

nzv_vars <- names(predictorz)[nzv$nzv]  # This should give you an empty vector since no nzv variables are found

predictorz_nz <- predictorz[, !(names(predictorz) %in% nzv_vars)]

predictorz_nz

dim(predictorz_nz)


# Handling highly correlated predictors

corMatrix <- cor(predictorz_nz)

highCor <- findCorrelation(corMatrix, cutoff = 0.90) # Adjust cutoff as needed

predictorz_cor <- predictorz_nz[, -highCor]

predictorz_cor


dim(predictorz_cor)


##################################### box cox and spatial transformation



# Specify the transformation methods

methods <- c( "BoxCox", "center", "scale", "spatialSign")


# Create a preProcess object for transformation

trans <- preProcess(predictorz_cor, method = methods)


# Apply the transformation to the data
```

```
predictorz_f <- predict(trans, predictorz_cor)
predictorz_f
dim(predictorz_f)
```

```
##################################### Histogram of transformed predictors

# Load necessary libraries
library(ggplot2)
library(tidyr)

# Reshape the data to long format
long_predictorz_f <- gather(predictorz_f, key = "variable", value = "value")

# Plotting all histograms in one figure using facet_wrap
ggplot(long_predictorz_f, aes(x = value)) +
  geom_histogram(bins = 30, fill = "blue", color = "black") +
  facet_wrap(~ variable, scales = "free_x", ncol = 3) +  # Adjust the number of columns with ncol
  theme_minimal() +
  ggtitle("Histograms of All Predictors")


##################################### Box plot of transformed predictors
```

```
# Load necessary libraries
library(ggplot2)
library(tidyr)

# Reshape the data to long format
long_predictorz_f <- gather(predictorz_f, key = "variable", value = "value")

# Plotting all box plots in one figure using facet_wrap
ggplot(long_predictorz_f, aes(x = variable, y = value)) +
  geom_boxplot(fill = "lightblue", color = "black") +
  facet_wrap(~ variable, scales = "free_y", ncol = 3) +  # Adjust ncol as needed
  theme_minimal() +
  ggtitle("Box Plots of All Predictors") +
  theme(axis.text.x = element_blank(), axis.ticks.x = element_blank())



#################################### Correlation plot of transformed predictors

# Load required library
library(corrplot)

# Calculate the correlation matrix for 'predictorz_f'
correlation_matrix <- cor(predictorz_f)

# Open a new plotting window (useful in RStudio or similar environments)
plot.new()
dev.off()
```

```
# Adjust graphical parameters
old_par <- par(no.readonly = TRUE) # Save current settings to restore later
par(mar = c(5, 5, 5, 5))  # Adjust margins

# Create a larger correlation plot with adjusted text size
corrplot(correlation_matrix, method = "circle", tl.cex = 0.6) # Adjust tl.cex as needed

# Restore previous graphical parameters
par(old_par)



############################################################ splitting the data

#Transforming respone to have data points

diagnosis_new <- diagnosis$diagnosis
diagnosis_new


# Convert 'diagnosis' into a factor with levels 'M', 'B'
diagnosis_new <- factor(diagnosis_new, levels = c( "M", "B"))
str(diagnosis_new)
print(length(diagnosis_new))


# Create an index for data partitioning
```

```
index <- createDataPartition(y = diagnosis_new, p = 0.8, list = FALSE)


# Split the data into training and testing sets
train_predictors <- predictorz_f[index, ]
train_response <- diagnosis_new[index]

test_predictors <- predictorz_f[-index, ]
test_response <- diagnosis_new[-index]


############################################################ control for cross validation

# Control for cross-validation
ctrl <- trainControl(method = "LGOCV",
                     summaryFunction = twoClassSummary,

                     classProbs = TRUE,
                     savePredictions = TRUE)


############################################################  Logistic regression model

# Train the model using glm logistic regression because we are only dealing with 2 y var
lrm <- train(train_predictors, train_response,
             method = "glm",
             trControl = ctrl,
```

```
        preProcess = c("center","scale"),
        metric = "ROC")

lrm

plot(lrm)

#model predictions
head(lrm$pred)

lrm_varImp <- varImp(lrm, scale = FALSE) ############variable importance
lrm_varImp

plot(lrm_varImp, top = 20, scales = list(y = list(cex = .95)))



# Predictions and performance evaluation confusion matrix

predictions_lrm <- predict(lrm, test_predictors)
predictions_lrm

confusionMatrix(predictions_lrm, test_response)



########################################################################## LINEAR
DISCRIMINANT ANALYSIS
```

```
# Training the LDA model
Lda<- train(train_predictors, train_response,
        method = "lda",
        metric = "ROC",
        preProcess = c("center","scale"),
        trControl = ctrl)

Lda
plot(Lda)

# Viewing initial predictions
head(Lda$pred)

##Variable of Importance
library(caret)

Lda_varImp <- varImp(Lda, scale = FALSE) ############variable importance
Lda_varImp

plot(Lda_varImp, top = 20, scales = list(y = list(cex = .95)))


# Predictions and performance evaluation confusion matrix

predictions_Lda <- predict(lrm, test_predictors)
predictions_Lda
```

```
confusionMatrix(predictions_Lda, test_response)



########################################################################## PLSDA

# Training the PLSDA model
Plsda<- train(train_predictors, train_response,
              method = "pls",
              metric = "ROC",
              preProcess = c("center","scale"),
              tuneGrid = expand.grid(.ncomp = 1:14),
              #tuneLength = 14,
              trControl = ctrl)

Plsda

plot(Plsda,main = "No. of components Vs Accuracy in PLSDA")
# Viewing initial predictions
head(Plsda$pred)

##Variable of Importance
library(caret)
Plsda_varImp <- varImp(Plsda, scale = FALSE) ############variable importance
Plsda_varImp
```

```
plot(Plsda_varImp, top = 20, scales = list(y = list(cex = .95)))


# Predictions and performance evaluation confusion matrix

predictions_Plsda <- predict(Plsda, test_predictors)
predictions_Plsda

confusionMatrix(predictions_Plsda, test_response)



############################################################ PENALIZED MODEL

glmnGrid <- expand.grid(.alpha = c(0, .1, .2, .4, .6, .8, 1),
                        .lambda = seq(.01, .2, length = 10))

set.seed(476)
glmnTuned <- train(train_predictors, train_response,
                   method = "glmnet",
                   tuneGrid = glmnGrid,
                   preProc = c("center", "scale"),
                   metric = "ROC",
                   trControl = ctrl)


glmnTuned
```

```
## The heat map in

head(glmnTuned$pred)

plot(glmnTuned, plotType = "level")


######################Variable of Importance
library(caret)
glmnImpSim <- varImp(glmnTuned, scale = FALSE) ############variable importance
glmnImpSim

plot(glmnImpSim, top = 20, scales = list(y = list(cex = .95)))


# Make predictions on the test set
glmn_predictions <- predict(glmnTuned, newdata = test_predictors)


######################### Evaluating the model Confusion Matrix
confusionMatrix(glmn_predictions, test_response)



########################################################################## NEAREST
SHRUNKEN CENTROIDS
```

```
## nscGrid <- data.frame(.threshold = 0:4)
nscGrid <- data.frame(.threshold = seq(0,4, by=0.1))
set.seed(476)
nscTuned <- train(train_predictors, train_response,
                  method = "pam",
                  preProc = c("center", "scale"),
                  tuneGrid = nscGrid,
                  metric = "ROC",
                  trControl = ctrl)

nscTuned

plot(nscTuned)



library(caret)
nsc_varImp <- varImp(nscTuned, scale = FALSE) ############variable importance
nsc_varImp

plot(nsc_varImp, top = 20, scales = list(y = list(cex = .95)))



##############Prediction

# Make predictions on the test set
nsc_predictions <- predict(nscTuned, newdata = test_predictors)
```

```
######################## Evaluating the model Confusion Matrix
confusionMatrix(nsc_predictions, test_response)




######################## NON LINEAR CLASSIFICATION MODELS

######## Nonlinear Discriminant Analysis ################################


#a) Mixture Discriminant Analysis

## do not need to save predictions)
set.seed(476)
Mda <- train(train_predictors, train_response,
             method = "mda",
             metric = "ROC",
             tuneGrid = expand.grid(.subclasses = 1:10),
             trControl = ctrl)

Mda

plot(Mda)
```

```
Mda_varImp <- varImp(Mda, scale = FALSE) ###########variable importance
Mda_varImp

plot(Mda_varImp, top = 20, scales = list(y = list(cex = .95)))
# Predictions and performance evaluation

predictions <- predict(Mda, newdata = test_predictors)
confusionMatrix(predictions, test_response)



#####################################################################

#b) Quadratic Discriminant Analysis



# Assuming you have already set the seed and defined your control parameters
# Train a QDA model

QdaModel <- train(train_predictors, train_response,
                  method = "qda",
                  metric = "ROC",
                  preProc = c("center", "scale"),
                  trControl = ctrl)



# To view the model summary
QdaModel
```

```
plot(QdaModel)


QdaModel_varImp <- varImp(QdaModel, scale = FALSE) ############variable importance
QdaModel_varImp

plot(QdaModel_varImp, top = 20, scales = list(y = list(cex = .95)))
# Predictions and performance evaluation

predictions <- predict(QdaModel, newdata = test_predictors)
confusionMatrix(predictions, test_response)


##################################################

# c) Regularized Discrimiant Analysis

#install.packages("rrcov")

#library(rrcov)
set.seed(300)
tunegrid <- expand.grid(
  .gamma = seq(0.1, 1, by = 0.1),    # Example values for gamma
  .lambda = seq(0.01, 0.1, by = 0.01) # Example values for lambda
)
```

```
rdaFit <- train(train_predictors, train_response,
                method = "rda",
                metric = "ROC",
                tuneGrid = tunegrid,
                trControl = ctrl)

rdaFit

plot(rdaFit)


rdaFit_varImp <- varImp(rdaFit, scale = FALSE) ############variable importance
rdaFit_varImp

plot(rdaFit_varImp, top = 20, scales = list(y = list(cex = .95)))
# Predictions and performance evaluation

predictions <- predict(rdaFit, newdata = test_predictors)
confusionMatrix(predictions, test_response)


################################################## Neural Networks ##############

# Expand grid for tuning the neural network
nnetGrid <- expand.grid(.size = 1:8, .decay = c(0, .1, 1, 2))
maxSize <- max(nnetGrid$.size)
numWts <- (maxSize * (81 + 1) + (maxSize + 1) * 3)
```

```
set.seed(476)

nnetFit <- train(train_predictors, train_response,
                 method = "nnet",
                 metric = "ROC",
                 preProc = c("center", "scale", "spatialSign"),
                 tuneGrid = nnetGrid,
                 trace = FALSE,
                 maxit = 2000,
                 MaxNWts = numWts,
                 trControl = ctrl)



nnetFit

plot(nnetFit)



nnetFit_varImp <- varImp(nnetFit, scale = FALSE) ############variable importance
nnetFit_varImp

plot(nnetFit_varImp, top = 20, scales = list(y = list(cex = .95)))
# Predictions and performance evaluation

predictions <- predict(nnetFit, newdata = test_predictors)
```

```
confusionMatrix(predictions, test_response)


################################## Flexible Discriminant Analysis #############


marsGrid <- expand.grid(.degree = 1:2, .nprune = 2:38)

fdaTuned <- train(train_predictors, train_response,
                  method = "fda",
                  metric = "ROC",
                  # Explicitly declare the candidate models to test
                  tuneGrid = marsGrid,
                  preProc = c("center", "scale", "spatialSign"),
                  trControl = ctrl)

fdaTuned

plot(fdaTuned)
plot(fdaTuned,main="FDA, degree = 1 and nprune = 6")

fdaTuned_varImp <- varImp(fdaTuned, scale = FALSE) ############variable importance
fdaTuned_varImp

plot(fdaTuned_varImp, top = 20, scales = list(y = list(cex = .95)))
# Predictions and performance evaluation
```

```
predictions <- predict(fdaTuned, newdata = test_predictors)
confusionMatrix(predictions, test_response)



################################################## Support Vector Machines ##########

set.seed(202)
library(kernlab)
library(caret)

sigmaRangeReduced <- sigest(as.matrix(biotrain))



svmRGridReduced <- expand.grid(.sigma = sigmaRangeReduced[1],
                               .C = 2^(seq(-4, 6)))
set.seed(476)
svmRModel <- train(train_predictors, train_response,
                   method = "svmRadial",
                   metric = "ROC",
                   preProc = c("center", "scale"),
                   tuneGrid = svmRGridReduced,
                   fit = FALSE,
                   trControl = ctrl)

svmRModel
```

```
plot(svmRModel)



svmRModel_varImp <- varImp(svmRModel, scale = FALSE) ###########variable importance
svmRModel_varImp

plot(svmRModel_varImp, top = 20, scales = list(y = list(cex = .95)))
# Predictions and performance evaluation

predictions <- predict(svmRModel, newdata = test_predictors)
confusionMatrix(predictions, test_response)



############################################## K-Nearest Neighbors ##############



library(caret)
set.seed(476)
knnFit <- train(train_predictors, train_response,
                method = "knn",
                metric = "ROC",
                preProc = c("center", "scale"),
                #tuneGrid = data.frame(.k = c(4*(0:5)+1, 20*(1:5)+1, 50*(2:9)+1)), ## 21 is the best
                tuneGrid = data.frame(.k = 1:50),
                trControl = ctrl)
```

```
knnFit

plot(knnFit)


knnFit_varImp <- varImp(knnFit, scale = FALSE) ############variable importance
knnFit_varImp

plot(knnFit_varImp, top = 20, scales = list(y = list(cex = .95)))
# Predictions and performance evaluation

predictions <- predict(knnFit, newdata = test_predictors)
confusionMatrix(predictions, test_response)



########################################################## Naive Bayes ##########

install.packages("klaR")
library(klaR)
set.seed(476)
nbFit <- train( train_predictors, train_response,
                method = "nb",
                metric = "ROC",
                preProc = c("center", "scale"),
                ##tuneGrid = data.frame(.k = c(4*(0:5)+1, 20*(1:5)+1, 50*(2:9)+1)), ## 21 is the best
```

```
        tuneGrid = data.frame(.fL = 2,.usekernel = TRUE,.adjust = TRUE),
        trControl = ctrl)


nbFit

plot(nbFit)


nbFit_varImp <- varImp(nbFit, scale = FALSE) ############variable importance
nbFit_varImp

plot(nbFit_varImp, top = 20, scales = list(y = list(cex = .95)))
# Predictions and performance evaluation

predictions <- predict(nbFit, newdata = test_predictors)
confusionMatrix(predictions, test_response)
```