A Comparative Analysis of Machine Learning Classification Methods Applied to Wisconsin Breast Cancer Database

Applied Regression Analysis

Final project

By

Soodabeh Ramezani

Instructor

Dr. Katherine Shoemaker

Fall 2019

1. Introduction

This report presents a comparative study on the performance of several modern machine learning algorithms used in a classification task. The Wisconsin breast cancer dataset [1 and 2], investigated in this study, is a collection of digitized images with several extracted features and their associated ranges (e.g., standard deviation). Each image represents a cell nucleus of breast mass being classified as Benign or Malignant lump. The classification task is to predict whether an unseen image of breast lump is benign or malignant. For diagnostics purposes, it is very important to develop a predicative model that ensures a high level of accuracy while predicting the cancer type. Therefore, it is worthwhile to develop and test several machine learning methods which help with choosing the most reliable predictive model.

The dataset includes 30 (correlative) feature columns which require a data preparation step to reduce the dimensionality of data. We make use of principle component analysis (PCA) to identify the directions of highest variations. Subsequently, we use this method to choose adequate number of principle components that explain majority of variations in the data. The outcomes of PCA are then fed to machine learning algorithms for training and performance predication.

In addition to PCA, we apply another data preparation method to lower the correlation/dependency between input features. This step, also considered as a variable selection step, is performed to identify highly-correlated features. We subsequently, cluster the highly-correlated features and select only one feature representing each cluster. This step helps to reduce 30 features down to only 10 features being fed to the machine learning algorithms.

We develop and test the following methods: (a) logistic regression, (b) kth nearest neighbors method (KNN), (c) linear discriminant analysis (LDA), (d) quadratic discriminant analysis (QDA), and (e) decision tree to perform classification task. The performance of these methods will be quantified using validation set approach as well as k-fold cross-validation technique. We present the results using confusion matrices to help comparing accuracy of those methods.

This report benefits those working in breast cancer research, and in general, those in diagnostics studies providing them with a clear idea of the performance of several currently available machine learning methods in predicting cancer stage. Even though we implement our algorithms based on breast cancer dataset, the same technique can be applied to any other classification task appealing to medical diagnostics industry.

2. Packages Required

```
#All packages
'Loading library corrplot in order to use corrplot function for
plotting dependency between input features using heat map'
library(corrplot)

'Loading library ggplot2 in order to use ggplot function for
advance plotting with variety of visualization options'
library(ggplot2)
```

```
'Loading library tibble in order to use function as_tibble to
convert data to tibble format which makes it easier to handle
tabular data including data merging and printing'
library(tibble)

'Loading library GGally in order to use ggpairs function to make a
generalized pair plot for  numerical and categorial data'
library(GGally)

'Loading library gmodels in order to use CrossTable to
show/compare the performance of classification'
library(gmodels)

'Loading library MASS in order to use lda function'
library(MASS)

'Loading library class in order to use knn function'
library(class)

'Loading caret library to use createFolds in order to
partition the data into k folds for cross-validation'
library(caret)

'Loading library tree for using tree function'
library(tree)
```

3. Data Preparation

The dataset used in the project was obtained from the University of California at Irvine (UCI) machine learning repository [1]. The dataset, referred to as Wisconsin Diagnostic Breast Cancer or WDBC, was originally created by the University of Wisconsin as a collaborative effort between General Surgery Department and Computer Science Department [2]. It was first published by W.N. Street et al. [3] in 1993 and then by O.L. Mangasarian [4] in 1995. The Authors in both papers used linear programming approaches to classify images of breast tumor and performed predication using their highly accurate models.

The dataset consists of digitized images of cell nuclei in breast mass extracted from fine needle aspirates. The images are characterized by 10 features and their associated mean, max (or worst value), and standard deviation. According to UCI website, those features are described as follows:

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

b) texture (standard deviation of gray-scale values)

c) perimeter

d) area

e) smoothness (local variation in radius lengths)
f) compactness (perimeter^2 / area - 1.0)
g) concavity (severity of concave portions of the contour)
h) concave points (number of concave portions of the contour)
i) symmetry
j) fractal dimension ("coastline approximation")

Each image is labeled as either benign or malignant type of breast mass. The objective of this project is to accurately predict cancer type by analyzing the relevant image features. Below we are showing the first few lines of the data:

```
head(Cancer)
```

```
##   ID number Diagnosis Radius Texture Perimeter   Area Smoothness
## 1    842302         M  17.99   10.38    122.80 1001.0    0.11840
## 2    842517         M  20.57   17.77    132.90 1326.0    0.08474
## 3  84300903         M  19.69   21.25    130.00 1203.0    0.10960
## 4  84348301         M  11.42   20.38     77.58  386.1    0.14250
## 5  84358402         M  20.29   14.34    135.10 1297.0    0.10030
## 6    843786         M  12.45   15.70     82.57  477.1    0.12780
##   Compactness Concavity Concave Symmetry Fractal SE_radius SE_texture
## 1     0.27760    0.3001 0.14710   0.2419 0.07871    1.0950     0.9053
## 2     0.07864    0.0869 0.07017   0.1812 0.05667    0.5435     0.7339
## 3     0.15990    0.1974 0.12790   0.2069 0.05999    0.7456     0.7869
## 4     0.28390    0.2414 0.10520   0.2597 0.09744    0.4956     1.1560
## 5     0.13280    0.1980 0.10430   0.1809 0.05883    0.7572     0.7813
## 6     0.17000    0.1578 0.08089   0.2087 0.07613    0.3345     0.8902
##   SE_perimeter SE_area SE_smoothness SE_compactness SE_concavity
## 1        8.589  153.40      0.006399        0.04904      0.05373
## 2        3.398   74.08      0.005225        0.01308      0.01860
## 3        4.585   94.03      0.006150        0.04006      0.03832
## 4        3.445   27.23      0.009110        0.07458      0.05661
## 5        5.438   94.44      0.011490        0.02461      0.05688
## 6        2.217   27.19      0.007510        0.03345      0.03672
##   SE_concave SE_symmetry SE_fractal W_radius W_texture W_perimeter W_area
## 1    0.01587     0.03003  0.006193    25.38     17.33      184.60 2019.0
## 2    0.01340     0.01389  0.003532    24.99     23.41      158.80 1956.0
## 3    0.02058     0.02250  0.004571    23.57     25.53      152.50 1709.0
## 4    0.01867     0.05963  0.009208    14.91     26.50       98.87  567.7
## 5    0.01885     0.01756  0.005115    22.54     16.67      152.20 1575.0
## 6    0.01137     0.02165  0.005082    15.47     23.75      103.40  741.6
##   W_smoothness W_compactness W_concavity W_concave W_symmetry W_fractal
## 1       0.1622        0.6656     0.7119    0.2654     0.4601    0.11890
## 2       0.1238        0.1866     0.2416    0.1860     0.2750    0.08902
## 3       0.1444        0.4245     0.4504    0.2430     0.3613    0.08758
```

```
## 4   0.2098   0.8663   0.6869   0.2575   0.6638   0.17300
## 5   0.1374   0.2050   0.4000   0.1625   0.2364   0.07678
## 6   0.1791   0.5249   0.5355   0.1741   0.3985   0.12440
```

Data was imported using “read.delim” command in R. The data is already clean however we added names to each column of the data for more clarity. Figure 1 shows several input features with their corresponding class labels using boxplot visualization. This figure shows significant distinctions between mean values for some of the features when plotting for each class. This distinction will help us separate the two classes with high accuracy when using machine learning methods. Figures 2 and 3 also show several interesting input data to be used for classification. We observe strong correlations between several input data requiring a variable selection method to remove dependencies before developing classification methods.

4. Exploratory Data Analysis

We have plotted the data in several ways to efficiently show the trend of the input, inter-relation between the data, as class separation observed in the data. Another way to look at the data is to visualize data correlation matrix using heatmap plot. The heatmap plot shown in Figure 4 visualizes the pairwise correlation between input features. The size and color of each circle in that figure corresponds to the correlation coefficients. The plot shows several clusters of highly correlative features. We initially choose 12 clusters and pick one input feature from each cluster. Subsequently, we reduce the input feature to 10 features to ensure minimal dependency between input features. Figure 5 shows the final heatmap plotted only for the final 10 selected features. This figure shows significantly lower correlations between the 10 input features to be used in subsequent machine learning methods.

PCA dimensionality reduction method is shown in Figures 6 through 10. The first two principle components shown to explain over 64% of the variations in the data. We use the first two components as inputs for future machine learning methods.

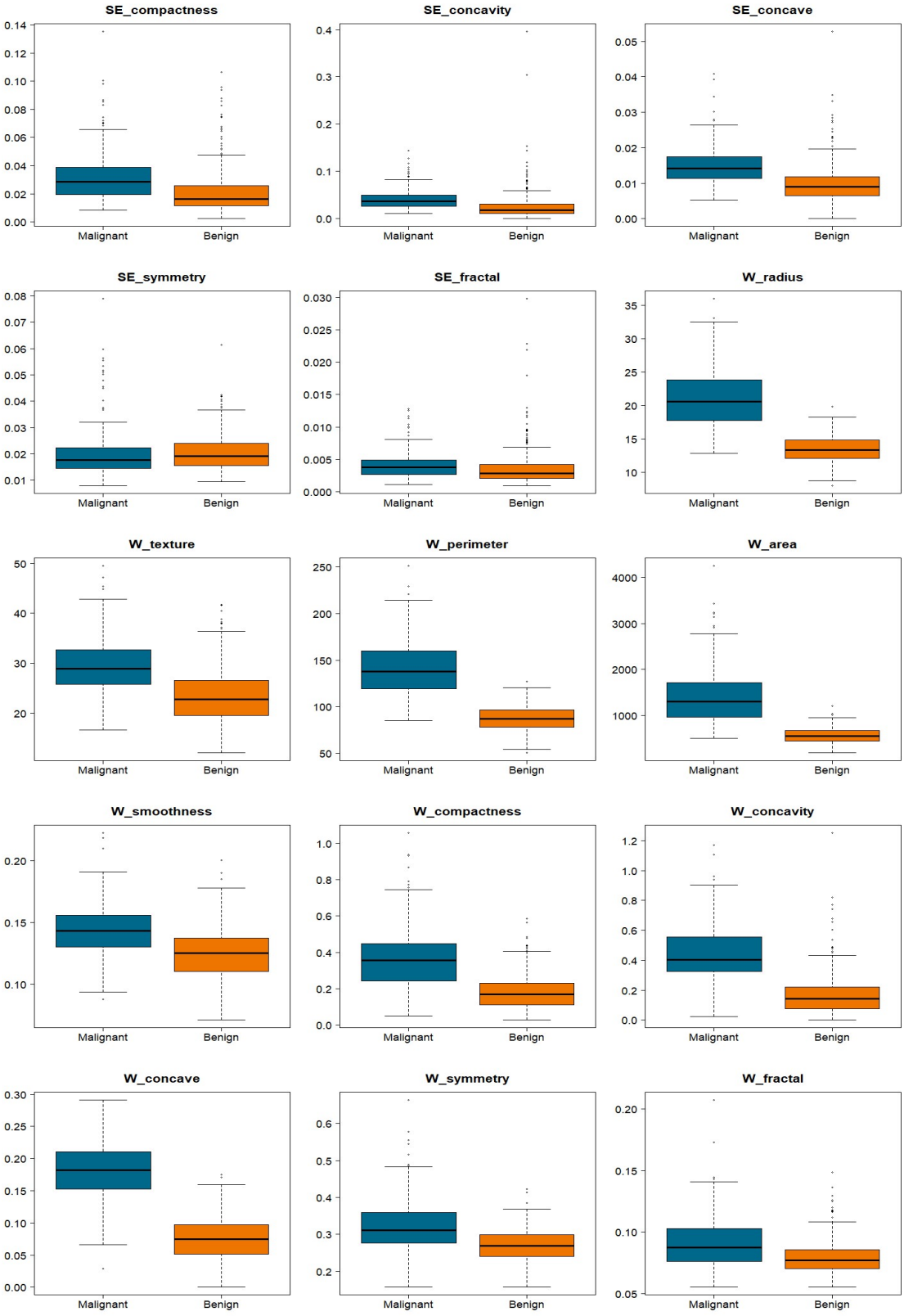


Figure 1: Boxplot of features as a function of Diagnosis

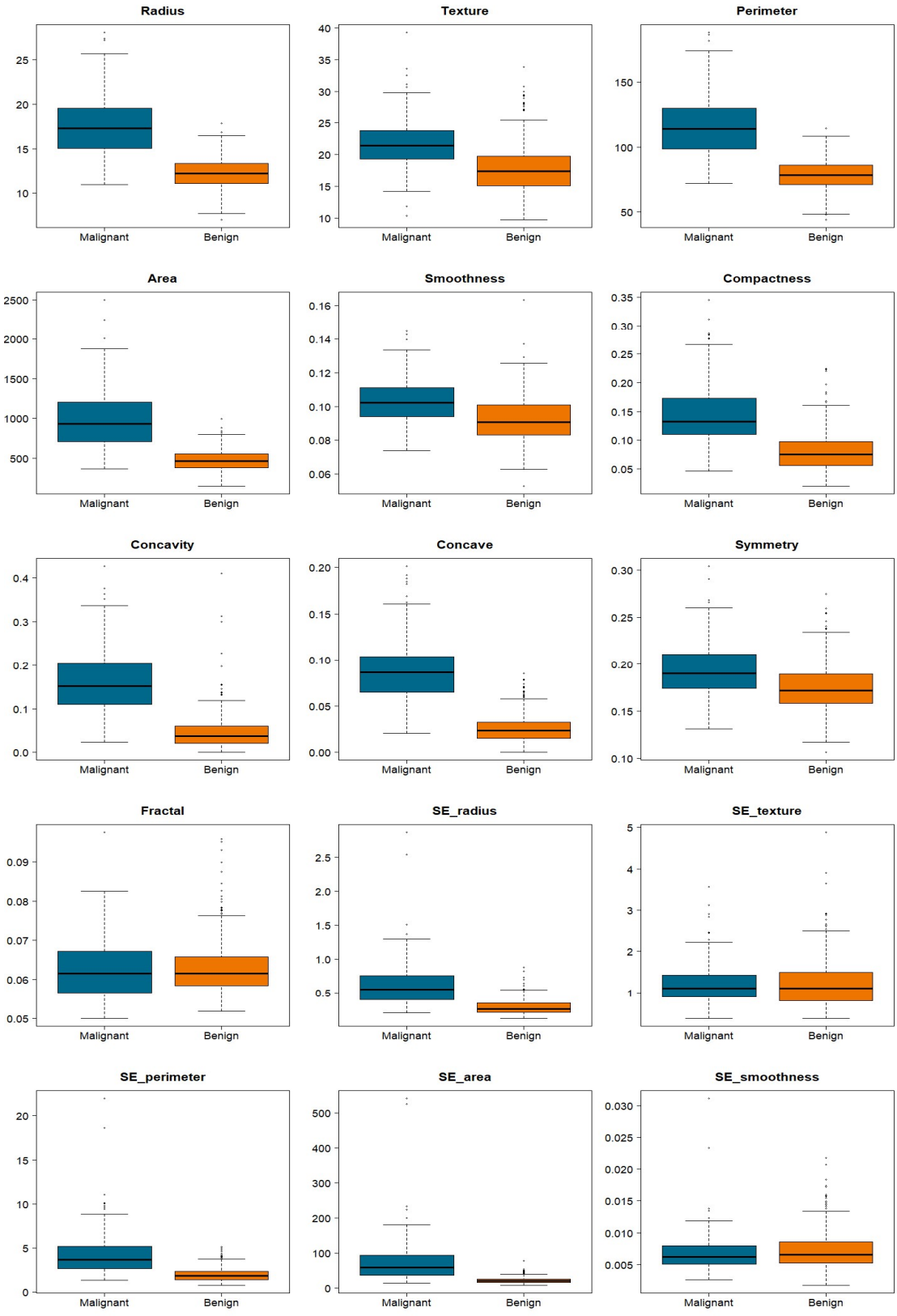


Figure 1: Boxplot of features as a function of Diagnosis

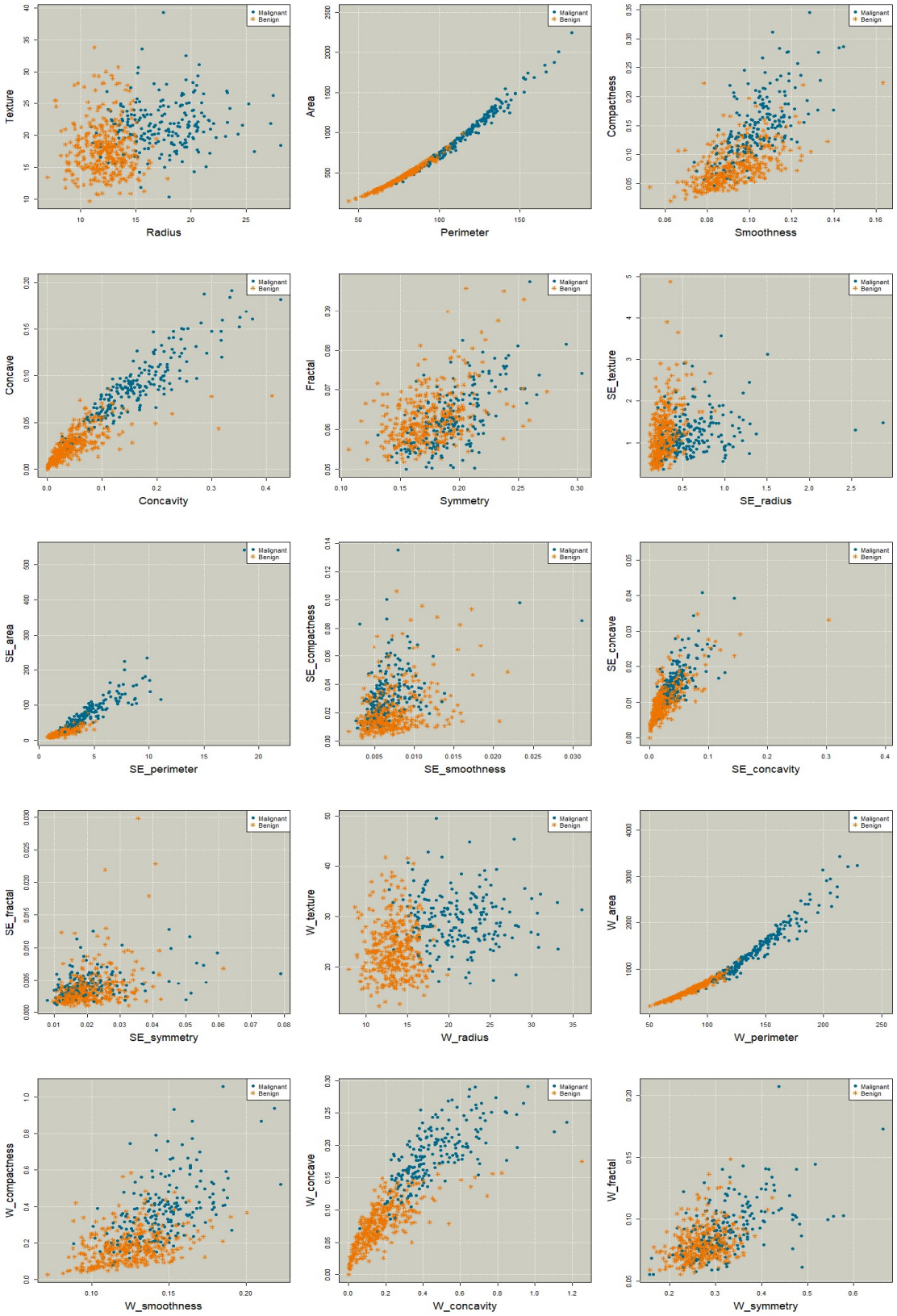


Figure 2: Pairwise scatter plot of select input features color coded by image type (Benign or Malignant)

Perimeter

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
43.79	75.17	86.24	91.97	104.10	188.50

worst _Perimeter

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
50.41	84.11	97.66	107.26	125.40	251.20

worst _Radius

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
7.93	13.01	14.97	16.27	18.79	36.04

Figure 3: Summary of a few important features: Perimeter, worst _Perimeter and worst _Radius

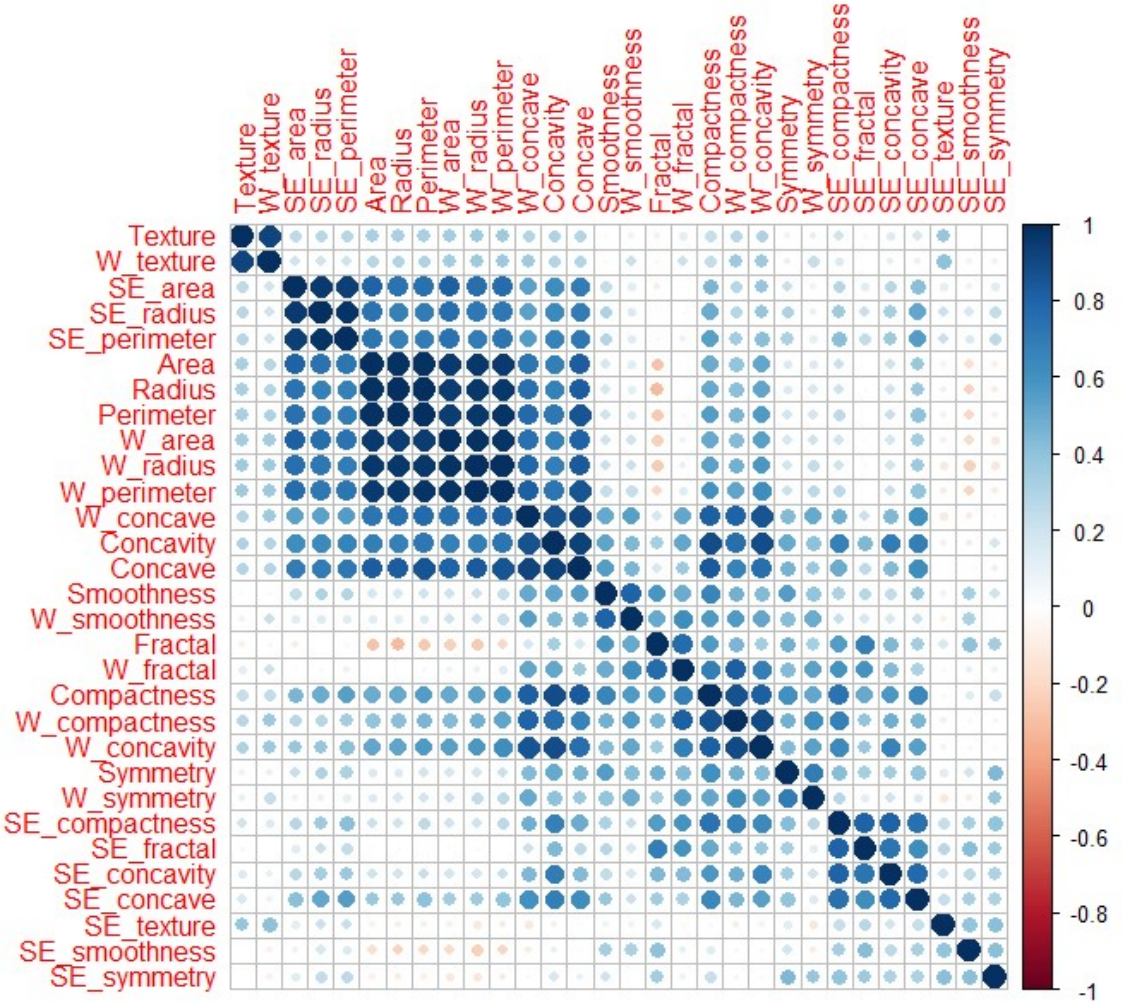


Figure 4: Pairwise correlation between input features

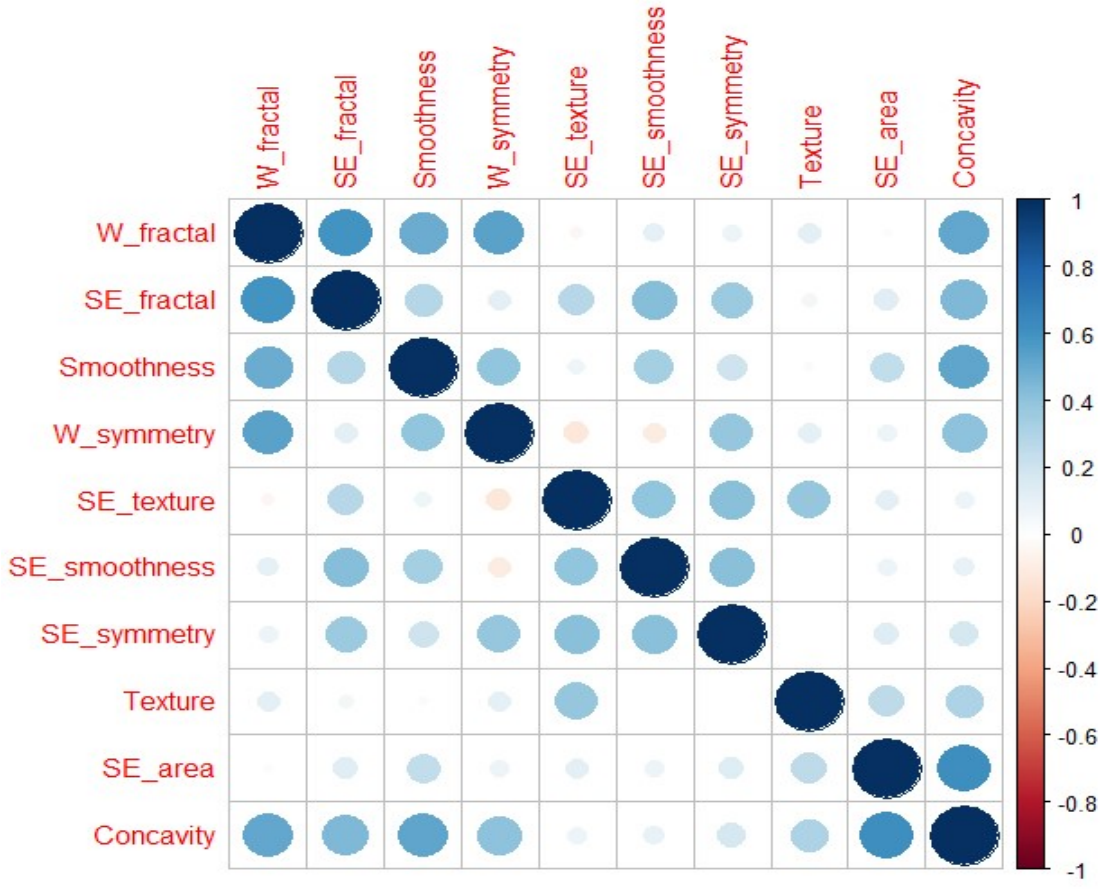


Figure 5: Pairwise correlation between input for the 10 selected features

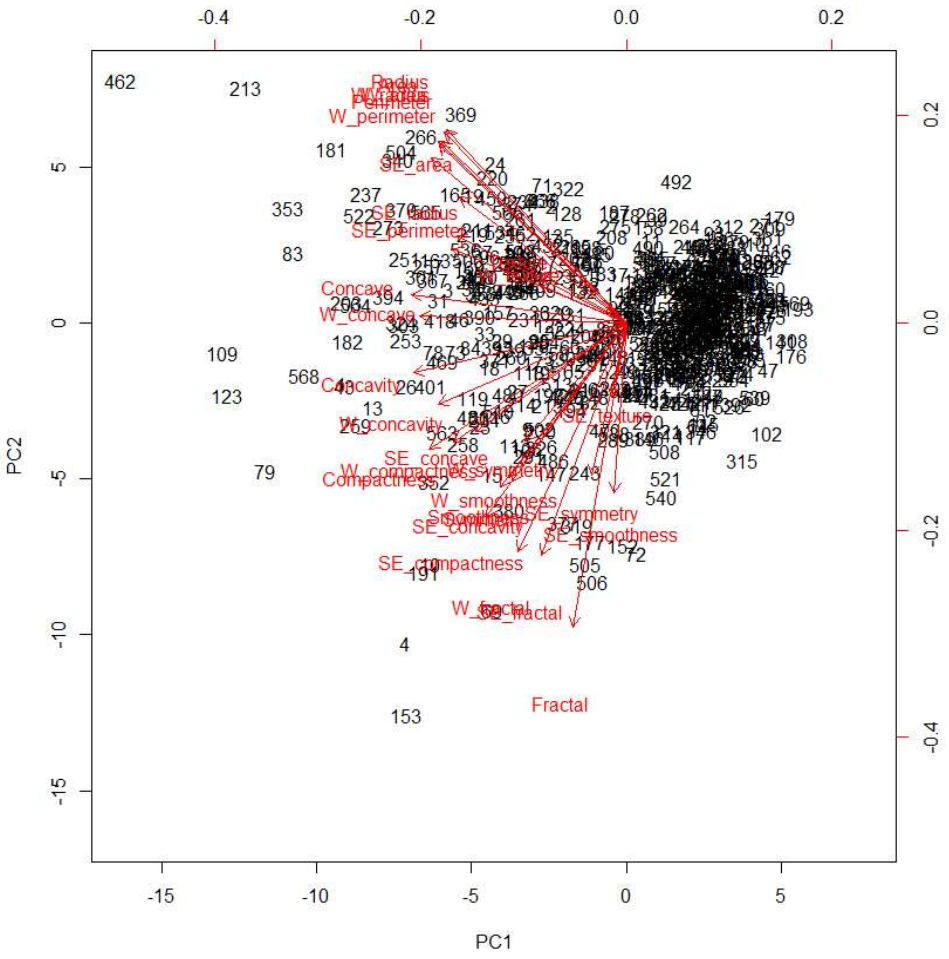


Figure 6: The first two principle components for the data. The red arrows indicate the first two principle component loading vectors with axes on the top and right.

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

Figure 7: Variance explained by each PC (30 principle components)

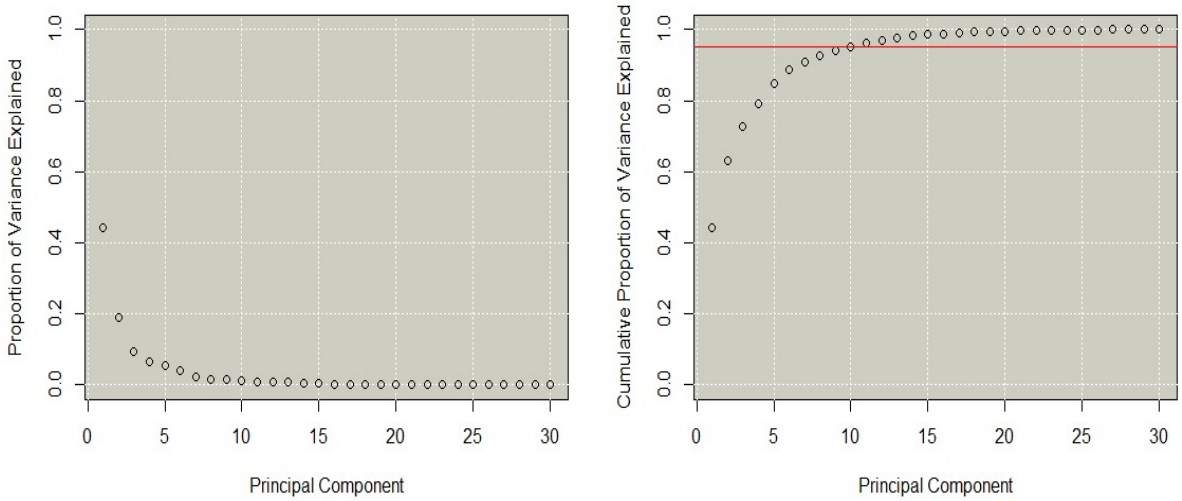


Figure 8: Left, Proportion of Variance Explained (PVE) by each PC. Right, cumulative PVE for each PC

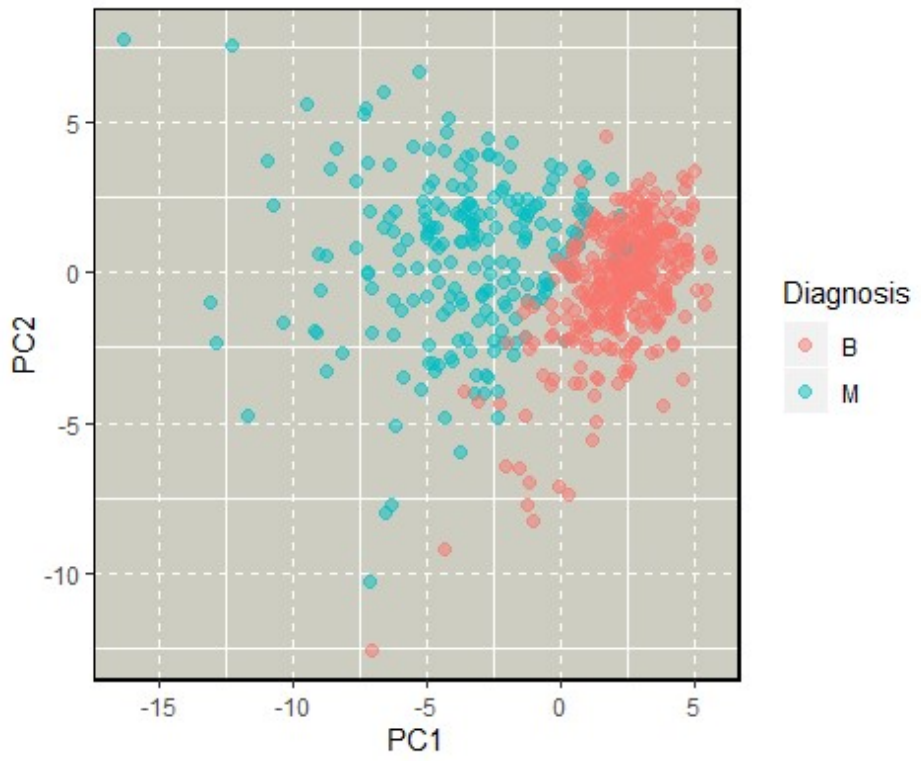


Figure 9: Pairwise scatter plot of PC1 vs. PC2, color coded by image type (Benign or Malignant)

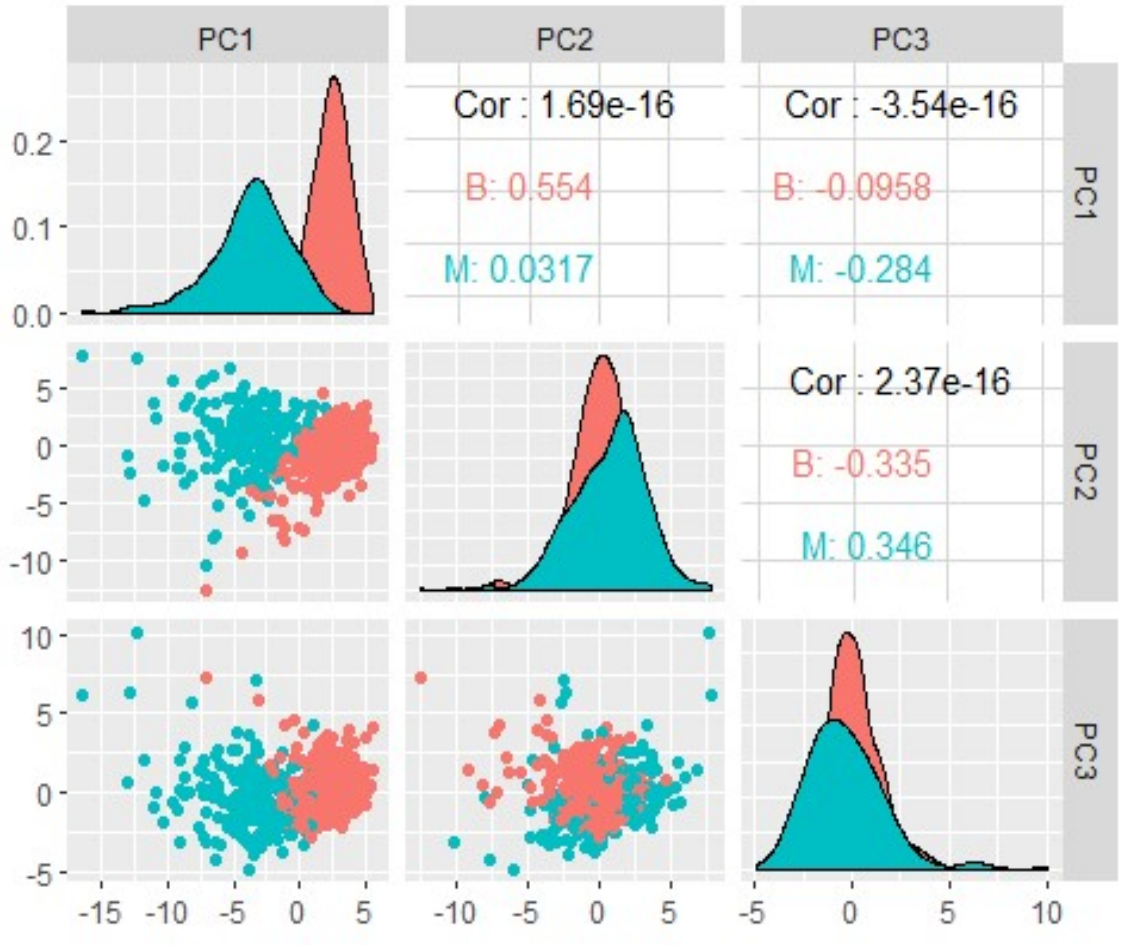


Figure 10: Pairwise scatter plots of the first three principle components after applying PCA, color coded by image type (Benign or Malignant). The distribution of rotated data is shown on the diagonal.

4.1. Logistic Regression

The first method to explore is logistic regression as shown in Figure 11. We only used the 10 selected features from previous step to perform training and prediction. Please note that a few of the features show p-values higher than 0.05 significant level. we attempted to delete those features in the hope of getting better results. However, the prediction accuracy of the reduced model did not improve, therefore, we decided to keep all the 10 input features.

```
##
## Call:
## glm(formula = Diagnosis ~ Texture + SE_area + Concavity + Smoothness +
##     W_fractal + W_symmetry + SE_fractal + SE_texture + SE_smoothness +
##     SE_symmetry, family = binomial, data = Cancer_train)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.58877  -0.04959  -0.00628   0.00010   2.68816
##
```

```
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -3.579e+01  7.615e+00  -4.699 2.61e-06 ***
## Texture       5.235e-01  1.344e-01   3.895 9.81e-05 ***
## SE_area       2.836e-01  5.799e-02   4.890 1.01e-06 ***
## Concavity     4.928e+01  1.080e+01   4.562 5.07e-06 ***
## Smoothness    1.734e+01  3.851e+01   0.450  0.65244
## W_fractal     1.037e+02  5.418e+01   1.914  0.05566 .
## W_symmetry    3.852e+01  1.232e+01   3.128  0.00176 **
## SE_fractal   -1.473e+03  4.519e+02  -3.261  0.00111 **
## SE_texture   -1.116e+00  8.248e-01  -1.352  0.17622
## SE_smoothness  3.222e+02  1.528e+02   2.109  0.03495 *
## SE_symmetry   -2.829e+02  9.109e+01  -3.106  0.00190 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 551.501  on 425  degrees of freedom
## Residual deviance:  61.575  on 415  degrees of freedom
## AIC: 83.575
##
## Number of Fisher Scoring iterations: 10
```

Figure 11: Logistic Regression Summary

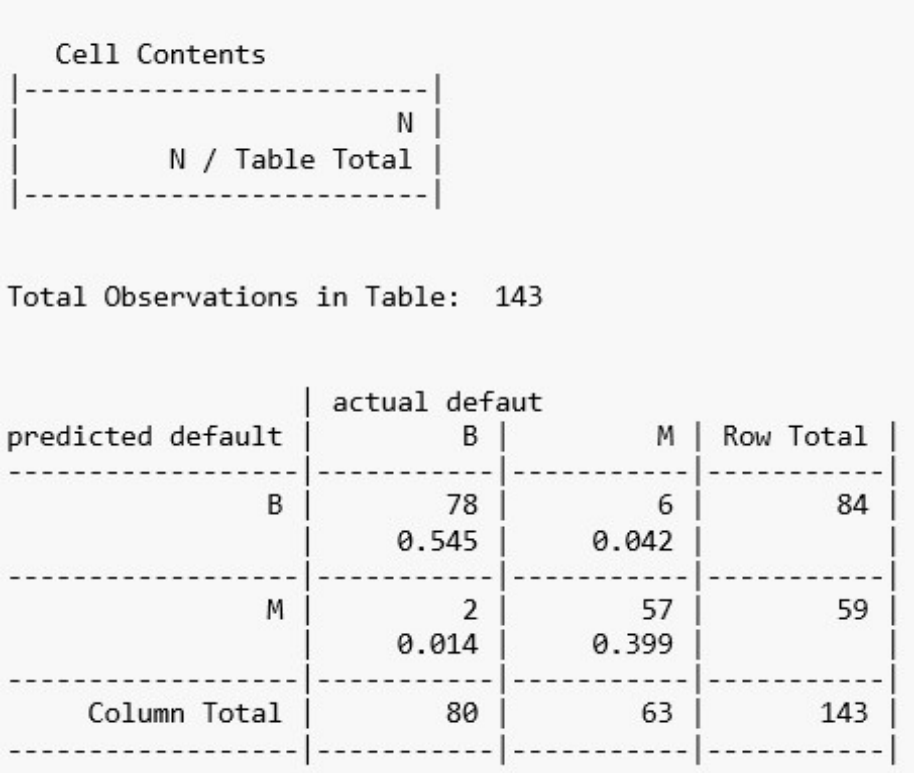
```
   Cell Contents
|-------------------------|
|                       N |
|         N / Table Total |
|-------------------------|


Total Observations in Table:  143


                    | actual defaut
predicted default   |         B |         M | Row Total |
--------------------|-----------|-----------|-----------|
                  B |        80 |         3 |        83 |
                    |     0.559 |     0.021 |           |
--------------------|-----------|-----------|-----------|
                  M |         0 |        60 |        60 |
                    |     0.000 |     0.420 |           |
--------------------|-----------|-----------|-----------|
     Column Total   |        80 |        63 |       143 |
--------------------|-----------|-----------|-----------|
```

Figure 12: Confusion matrix of Logistic Regression

We also investigated logistic regression performance when using the first two principle components of the data as shown in Figure 13. This approach does not result in a better accuracy compared to the previous logistic regression model (Figure 14).

```
## Call:
## glm(formula = Cancer_train_pc$Diagnosis ~ PC1 + PC2, family = binomial,
##     data = Cancer_train_pc)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.96425  -0.27853  -0.11471   0.00433   2.96855
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.575027   0.291275   1.974   0.0484 *
## PC1         -0.011990   0.001430  -8.386  < 2e-16 ***
## PC2          0.030533   0.005604   5.448 5.09e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 551.50  on 425  degrees of freedom
## Residual deviance: 154.02  on 423  degrees of freedom
## AIC: 160.02
##
## Number of Fisher Scoring iterations: 8
```

Figure 13: Logistic Regression Summary (used the first two principle components of the data as inputs)



```
   Cell Contents
|-------------------------|
|                       N |
|         N / Table Total |
|-------------------------|


Total Observations in Table:  143


                  | actual defaut
predicted default |         B |         M | Row Total |
------------------|-----------|-----------|-----------|
                B |        78 |         6 |        84 |
                  |     0.545 |     0.042 |           |
------------------|-----------|-----------|-----------|
                M |         2 |        57 |        59 |
                  |     0.014 |     0.399 |           |
------------------|-----------|-----------|-----------|
     Column Total |        80 |        63 |       143 |
------------------|-----------|-----------|-----------|
```

Figure 14: Confusion matrix of Logistic Regression (used the first two principle components of the data as inputs)

4.2. Kth Nearest Neighbors

The next method to explore is KNN as shown in Figure 15. The input features are standardized to remove any bias prior to feeding to KNN method. The prediction accuracy, as shown by confusion matrix, is around 88% with significant false negative error (Malignant cases being misclassified as Benign).

Cell Contents			
N			
N / Table Total			

Total Observations in Table: 143

	actual defaut		
predicted default	B	M	Row Total
B	80	17	97
	0.559	0.119	
M	0	46	46
	0.000	0.322	
Column Total	80	63	143

Figure 15: Confusion matrix of K-Nearest Neighbors

As shown in Figure 16, using the fist two principle components of the input features (without any standardization) helps to significantly improve the accuracy of prediction (the predication accuracy is around 94% using this method).

Cell Contents			
N			
N / Table Total			

Total Observations in Table: 143

	actual defaut		
predicted default	B	M	Row Total
B	78	6	84
	0.545	0.042	
M	2	57	59
	0.014	0.399	
Column Total	80	63	143

Figure 16: Confusion matrix of K-Nearest Neighbors (used the first two principle components of the data as inputs)

To further improve the accuracy of the KNN model we perform sensitivity analysis on k (number of nearest neighbors) to determine the optimum value of k giving rise to minimum test error (Figure 17). As shown in Figure 18, k equal to 6 results in around 96% predication accuracy.

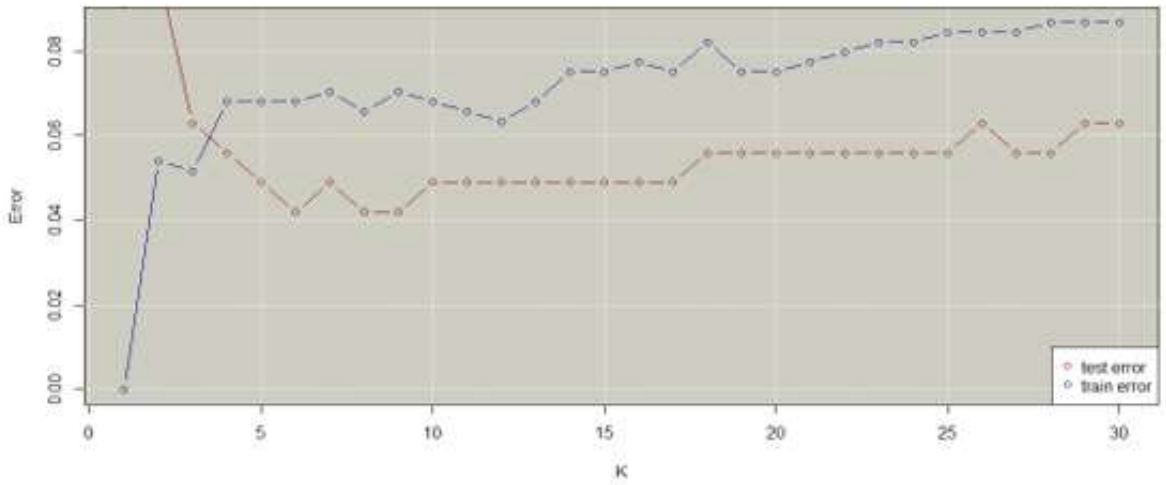


Figure 17: Test and train error as a function of number of k

Cell Contents			
	N		
N / Table Total			

Total Observations in Table: 143

	actual defaut		
predicted default	B	M	Row Total
B	78	4	82
	0.545	0.028	
M	2	59	61
	0.014	0.413	
Column Total	80	63	143

Figure 18: Confusion matrix of K-Nearest Neighbors (used the first two principle components of the data as inputs) with k=6

4.3. Decision Tree Based Methods

We additionally investigate the performance of a more complicated classification method, i.e., decision tree. As shown in Figures 19 and 20, the unpruned tree results in around 88%. We test if the model is overfitting the training data using cross-validation analysis. This method helps to decide the best pruning level to obtain minimum possible misclassification error rate. Figure 21 shows that two-node pruning results in the minimum misclassification error. However, the results do not show any improvement compared to the original (unpruned) tree. To further improve the predication accuracy, we use the first two principle components as inputs to decision tree classifier. Figures 23 through 26 show the final unpruned and pruned decision-tree-based models. We observed that the best performance is achieved when using the first two principle components as inputs with 6-node pruning of the decision tree. The predication accuracy for this case (shown in Figure 26) is around 95%.

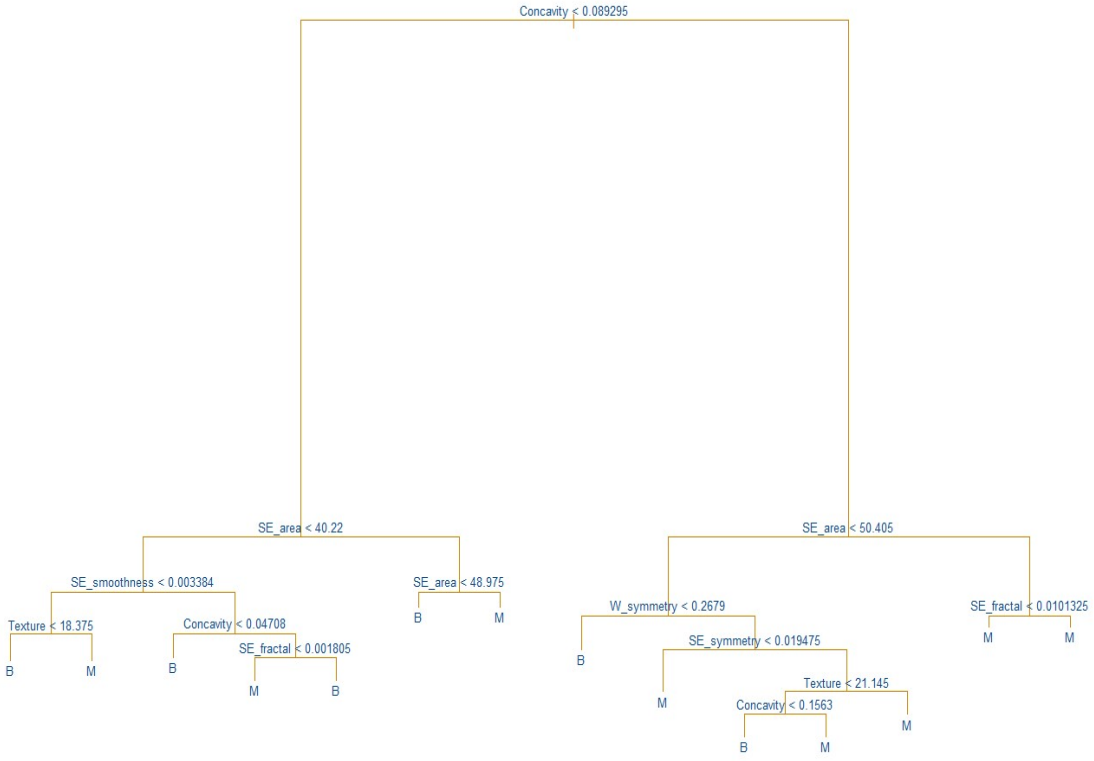


Figure 19: Regression tree analysis for the data set (unpruned tree)

```
   Cell Contents
|-------------------------|
|                       N |
|         N / Table Total |
|-------------------------|


Total Observations in Table:  143


                  | actual defaut
predicted default |         B |         M | Row Total |
------------------|-----------|-----------|-----------|
                B |        73 |         9 |        82 |
                  |     0.510 |     0.063 |           |
------------------|-----------|-----------|-----------|
                M |         7 |        54 |        61 |
                  |     0.049 |     0.378 |           |
------------------|-----------|-----------|-----------|
     Column Total |        80 |        63 |       143 |
------------------|-----------|-----------|-----------|
```

Figure 20: Confusion matrix of Regression tree analysis (unpruned tree)

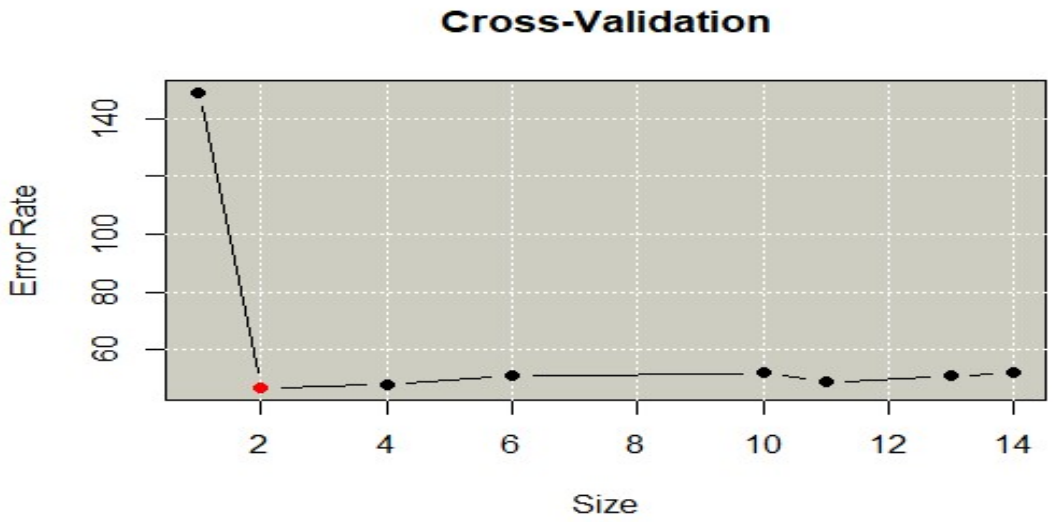


Figure 21: Cross-validation error as a function of the number of terminal nodes in the pruned tree

Cell Contents
N
N / Table Total

Total Observations in Table: 143

predicted default	actual defaut B	M	Row Total
B	71	10	81
	0.497	0.070	
M	9	53	62
	0.063	0.371	
Column Total	80	63	143

Figure 22: Confusion matrix of Regression tree analysis (two-node pruned tree)

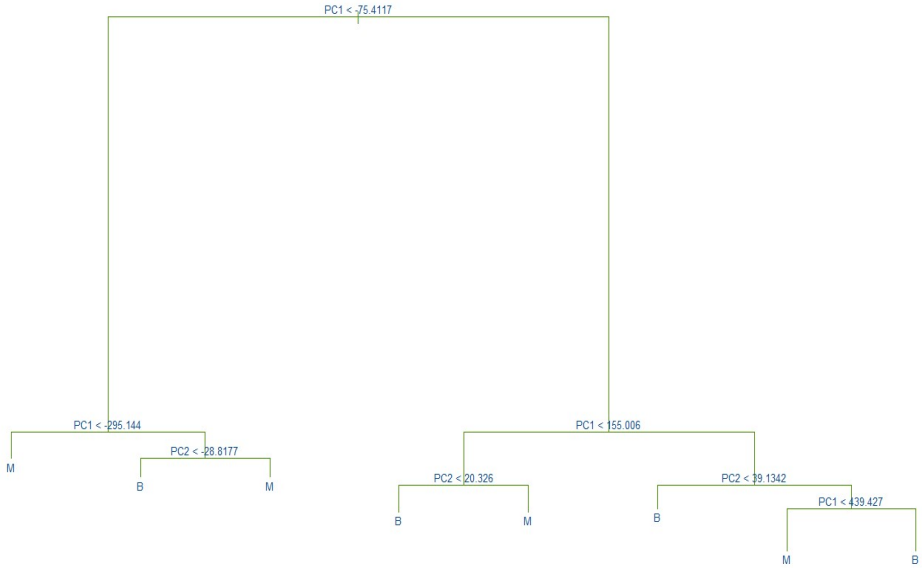


Figure 23: Regression tree analysis for the data set (unpruned tree)
Used the first two principle components of the data as inputs

```
   Cell Contents
|-------------------------|
|                       N |
|         N / Table Total |
|-------------------------|


Total Observations in Table:  143


                    | actual defaut
predicted default   |         B |         M | Row Total |
--------------------|-----------|-----------|-----------|
                  B |        78 |         9 |        87 |
                    |     0.545 |     0.063 |           |
--------------------|-----------|-----------|-----------|
                  M |         2 |        54 |        56 |
                    |     0.014 |     0.378 |           |
--------------------|-----------|-----------|-----------|
       Column Total |        80 |        63 |       143 |
--------------------|-----------|-----------|-----------|
```

Figure 24: Confusion matrix of Regression tree analysis (unpruned tree). Used the first two principle components of the data as inputs

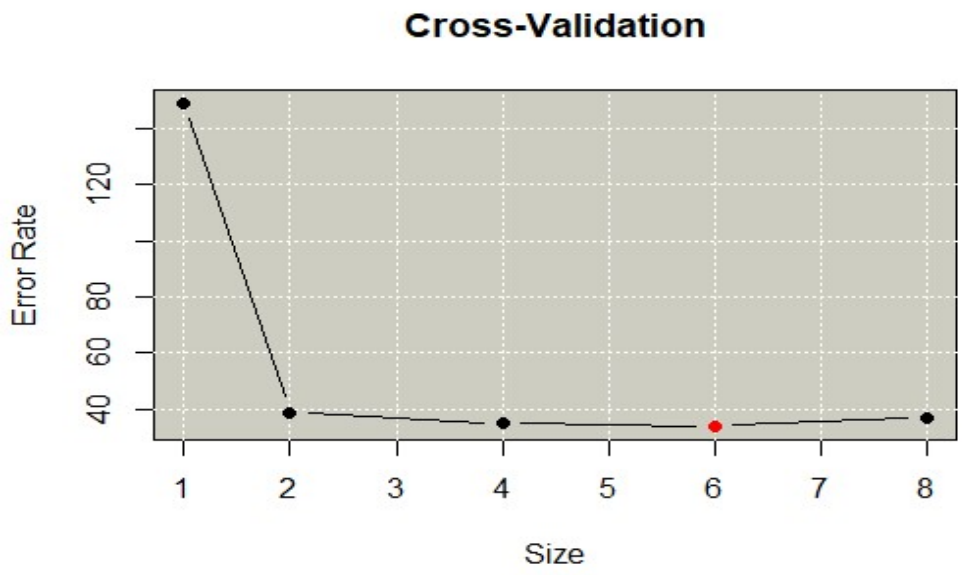


Figure 25: Cross-validation error as a function of the number of the terminal nodes in the pruned tree

Cell Contents
N
N / Table Total

Total Observations in Table: 143

predicted default	actual defaut B	actual defaut M	Row Total
B	77	5	82
	0.538	0.035	
M	3	58	61
	0.021	0.406	
Column Total	80	63	143

Figure 26: Confusion matrix of Regression tree analysis (six-node pruned tree)
Used the first two principle components of the data as inputs

4.4. Linear and Quadratic Discriminant Analyses

Even though not shown in this report, we have performed LDA and QDA classification methods to predict breast mass classes. The results are included in the R codes with theirs associated predication accuracies.

5. Summary

This study presented several automated methods for analyzing features associated with nuclear size breast mass images. Each method uses a set of analyzed features to classify their associated breast lump as benign or malignant. Various levels of predication accuracy can be achieved for each automated algorithm depending on the type and number of analyzed input features. The main challenge is the internal dependency of input features giving rise to unreliable models with high misclassification errors. Therefore, a consistent variable selection method is deemed necessary to remove redundant input features and focus on the main directions of variation in the data. To that end, we introduced two variable selection approaches, one based on PCA and the other based on clustering input features using their correlation matrix.

We subsequently developed several machine learning models with different levels of complexity to fully investigate the range of predication outcomes. We performed several analyses to test which method reach the best prediction performance. Our investigation shows that the most accurate model is logistic regression using manual variable selection approach with close to 98% accuracy. This method also gave rise to minimum false negative error rates (Malignant cases

being misclassified as Benign). There are also several other models with acceptable accuracies and various levels of complexity (e.g., pruned decision tree using principle components as input). As an interesting observation, decision tree and KNN methods showed significant predication improvement when combined with PCA dimensionality reduction technique. This confirmed the efficiency of PCA method for variable selection.

A limitation to the developed algorithms is the number of available observations used for supervised classification. A larger database could ensure a larger range of variations in the data thus capturing the true range of variations in the population. It is therefore recommended to collect a more comprehensive database to overcome that limitation.

As another recommendation, the manual variable selection technique used in this study can be replaced by an automated workflow to perform systematic grid search on the input data. This can be achieved by importing relevant R packages and performing grid search for each machine learning method.

6. References

[1] https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic)

[2] http://ftp.cs.wisc.edu/math-prog/cpo-dataset/machine-learn/cancer/WDBC/

[3] W.N. Street, W.H. Wolberg and O.L. Mangasarian. Nuclear feature extraction for breast tumor diagnosis. IS&T/SPIE 1993 International Symposium on Electronic Imaging: Science and Technology, volume 1905, pages 861-870, San Jose, CA, 1993.

[4] O.L. Mangasarian, W.N. Street and W.H. Wolberg. Breast cancer diagnosis and prognosis via linear programming. Operations Research, 43(4), pages 570-577, July-August 1995.