***Wisconsin Breast Cancer Data Set – Capstone Project- Reza Nickmanesh***

**1. Project overview:**

Breast cancer is the second most prevalent cancer diagnosed among US women, accounting for nearly one in three cancers (Siegel, Miller, & Jemal, 2016). It is also the second leading cause of cancer death among women after lung cancer (Siegel et al., 2016). The two most commonly used screening methods for breast cancer, physical examination of the breasts and mammography, offer an approximate likelihood that a lump is cancer. When these examinations are inconclusive, a procedure known as fine needle aspiration (FNA) is performed to help establish the diagnosis. This procedure involves removing a sample of fluid in the lump and the sample is investigated under microscope. The physician assesses the health of the sample based on various features of the cells (e.g. area, radius, texture, and so forth) visible in microscopic images. The objective of the procedure is to determine if the suspicious mass is benign (not life threatening) or malignant (cancerous) and accurate diagnosis at this stage could lead to implementing effective treatment to either decelerate or stop the cancer. However, this procedure lacks the repeatability required for a firm statement on the disease stage. Advancements in computer analysis of microscope images and development of powerful machine learning methods could significantly improve the accuracy of breast cancer diagnosis. Therefore, the objective of the current project is to develop predictive models that could distinguish benign and malignant cases based on features computed from a digitized image of a fine needle aspirate of a breast mass.

**2. Breast cancer Wisconsin data set**

The dataset is published by Kaggle and taken from the University of California Irvine (UCI) machine learning repository. Features are computed from a digitized image of a fine needle aspirate of a breast mass. They describe characteristics of the cell nuclei present in the image (Fig. 1).

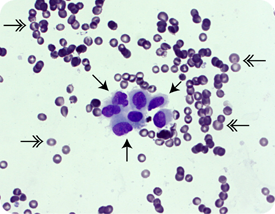


Figure 1. A representation of a digitized image of cells in FNA procedure. Various characteristics such as texture, radius, perimeter, and many others show detectable differences between benign (light purple cells) and malignant (dark purple cells) cells.

Attribute Information are listed below:

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

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

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" - 1)

The mean, standard error and "worst" or largest (mean of the three largest values) of these features were computed for each image, resulting in 30 features. All feature values are recoded with four significant digits.

**3. Data wrangling**

This data set includes 569 observations of 33 variables. There are no missing values in the main 30 variables (features obtained from imaging data). The column names were correct and there was no need for renaming. The last column of the data set was removed as it included an unknown variable with NA values. Additionally, the column for patient ID was removed as it contained no useful information for our analysis. Other than that, the data set was tidy and consisted of no missing values. Here is an overview of the structure of this data set:

'data.frame': 569 obs. of 31 variables:

$ diagnosis : Factor w/ 2 levels "B","M": 2 2 2 2 2 2 2 2 2 2 ...

$ 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 ...

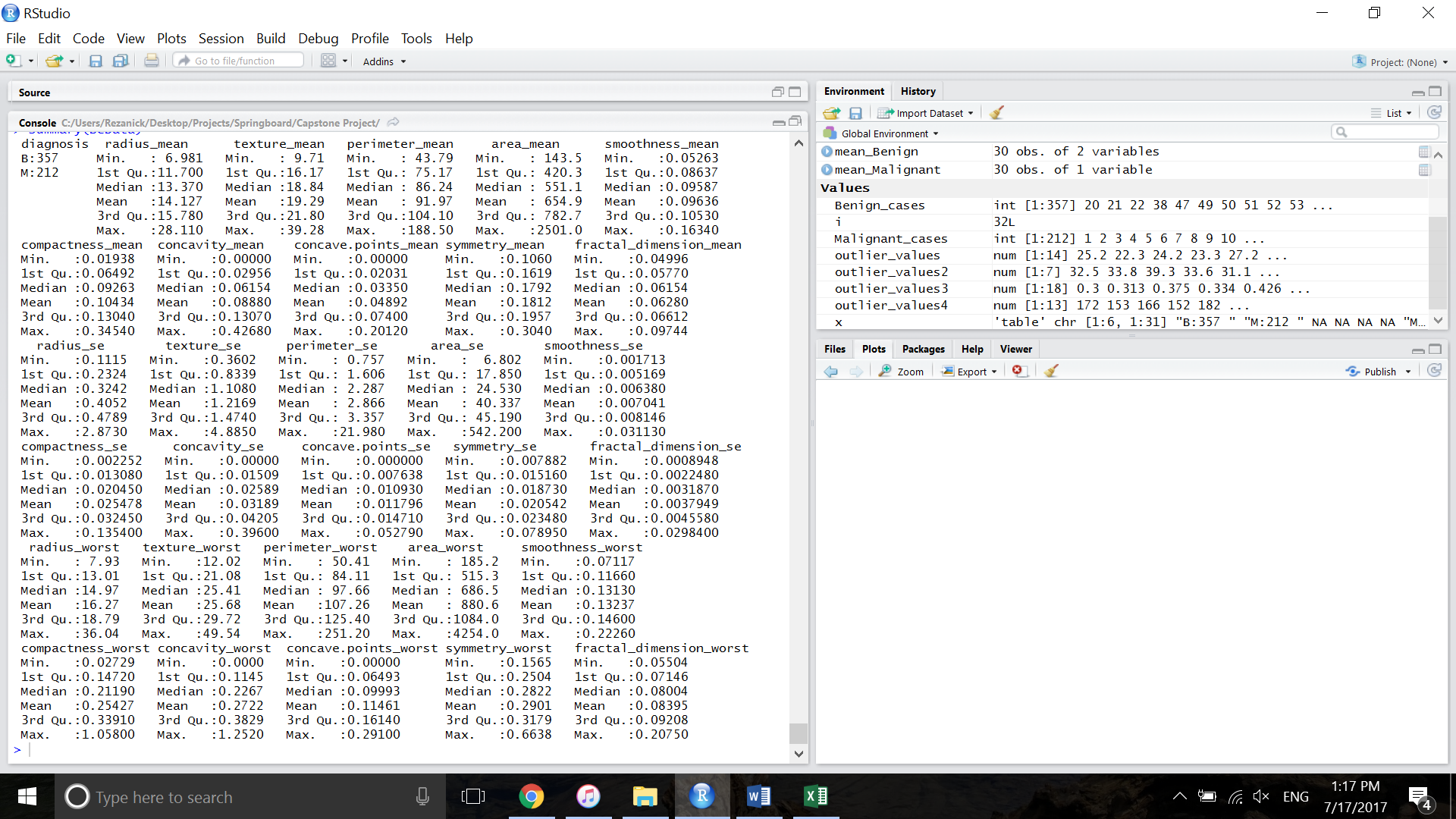
$ concave.points\_worst : num 0.265 0.186 0.243 0.258 0.163 ...

$ symmetry\_worst : num 0.46 0.275 0.361 0.664 0.236 ...

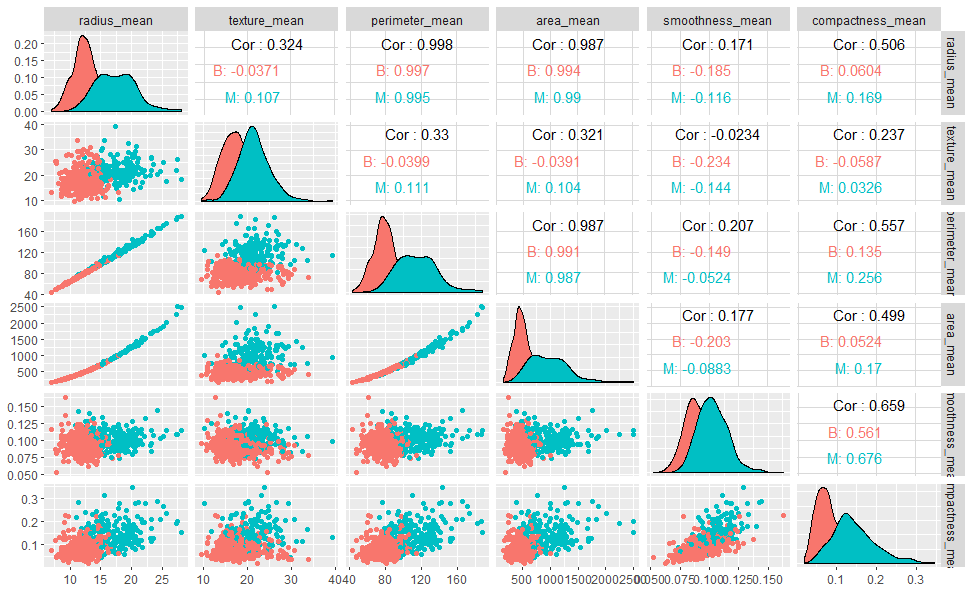
$ fractal\_dimension\_worst: num 0.1189 0.089 0.0876 0.173 0.0768 ...

**4. Data exploration and visualization**

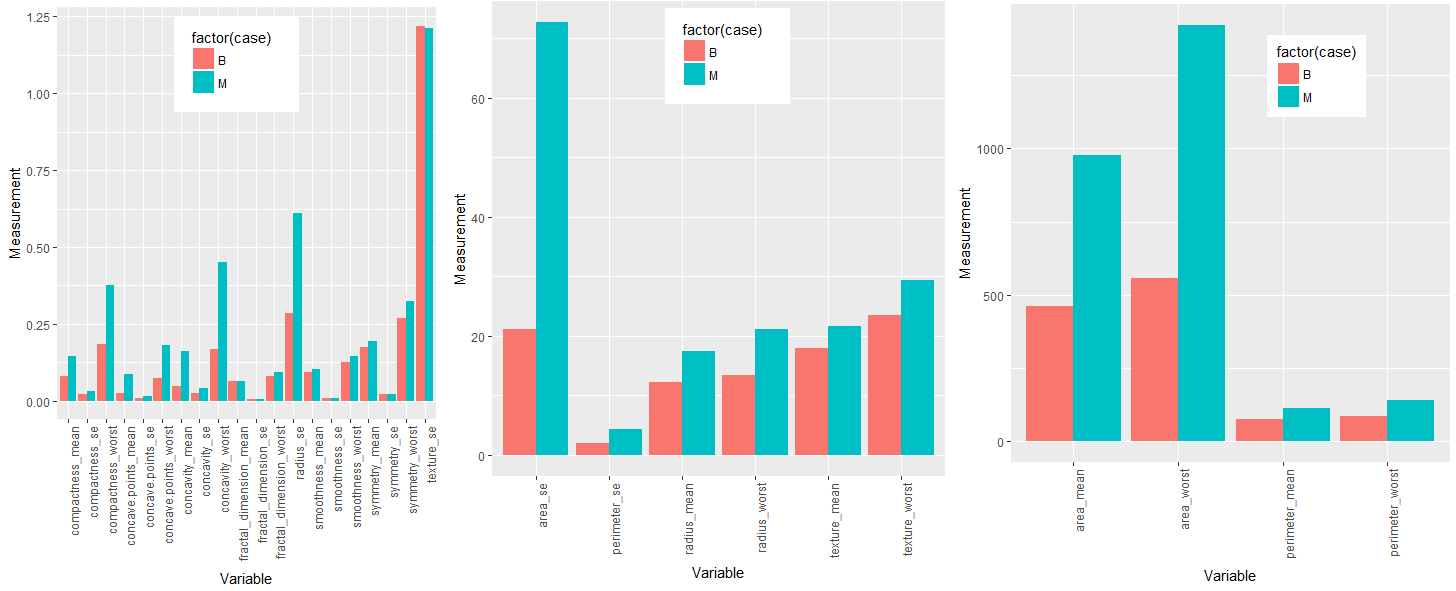
We summarize the values for each of the variables in our data set in the table below:



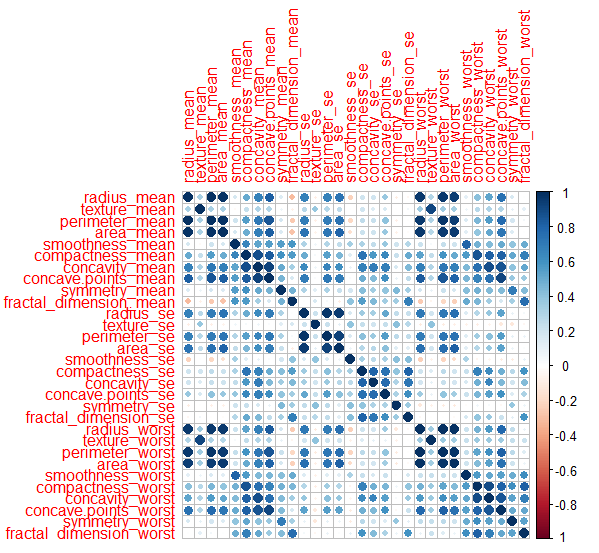
Visualization below shows the pairwise scatter plots of some of the selected features (mean radius, texture, perimeter, area, compactness, and smoothness). Number of plotted features was limited in order to keep the plots readable. The red and green points correspond to benign and malignant cases, respectively. The data suggests the existence of highly correlated variables in the data set which should be considered in development of regression models to avoid collinearity. The plot illustrates the fact that features in malignant cases have higher values than benign cases and this suggests the predictive potential of these features for classification of benign and malignant cases.



The graph below represents the mean value of the variables in B and M cases. The graphs are plotted in 3 separate graphs as the scales of variables are different. Generally speaking, variables have higher values in M cases compared to B cases (green bars vs. red bars) which suggest these computed features have potential to distinguish between B and M cases.



We need to quantify the correlation coefficients among the variables so that we could identify highly-correlated variables. This identification could help us avoid collinearity during variable selection process.



As shown above, many of the variables are highly correlated and it therefore would not be surprising if the final subset of variables (best subset for prediction) for the predictive model contain only a few of these 30 variables.

**5. Data analysis**

The objective of this capstone project is to develop predictive models that could distinguish between benign (B) and malignant (M) cases. This data set consists of features that showed promise to distinguish between benign and malignant cases. These features combined with advanced machine learning models could yield to a more robust and reliable method of diagnosis than conventional diagnostic methods.

**5.1 Preprocessing**

Values of the features have different ranges in this data set which is problematic for classification methods. Therefore, the data set was normalized to zero mean and unit variance. In addition to that, we created a principle component analysis (PCA) version of the data. The reason was that because of the correlations among the variables, some of the machine learning models might run into problem.

Importance of components%s:

PC1 PC2 PC3 PC4 PC5 PC6 PC7 PC8

Standard deviation 3.6152 2.2999 1.62978 1.40240 1.27742 1.09518 0.75951 0.68905

Proportion of Variance 0.4507 0.1824 0.09159 0.06782 0.05627 0.04136 0.01989 0.01637

Cumulative Proportion 0.4507 0.6331 0.72467 0.79249 0.84876 0.89012 0.91001 0.92638

PC9 PC10 PC11 PC12 PC13 PC14 PC15 PC16

Standard deviation 0.63653 0.59212 0.53767 0.51073 0.48148 0.38606 0.30128 0.25328

Proportion of Variance 0.01397 0.01209 0.00997 0.00899 0.00799 0.00514 0.00313 0.00221

Cumulative Proportion 0.94035 0.95244 0.96241 0.97141 0.97940 0.98454 0.98767 0.98988

PC17 PC18 PC19 PC20 PC21 PC22 PC23 PC24

Standard deviation 0.23050 0.22751 0.2223 0.17607 0.17283 0.15602 0.13571 0.13334

Proportion of Variance 0.00183 0.00178 0.0017 0.00107 0.00103 0.00084 0.00064 0.00061

Cumulative Proportion 0.99171 0.99350 0.9952 0.99627 0.99730 0.99814 0.99878 0.99939

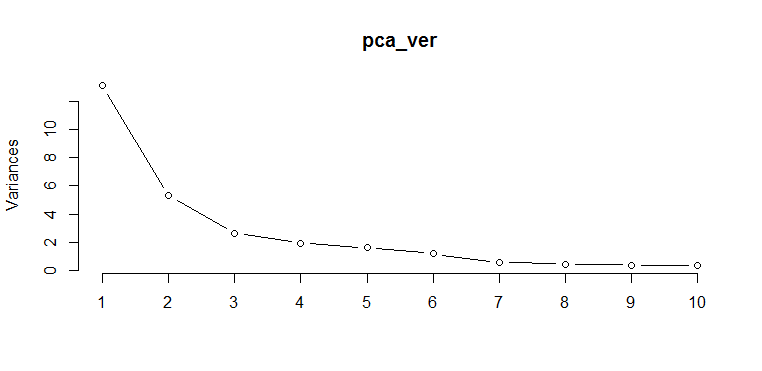
PC25 PC26 PC27 PC28 PC29

Standard deviation 0.09043 0.08392 0.04022 0.02749 0.01154

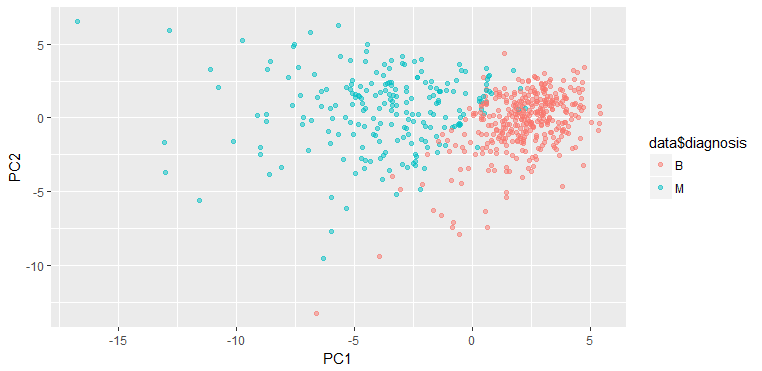
Proportion of Variance 0.00028 0.00024 0.00006 0.00003 0.00000

Cumulative Proportion 0.99967 0.99991 0.99997 1.00000 1.00000

The two first components explain 63% of the variance. We need 10 and 20 principal components to explain more than 95% and 99% of the variance, respectively. The graph below also demonstrates the contribution of the first 10 principal components to the variance.



The scatterplot below shows the distribution of the first two principal components in benign (red) and malignant (green) groups:



**5.2 Benchmark model**

The table below summarizes previous investigations which developed models based on features obtained from different image modalities to predict benign and malignant cancer cases.

|  |  |  |
| --- | --- | --- |
| Paper | Image modality | Accuracy |
| (Zhang, Wang, Liu, & Yang, 2016) | Mammography | 92.16 |
| (Zhang et al., 2016) | Mammography | 91.37 |
| (Tozaki & Fukuma, 2011) | Mammography | 90.7 |
| (Görgel, Sertbas, & Uçan, 2015) | Mammography | 90.1 |
| (Chung et al., 2013) | MRI | 93.6 |
| (Chung et al., 2013) | Ultrasound | 79.1 |

We chose the model with the highest level of accuracy in our literature review (93.6%) as our benchmark model to ensure that our final model outperforms the existing models.

**5.3 Algorithms and techniques**

Wisconsin Breast Cancer data set consists of 30 features obtained from a digitized image of a breast mass that could act as predictors in our predictive model. The objective is to predict the stage of the breast cancer that could be either benign (B) or malignant (M). Therefore, we predict a binary outcome (B/M) and multiple machine learning models could be useful in our case. Below, I briefly overview the machine learning models that will be used in this capstone project and the challenges associated with them:

1. **Logistic Regression:** Logistic regression is the method of choice since the outcome of prediction is a binary parameter. This analysis is a supervised problem as the output dataset is provided and could be used to train the machine learner. The dataset is split into two subsets of training set and test set (the split ratio is defined based on the value of 1/sqrt(number of predictors) which would be 20% for test set in our case). The most important step in this part is to select the subset of predictors that lead to the best predictive model. It has to be noted that many of the variables are highly correlated to each other as they measure various morphology parameters that are related (e.g. area is correlated with perimeter). We would use different algorithms to choose the best subset of predictors including best subset regression algorithm and also removal of highly correlated variables with a specific threshold. The measures we use to define the best model include: accuracy, sensitivity, specificity, misclassification error, area under curve (AUC), BIC, AIC, and adjusted R squared.
2. **K-means clustering:** This method is used to classify our data points into two main clusters (B and M). K-means clustering is an unsupervised algorithm as it does not require the data to be labelled. However, we would still split the data to assess the model in predicting the data that it has never seen before. The outcome measures we use include: AUC, accuracy, sensitivity, specificity, and misclassification error. In this dataset, we already knew the number of clusters (B and M), otherwise we had to use algorithms to identify the optimal number of clusters.
3. **K-nearest neighbor algorithm:** This method is considered as supervised learning as a subset of data (training set) with their outcome variable is used to train the machine learner. The outcome measures we use include: AUC, accuracy, sensitivity, specificity, and misclassification error. We also use k-fold cross validation method to identify the optimal number of nearest neighbors. Additionally, the method will be used to define the number of folds that yield minimum misclassification error.
4. **Classification and Regression Tree:** This method is a more advanced algorithm for classification that could be considered as a supervised learning algorithm. We use decision trees to train a machine learner using the training set and assess the predictive power of the model in the test set based on these measures: AUC, accuracy, sensitivity, specificity, and misclassification error. We also combine decision tree algorithm with bagging (bootstrap aggregating) algorithm and boosting (Adaboost) to improve the accuracy of our decision tree model.
5. **Random Forest:** Random forest is a learning method which constructs a multitude of decision trees at training and outputting the prediction (B/M). Random decision forests correct for decision trees’ habit of overfitting to their training set. Nodesize is a parameter needed to be determined for this algorithm and we identify its optimal value by relating the misclassification error to that parameter.

**5.3.1 Logistic Regression**

In this section, the logistic regression models are described first and they will be compared at the end according to various measures (accuracy, sensitivity, specificity, AUC, AIC, BIC, and adjusted R squared). We stated with a generalized linear model which used all 30 variables as predictors. We call this model “model.glm0”. The summary of model.glm0 surprisingly revealed that none of the variables were significant in the model. This observation could be attributed to the highly correlated variables in the data set.

Call: glm(formula = diagnosis ~ ., family = binomial, data = train)

Coefficients:

Estimate Std. Error z value Pr(>|z|)

(Intercept) -3.026e+03 1.110e+06 -0.003 0.998

radius\_mean -4.204e+02 1.707e+05 -0.002 0.998

texture\_mean -8.403e+00 3.515e+03 -0.002 0.998

perimeter\_mean 3.491e+01 2.504e+04 0.001 0.999

area\_mean 1.650e+00 5.113e+02 0.003 0.997

smoothness\_mean 3.991e+03 9.458e+05 0.004 0.997

compactness\_mean -6.220e+03 1.648e+06 -0.004 0.997

concavity\_mean -3.317e+03 5.606e+05 -0.006 0.995

concave.points\_mean 1.003e+04 8.417e+05 0.012 0.990

symmetry\_mean -3.114e+03 4.109e+05 -0.008 0.994

fractal\_dimension\_mean -3.123e+03 5.852e+06 -0.001 1.000

radius\_se 2.353e+03 3.311e+05 0.007 0.994

texture\_se -4.383e+01 1.544e+04 -0.003 0.998

perimeter\_se -2.679e+02 4.653e+04 -0.006 0.995

area\_se 1.137e+00 3.359e+03 0.000 1.000

smoothness\_se -2.234e+04 2.995e+06 -0.007 0.994

compactness\_se 5.948e+02 1.183e+06 0.001 1.000

concavity\_se 1.294e+03 9.850e+05 0.001 0.999

concave.points\_se 4.330e+04 3.796e+06 0.011 0.991

symmetry\_se 9.042e+03 2.861e+06 0.003 0.997

fractal\_dimension\_se -9.369e+04 9.981e+06 -0.009 0.993

radius\_worst -3.063e+01 3.240e+04 -0.001 0.999

texture\_worst 2.292e+01 3.736e+03 0.006 0.995

perimeter\_worst 4.829e+01 1.239e+04 0.004 0.997

area\_worst -1.843e+00 6.860e+02 -0.003 0.998

smoothness\_worst 2.240e+03 6.948e+05 0.003 0.997

compactness\_worst -1.169e+03 2.982e+05 -0.004 0.997

concavity\_worst 1.695e+03 1.879e+05 0.009 0.993

concave.points\_worst -4.088e+03 4.404e+05 -0.009 0.993

symmetry\_worst 7.361e+02 3.331e+05 0.002 0.998

fractal\_dimension\_worst 1.604e+04 1.851e+06 0.009 0.993

(AIC: 62)

Analysis of deviance table (using anova function), however, identified some of the variables as significant in the model.

Analysis of Deviance Table, Model: binomial, link: logit, Response: diagnosis

Terms added sequentially (first to last)

Df Deviance Resid. Df Resid. Dev Pr(>Chi)

NULL 455 602.31

radius\_mean 1 333.35 454 268.97 < 2.2e-16 \*\*\*

texture\_mean 1 28.87 453 240.09 7.728e-08 \*\*\*

perimeter\_mean 1 61.33 452 178.76 4.821e-15 \*\*\*

area\_mean 1 6.13 451 172.63 0.0133061 \*

smoothness\_mean 1 39.11 450 133.53 4.007e-10 \*\*\*

compactness\_mean 1 0.91 449 132.62 0.3402927

concavity\_mean 1 10.21 448 122.41 0.0013977 \*\*

concave.points\_mean 1 4.70 447 117.71 0.0301781 \*

symmetry\_mean 1 4.05 446 113.66 0.0441090 \*

fractal\_dimension\_mean 1 0.29 445 113.36 0.5874081

radius\_se 1 1.13 444 112.24 0.2887021

texture\_se 1 10.98 443 101.26 0.0009216 \*\*\*

perimeter\_se 1 0.02 442 101.24 0.8952215

area\_se 1 11.47 441 89.77 0.0007065 \*\*\*

smoothness\_se 1 6.73 440 83.04 0.0095049 \*\*

compactness\_se 1 3.81 439 79.23 0.0510127 .

concavity\_se 1 9.88 438 69.35 0.0016668 \*\*

concave.points\_se 1 0.02 437 69.33 0.8926335

symmetry\_se 1 1.08 436 68.25 0.2984668

fractal\_dimension\_se 1 1.62 435 66.63 0.2032144

radius\_worst 1 31.15 434 35.48 2.391e-08 \*\*\*

texture\_worst 1 1.27 433 34.21 0.2590317

perimeter\_worst 1 5.97 432 28.24 0.0145730 \*

area\_worst 1 0.01 431 28.23 0.9234020

smoothness\_worst 1 4.05 430 24.19 0.0442791 \*

compactness\_worst 1 24.19 429 0.00 8.742e-07 \*\*\*

concavity\_worst 1 0.00 428 792.96 1.0000000

concave.points\_worst 1 792.96 427 0.00 < 2.2e-16 \*\*\*

symmetry\_worst 1 0.00 426 0.00 0.9990246

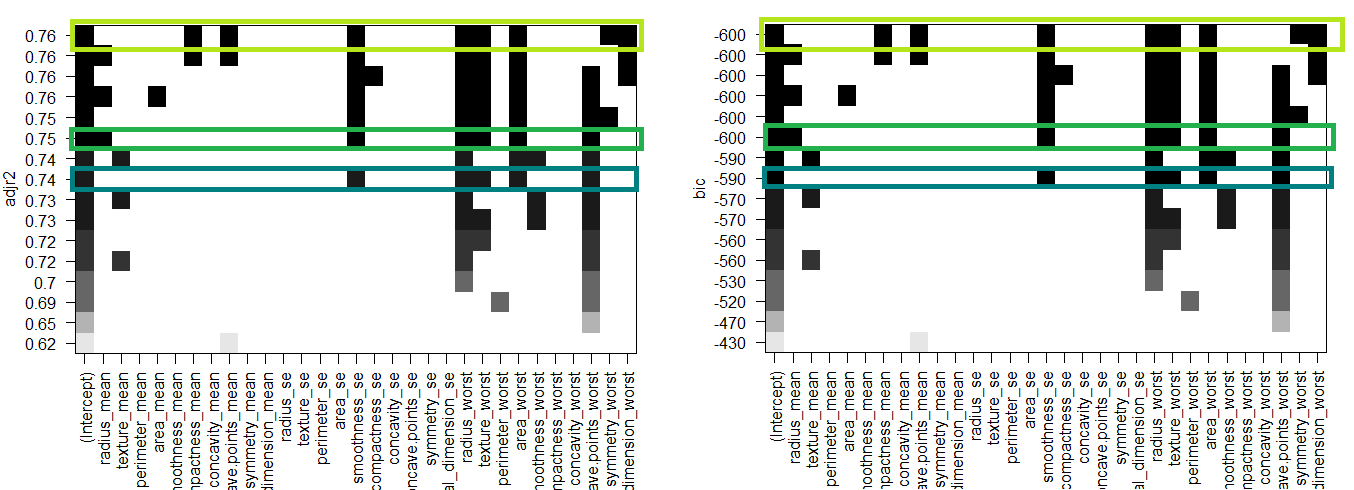
fractal\_dimension\_worst 1 0.00 425 0.00 0.9990360

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

This observation is not surprising as summary function and anova function evaluate the model based on different approaches. Summary function is comparing the marginal significance of each variable while the anova test is comparing their significance sequentially (is variable A significant in the presence of only the intercept; is variable B significant in the presence of the intercept and variable A; and so on).

In the next step, we evaluated all possible combinations of the variables and their corresponding logistic regression models (leaps function). We rank the subset of regression models according to their BIC and adjusted R squared value. BIC is a criterion for model selection and the model with lowest BIC is preferred. When fitting models, it is possible to increase the likelihood by adding parameters, but doing so may result in overfitting. Both BIC and AIC attempt to resolve this problem by introducing a penalty term for the number of parameters in the model; the penalty term is larger in BIC than in AIC. The graph below shows the best 2 regression models for each possible number of variables.



The first 8 subsets have approximately same values for BIC and adjusted r square. Three subsets of variables with good performance (based on BIC and adjusted R squared metrics) are highlighted with green boxes. A simpler model is preferred and here we therefore have the selected variables among all 30 variables: “smoothness\_se”, “radius\_worst”, “texture\_worst”, “area\_worst”, “concave.points\_worsts”. A new logistic regression model with these selected variables was generated.

Call: glm(formula = diagnosis ~ smoothness\_se + radius\_worst + texture\_worst +

area\_worst + concave.points\_worst, family = "binomial", data = train)

Coefficients:

Estimate Std. Error z value Pr(>|z|)

(Intercept) -20.42629 11.46848 -1.781 0.0749 .

smoothness\_se 257.36554 185.29825 1.389 0.1649

radius\_worst -1.13639 1.32747 -0.856 0.3920

texture\_worst 0.30362 0.06798 4.466 7.97e-06 \*\*\*

area\_worst 0.02481 0.01367 1.815 0.0695 .

concave.points\_worst 68.88804 14.02733 4.911 9.06e-07 \*\*\*

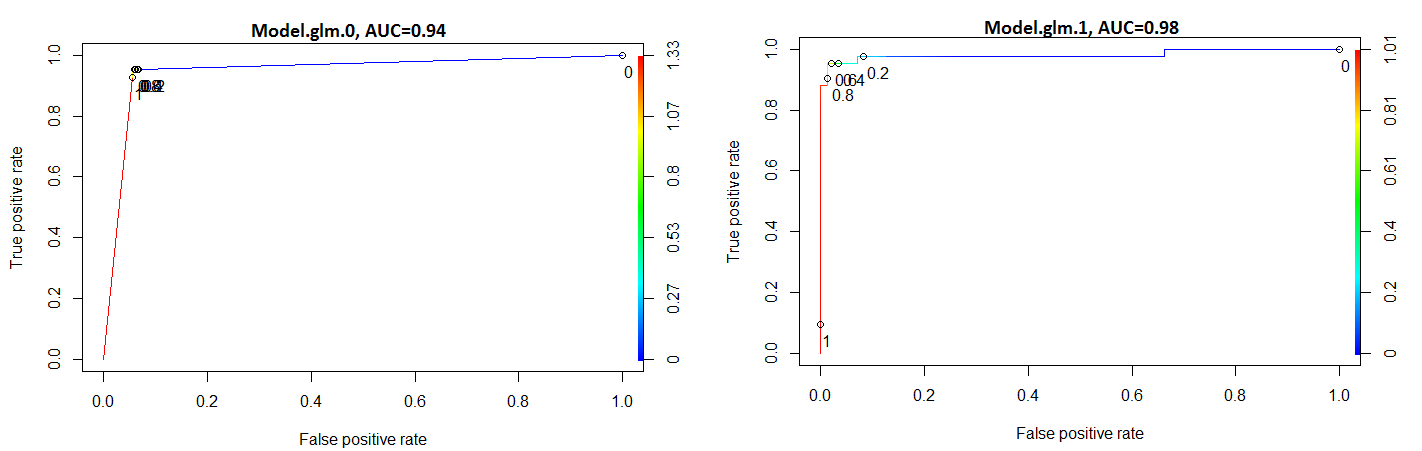
---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(AIC: 75.221)

The table below compares the two logistic regression models with respect to various measures:

|  |  |  |
| --- | --- | --- |
| Measure/Model | Model.glm.0 | Model.glm.1 |
| Accuracy | 0.95 | 0.97 |
| Sensitivity | 0.93 | 0.95 |
| Specificity | 0.97 | 0.98 |
| Misclassification Error (%) | 4.4% | 2.65% |
| AUC | 0.94 | 0.98 |
| AIC | 62 | 75 |
| BIC | -510 | -590 |
| Adjusted R squared | 0.77 | 0.74 |



Based on these comparisons, the logistic regression model which included the selected variables (based on best subset regression variables) proved to have a better performance than the model with all variables.

We created logistic regression models using principal component version of the data set. The table below summarizes the performance of the three models developed in this part:

|  |  |  |  |
| --- | --- | --- | --- |
| Measure/Model | Model.glm.pca.0  (diagnosis~.) | Model.glm.pca.1  (diagnosis~PC(1:10)) | Model.glm.pca.2  (diagnosis~PC(1:20)) |
| Accuracy | 0.95 | 0.95 | 0.96 |
| Sensitivity | 0.95 | 0.95 | 0.95 |
| Specificity | 0.95 | 0.95 | 0.97 |
| Misclassification Error (%) | 5.3% | 5.3% | 3.5% |
| AUC | 0.95 | 0.99 | 0.99 |

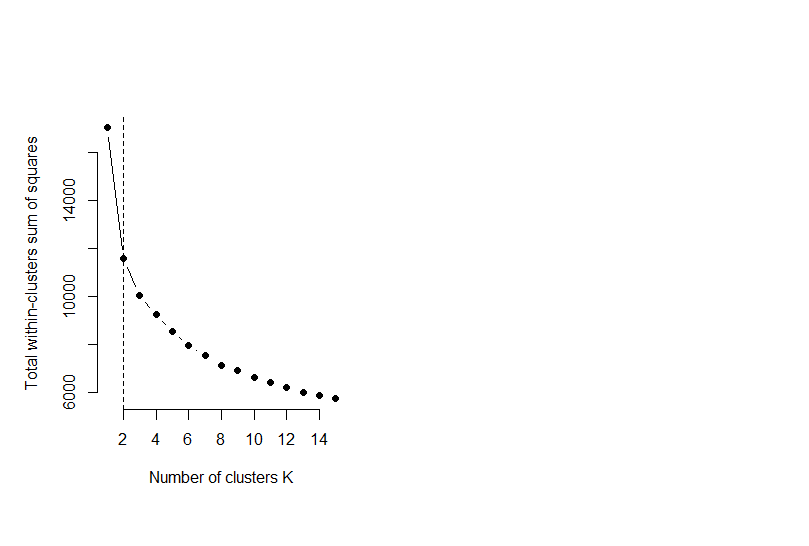
The model with the first 20 principal components (explaining more than 99% of the variance) had a better performance over the other two. It should be noted that the model with all principle components showed a slightly weaker performance (5.3% vs. 4.4% error) than the model with all the original variables.

In conclusion, among all the logistic regression models, the model.glm.0 which used the best subset of variables had the best performance.

**5.3.2 K-means Clustering**

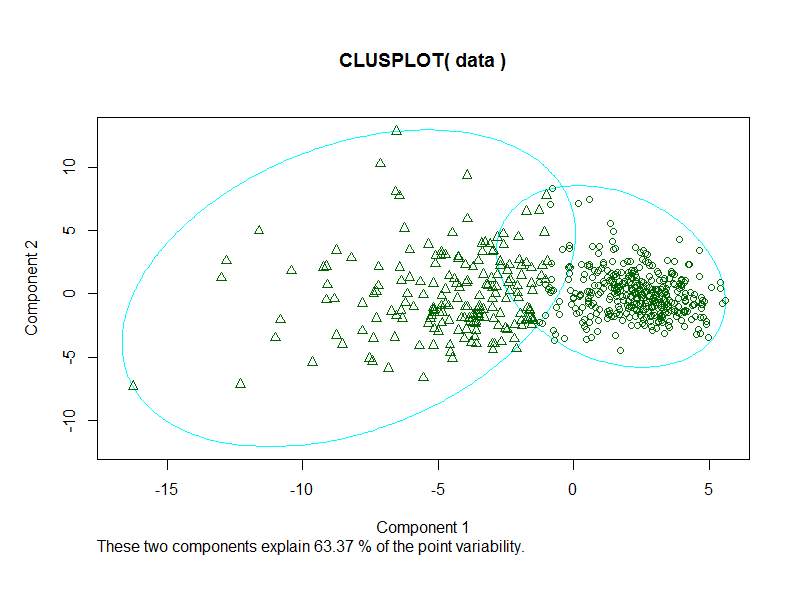
K Means Clustering is an unsupervised learning algorithm that tries to cluster data based on their similarity. Unsupervised learning means that there is no outcome to be predicted, and the algorithm just tries to find patterns in the data. The first step is to determine the number of clusters which in our case we already knew that the data points could be classified into two clusters (B and M). For educational purposes, we check if the common methods for determining the number of clusters also confirm the selection of 2 clusters.

The plot of total within-groups sums of squares against the number of clusters is generated to identify the number of clusters. The graph below suggests k=2 as an abrupt change in the slope is observed at that point.



We created an scaled version of the data and used kmeans() function and applied it to the whole data set to classify the points. The table summarizes the performance of the clustering algorithm.

|  |  |
| --- | --- |
| Measure/Model | Model.kmeans |
|  |  |
| Accuracy | 0.91 |
| Sensitivity | 0.82 |
| Specificity | 0.96 |
| Misclassification Error (%) | 8.9% |



**5.3.3 k-nearest algorithm**

Knn is a supervised learning algorithm where the result of new instance query is classified based on majority of k-nearest neighbor category. The classification is using majority vote among the classification of the k objects. An essential step is to determine the optimal k value in this algorithm. We used 10-fold validation approach with 3 repetitions and used the accuracy was used as the metric to be compared amongst the subsets. Train function in “caret” package was used to tune the k parameter.

Based on the 10-fold validation procedure, the top three models had k=5, 7 , and 9 with the corresponding accuracies of 0.92, 0.93, and 0.931, respectively. K=9 was selected for our final knn analysis.

|  |  |
| --- | --- |
| Measure/Model | Model.knn |
| Accuracy | 0.96 |
| Sensitivity | 0.90 |
| Specificity | 0.99 |
| Misclassification Error (%) | %4.4 |
| AUC | 0.99 |

* + 1. **Tree-based Algorithms**

Decision tree learning uses a decision tree (as a predictive model) to go from observations about an item (represented in the branches) to conclusions about the item's target value (represented in the leaves). Decision tree algorithm is not a robust method for prediction and is prone to overfitting. Here, we use three common methods that could enhance the performance of a decision tree.

1. **Random forest** is an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of overfitting to their training se
2. **Bagging** builds multiple models (decision trees) from different subsamples of the training dataset.
3. **Boosting** builds multiple models (typically of the same type) each of which learns to fix the prediction errors of a prior model in the chain.

The table below represents the performance of each of the models described above. We used adaptive boosting algorithm for boosting.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Model/Measure | Cross-Validation Accuracy | Out of sample Accuracy | Out of sample Sensitivity | Out of sample Specificity | Out of sample Misclassification Error (%) |
| Decision Tree | 0.91 | 0.92 | 0.88 | 0.94 | 7.9% |
| Bagging | 0.95 | 0.95 | 0.95 | 0.94 | 5.3% |
| Boosting | 0.96 | 0.98 | 0.95 | 1 | 1.8% |
| Random Forest | 0.95 | 0.98 | 0.95 | 1 | 1.8% |

**5.4 Model Comparison**

The table below shows the comparison among the models developed in this project for prediction of breast cancer in our data set. Random forest and adaptive boosting algorithms demonstrated the best performance with respect to the metrics of accuracy, sensitivity, specificity, and misclassification error.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model/Measure | Accuracy | Sensitivity | Specificity | Misclassification Error (%) |
| Best logistic model | 0.97 | 0.95 | 0.98 | 2.7% |
| Best logistic model with principal components | 0.96 | 0.95 | 0.97 | 3.5% |
| Kmeans | 0.91 | 0.82 | 0.96 | 8.9% |
| Knn | 0.96 | 0.90 | 0.99 | 4.4% |
| Decision Tree | 0.92 | 0.88 | 0.94 | 7.9% |
| Bagging | 0.95 | 0.95 | 0.94 | 5.3% |
| Boosting | 0.98 | 0.95 | 1 | 1.8% |
| Random Forest | 0.98 | 0.95 | 1 | 1.8% |

**6 Result and Discussion**

In this capstone project, we attempted to develop predictive models for prediction of the stage of breast cancer (benign and malignant) using various algorithms. Except for the Kmeans and decision tree methods, all other models successfully outperformed the baseline model with the accuracy of 93.6%. Our results suggest that both adaptive boosting and random forest algorithms had the best performance with the accuracy of 98%. Kmeans algorithm as the only unsupervised learning method used in this project had the lowest accuracy (92%). Decision tree method had the second lowest accuracy (92%). However, implementing other machine learning methods (bagging, boosting, and random forest) combined with decision tree learning significantly improved the accuracy of the model (92% vs. 95%, 98%, and 98%). Considering the equal performance of the best top two models (random forest and boosting), random forest could be selected as the best method for diagnosis of breast cancer in our data set because of its faster processing time compared to adaptive boosting algorithm.

**7 Future Work**

This data lacks the information about the progression of breast cancer in benign and malignant cases. Therefore, some questions such as “when/what percentage of benign cases develop breast cancer in future?” or “when is the best time for intervention in malignant cases?” could not be answered with this data set. Including those information in the data set could potentially lead to development of more accurate and reliable methods for diagnosis.

It would worthwhile to use ensemble learning methods used in this study and combined with other potential methods (e.g. SVM, neural network, and so forth) to see if using a voting model based on all of the methods could improve the accuracy of our current model.

**8 Acknowledgement**

I would like to thank my mentor, Tom Hopper, for his valuable feedback throughout this course.

**9 Addendum – R code**

#------------------------------------Load and clean the data set-----------------------------------

data=read.csv("data.csv")

str(data)

data <- data[,-1]

data <- data[,-32]

contrasts(data$diagnosis)

str(data)

#------------------------------------Splitting the data into training and test sets---------

set.seed(22)

split=sample.split(data$diagnosis, SplitRatio = 0.80)

train=subset(data, split==TRUE)

test=subset(data, split==FALSE)

#-----------------------Logistic Regression------------------------

model.glm.0=glm(diagnosis~.,data=train, family = binomial)

predict.glm.0=predict(model.glm.0, type = 'response', newdata = test)

table(test$diagnosis, predict.glm.0 > 0.5)

tapply(predicttest, test$diagnosis, mean)

#Plot ROC curve using ROCR and calculation of auc

pr <- prediction(predicttest, test$diagnosis)

prf <- performance(pr, measure = "tpr", x.measure = "fpr")

plot(prf)

plot(prf, colorize=TRUE, print.cutoffs.at=seq(0,1,0.1), text.adj=c(-0.2, 1.7))

auc <- performance(pr, measure = "auc")

auc <- auc@y.values[[1]]

auc

# anova test

anova(model.glm.0, test="Chisq")

#-------------------Variable Selection----------------------------

leaps <- regsubsets(diagnosis~., data=train, nbest = 2)

plot(leaps, scale="adjr2")

plot(leaps, scale="bic")

#------New Logistic Regression Model with the best subset of variables---------

model.glm.1=glm(diagnosis~smoothness\_se+radius\_worst+texture\_worst+area\_worst+concave.points\_worst, data=train, family = "binomial")

predict.glm.1=predict(model.glm.1, type = 'response', newdata = test)

table(test$diagnosis, predict.glm.1 > 0.5)

pr <- prediction(predicttest, test$diagnosis)

prf <- performance(pr, measure = "tpr", x.measure = "fpr")

plot(prf)

plot(prf, colorize=TRUE, print.cutoffs.at=seq(0,1,0.1), text.adj=c(-0.2, 1.7))

auc <- performance(pr, measure = "auc")

auc <- auc@y.values[[1]]

auc

#--------PCA and Logistic Regression-----

pca\_ver=prcomp(data[,2:31], center = TRUE, scale. = TRUE)

plot(pca\_ver, type="l")

summary(pca\_res)

pca\_df <- as.data.frame(pca\_res$x)

pca\_df$diagnosis=data$diagnosis

train=subset(pca\_df, split==TRUE)

test=subset(pca\_df, split==FALSE)

model.glm.pca.0=glm(diagnosis~., data=train, family = "binomial")

predict.glm.pca.0=predict(model.glm.pca.0, type = 'response', newdata = test)

table(test$diagnosis, predict.glm.0 > 0.5)

model.glm.pca.1=glm(diagnosis~., data=train, family = "binomial")

predict.glm.pca.1=predict(model.glm.pca.1, type = 'response', newdata = test)

table(test$diagnosis, predict.glm.1 > 0.5)

model.glm.pca.2=glm(diagnosis~., data=train, family = "binomial")

predict.glm.pca.2=predict(model.glm.pca.2, type = 'response', newdata = test)

table(test$diagnosis, predict.glm.2 > 0.5)

#---------------------K-means Clustering-------------------

scale\_ver=as.data.frame(scale(data[2:31]))

# Compute and plot wss for k = 2 to k = 15

k.max <- 15 # Maximal number of clusters

wss <- sapply(1:k.max,

function(k){kmeans(scale\_ver, k, nstart=10 )$tot.withinss})

plot(1:k.max, wss,

type="b", pch = 19, frame = FALSE,

xlab="Number of clusters K",

ylab="Total within-clusters sum of squares")

abline(v = 2, lty =2)

model.kmeans <- kmeans(scale\_ver, 2, nstart = 20)

clusplot(data, model.kmeans$cluster)

table(data$diagnosis, model.kmeans$cluster)

#---------------------K nearest algorithm-----------------

train0=train[,-1]

test0=test[,-1]

train\_lables=train$diagnosis

control <- trainControl(method="repeatedcv", number=10, repeats=3)

seed <- 7

metric <- "Accuracy"

set.seed(seed)

#Finding the optimal knn parameters

fit.knn <- train(diagnosis~., data=train, method="knn", metric=metric, trControl=control)

summary(fit.knn)

model.knn <- knn(train = train0, test = test0, cl=train\_lables, k=9)

#------------Tree based algorithms------------

fit.tree <- train(diagnosis~., data=train, method="rpart", metric=metric, trControl=control)

prp(tree)

predicttree=predict(tree, newdata = test, type = "class")

table(test$diagnosis, predicttree)

fit.ada <- train(diagnosis~., data=train, method="adaboost", metric=metric, trControl=control)

fit.treebag <- train(diagnosis~., data=train, method="treebag", metric=metric, trControl=control)

fit.rf <- train(diagnosis~., data=train, method="rf", metric=metric, trControl=control)

#-------------------------------------------