Name: Han Han

Class Section: CSE 5243

Semester: Spring 2019

Instructor: Jason Van Hulse

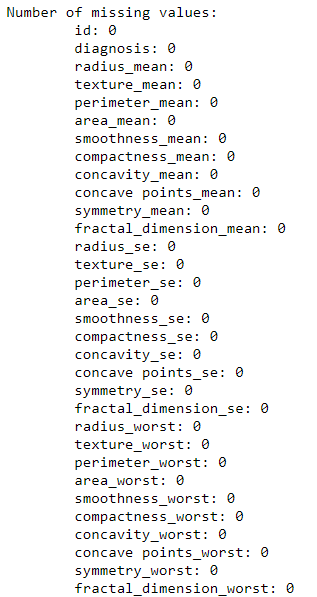
Date: 04/11/2019

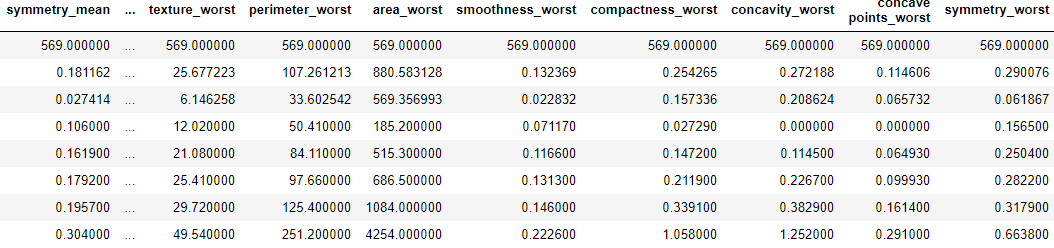
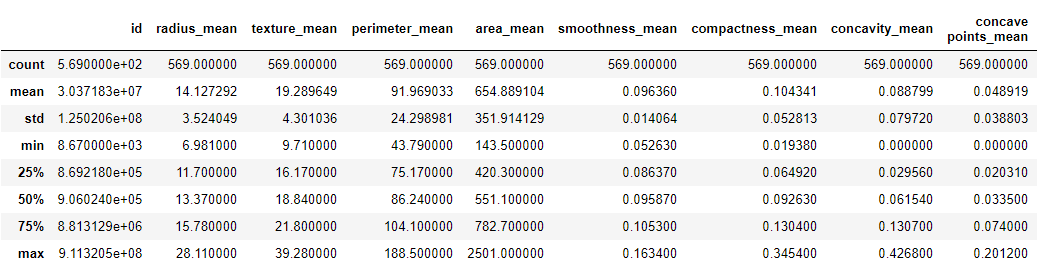
**Business Understanding:**

The dataset is from the Kaggle Website. Each row represents each person with specific breast mass information. It contains total 357 benign and 212 malignant data and 32 attributes including the person’s ID, the class label Diagnosis, radius\_mean, texture\_mean, perimeter\_mean, area\_mean, smoothness\_mean, compactness\_mean, concavity\_mean, concave points\_mean, symmetry\_mean, fractal\_dimension\_mean, radius\_se, texture\_se, perimeter\_se, area\_se, smoothness\_se, compactness\_se, concavity\_se, concave points\_se, symmetry\_se, fractal\_dimension\_se, radius\_worst, texture\_worst, perimeter\_worst, area\_worst, smoothness\_worst, compactness\_worst, concavity\_worst, concave points\_worst, symmetry\_worst, and fractal\_dimension\_worst. In general, the attributes from the third one to the last consist of three main categories that is the mean, standard error, and the worst of the mass information including radius of the mass, the texture of mass, the perimeter of the mass, the area of the mass, the smoothness of the mass, the compactness of the mass, the concavity of the mass. In details, the radius is the mean of distances from center to points on the perimeter. The texture is the standard deviation of gray-scale values. The perimeter and area are the just the perimeter and the area of the mass. The smoothness is the local variation in radius lengths. The compactness is the square of perimeter divides area and minus 1.0. The concavity is the severity of concave portions of the contour. The concave points are the number of concave portions of the contour. The fractal dimension is the coastline approximation -1. I come up with some questions in my mind by looking through this dataset. One of the questions I think about is that are there any correlated attributes? Because it is hard to analyze 32 attributes. I can take out some attributes if they are highly correlated. One of the important questions is that what is the best model for predicting the breast cancer? It is very necessary to analyze multiple models for this dataset to conclude the best model to fit in this dataset.

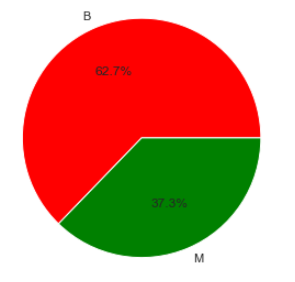
**Data understanding:**





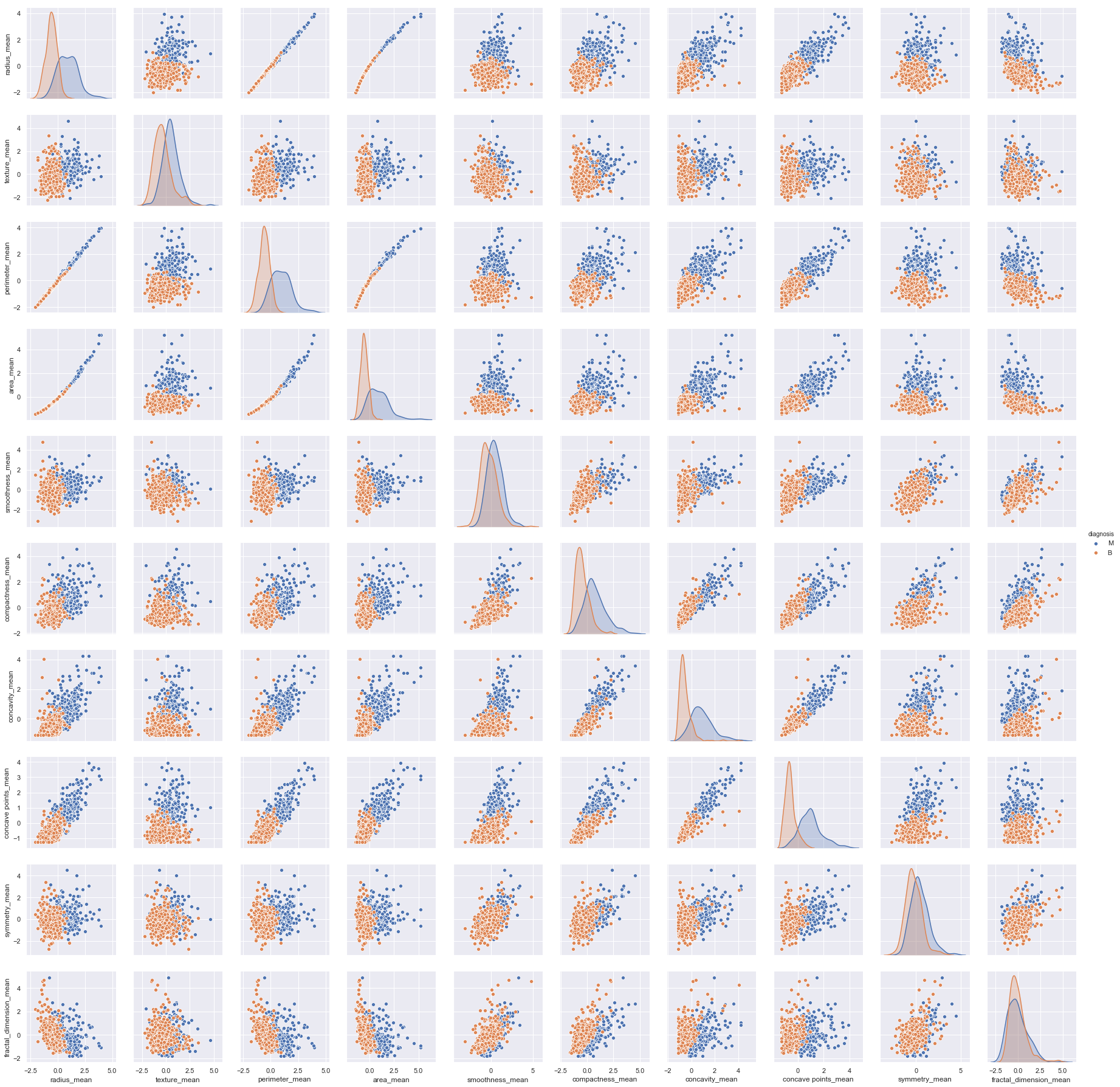


Since there is no missing or duplicated data, I don’t have to do data cleaning. All these data are float point data type based. In addition, data transformation is not needed since all the data are numeric value. To have a general idea of the distribution of this dataset, I first look at the class label Diagnosis. The pie graph of the class label Diagnosis shows the distribution is fair and reasonable.



Furthermore, Since the dataset has three general categories which are the mean, the standard error, and the worst of these attributes, I partition these attributes into three data frames. To understand the relationship of the attributes in visual forms, I make pair graphs for these three data categories.

**The mean of the attributes:**



**The standard error of the attributes:**

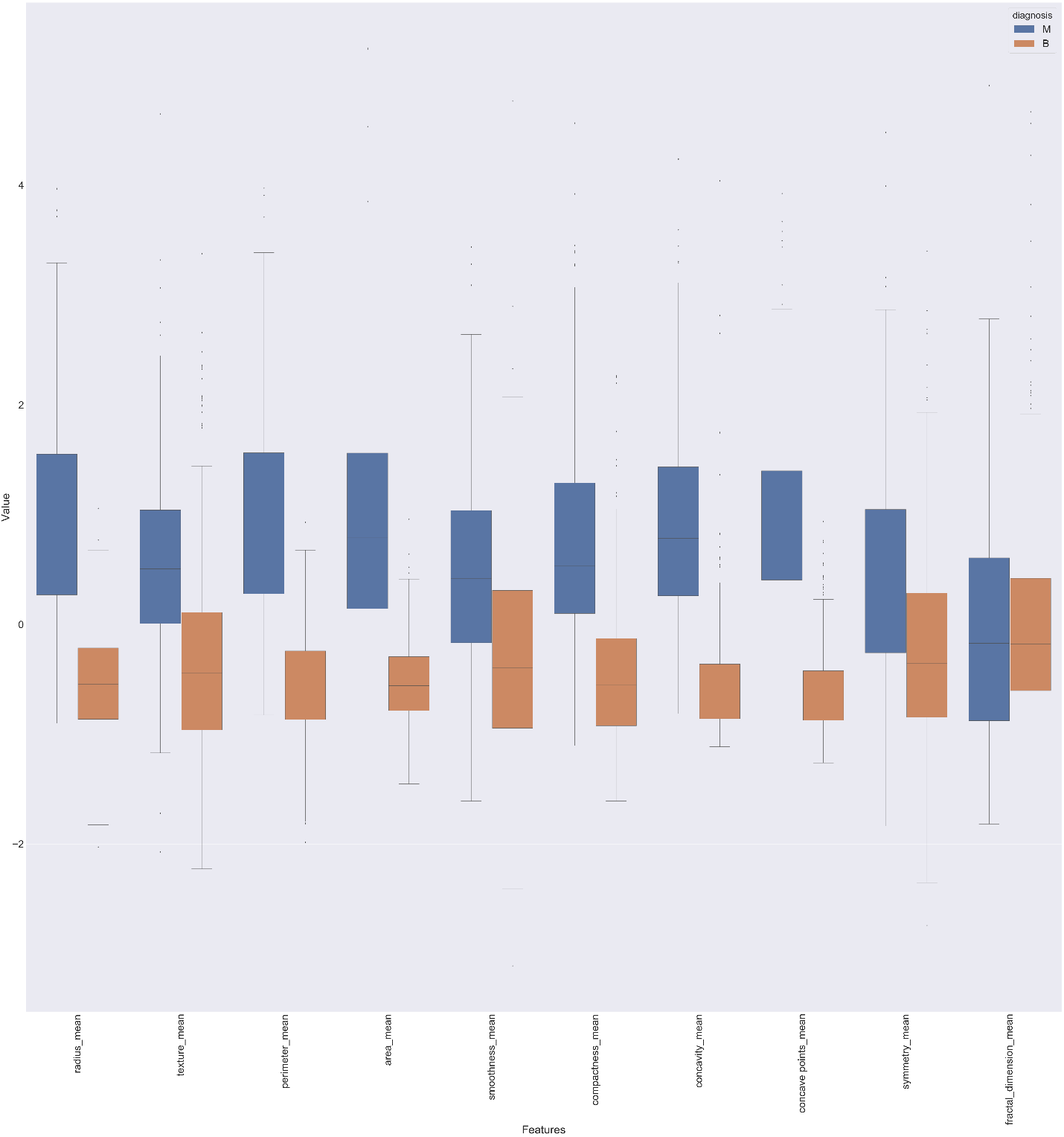


**The worst of the attributes:**

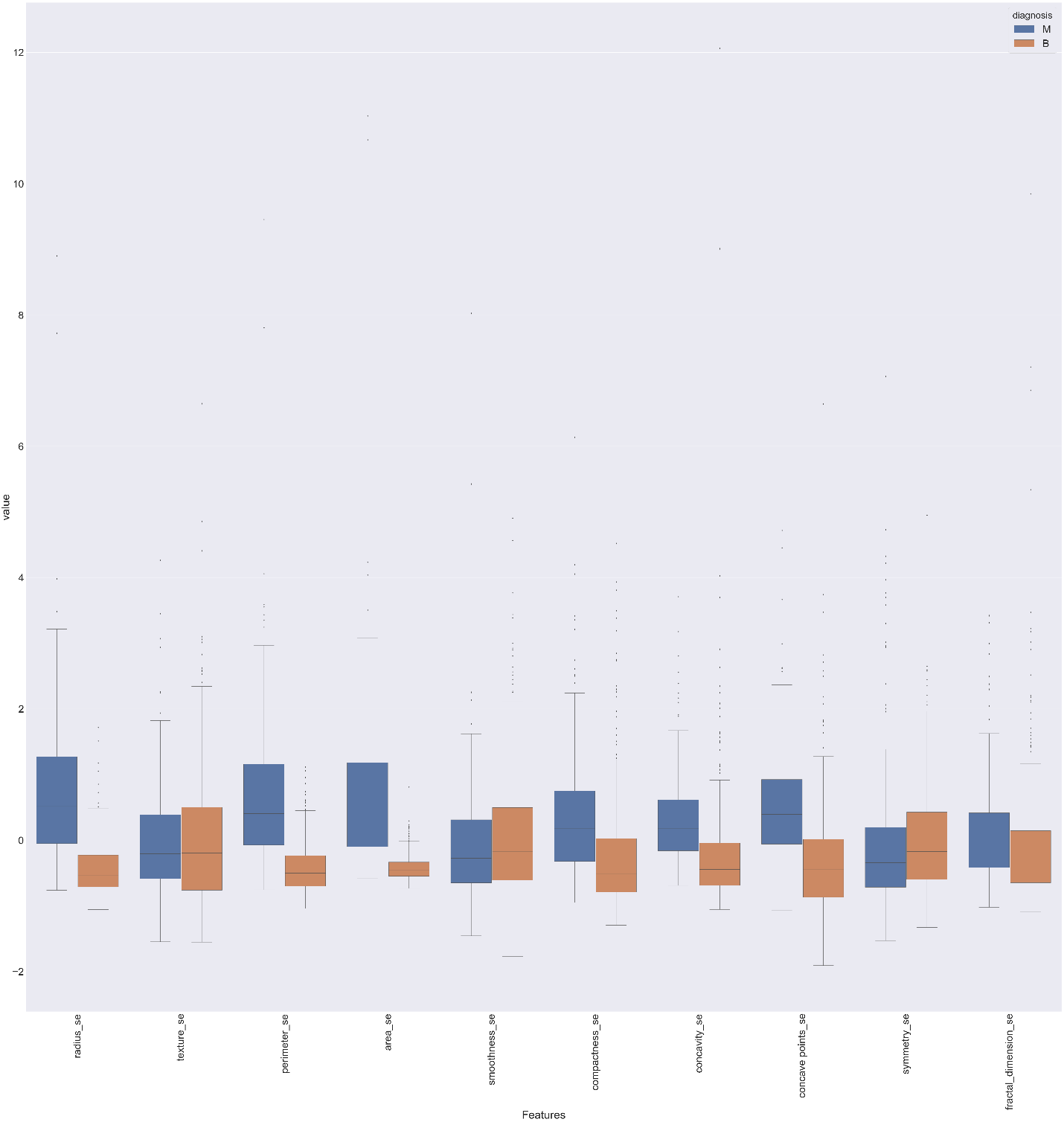


By looking through the all of these three graphs, I can see clearly that there are very strong positive correlations between the perimeter and the radius, the area and the perimeter, and the area and the perimeter. Furthermore, of the mean, standard error, and worst categories, the compactness and the concavity are also strongly positive correlated. The compactness and the concave points are also strongly positive correlated. The concavity and the concave points are also strongly positive correlated. Other attributes are not highly correlated in the pair graph visually.

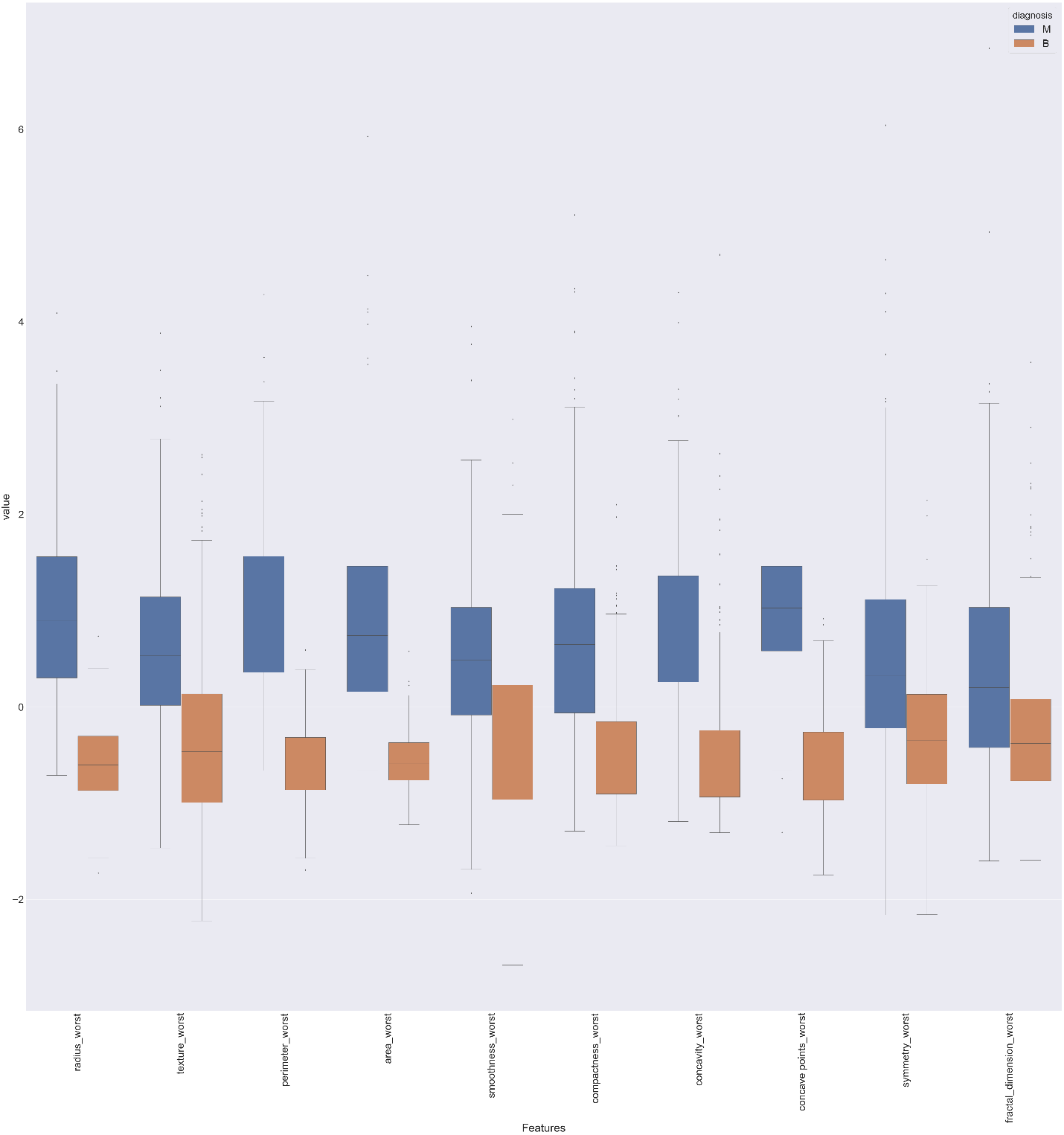
**The boxplot of the mean of attributes with the class label.**



**The boxplot of the standard error of attributes with the class label.**



**The boxplot of the worst of attributes with the class label.**



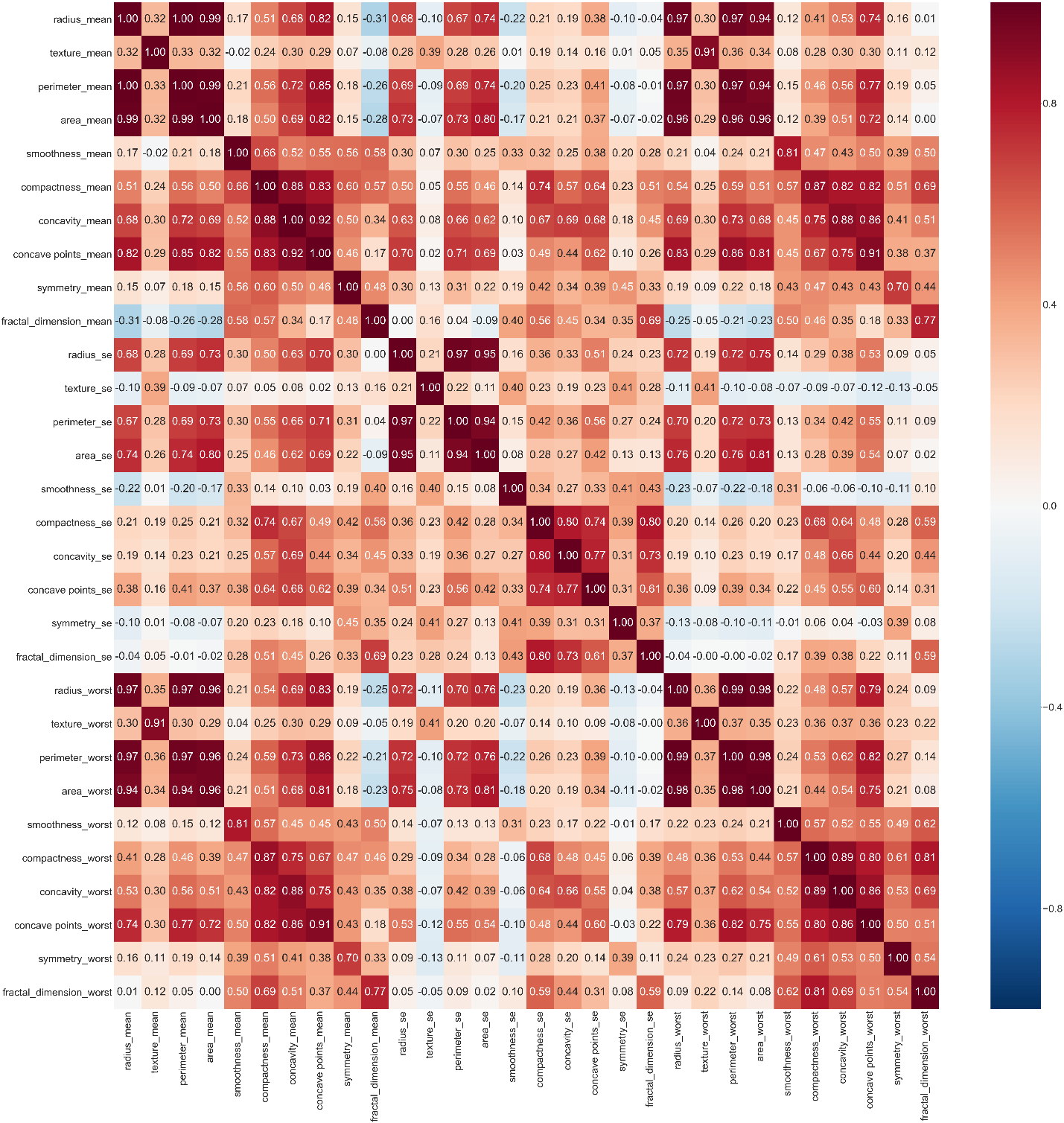
From the boxplot graph of the mean attributes, I can clearly see that the attributes are very closely related to the class label Diagnosis except for the last two attributes, the symmetry\_mean and the fractal\_dimension\_mean, since these attributes don’t have large overlap with the respect of the class label Diagnosis.

From the boxplot graph of the standard error attributes, I can clearly see that the attributes are very closely related to the class label Diagnosis except for the texture\_se and the smoothnes\_se, the symmetry\_se, and the fractal\_dimension\_se, since these attributes don’t have large overlap with the respect of the class label Diagnosis.

From the boxplot graph of the worst attributes, I can clearly see that all attributes are very closely related to the class label Diagnosis since these attributes don’t have large overlap with the respect of the class label Diagnosis.

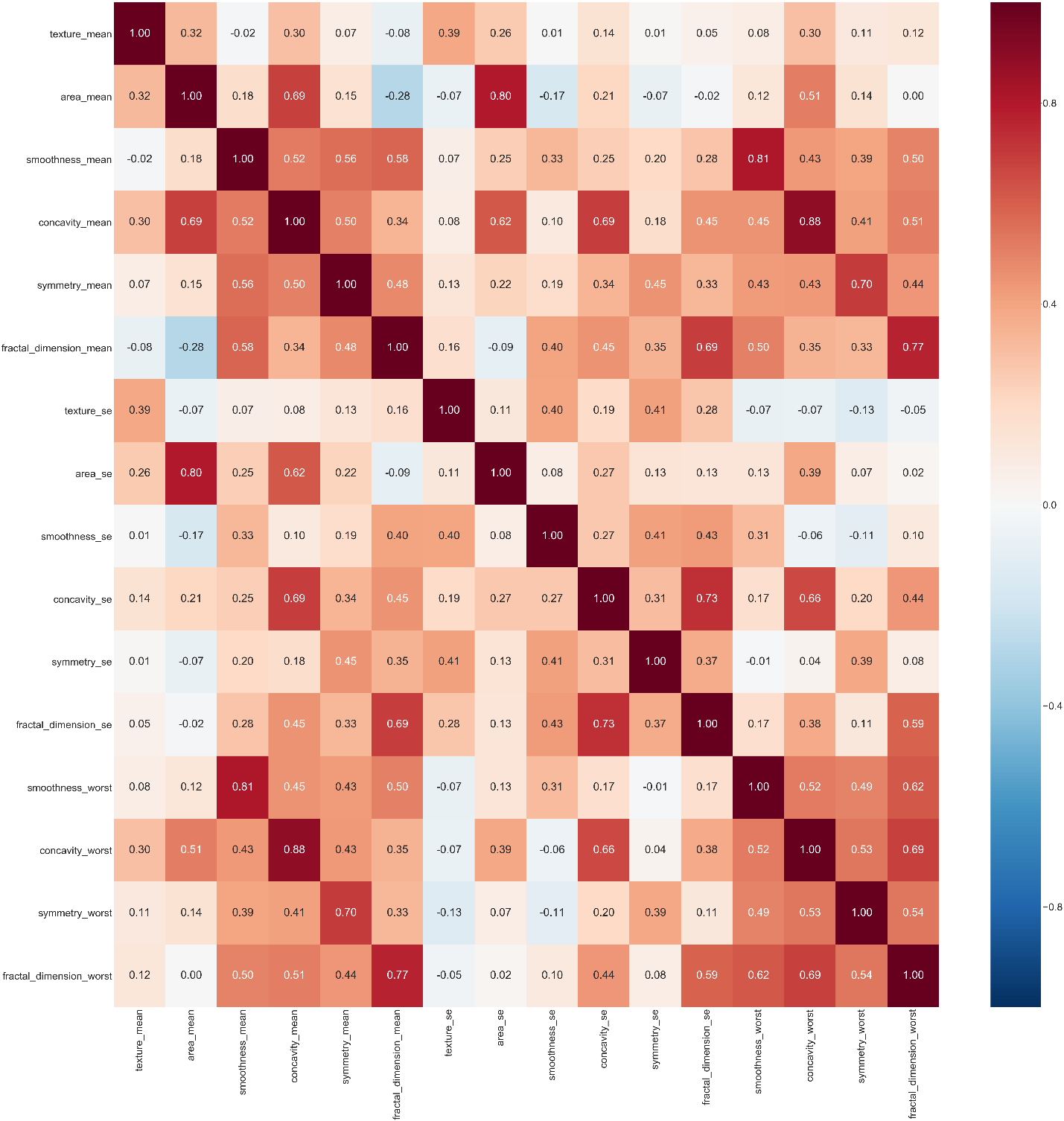
Therefore, I can conclude that all the attributes are useful and important to evaluate the class label Diagnosis.

**The heap map of all attributes before attributes deleted:**



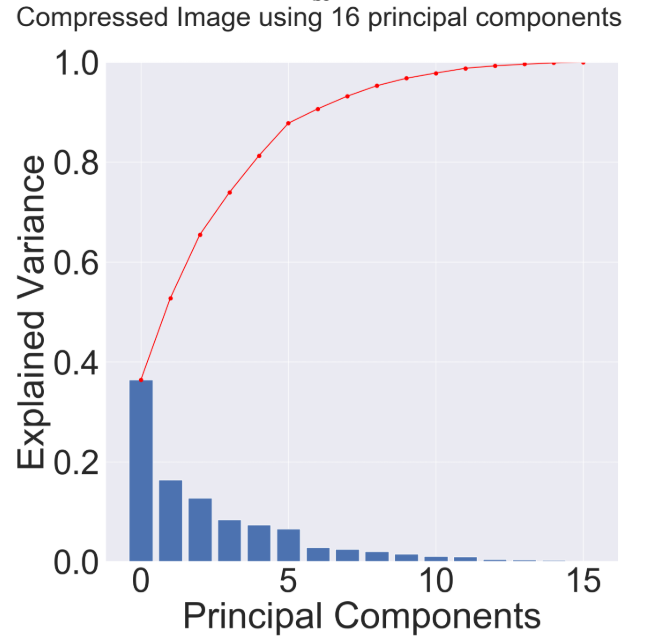
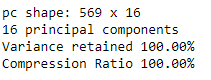
The heap map of all the attributes can provide a clear vision of the correlations between two attributes. In the heap map graph, of all the mean, standard error, and worst categories, the radius and the perimeter are highly correlated. The area and the radius are highly correlated. The perimeter and area are highly correlated. Compactness\_mean, concavity\_mean and concave points\_mean are correlated. radius\_se, perimeter\_se and area\_se are correlated. radius\_worst, perimeter\_worst and area\_worst are correlated. Compactness\_worst, concavity\_worst and concave points\_worst. Compactness\_se, concavity\_se and concave points\_se are correlated. texture\_mean and texture\_worst are correlated. area\_worst and area\_mean are correlated. Then based on the boxplot graphs and considered the importance to the class label Diagnosis, I decide to take out a list of attributes, perimeter\_mean, radius\_mean, compactness\_mean, concave points\_mean, radius\_se, perimeter\_se, radius\_worst, perimeter\_worst, compactness\_worst, concave points\_worst, compactness\_se, concave points\_se, texture\_worst, area\_worst. Therefore, I only need to analyze 16 attributes with the class label Diagnosis.

**The heat map of the attributes after some attributes deleted:**

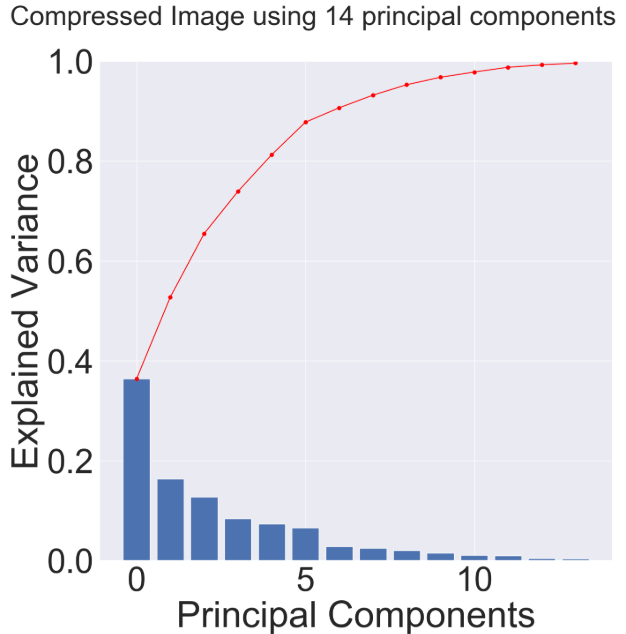
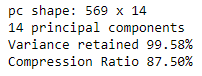


Even though there are still some relatively high correlations, these attributes are necessary factors for the further analysis. Therefore, I decide to keep the attributes as stated above. In addition, considering the amount of the 16 attributes, I decide to use principle component analysis to reduce the dimension of this dataset for a better performance in the classification stage.

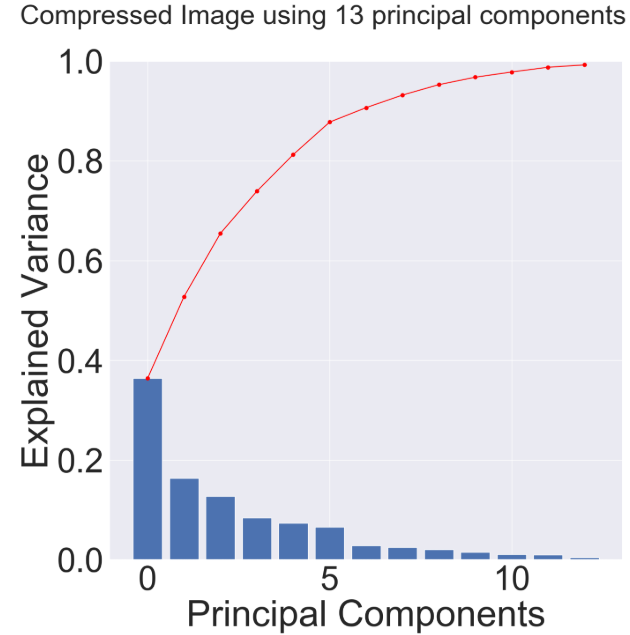
**Principle Component Analysis of 16 attributes:**



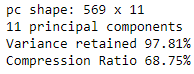
**Principle Component Analysis of 14 attributes:**

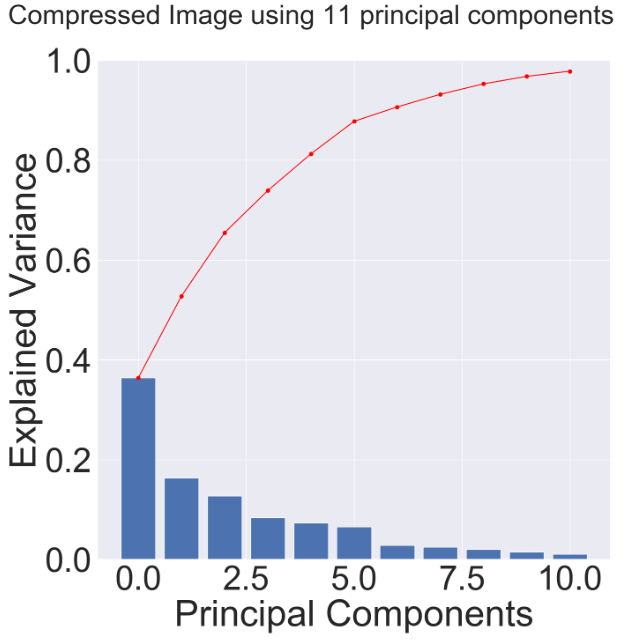


**Principle Component Analysis of 13 attributes:**

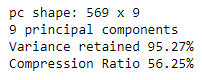


**Principle Component Analysis of 11 attributes:**

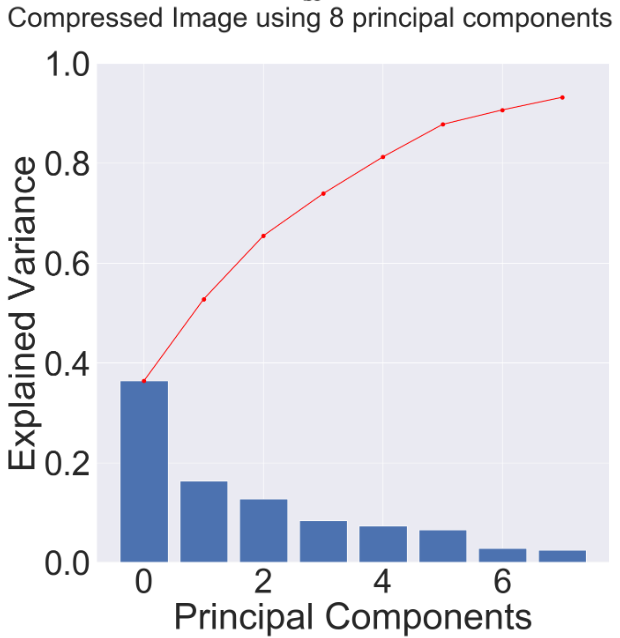
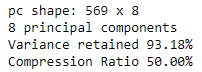




**Principle Component Analysis of 9 attributes:**



**Principle Component Analysis of 8 attributes:**



In order to keep the originality of the dataset as much as possible, I decide to keep 9 principal components with 95.27% of variance retained and 56.25%.

**Data Normalization:**

Before the classification stage, it is reasonable and necessary to normalize the whole dataset. Because the differences between values of the attributes are very high. It will interfere the classification performance. For example, SVM is distance-based classification algorithm, large differences between each attribute value will affect the classification performance. Therefore, I normalize the whole dataset from 0 to 1 for a better classification performance.

**Evaluation Approach:**

In this adult dataset, SVM, Decision tree, and Naïve Bayes are the three models to analyze this dataset to find the best model. In classification stage, I decide to use two validation methods to find a better performance model: The Simple Split approach, and the Cross-Validation approach. Based on these two validation approaches, I can have a solid evidence about the performance of the classifiers.

By using the Single partition evaluation approach for the training set, I am able to divide the 70% of the total dataset into a training set and 30% of the total dataset into a test set. Furthermore, in the training set, I divide 30% of training data into a validation set and use the rest of the training set to train models. In this way, I can use my validation test set to compare the performance of my four models so that I can choose the best one to feed in the test set.

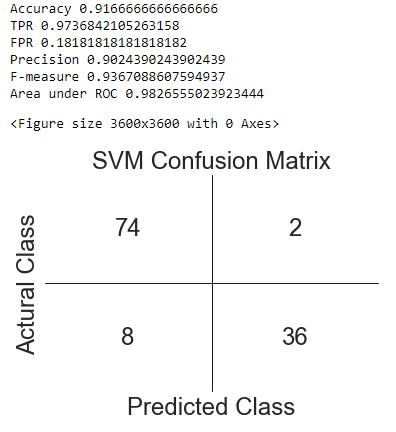
**Simple split into train, validation, and test sets:**

**SVM:**

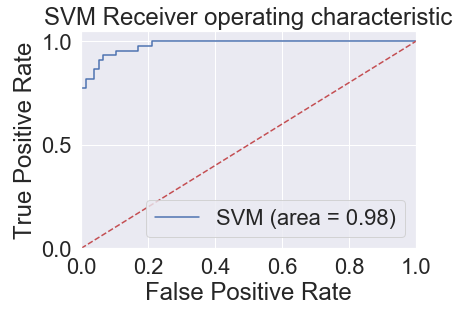
After splitting the dataset into the training set, validation set, and test set explained above, I import SVM classifier from the sklearn package and fit the training data into SVM and use validation test set to determine the statistic information. For the SVM model, all the parameters are default except for the probability parameter. In order to calculate the ROC curve, I have to get the probability from the SVM model. Therefore, I set the probability parameter to true. I also set up a timer to monitor the time spent during prediction. Since the dataset is not very huge, the time on prediction is relatively small. The first graph below shows the classification report and the second graph contains the accuracy, True Positive, and False Positive rates, Precision rate, and F-measure of SVM model on Validation Dataset. The third graph shown below is the ROC of the SVM model for the Validation set.

**Statistic Information and the Confusion Matrix:**





**ROC Curve:**

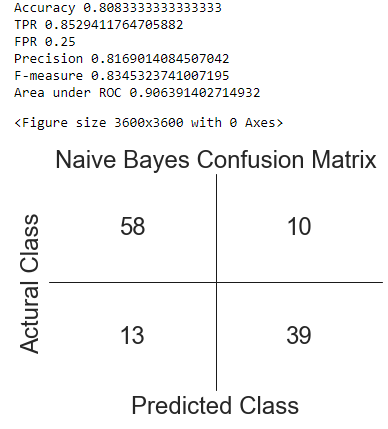


**Naïve Bayes:**

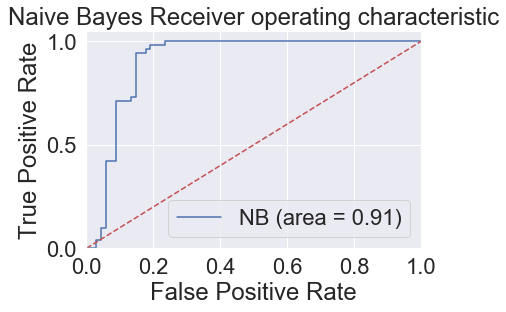
With the same splitting approach, I feed the training data into the Naïve Bayes model from sklearn.naive\_bayes package and use validation test set to determine the statistic information. I also set up a timer to monitor the time spent during prediction. Since the dataset is not very huge, the time on prediction is relatively small. The first graph below shows the classification report and the second graph contains the accuracy, True Positive, and False Positive rates, Precision rate, and F-measure of Naïve Bayes model. The third graph shown below is the ROC of the Naïve Bayes model.

**Statistic Information and the Confusion Matrix:**





**ROC Curve:**

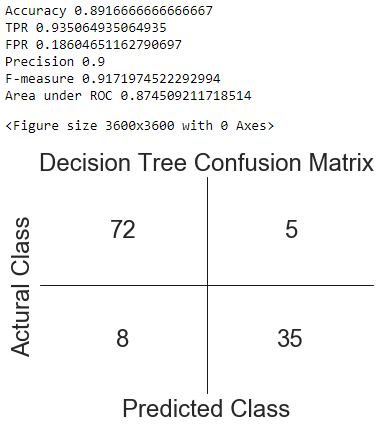


**Decision Tree:**

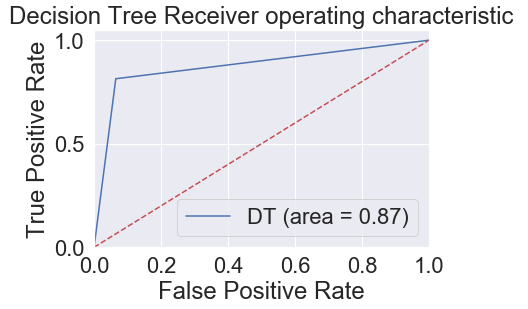
With the same splitting approach, I feed the training data into the Decision Tree model from the sklearn package and use the validation test set to determine the statistic information. I also set up a timer to monitor the time spent during prediction. Since the dataset is not very huge, the time on prediction is relatively small. The first graph below shows the classification report and the second graph contains the accuracy, True Positive, and False Positive rates, Precision rate, and F-measure of the Decision Tree model. The third graph shown below is the ROC of the Decision Tree model.

**Statistic Information and the Confusion Matrix:**





**ROC Curve:**



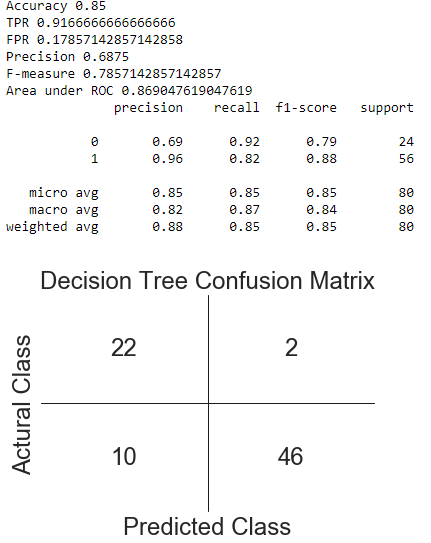
**Cross Validation with 5-Fold:**

**Decision Tree:**

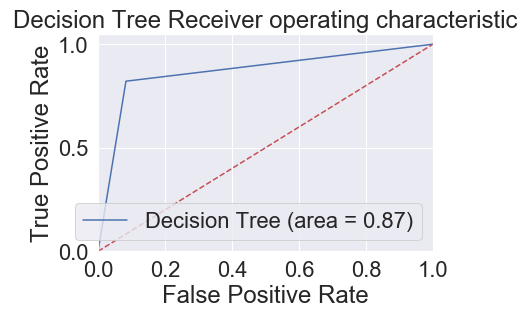
With the same splitting approach, I feed the training data into the Decision Tree model from the sklearn package and use the validation test set to determine the statistic information. I also set up a timer to monitor the time spent during prediction. Since the dataset is not very huge, the time on prediction is relatively small. The first graph below shows the classification report and the second graph contains the accuracy, True Positive, and False Positive rates, Precision rate, and F-measure of the Decision Tree model. The third graph shown below is the ROC of the Decision Tree model.

**First iteration: Statistic Information and Confusion Matrix**



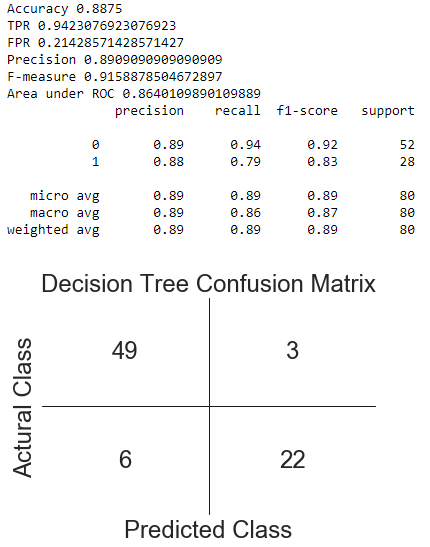


**ROC Curve:**

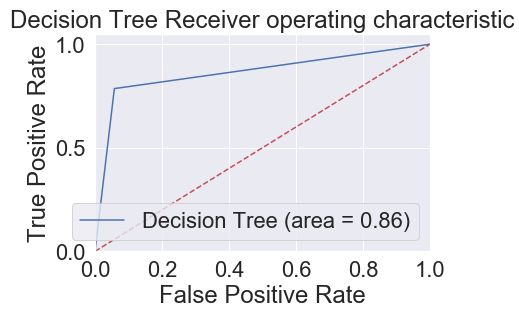


**Second iteration: Statistic Information and Confusion Matrix**



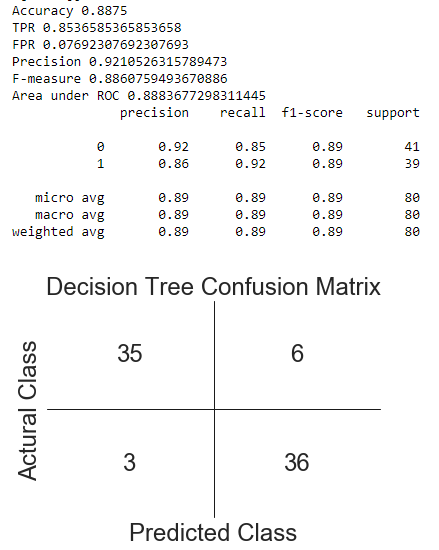


**ROC Curve:**

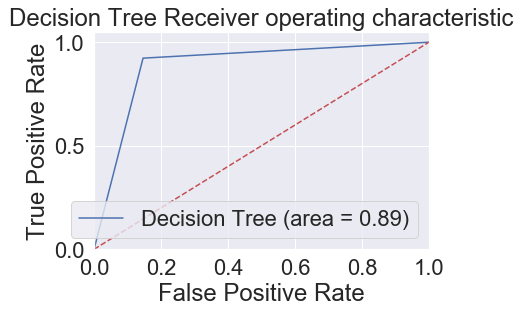


**Third iteration: Statistic Information and Confusion Matrix**



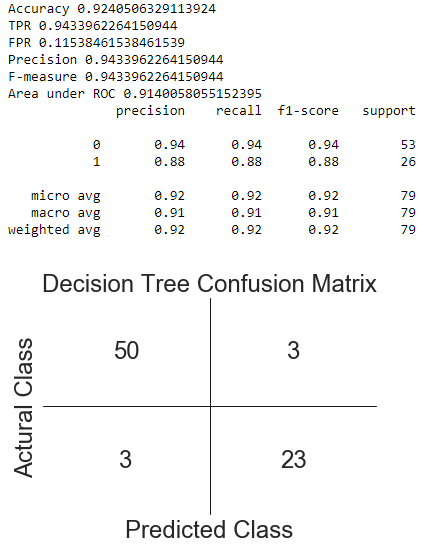


**ROC Curve:**



**Forth iteration: Statistic Information and Confusion Matrix**



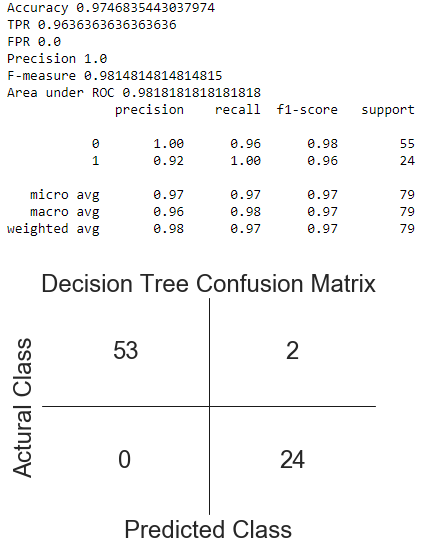


**ROC Curve:**

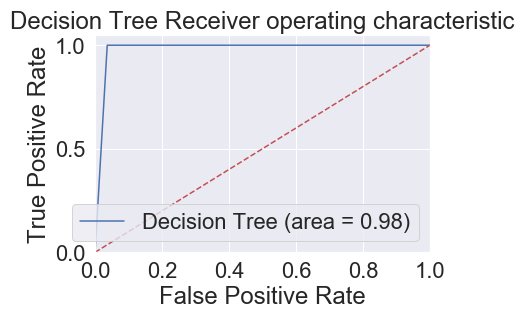


**Fifth iteration: Statistic Information and Confusion Matrix**





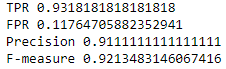
**ROC Curve:**

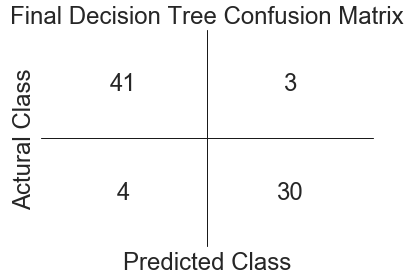


After five iterations, I have the statistic information including confusion matrix with the accuracy, True Positive and False Positive rates, Precision, and F-measure. In addition, I also plot the ROC curve and compute the area under the ROC curve. Then, I sum all the data in each iteration. I have a new confusion matrix that is the average of the ten confusion matrices. Then I recalculate the statistic information.

**Statistic Information and Confusion Matrix:**





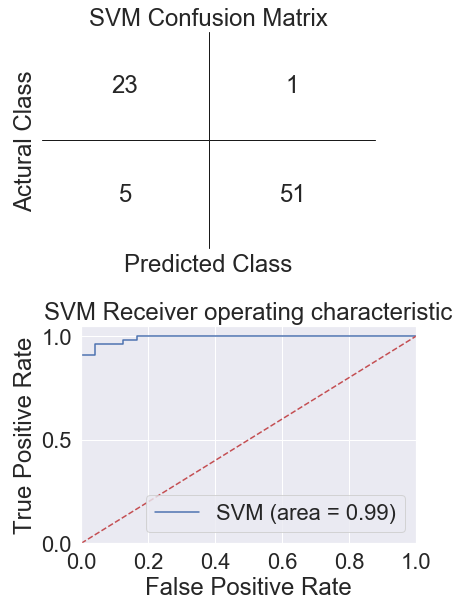
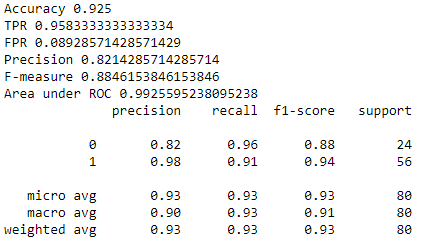


**SVM:**

With the 5-Fold Cross Validation evaluation approach, I import SVM classifier from the sklearn package and fit the training data into SVM and use validation test set to determine the statistic information. For the SVM model, all the parameters are default except for the probability parameter. In order to calculate the ROC curve, I have to get the probability from the SVM model. Therefore, I set the probability parameter to true. I also set up a timer to monitor the time spent during prediction. Since the dataset is not very huge, the time on prediction is relatively small. The first graph below shows the classification report and the second graph contains the accuracy, True Positive, and False Positive rates, Precision rate, and F-measure of SVM model on Validation Dataset. The third graph shown below is the ROC of the SVM model for the Validation set.

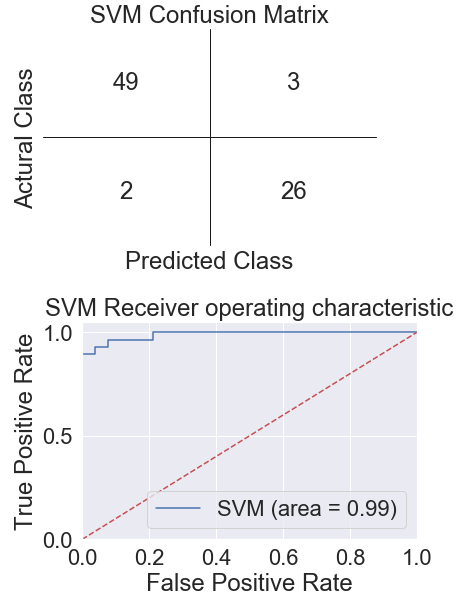
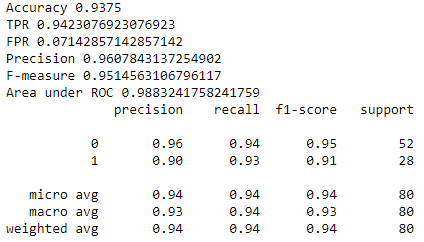
**Statistic Information, Classification Report, Confusion Matrix and ROC Curve of the SVM model for the First Iteration:**





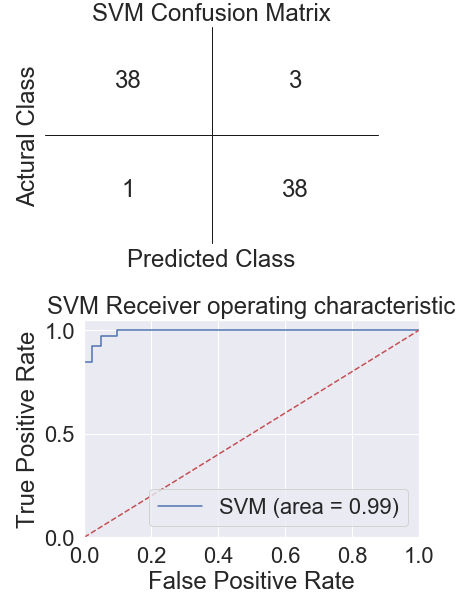
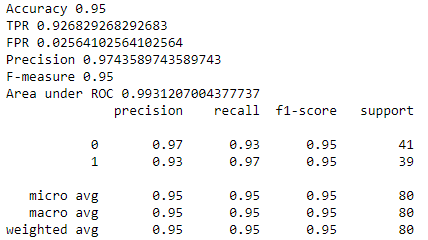
**Statistic Information, Classification Report, Confusion Matrix and ROC Curve of the SVM model for the Second Iteration:**





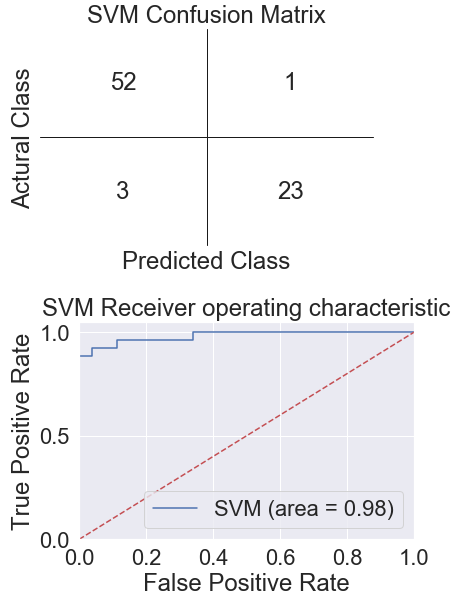
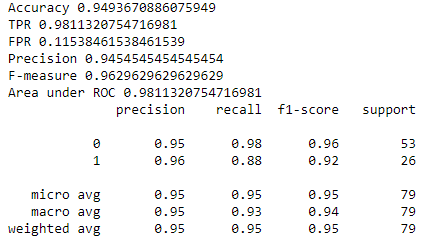
**Statistic Information, Classification Report, Confusion Matrix and ROC Curve of the SVM model for the Third Iteration:**





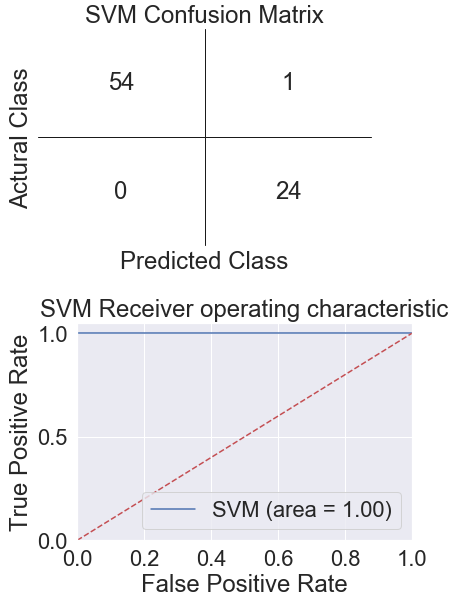
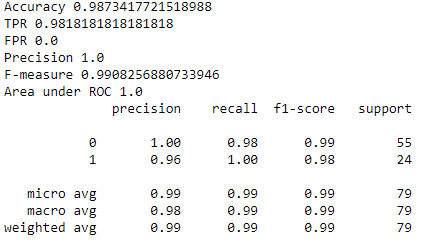
**Statistic Information, Classification Report, Confusion Matrix and ROC Curve of the SVM model for the Forth Iteration:**





**Statistic Information, Classification Report, Confusion Matrix and ROC Curve of the SVM model for the Fifth Iteration:**

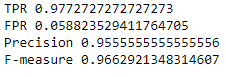


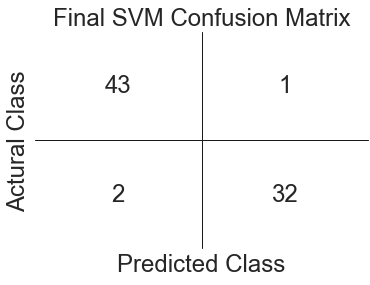


After five iterations, I have the statistic information including confusion matrix with the accuracy, True Positive and False Positive rates, Precision, and F-measure. In addition, I also plot the ROC curve and compute the area under the ROC curve. Then, I sum all the data in each iteration. I have a new confusion matrix that is the average of the ten confusion matrices. Then I recalculate the statistic information.

**Statistic Information and Confusion Matrix:**





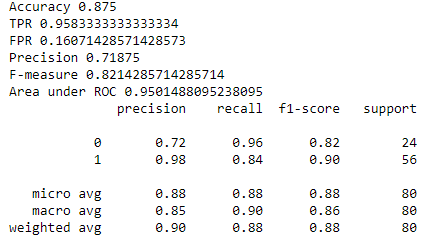


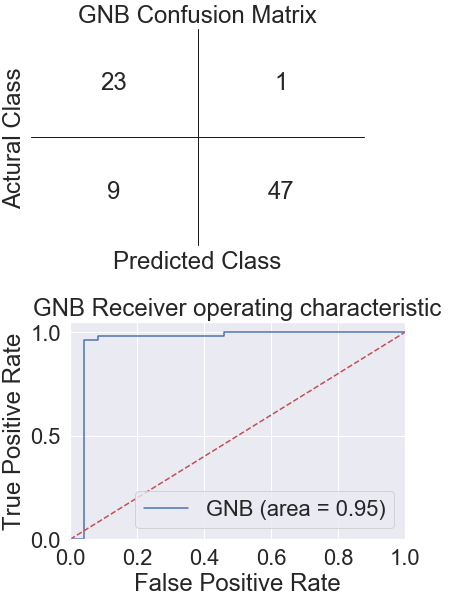
**Naïve Bayes:**

With the same splitting approach, I feed the training data into the Naïve Bayes model from sklearn.naive\_bayes package and use validation test set to determine the statistic information. I also set up a timer to monitor the time spent during prediction. Since the dataset is not very huge, the time on prediction is relatively small. The first graph below shows the classification report and the second graph contains the accuracy, True Positive, and False Positive rates, Precision rate, and F-measure of Naïve Bayes model. The third graph shown below is the ROC of the Naïve Bayes model.

**Statistic Information, Classification Report, Confusion Matrix and ROC Curve of the Naïve Bayes model for the First Iteration:**

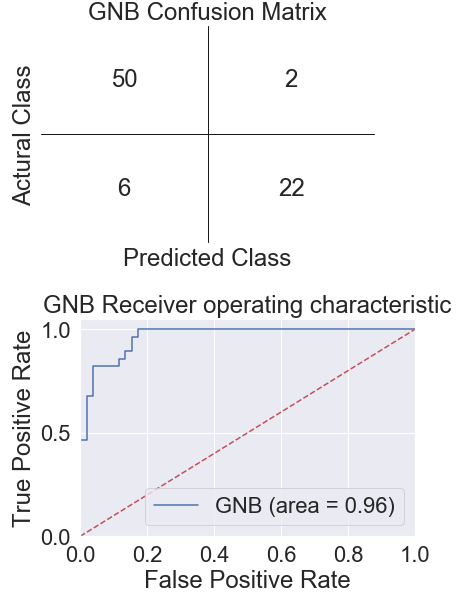






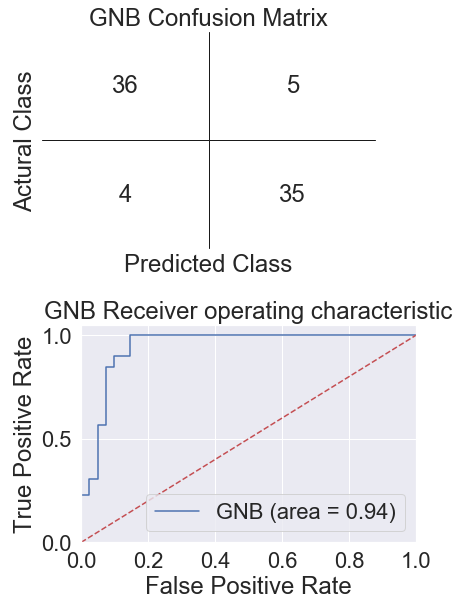
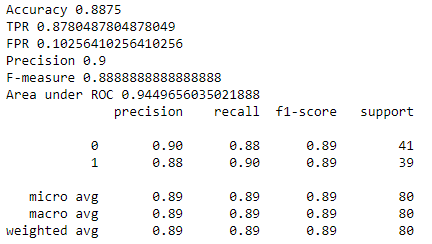
**Statistic Information, Classification Report, Confusion Matrix and ROC Curve of the Naïve Bayes model for the Second Iteration:**





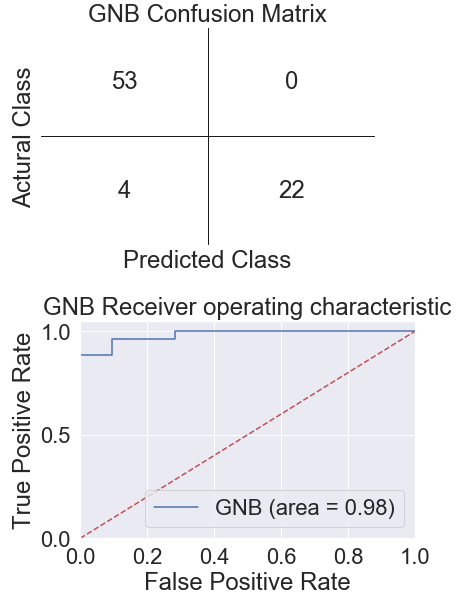
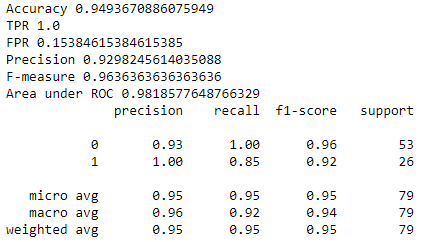
**Statistic Information, Classification Report, Confusion Matrix and ROC Curve of the Naïve Bayes model for the Third Iteration:**





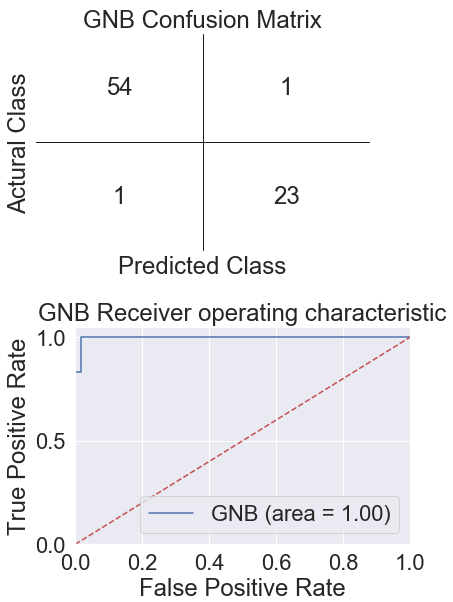
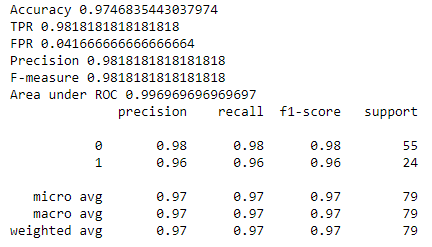
**Statistic Information, Classification Report, Confusion Matrix and ROC Curve of the Naïve Bayes model for the Forth Iteration:**





**Statistic Information, Classification Report, Confusion Matrix and ROC Curve of the Naïve Bayes model for the Fifth Iteration:**

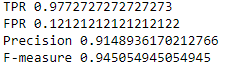


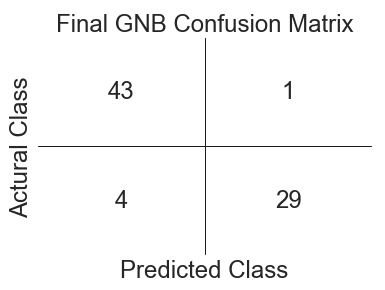


After five iterations, I have the statistic information including confusion matrix with the accuracy, True Positive and False Positive rates, Precision, and F-measure. In addition, I also plot the ROC curve and compute the area under the ROC curve. Then, I sum all the data in each iteration. I have a new confusion matrix that is the average of the ten confusion matrices. Then I recalculate the statistic information.

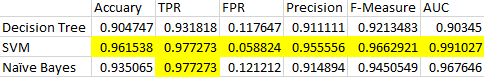
**Statistic Information and Confusion Matrix:**







**Best Model SVM:**

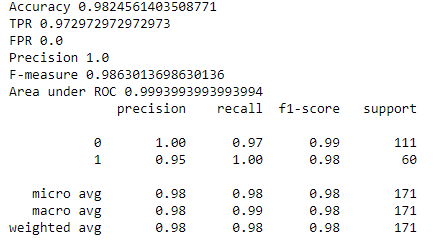
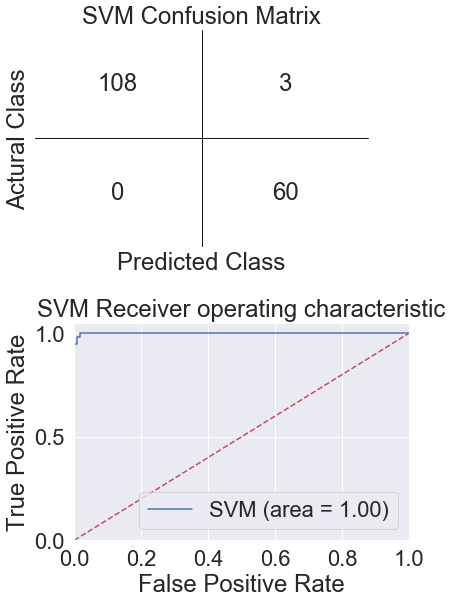


Based on the performance of the four model, accuracy, TPR, FPR, Precision, F-Measure, and the area under ROC of the model can be the factors to determine the best model fit in the dataset. In order to have the best performance, compared with the accuracy, TPR, FPR, Precision, F-Measure, and AUC, the best model is SVM among the four models. The performance of SVM Model is the best among the three models. The Accuracy of the SVM model 96.2% can have the best prediction for the class label and less False Positive Error 5.9%. Furthermore, the Highest True Positive Rate 97.7% can give me the highest possible correct prediction for the True class. The Highest Precision 95.6% can also give me the best possible prediction precisely. The highest F-measure 96.6% and AUC 99.1% also shows that this model has better performance among the four models. Therefore, the SVM model still has the best performance among the three models based on the best performance on Accuracy, TPR, FPR, Precision, F-Measure, and AUC. Then I use the SVM model as the best model. Then I feed the test set into SVM to check the performance on the test set.

The performance of SVM model on the test set has 98.8% accuracy with 99.1% AUC. I set up a timer to monitor the time spent during prediction. Since the dataset is not very huge, the time on prediction is relatively small. The first graph shows the classification report and the second graph below contains the accuracy, True Positive, and False Positive rates, Precision rate, and F-measure of Naïve Bayes model. The third graph shown below is the ROC of the SVM model for the test set.

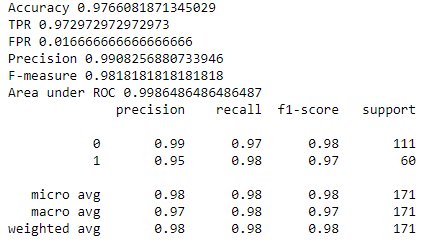
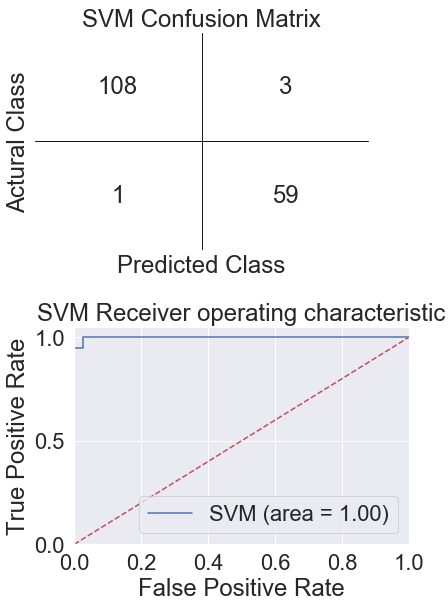
**Statistic Information, Classification Report, Confusion Matrix and ROC Curve of the SVM model for the First Iteration:**



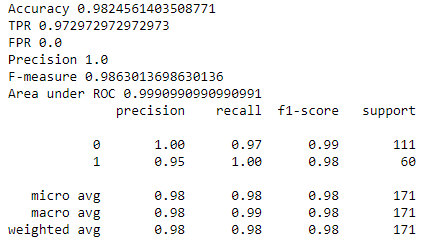
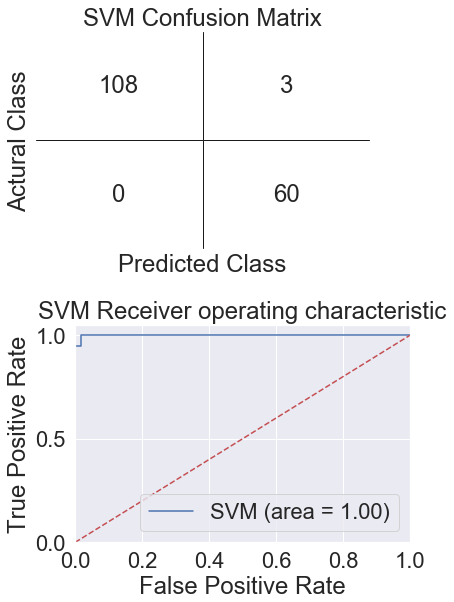
**Statistic Information, Classification Report, Confusion Matrix and ROC Curve of the SVM model for the Second Iteration:**



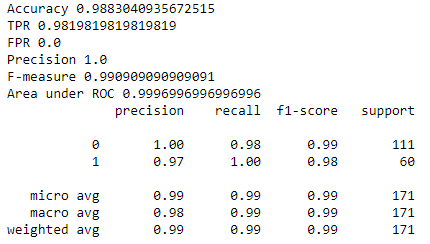
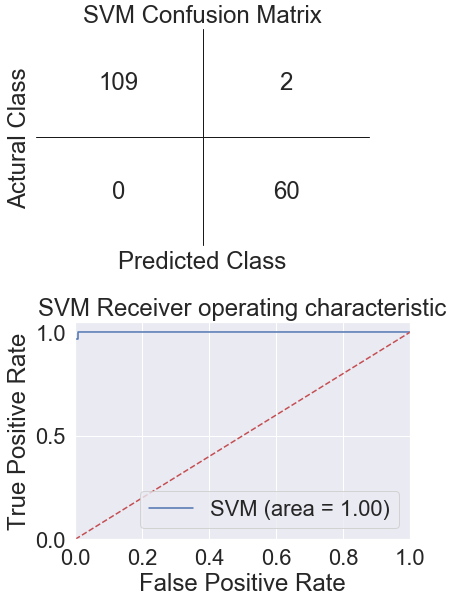
**Statistic Information, Classification Report, Confusion Matrix and ROC Curve of the SVM model for the Third Iteration:**



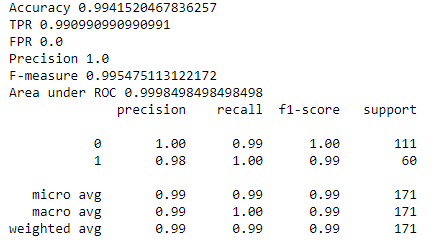
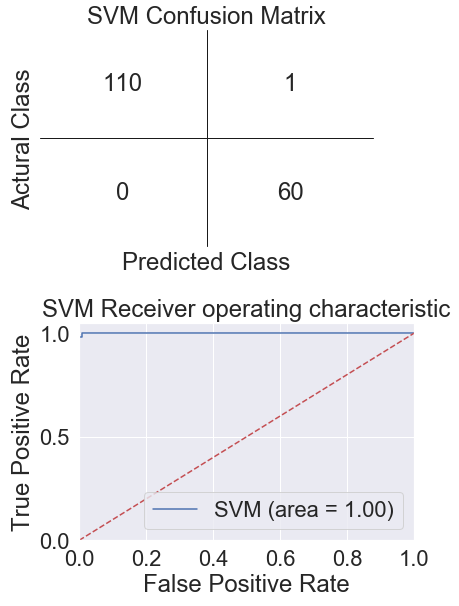
**Statistic Information, Classification Report, Confusion Matrix and ROC Curve of the SVM model for the Forth Iteration:**



**Statistic Information, Classification Report, Confusion Matrix and ROC Curve of the SVM model for the Fifth Iteration:**

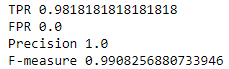


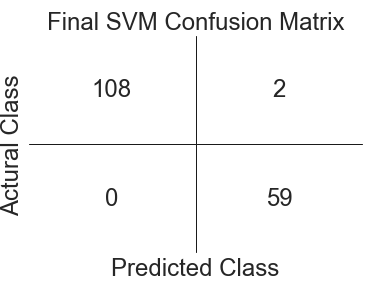
 

After five iterations, I have the statistic information including confusion matrix with the accuracy, True Positive and False Positive rates, Precision, and F-measure. In addition, I also plot the ROC curve and compute the area under the ROC curve. Then, I sum all the data in each iteration. I have a new confusion matrix that is the average of the ten confusion matrices. Then I recalculate the statistic information.

**Statistic Information and Confusion Matrix:**







**In conclusion:**

From the model selection perspective, SVM Classifier is the best model for this dataset since SVM is based on the distribution of the coordinate positions in higher dimensions. SVM can perform better in the dataset because all the attributes in the dataset are numeric values. Therefore, SVM can perform more sophisticated classification. However, Naive Bayes Classifier requires prior knowledge about the dataset and the prior knowledge is very crucial for the prediction. Furthermore, Naive Bayes Classifier has better performance for categorical data instead of numerical data. In addition, Decision Tree Classifier is a non-parametric classification, which mean Decision Tree Classifier does not depend on the probability distribution. Furthermore, Decision Tree Classifier has pruning issues that can significantly affect the generalization error.

The practical meaning for this project can be an assistant for doctors or patients to understand the potential risk of breast cancer, based on the information of the specific patient. This project filters the best classification model among three models and can provide a high prediction for the probability of having breast cancer. Patients can even diagnose themselves to have a better idea of their breast condition, based on the feature information.