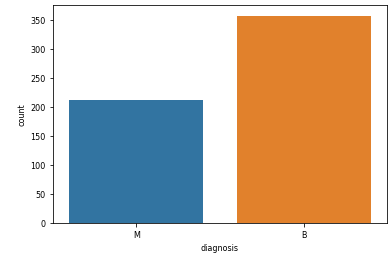
**Problem Statement**

Breast cancer is the most common type of cancer for women. Luckily, breast cancer death rates have declined 40% from 1989 to 2016. The progress is attributed to improvements in early detection. By being able to detect all the people who have breast cancer, the death rate can be lowered even more. This is serious problem because you are dealing with matters of life and death. Using machine learning classification methods would be a great option for this dataset because the data already contains the labels of malignant or benign for the tumor biopsies and you want to be able to tell people if already have or do not have breast cancer.

**Data**

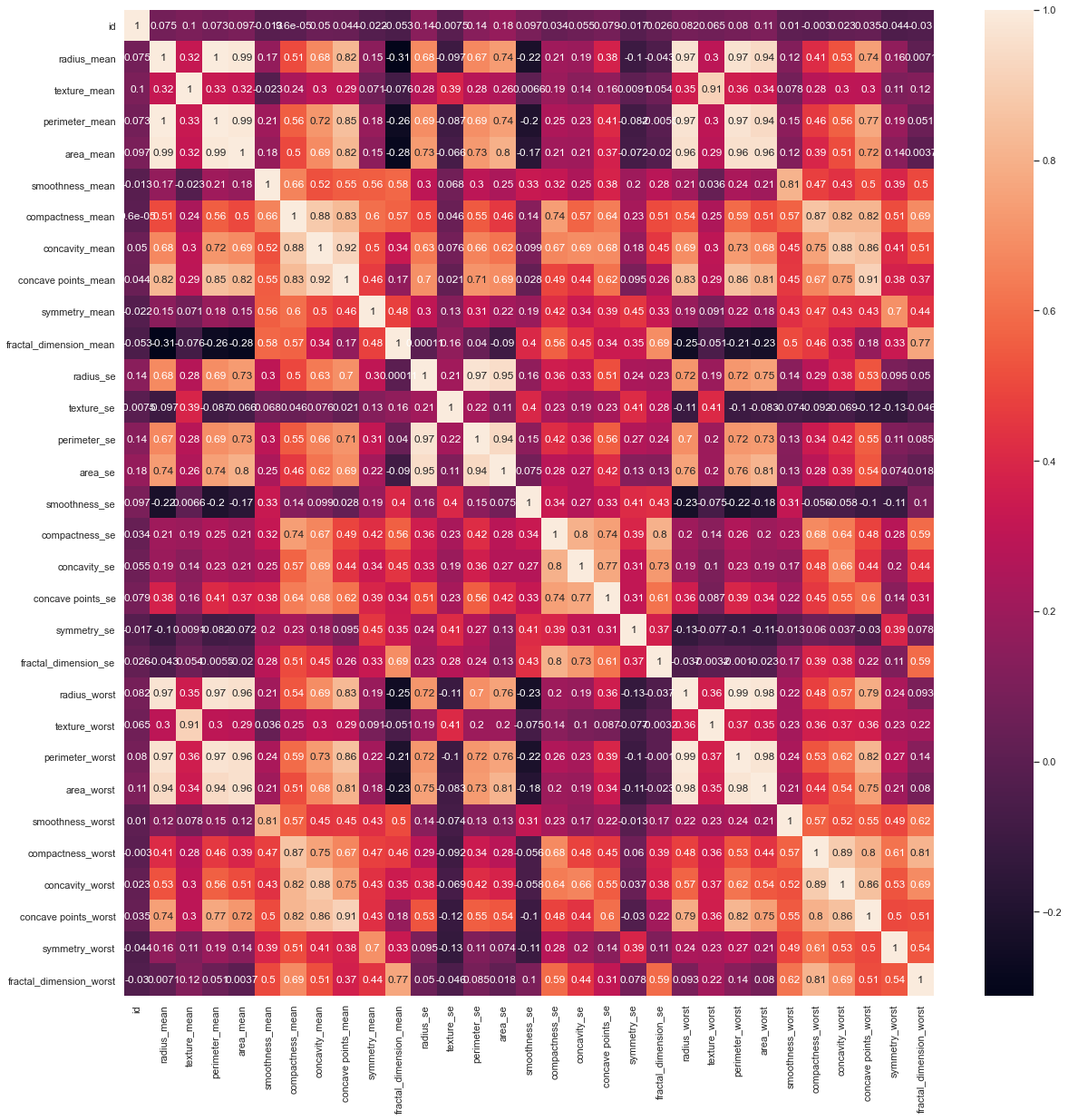
I used the breast cancer Wisconsin (Diagnostic) data set that is available on UCI machine learning repository. (<https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic)>) The data was collected by the University of Wisconsin General Surgery Department. There was an academic paper written about the data set titled: "Robust Linear Programming Discrimination of Two Linearly Inseparable Sets". I am sure the data was collected in order to figure out a method to determine if someone had a malignant (bad result) or benign (good result) form of a breast tumor after a biopsy was performed. There are 32 columns and 569 rows of data. This data probably only came from one hospital in Madison, WI. This means that the results might not be applicable outside of Wisconsin, Midwest, or the USA. The first column in the data set is the subject’s id. The second column is the diagnosis (answer/label) of whether the tumor is malignant or benign. The other 30 columns are made up of three variations of 10 variables: radius, texture, perimeter, area, smoothness, compactness, concavity, concave points, symmetry, and fractal dimension. The variables are computed from a digitized image of a fine needle aspirate (FNA) biopsy of a breast mass. The three variations of those 10 variables are: mean, se (standard error), and worst. Mean is the average value of the measure detected by the computer, se is the standard error of variable being detected, and worst is the average of the three largest values detected. Most of the values in each variables are in in decimal format meaning less than one or up to around a thousand except for the id variables which has up to 9 digits present. The id variable will not be useful in helping with our classification procedure so it was eliminated.

I first wanted to see the amount of data that was in the diagnosis column because that is the response variable that we are interested in. There were 357 cases of those who are benign (62.7%) and 212 of those who are malignant (37.3%). This is part of the data understanding phase of the data mining process.



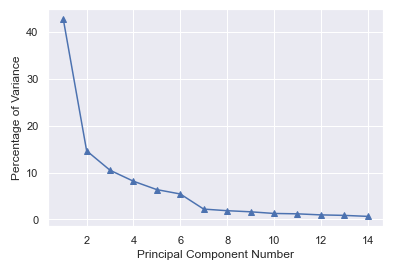
**Data Preparation**

I needed to make sure that there were no missing values in the data. This was done by checking for missing values in Python. Luckily, there were no missing values present within my data set. Further checking of the variables were done by checking the correlation among all the variables with one another as shown on the correlation table shown below. Many variables were seen to be correlated. This is not good because they would be providing redundant information and cause multicollinearity. All those pairs with correlation higher than 0.9 were compared and eliminated. I found one group of variables six variables (radius\_mean, perimeter\_mean, area\_mean, radius\_worst, perimeter\_worst, and area\_worst) being highly correlated with one another as well as another group of three (radius\_se, perimeter\_se, and area\_se) with the same issue. I kept the radius\_mean and radius\_se variables while eliminating all the others.



We are down to 24 columns in our data set from the initial 32 columns of data due to the elimination of the 7 highly correlated variables and the id variable. There was no change to the number of observations (rows).

The data was divided into X and y variables to signify the explanatory and response variables, respectively. The response variables were ranging in many orders of magnitude from one another so to make sure one variable did not have a bigger effect on the data, a standardization was performed. PCA was done manually to determine how many principal component values would be needed to explain 95% of the variability in the data. This would found to be at 10 principal component variables. The individual contributions of each principal component variable are seen below. 23 explanatory variables were able to be described by 10 PCs yet still explain the 95% variability of the data!

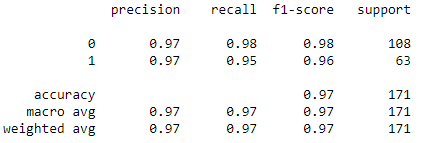


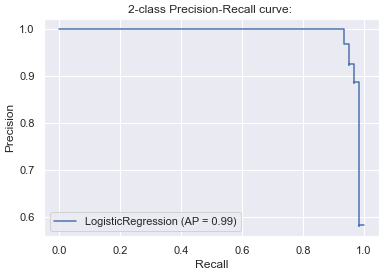
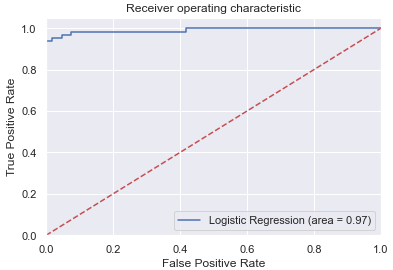
I split the data into p = 0.7 for the training set (399 observations) and the remaining data points were put in the testing set (170 observations). I tried using the many classification methods mentioned in class such as Logistic Regression, k-Nearest Neighbors (kNN), Random Forests, Decision Trees, Support Vector Machines, and a stacked architecture method using multiple classification techniques in one. Whenever possible, I ran gridsearchCV to be able to find the optimal hyperparameters for the classification method to try to maximize my results.

**Modeling**

Logistic Regression:

 (Confusion Matrix on the left)

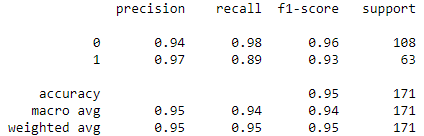


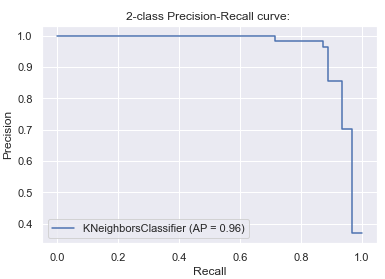
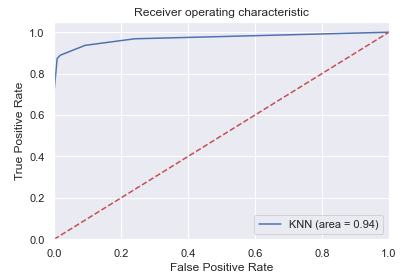


kNN:

The GridSearchCV found out that using the Euclidean metric with k equal to 5 would make the best random forest classifier out of the parameter values tested.



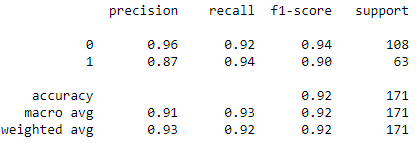


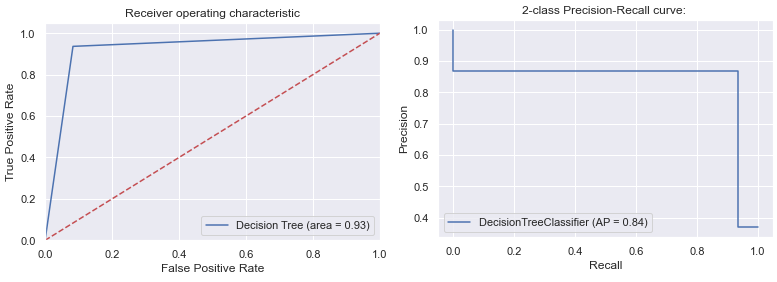


Decision Tree:

A Decision Tree Classifier with the entropy measure was used.

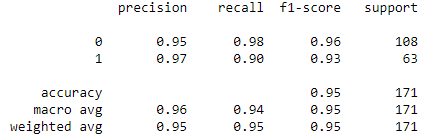


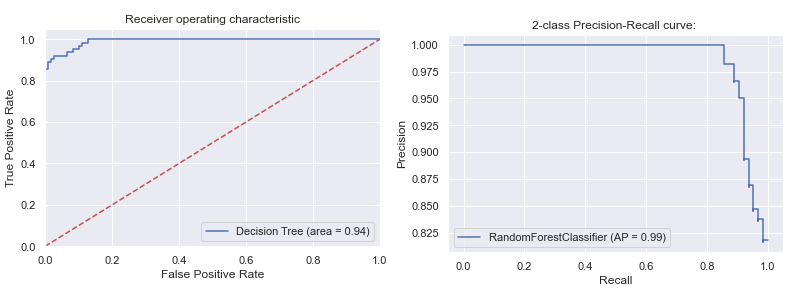




Random Forest:

A Random Forest Classifier with the following parameters were determined to the best 'max\_depth': 45, 'max\_features': 6, 'min\_samples\_leaf': 1, 'min\_samples\_split': 2, 'n\_estimators': 700.

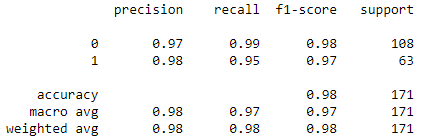


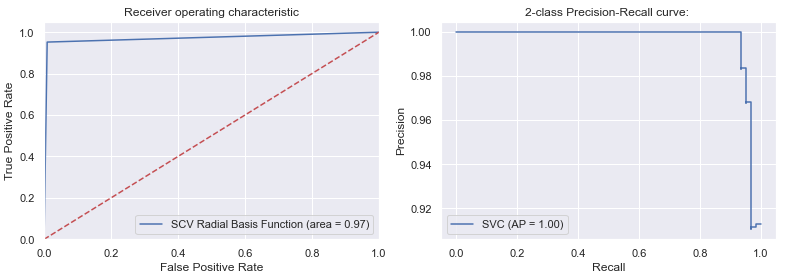


Support Vector Machine (Radial Basis Function kernel):

Support Vector Machine classifier were tried with linear, polynomial, and radial basis function (rbf) kernels. The rbf kernel with gamma at 0.01 and C equal to 10 was determined to be the best possible method for this classification technique. This method ended up being the best method out of all the methods that I tried for this dataset.







Stacked Architecture:

A stacked architecture where the first layer (level 1) had a Logistic Regression and Random Forest algorithms applied to the data set. Next a Support Vector Machine Regression (level 2) was conducted on the dataset. My hope was that this method of combining multiple regression methods would result in a better accuracy and classification of the dataset, but this ended up not being as good as the support vector machine classification above.

**Evaluation**

All models performed admirably but the SVM classification ended up being the best model. I wanted to measure specificity because we do not want to miss any of those who might have the disease even if we get some false positives. The SVM model was not only better in specificity, but also in sensitivity and overall accuracy. All models had some malignant tumors predicted as benign. The biggest disappointment and surprise was that the decision tree had a precision recall curve value of .84 while every other model ended up in the .9 or higher range.

**Discussion and Conclusions**

All models were able to come up with high specificity for breast cancer detection. I would hope that the specificity number is much higher in the real world so that you do not miss many malignant tumors which could result in the loss of lives of loved ones. Getting some more data points and seeing if the model could become even more accurate for the classification methods would be the next step. Furthermore, I believe with fine tuning the stacked architecture method could become better than just a single classification technique.

**References**

<https://www.kaggle.com/uciml/breast-cancer-wisconsin-data/activity>

<https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Diagnostic%29>

<https://towardsdatascience.com/building-a-simple-machine-learning-model-on-breast-cancer-data-eca4b3b99fa3>