Breast Cancer Detection Model

Arjun Kannan

11th September 2020

1. Introduction

1.1 Background

Breast cancer is the most frequent cancer among women, impacting 2.1 million women each year, and also causes the greatest number of cancer-related deaths among women. In 2020, it still remains a worldwide public health dilemma and is currently the most common tumor in the globe. To define, Breast cancer is an uncontrolled growth of breast cells.

1.2 Problem

In 2018, it is estimated that 627,000 women died from breast cancer – that is approximately 15% of all cancer deaths among women. While breast cancer rates are higher among women in more developed regions, rates are increasing in nearly every region globally. **In order to improve breast cancer outcomes and survival, early detection is critical.**

1.3 Interest

The interest stems from the various Breast Cancer Research Groups across the world focusing on new methods to better understand breast disease and to develop new approaches to characterize this disease.

2. Data Acquisition and Cleaning

2.1 Data Sources

Data has been acquired from a Kaggle Dataset. Link to Dataset

2.2 Data Cleaning

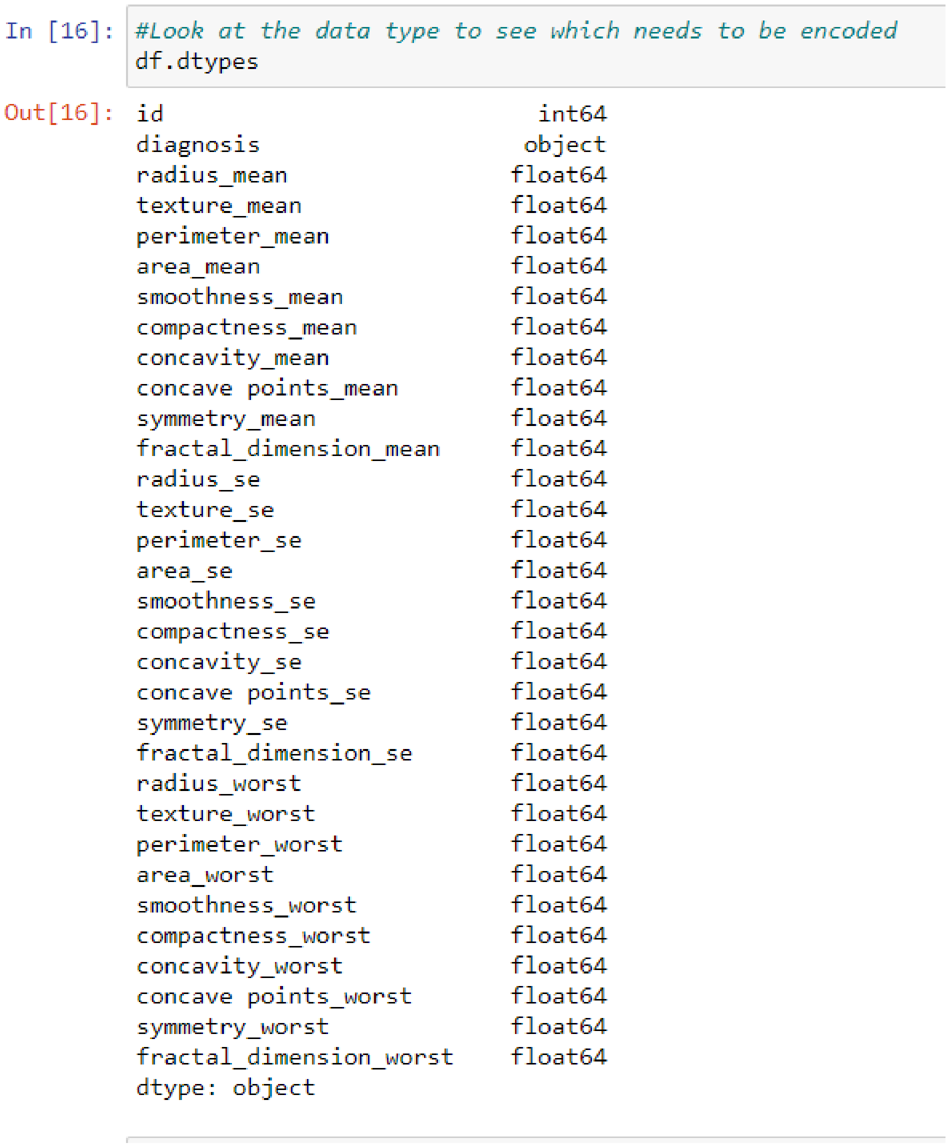
Data was downloaded from a single data source. “Nan” or Null data was present in only column named “Unamed:32” which was deleted. The data was faultless and ready to use.

2.3 Feature Selection

After data cleaning, there were 569 rows and 33 columns. As mentioned above, data was accurate with no redundancies. I performed an analysis of all the columns and identified that we will be focusing on “Diagnosis” which is critical in identifying whether the cancer cell is benign or malignant.

3. Data Analysis

Primarily, it was vital to ascertain what type of values each column contained. So, I used the “*dtype*” function to do the same.

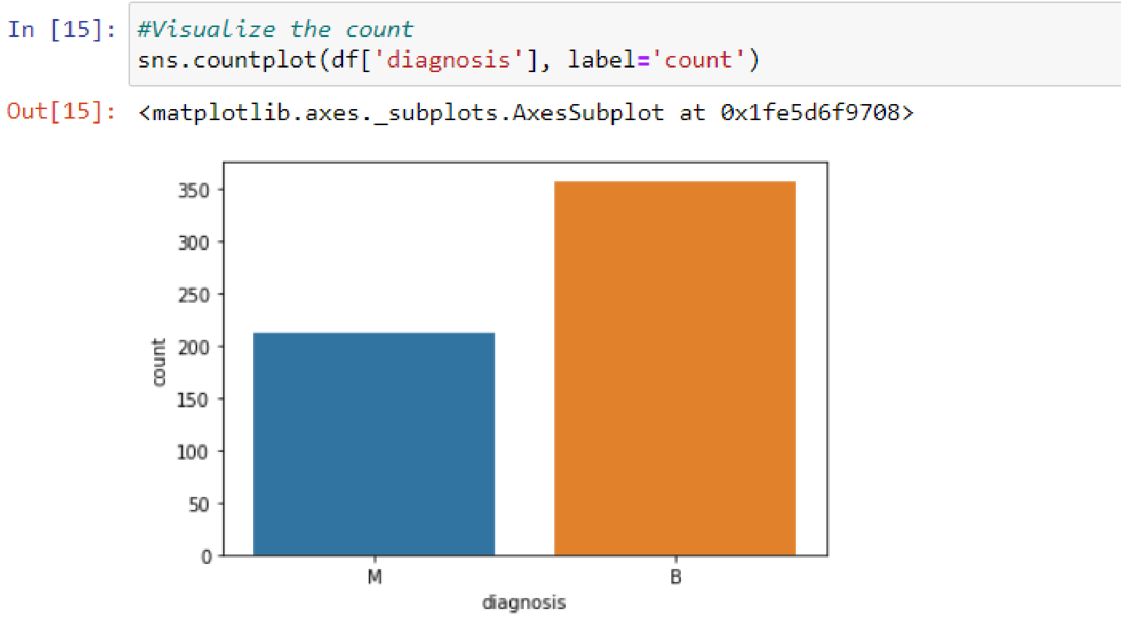


```
In [16]: #Look at the data type to see which needs to be encoded
         df.dtypes

Out[16]: id                           int64
         diagnosis                   object
         radius_mean                float64
         texture_mean               float64
         perimeter_mean             float64
         area_mean                  float64
         smoothness_mean            float64
         compactness_mean           float64
         concavity_mean             float64
         concave points_mean        float64
         symmetry_mean              float64
         fractal_dimension_mean     float64
         radius_se                  float64
         texture_se                 float64
         perimeter_se               float64
         area_se                    float64
         smoothness_se              float64
         compactness_se             float64
         concavity_se               float64
         concave points_se          float64
         symmetry_se                float64
         fractal_dimension_se       float64
         radius_worst               float64
         texture_worst              float64
         perimeter_worst            float64
         area_worst                 float64
         smoothness_worst           float64
         compactness_worst          float64
         concavity_worst            float64
         concave points_worst       float64
         symmetry_worst             float64
         fractal_dimension_worst    float64
         dtype: object
```

Once each and every column and its type has been identified, I presumed by analyzing various the similarities and relevance of the data set. The following were

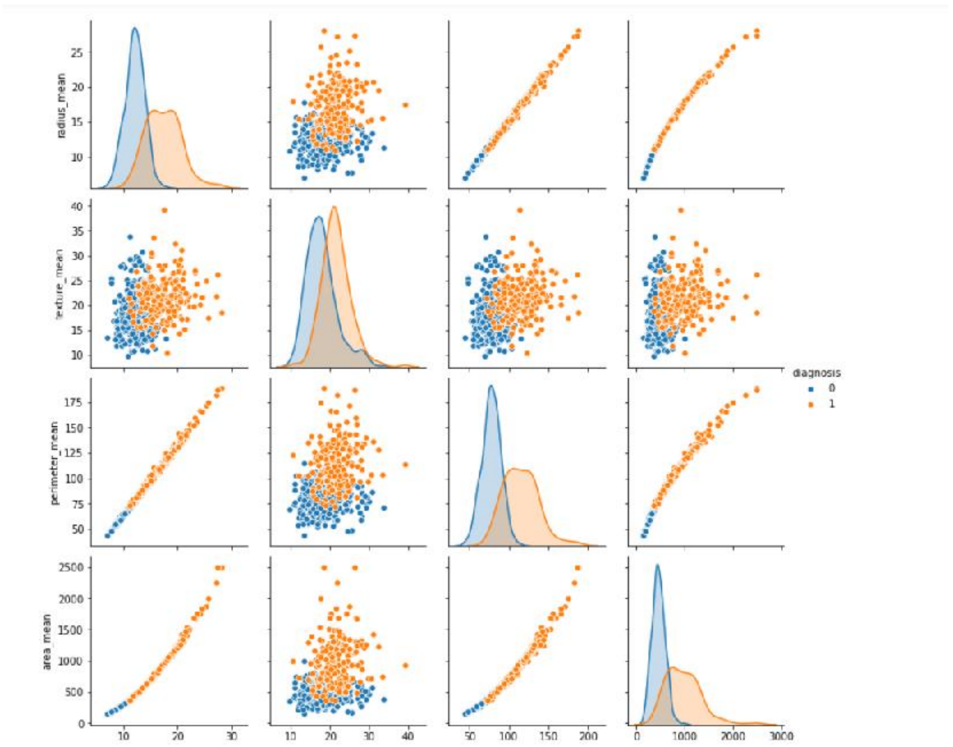
a) Relationship between Benign and Malignant cells

The image below clearly states that there are 357 Benign cases and 212 Malignant cases totally.

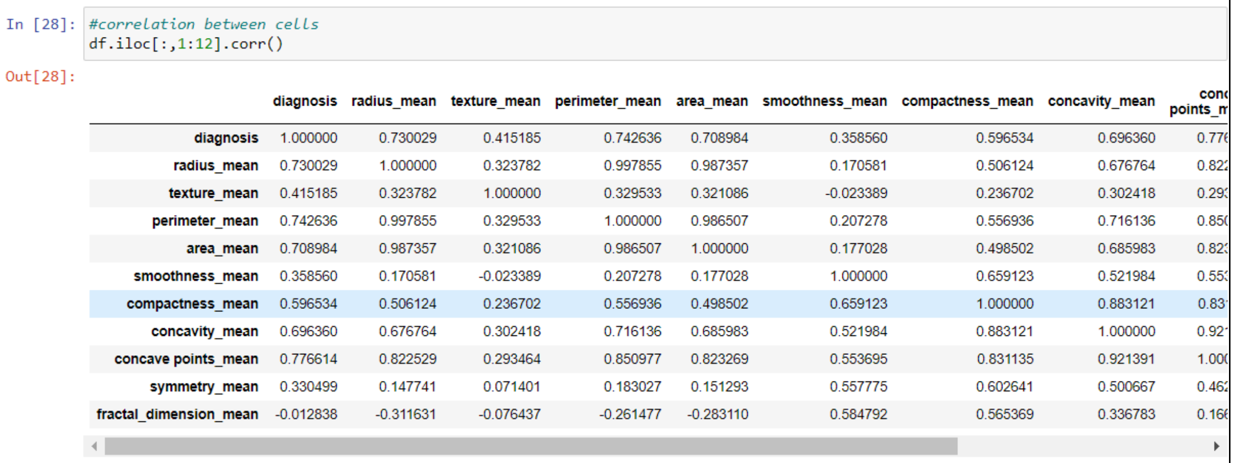


```
In [15]: #Visualize the count
         sns.countplot(df['diagnosis'], label='count')

Out[15]: <matplotlib.axes._subplots.AxesSubplot at 0x1fe5d6f9708>
```

b) Pair Plot of the first 5 columns



c) Correlation between the cells

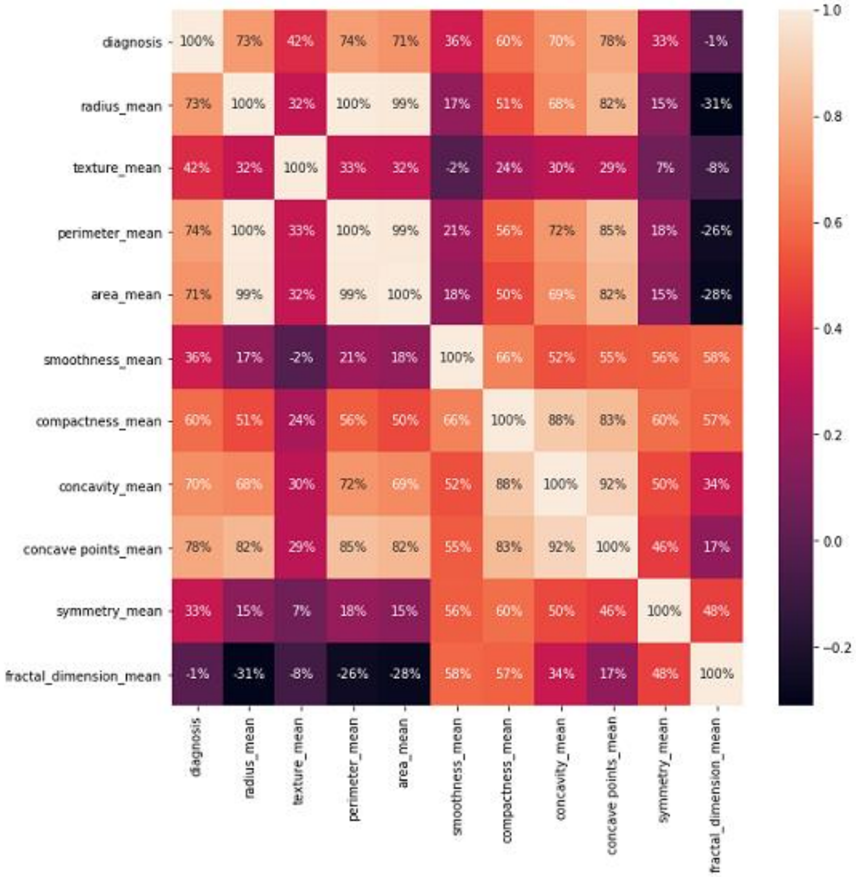


```
In [28]: #correlation between cells
         df.iloc[:,1:12].corr()
```

Out[28]:

	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	conc points_m
diagnosis	1.000000	0.730029	0.415185	0.742636	0.708984	0.358560	0.596534	0.696360	0.776
radius_mean	0.730029	1.000000	0.323782	0.997855	0.987357	0.170581	0.506124	0.676764	0.822
texture_mean	0.415185	0.323782	1.000000	0.329533	0.321086	-0.023389	0.236702	0.302418	0.293
perimeter_mean	0.742636	0.997855	0.329533	1.000000	0.986507	0.207278	0.556936	0.716136	0.850
area_mean	0.708984	0.987357	0.321086	0.986507	1.000000	0.177028	0.498502	0.685983	0.823
smoothness_mean	0.358560	0.170581	-0.023389	0.207278	0.177028	1.000000	0.659123	0.521984	0.553
compactness_mean	0.596534	0.506124	0.236702	0.556936	0.498502	0.659123	1.000000	0.883121	0.83
concavity_mean	0.696360	0.676764	0.302418	0.716136	0.685983	0.521984	0.883121	1.000000	0.92
concave points_mean	0.776614	0.822529	0.293464	0.850977	0.823269	0.553695	0.831135	0.921391	1.000
symmetry_mean	0.330499	0.147741	0.071401	0.183027	0.151293	0.557775	0.602641	0.500667	0.462
fractal_dimension_mean	-0.012838	-0.311631	-0.076437	-0.261477	-0.283110	0.584792	0.565369	0.336783	0.166

d) Visualizing the correlation between cells



4. Machine Learning

4.1 Independent and Dependent Data

As per the dataset, it was obvious to use the “Diagnosis” column consisting of 2 values – Benign and Malignant as the dependent variable – Y. Meanwhile, the rest was considered as Independent data.

4.2 Techniques of ML

The following techniques were used to train the model.

A. **Logistic Regression** (Model 1)- Logistic regression is a supervised learning classification algorithm used to predict the probability of a target variable.

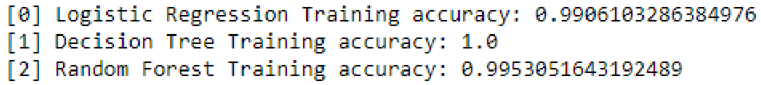
B. **Decision Tree** (Model 2)- Decision Tree Analysis is a general, predictive modelling tool that has applications spanning a number of different areas. In general, decision trees are constructed via an algorithmic approach that identifies ways to split a data set based on different conditions. It is one of the most widely used and practical methods for supervised learning.

C. **Random Forest** (Model 3)- Random forests or random decision forests are 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 or mean prediction of the individual trees.

4.3 Testing Accuracy

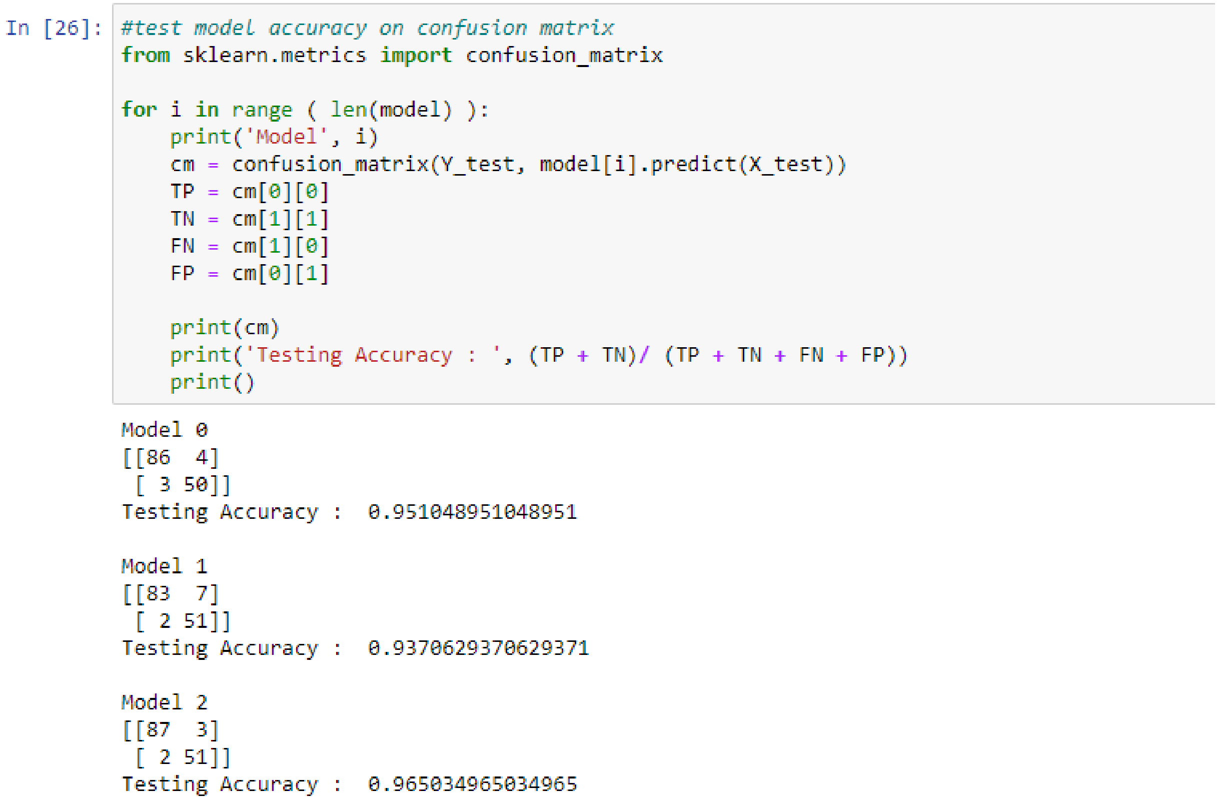
In order to run the model, I have used 75% data as training while 25% is testing.

Initial Training provided a good accuracy level. All models almost accurately providing the values.



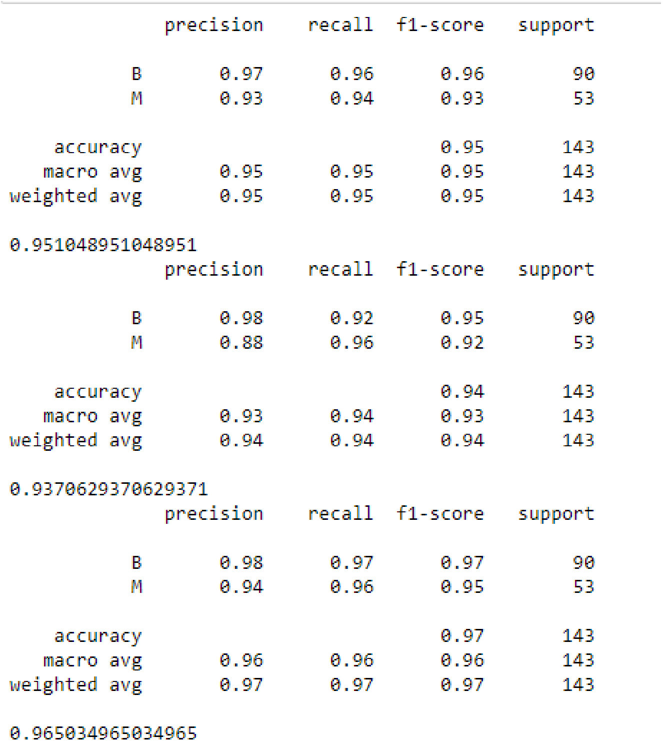
```
[0] Logistic Regression Training accuracy: 0.9906103286384976
[1] Decision Tree Training accuracy: 1.0
[2] Random Forest Training accuracy: 0.9953051643192489
```

To broaden the analysis, with the help of a **confusion matrix**, I have identified the True Positive, True Negative, False Positive and False Negative.



```
In [26]: #test model accuracy on confusion matrix
         from sklearn.metrics import confusion_matrix

         for i in range ( len(model) ):
             print('Model', i)
             cm = confusion_matrix(Y_test, model[i].predict(X_test))
             TP = cm[0][0]
             TN = cm[1][1]
             FN = cm[1][0]
             FP = cm[0][1]

             print(cm)
             print('Testing Accuracy : ', (TP + TN)/ (TP + TN + FN + FP))
             print()

         Model 0
         [[86  4]
          [ 3 50]]
         Testing Accuracy :  0.951048951048951

         Model 1
         [[83  7]
          [ 2 51]]
         Testing Accuracy :  0.9370629370629371

         Model 2
         [[87  3]
          [ 2 51]]
         Testing Accuracy :  0.965034965034965
```

Finally, the ascertain a conclusion on the accuracy, I used "*accuracy_score*" to provide a final report card of how the 3 models performed.



```
              precision    recall  f1-score   support

           B       0.97      0.96      0.96        90
           M       0.93      0.94      0.93        53

    accuracy                           0.95       143
   macro avg       0.95      0.95      0.95       143
weighted avg       0.95      0.95      0.95       143

0.951048951048951
              precision    recall  f1-score   support

           B       0.98      0.92      0.95        90
           M       0.88      0.96      0.92        53

    accuracy                           0.94       143
   macro avg       0.93      0.94      0.93       143
weighted avg       0.94      0.94      0.94       143

0.9370629370629371
              precision    recall  f1-score   support

           B       0.98      0.97      0.97        90
           M       0.94      0.96      0.95        53

    accuracy                           0.97       143
   macro avg       0.96      0.96      0.96       143
weighted avg       0.97      0.97      0.97       143

0.965034965034965
```

5. Conclusion

It is evident that the Random Forest Training is the most accurate at 96.5% followed by Logistic Regression at 95.1% and lastly Decision Tree at 93.7%. When tested, there is a few errors that the Machine makes but provides a very good accuracy rate. This study is only the beginning of research conducted by the Breast Cancer Research Group and can be pivotal for predicting breast cancer in the future.