BREAST CANCER DIAGNOSTIC

Disease Prediction Based on FNA Tissue Imaging

Paul Gagliardi – University of Denver Data Science

INTRODUCTION

The use of machine learning models in visual data fields is increasing, and healthcare is no exception. It is in the best interests of the patients, healthcare professionals, and insurance companies to make the most accurate diagnosis possible, and machine learning can be used as a powerful tool for that.

DATASET

This dataset was given on Kaggle here and describes digitized images of an FNA (Fine Needle Aspirate) of breast mass. FNA is a minimally invasive and cost-effective sampling that obtains tissues from a part of the body. The alternative to this a core biopsy, which provides more information but is more invasive, expensive, and takes longer to process. To improve medical outcomes for citizens, better access to healthcare is paramount. Better predictions with an FNA tissue sample could lead to cheaper tests and faster results with the same sensitivity as a Biopsy.

Features

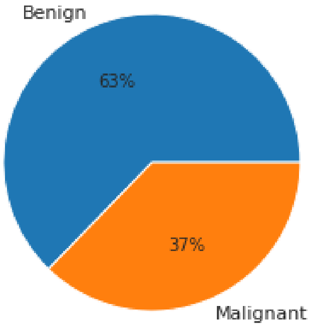
Dataset has 32 features, 1 target class, and 569 observations. There are 10 continuous variables, and for each a mean, standard error, and 'worst' measurement is taken (worst being the mean of the largest 3 measurements).

- *ID number (Meaningless, dropped)*
- ***Diagnosis (M = Malignant, B = Benign)***
  - *Binary Target Class, later M and B will be encoded as 1 and 0 respectively*
- *Continuous variables (3 of each – mean, standard error, 'worst')*
  - *radius (mean of distances from center to points on the perimeter)*
  - *texture (standard deviation of gray-scale values)*
  - *perimeter*
  - *area*

- ○ *smoothness (local variation in radius lengths)*
  - ○ *compactness (perimeter^2 / area - 1.0)*
  - ○ *concavity (severity of concave portions of the contour)*
  - ○ *concave points (number of concave portions of the contour)*
  - ○ *fractal dimension ("coastline approximation" - 1)*
- *Unnamed (dropped)*

When building the model later, we will only use the 30 continous features and the Diagnosis target variable.

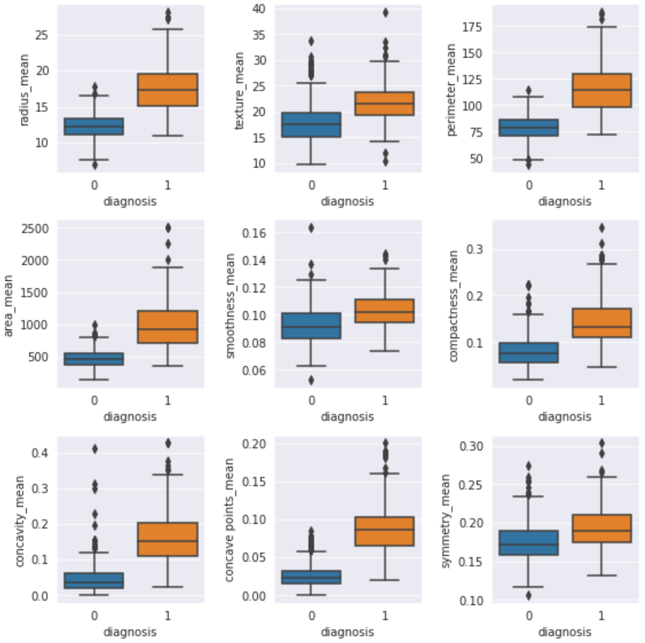
Data Exploration

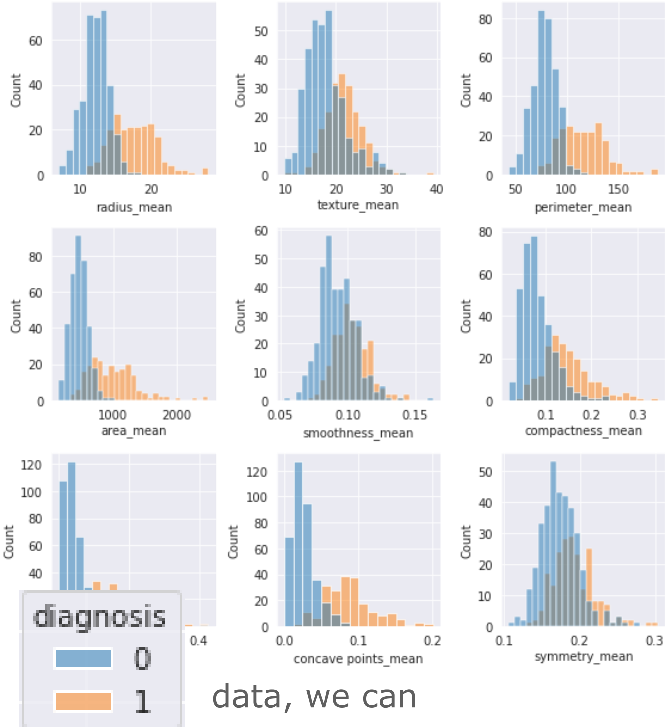


The diagnosis slightly favored Benign classifications, with Benign and Malignant observations

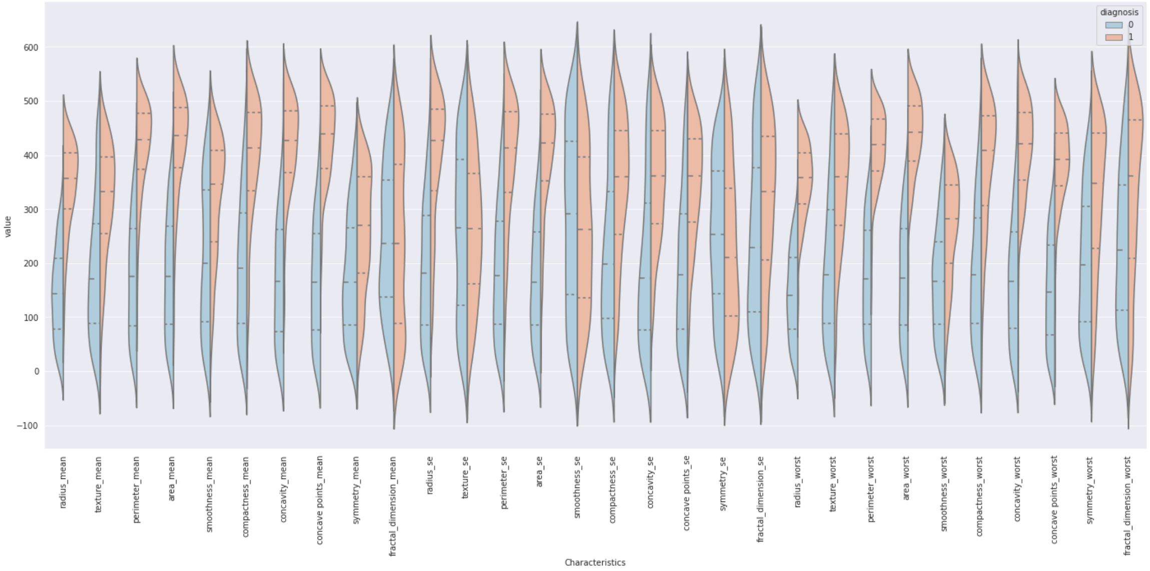
Examining the mean observations for the cells, the data distribution looks fairly normal with some of the Benign cells having a systematically larger measurement. Here 0 represents Benign and 1 Malignant.

Examining some box plots of the data, we can





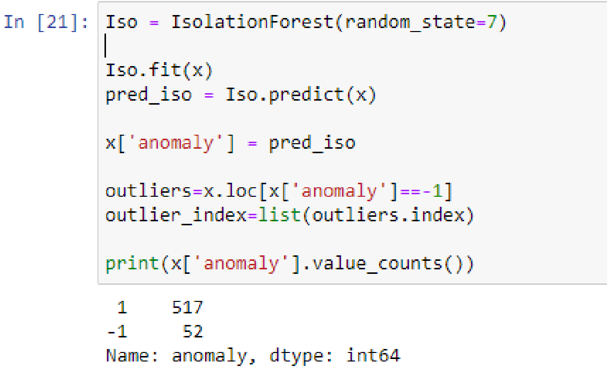
clearly see there is a wide distribution of the data and possibly even some outliers.



This Violin Plot gives an overview of how the data is distributed.

Outlier Detection

Outliers can harm our models or reduce accuracy. The boxplots indicated that some could exist, so I opted to try an “Isolation Forest” to classify some outliers.



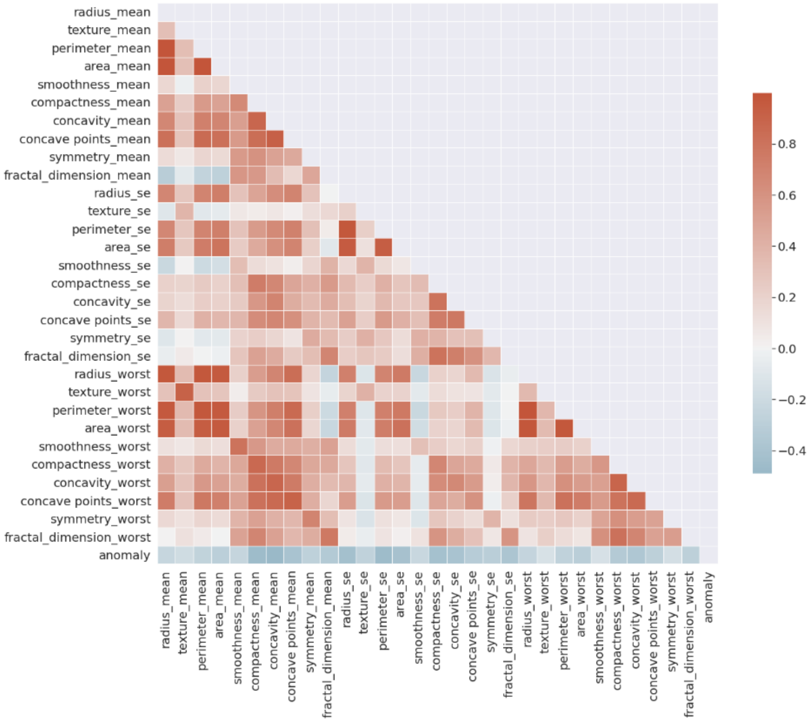
```
Iso = IsolationForest(random_state=7)

Iso.fit(x)
pred_iso = Iso.predict(x)

x['anomaly'] = pred_iso

outliers=x.loc[x['anomaly']==-1]
outlier_index=list(outliers.index)

print(x['anomaly'].value_counts())
```

```
 1    517
-1     52
Name: anomaly, dtype: int64
```

Isolation Forest acts like a decision tree, but selects a feature and randomly splits along that value. This quickly isolates outliers due to their more extreme values, and they can be labeled as such and counted. This dataset had 52 outliers out of 569 observations which is a little over 9%, too much information to throw away.

Collinearity

Since many of the features describe the same geometric shapes, it is likely that there is collinearity among them. Creating a correlation matrix, we can see there are 5 features with a correlation of about 0.9. This is an indication that I will likely need feature selection to produce a useful model.



Data for this project was given a 70:30 train test split.

FEATURE ENGINEERING

Feature Scaling

To avoid outliers impacting feature scaling or model results, Robust Data Scaling was needed. This type of scaling is similar to minmax scaling, but rather than subtracting and scaling to

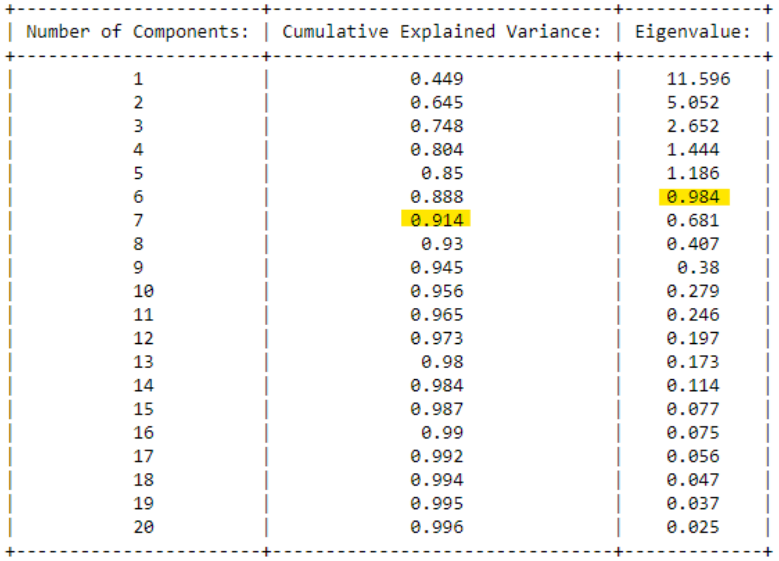
$$\frac{x_i - Q_1(x)}{Q_3(x) - Q_1(x)}$$

the minimum and maximum (which could be outliers), the data is scaled to the first and third quartile.

Dimensionality Reduction

A PCA seemed like a great method for a dataset with high dimensions, but to determine how many components to reduce we must examine how much variance is explained by each principal component and its eigen value.

Considering the high dimensionality, I wanted my PCA to be able to explain at leaset 90% of the variance. The Kaiser rule regarding PCA dictates that once the eigen value of an added component drops below 1.0, it is no longer adding value. I rad a 20 component PCA on the dat set and ran tests for explained variance. Examining the table below we have a quandary.



```
+-----------------------+---------------------------------+-------------+
| Number of Components: | Cumulative Explained Variance: | Eigenvalue: |
+-----------------------+---------------------------------+-------------+
|           1           |              0.449              |    11.596   |
|           2           |              0.645              |    5.052    |
|           3           |              0.748              |    2.652    |
|           4           |              0.804              |    1.444    |
|           5           |               0.85              |    1.186    |
|           6           |              0.888              |    0.984    |
|           7           |              0.914              |    0.681    |
|           8           |               0.93              |    0.407    |
|           9           |              0.945              |     0.38    |
|           10          |              0.956              |    0.279    |
|           11          |              0.965              |    0.246    |
|           12          |              0.973              |    0.197    |
|           13          |               0.98              |    0.173    |
|           14          |              0.984              |    0.114    |
|           15          |              0.987              |    0.077    |
|           16          |               0.99              |    0.075    |
|           17          |              0.992              |    0.056    |
|           18          |              0.994              |    0.047    |
|           19          |              0.995              |    0.037    |
|           20          |              0.996              |    0.025    |
+-----------------------+---------------------------------+-------------+
```

We reach an eigen value of below 1.0 at 6 components but only explain 90% of variance at 7 components. I opted to pick 7 components rather than 6 since adding an additional component to 6 versus the 30 features I initially had seamed arbitrary.

ANALYSIS

According to this study at the National Library of Medicine, sensitivity for Malignant cells can range between 65.4 – 92.4% for FNA and between 88.7 - 100% for core biopsies. This is what we will be evaluating our results against.

We will be evaluating seven classification techniques against each other, and picking the best two to tune.

Models

Before models were tuned they had the following accuracy on the training set

- *SVC: 97.5%*
- ***Logistic Regression: 98.0%***
- *K-Neighbors Classifier: 96.7%*
- ***MLP Classifier: 98.7%***
- *Gaussian Naïve-Bayes: 94.4%*
- ***Random Forest: 100%***
- *Decision Tree: 100%*

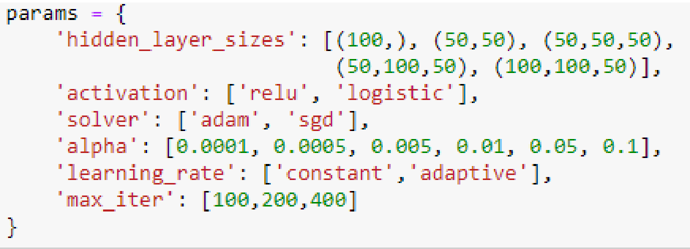
Cross Validation

Decision Trees are prone to overfitting and Random Forests are similar in nature, so I opted to take Random Forest, MLP, and Logistic Regression Classifiers as my analysis tools and to **cross validate** them further to see which fit best.

- ***MLP Classifier: 96.7%***
- ***Logistic Regression: 97.2%***
- *Random Forest: 93.4%*

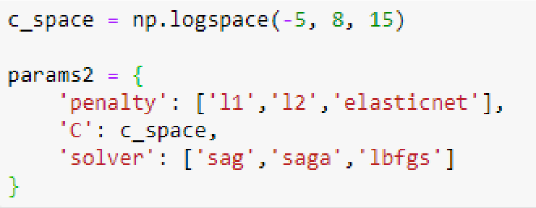
Narrowed down to Logistic Regression and MLP, I tuned both with the following sets of Hyperparameters.

MLP



```
params = {
    'hidden_layer_sizes': [(100,), (50,50), (50,50,50),
                           (50,100,50), (100,100,50)],
    'activation': ['relu', 'logistic'],
    'solver': ['adam', 'sgd'],
    'alpha': [0.0001, 0.0005, 0.005, 0.01, 0.05, 0.1],
    'learning_rate': ['constant','adaptive'],
    'max_iter': [100,200,400]
}
```

Logistic Regression



```
c_space = np.logspace(-5, 8, 15)

params2 = {
    'penalty': ['l1','l2','elasticnet'],
    'C': c_space,
    'solver': ['sag','saga','lbfgs']
}
```

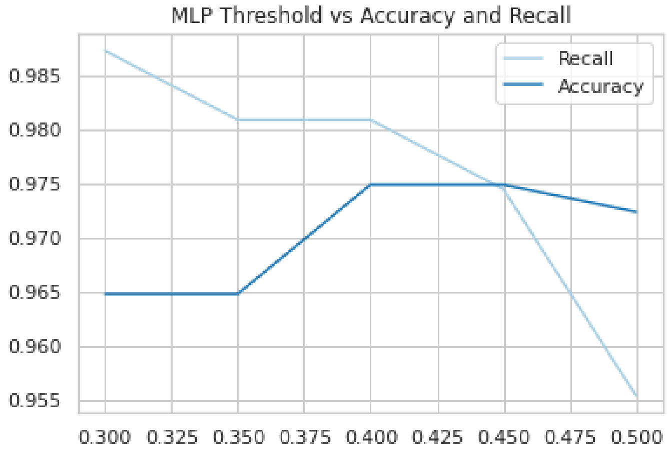
I performed two **GSCVs** (Grid Search Cross Validation) on each model, with accuracy optimized for one and recall optimized in the other. This produced two optimized two for each method

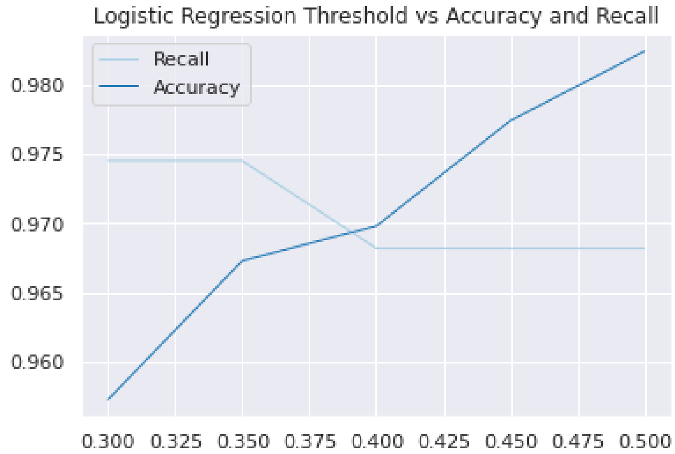
Recall in this problem is a much more significant issue that specificity, since a false negative represents an incorrect diagnosis on a tumor of benign when it is actually malignant, which could result in the death of the patient. Maximizing recall limits false negatives.I am tuning these models then for the ideal compromise between recall while maintaining a high accuracy.

Of the 4 models produced via grid search, the **MLP optimized for recall** and **Logistic Regression optimized for accuracy** proved best.

Classification Threshold Tuning

When making a classification these methods use the continous input variables to create a probability output variable. A probability over 0.5 is classified as 1, and under is classified as 0. Changing this threshold of 0.5 can greatly impact misclassifications. In this case, lowering the threshold will increase the number of false positives but decrease false negatives and improve recall. Since that is a goal of this analysis, the two remaining models will be further tuned along decreasing thresholds.





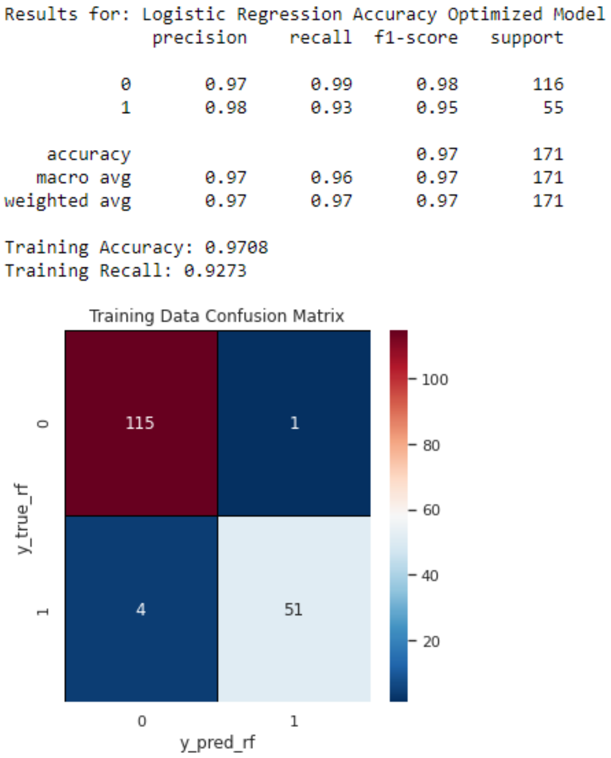
Though the MLP seemed to benefit from the threshold change from 0.5 to 0.35 (gained 3 true positives), the Logistic Regression classifier did not benefit as much at 0.35 (gained 1 true positives). Both had the best recall accuracy tradeoff at 0.35, so for the final models this was used.

TESTING AND CONCLUSION

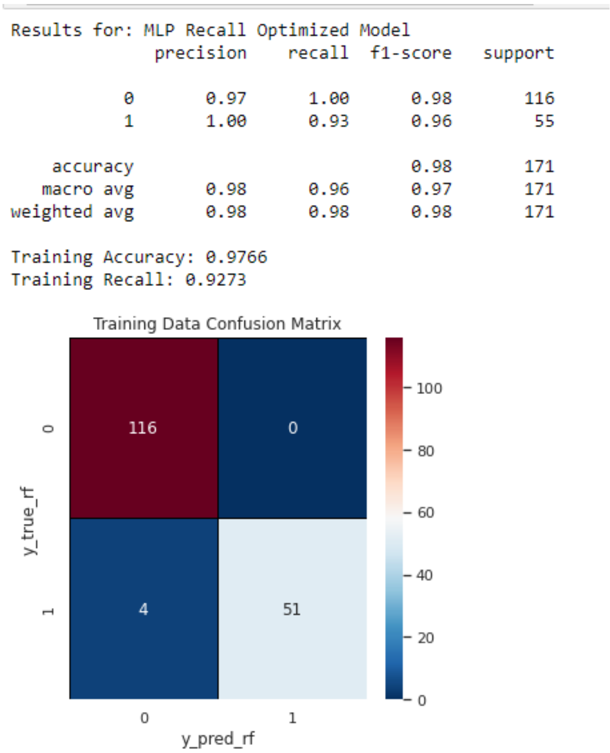
Testing the final models and their threshold values on the test set the following results were obtained

	MLP RECALL OPTIMIZED	LOGISTIC REGRESSION ACCURACY OPTIMIZED	STANDARD FNA	CORE BIOPSY
Accuracy	97.7%	97.1%	–	–
Recall/Sensitivity	92.7%	92.7%	65 – 92.4%	88.7 – 99.6%
Precision/Specificity	100%	98%	60 – 100%	98 – 100%

Though the recall for both is better than only an FNA analysis, the amount of false negatives that remain in the test data (4 each) suggest that both final models should be paired with FNA and human analysis if they do not detect a malignant tumor. More observations or more components in the PCA could help improve this model as well. Attempting another model with a random forest classifier without PCA could also provide useful results.



```
Results for: Logistic Regression Accuracy Optimized Model
              precision    recall  f1-score   support

           0       0.97      0.99      0.98       116
           1       0.98      0.93      0.95        55

    accuracy                           0.97       171
   macro avg       0.97      0.96      0.97       171
weighted avg       0.97      0.97      0.97       171

Training Accuracy: 0.9708
Training Recall: 0.9273
```



```
Results for: MLP Recall Optimized Model
              precision    recall  f1-score   support

           0       0.97      1.00      0.98       116
           1       1.00      0.93      0.96        55

    accuracy                           0.98       171
   macro avg       0.98      0.96      0.97       171
weighted avg       0.98      0.98      0.98       171

Training Accuracy: 0.9766
Training Recall: 0.9273
```