MA463X: Data Analytics & Statistical Learning

Project Report

Michael Giancola, Ranier Gran, Cassidy Litch, Charles Lovering, Cuong Nguyen

July 20, 2017

1 Overview

We chose to analyze the diagnostic Breast Cancer dataset from the UCI Dataset Repository[1]. This data set can be used to classify tumors to be either benign or malignant. The data set contains 30 predictors determined from digitized image of a fine needle aspirate (FNA) of a breast mass. The predictors describe the characteristics of the cell nuclei present in the tumor. The features include:

1. radius (mean of distances from center to points on the perimeter)
2. texture (standard deviation of gray-scale values)
3. perimeter
4. area
5. smoothness (local variation in radius lengths)
6. compactness ($\frac{perimeter^2}{area - 1.0}$)
7. concavity (severity of concave portions of the contour)
8. concave points (number of concave portions of the contour)
9. symmetry
10. 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 tumor. Therefore each feature contains three predictors which totals 30 predictors in the data set to determine the classification of the tumor, whether the tumor is malignant or benign. Additionally, these predictors can be used to determine which factors affect whether or not a tumor is malignant. Therefore these predictors can also be used in the inference problem.

2 Preliminary Analysis

Before doing any analysis and training the models, we sample a subset of the dataset to use as a test set and leave it untouched until evaluating the final models. In this section, we first investigate some aspects of the training data to gain a better understanding about the data and carry out appropriate preprocessing.

Figure 1 shows the histograms of the predictors. We can see that there is big difference in scale between some predictors. For example, *mean_area* has value over the range from 0 to over 2000, whereas *mean_concav* is only from 0 to 0.4. This observation prompts us to normalize the data by subtracting the mean and divided by the standard deviation.

[1]The data set cites the following sources:[1][3][4][2]

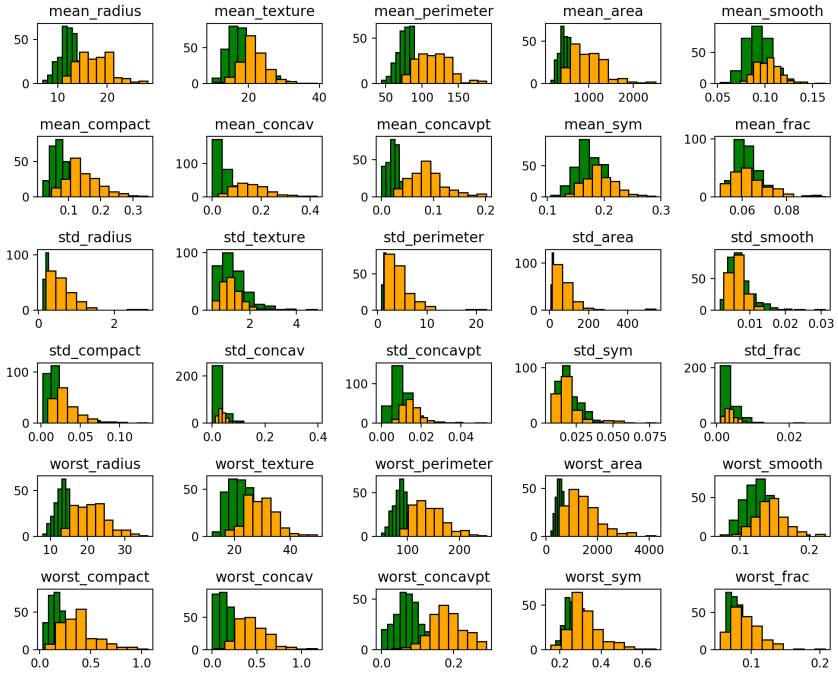

Figure 1: Histograms of the predictors. Green columns correspond to examples labeled with "benign". Red columns correspond to examples labeled with "malignant".

Next, we perform PCA on the normalized training data and reduce the dimensions for visualization. The plot of explained variance by number of principal components in Figure 2 shows us that using 15 components is enough to explain almost 99% of variance in the data. By using the first 2 and 3 principal components, we can visualize the data in 2 and 3 dimensional plots as shown in Figure 3 and Figure 4. The plots show promising structure of the data where linear classifiers such as LDA or Logistic Regression can perform well.

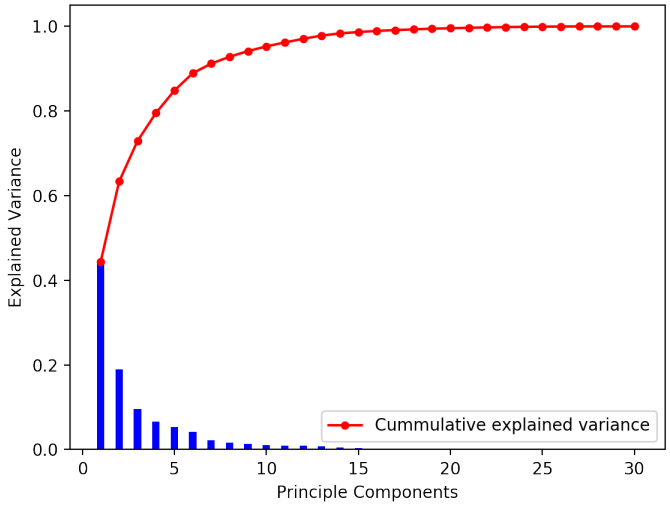

Figure 2: Explained variance by number of principal components.

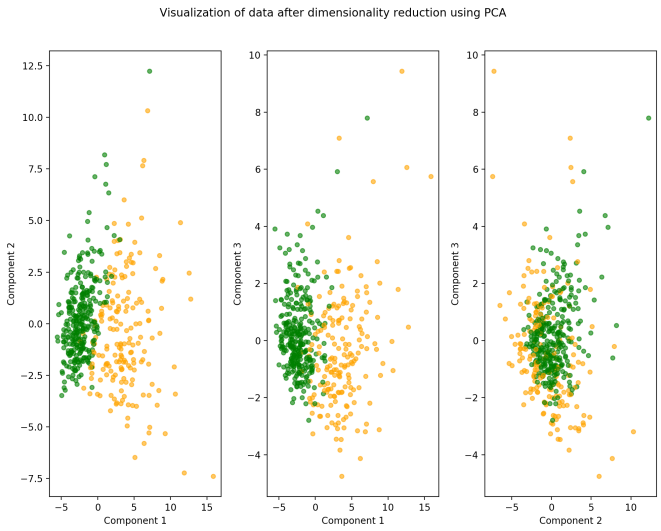

Figure 3: Visualization of the training data using 2 principal components. Green dots are examples labeled with "benign". Orange dots are examples labeled with "malignant".

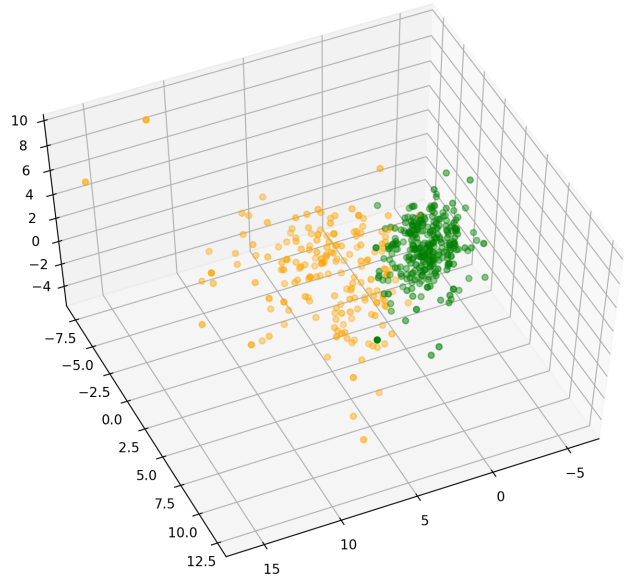

Figure 4: Visualization of the training data using 3 principal components. Green dots are examples labeled with "benign". Orange dots are examples labeled with "malignant".

3 Approaches

In this section, we present the results of the different models that we experimented with. Overall, we tried 6 different models: K-Nearest Neighbors, Logistic Regression, LDA & QDA, Random Forest and an ensemble of the 5 previous models. For each model, we tuned the hyper-parameters using k-fold cross-validation. The best set of hyper-parameters is used in the ensemble.

Metrics

The three metrics we used were accuracy, recall, and precision. Recall is how well a model is able to label positive points, how many positive samples it classifies as positive. In our case positive is malignant. Thus recall is important to us because **False Negatives** (classifying a malignant tumor as benign) will kill patients. Recall is the ratio of True Positives to All Positive Values. Precision is the success rate of your positive classifications - True Positives to Proposed Positives. Accuracy is the number of correct classifications (True Positive and True Negative) to total classifications.

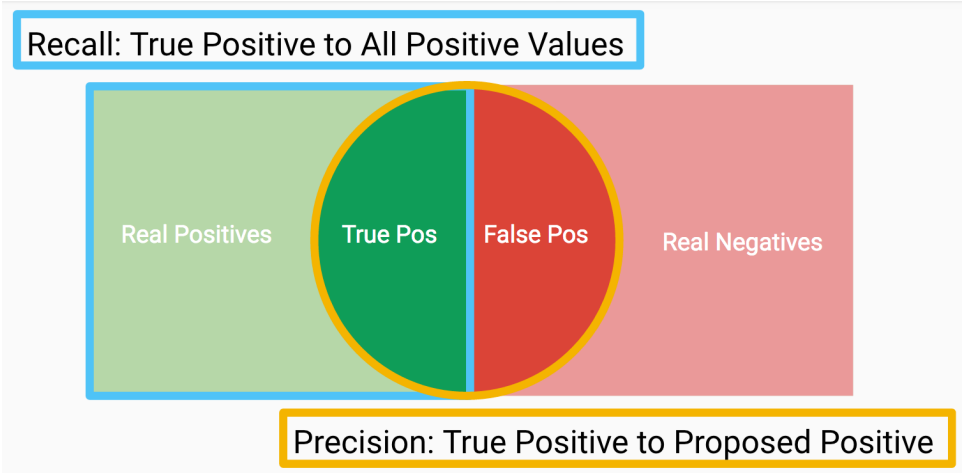

Figure 5: Visually Understanding Recall and Precision

Validation Methodology

In order to avoid data snooping we employed a uniform strategy for evaluating the models and tuning hyper-parameters. Firstly we split into train and test data. Then for each model and hyper parameter we would bootstrap to create 100 sampled datasets and then perform 10-fold cross validation on each sample. We are careful to perform preprocessing steps such as lasso, PCA, and normalization solely on the training data and then transform the test data based on training transformation information (mean, standard deviation, principal components). Thus, we applied preprocessing steps for each fold.

3.1 K-Nearest Neighbors (KNN)

First, we tested possible numbers of principal components between 1 and 30 with neighborhood size = 3. These results are found in the left graph in figure 6. From this graph, the accuracy, recall, and precision all increase as the number of principal components increase. Therefore we chose to not use PCA for KNN. We tested all odd k values between 3 and 21 inclusive, then 31, 51, 71, and 91. The reason that we tested above 21 was because, as shown in the figure below, the accuracy for $k \in \{3, ..., 21\}$ is within a single percentage. Performance begins to drop significantly after $k = 31$. Ultimately, we chose $k = 3$ as the final value as KNN performed well for $k = 3$ under both 10-fold and leave one out cross validation.

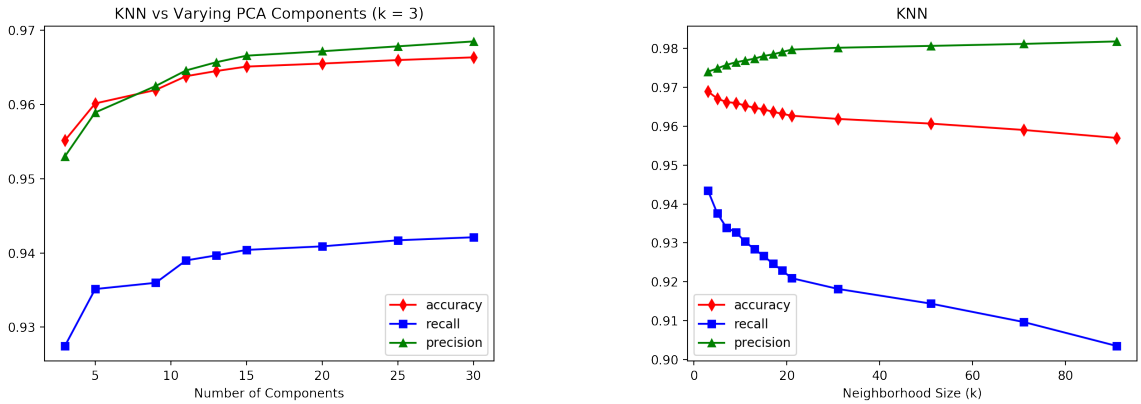

Figure 6: KNN Accuracy using Cross Validation

3.2 LDA & QDA

We first tested using PCA, testing with $[3, 5, 9, 11, 13, 15, 20, 25, 30]$ predictors. We then tested using Lasso using *alphas* $\in \{0.1, 0.2, 0.25\}$ we also included using no Lasso on the data for consistency. We then applied Lasso after choosing our "best" number of predictors (30) from PCA. We saw, by combining the two methods, we actually lost accuracy.

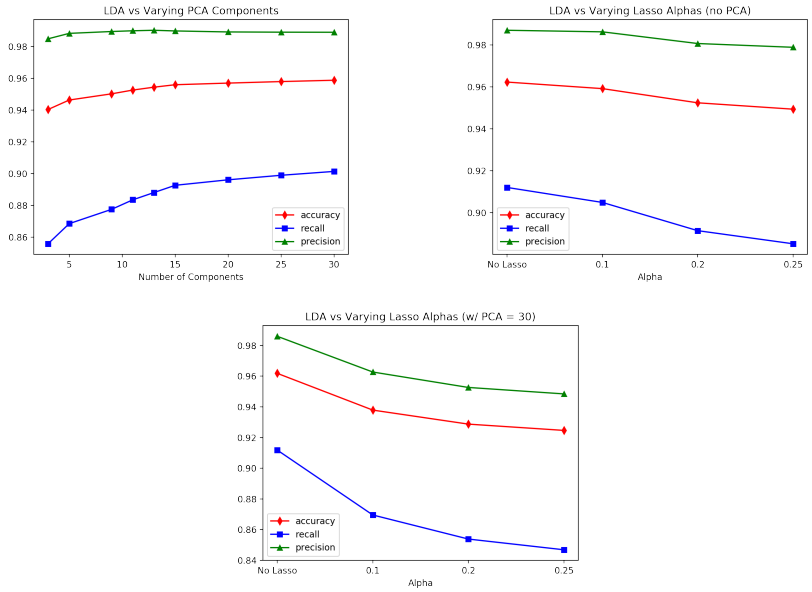

Figure 7: LDA Accuracy using Cross Validation

We found that the same results applied to QDA as well. Upon running Lasso on the data, both with PCA and without, our accuracies decreased again.

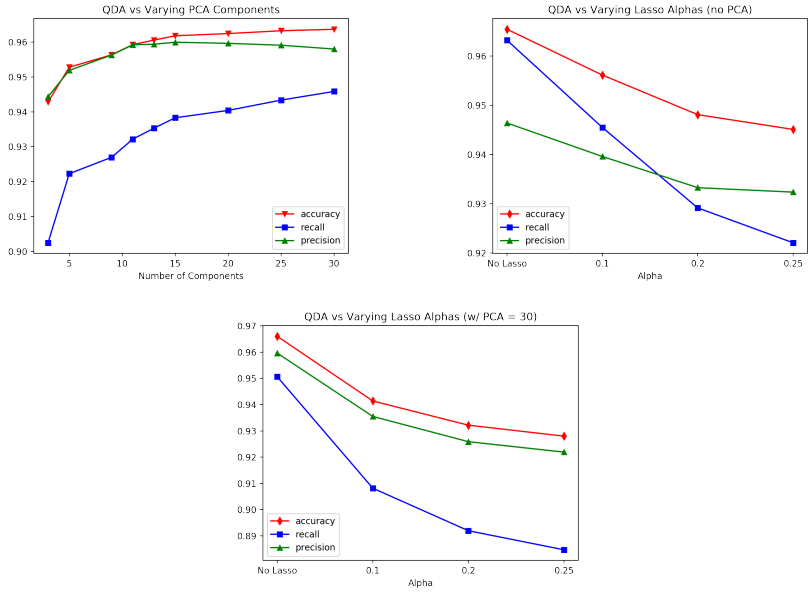

Figure 8: QDA Accuracy using Cross Validation

Regardless, through our comparison we believed that QDA would be a better fit on the data than LDA for this classification problem. The overall accuracy of QDA was slightly higher in every case and did not seem to be in danger of overfitting the data.

3.3 Logistic Regression

We experimented with 6 set of hyper-parameters. Here, C is the "budget" of the weights:

1. L2 regularization with C = 1
2. L2 regularization with C = 10
3. L2 regularization with C = 100
4. L1 regularization with C = 1
5. L1 regularization with C = 5
6. L1 regularization with C = 10

For each set of hyper-parameter, we also cross-validated the model over the number of used principal components. Figure 9 shows the accuracies obtained from the validation. Table 1 summarizes the best number of components for each set of hyper-parameters.

Table 1: Number of principal components that gives the best validation accuracy for each set of hyper-parameters

Regularization	C	#Principal Components	Accuracy (%)
L2	1	29	98.3
L2	10	24	98.3
L2	100	21	98.3
L1	1	12	98.3
L1	5	25	98.3
L1	10	26	98.4

It is clear that the best set of hyper-parameters is **L1 regularization with C = 01 using 26 principal components**, which gives a cross-validation accuracy of **98.4%**.

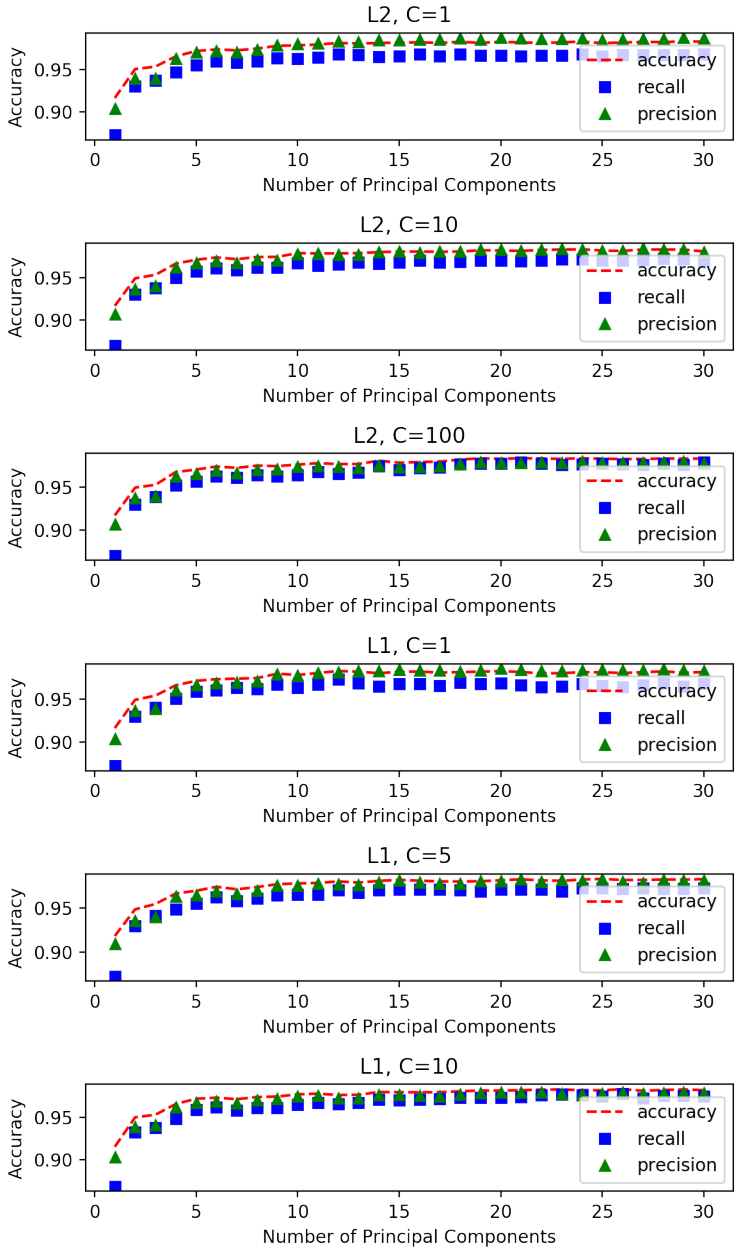

Figure 9: Validation accuracy of different set of hyper-parameters over the number of used principal components.

3.4 Random Forest Classification

We experimented with the number of principal components for PCA computed before random forest classification. Then, we experimented with the number of trees in the random forest. When 10-fold cross-validation was computed for 9 possible principal components, between 1 and 30 using a forest of 50 trees, the result is depicted in figure 10. In this figure, the highest combined accuracy, recall, and precision occurs when there are 5 principal components. Therefore we chose 5 principal components to perform testing on the optimal number of trees in the forest.

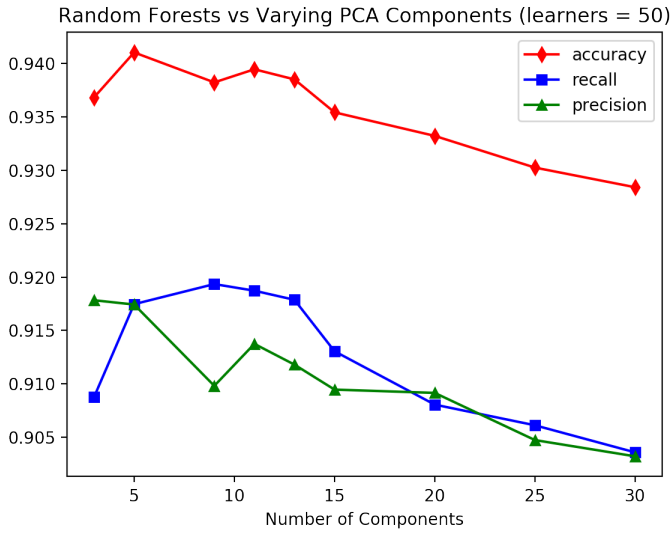

Figure 10: Random Forest Accuracy using Cross Validation and to find the optimal number of principal components.

Once the optimal number of principal components was determined to be 5, the optimal number of trees needed to be determined. We performed 10-fold cross-validation on PCA with 5 principal components and random forest classification with 15 different possible numbers of trees between 1 and 500. The accuracy of each possible number of trees is depicted in figure 11.

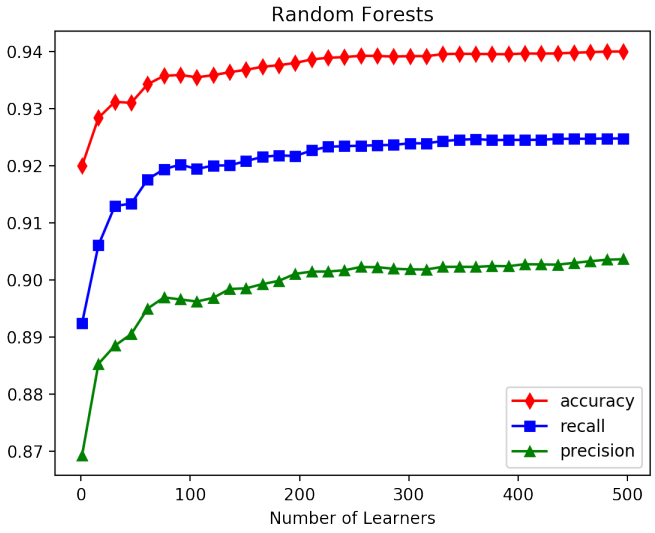

Figure 11: Random Forest Accuracy using 10-fold cross-validation and to find the optimal number of trees.

In this figure, the accuracy continuously increases as the number of trees in the forest increases. Therefore we decided to pick the number of trees based on the location where the slope of each recall and precision appeared to decrease. Therefore we selected 250 trees as the final number of trees for random forest classification on PCA with 5 principal components. Therefore the best accuracy is around **94%** with the **number of trees = 250 using 5 principal components**.

3.5 Bagging

Bagging is training an ensemble of models trained on samples from bootstrap aggregation. The setup for this is represented in figure 12. We cross-validated over which base learner to use.

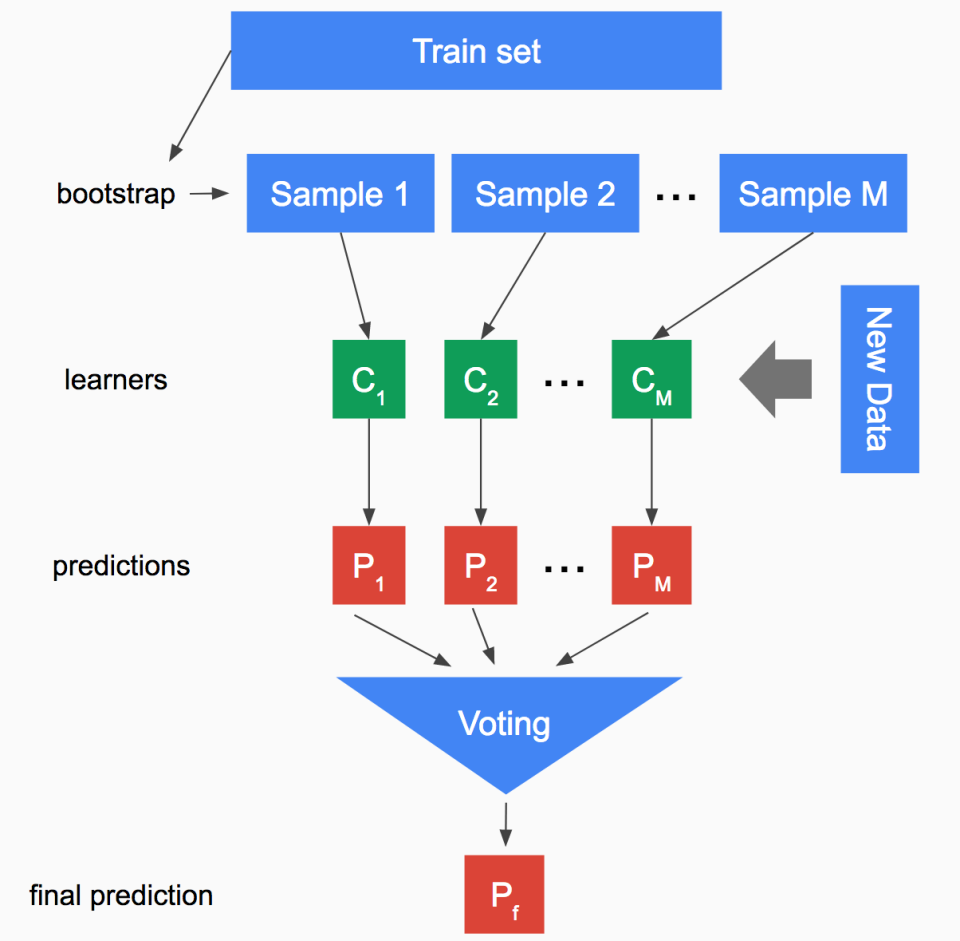

Figure 12: Outline of Bagging

We found that logistic regression performed the best for all metrics, as show the Figure 13 below.

3.6 Boosting

We also tested an ensemble model that used boosting with logistic regression as the base learner. It performed worse than bagging with an accuracy of 96.67%, recall 95.6%, and precision 94.3%.

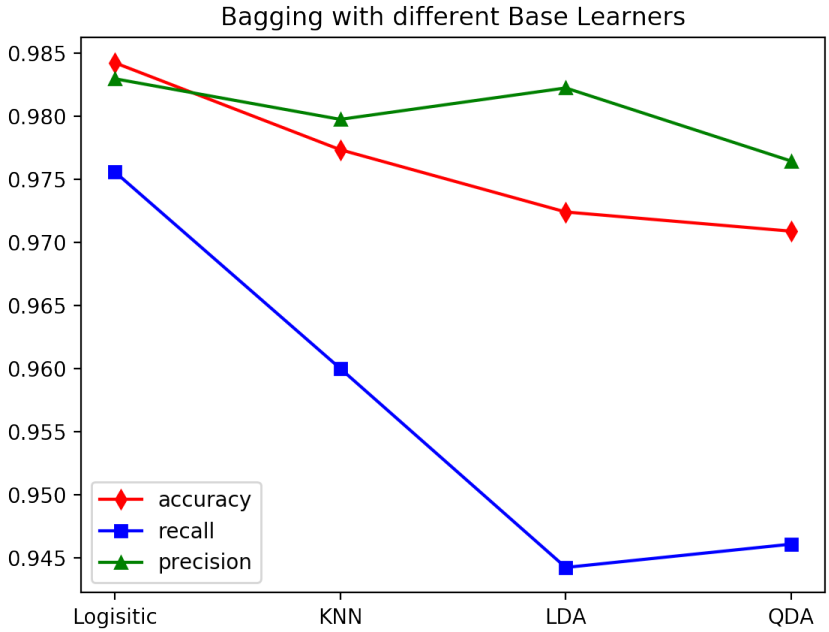

Figure 13: Visualization of performance of Boosting with different learners

4 Results

4.1 Final Approach

After trying the models described above, the final model we would *sell to the client* would be bagging with logistic regression. Although the accuracy was equal to that of the regular logistic regression, and the training time was greatly increased, we expect it to be a more stable as it has seen a greater diversity of data.

Table 2: Performances of Each Model-Results

	Accuracy	Recall	Precision
Bagging	96.70%	93.33%	96.55%
Forest	90.10%	86.67%	83.87%
KNN	96.70%	96.66%	93.54%
LDA	93.40%	80.00%	100.00%
QDA	95.60%	93.33%	93.33%
Logistic	92.30%	90.00%	87.09%

In figure 14, the accuracy, recall, and precision for each algorithm on the testing set is depicted. Our recommended algorithm of bagging with logistic regression performed better than most of the algorithms in terms of accuracy, bagging had the same accuracy as KNN. Interestingly, the recall for KNN was higher than the recall for bagging, but the precision for bagging was higher than the precision for KNN. In this particular problem, the True Positive Rate is more important than the True Negative Rate. This is because when detecting types of cancer, telling someone who has a malignant tumor that it is benign is much more significant than incorrectly marking a benign tumor. Therefore recall is more significant than precision for this problem, and KNN is the true best algorithm on the testing data. Although KNN is best when tested on the testing data, the bagging example is also very close second best detection algorithm.

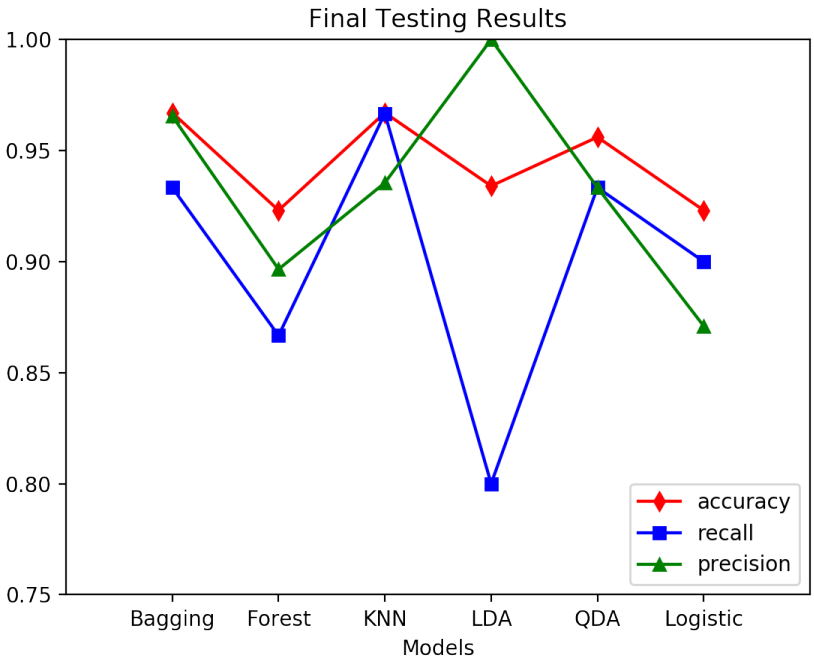

Figure 14: Performance of Each model

4.2 KNN

Figure 15 depicts the testing of the optimal KNN algorithm, from the training data, on the testing data. KNN was able to accurately predict benign tumors 97% of the time and malignant tumors 97% of the time. As stated in section 4.1, KNN had the highest accuracy in the testing data.

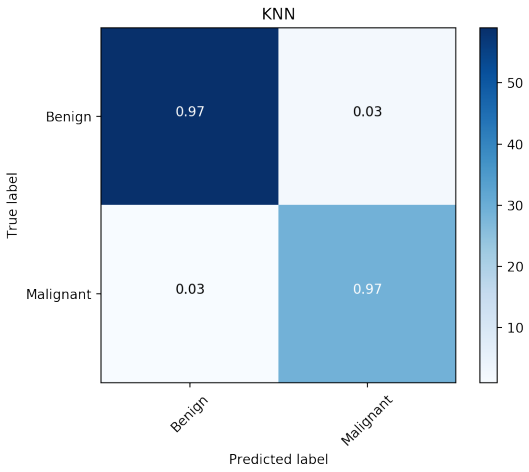

Figure 15: Confusion Matrix of KNN

4.3 LDA & QDA

As seen in figure 16 in checking with our testing data we found that QDA performed significantly better than LDA. While LDA gave as high as a 20 false negative rate, QDA had only a 7 false negative rate, overall performing much better on the data.

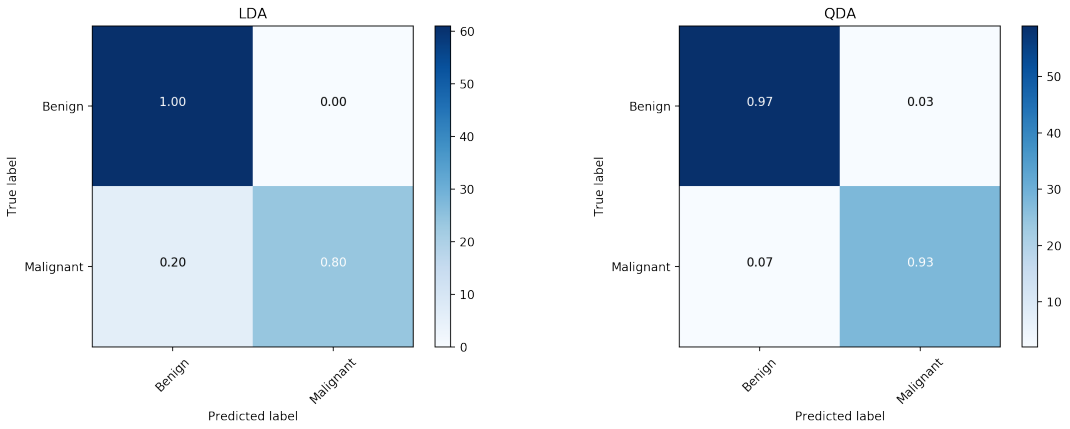

Figure 16: Confusion Matrices of LDA and QDA for Comparison

4.4 Logistic Regression

Figure 17 depicts the testing of the optimal logistic regression algorithm, from the training data, on the testing data. Logistic regression was able to accurately predict the tumor to be benign 97% of the time, but was only able to accurately predict the tumor to be malignant 90% of the time. Therefore, even though logistic regression had one of the highest accuracies in the training data, the method was not as accurate on the testing data. As shown in section 4.3, QDA had a higher True Positive Rate and True Negative Rate than logistic regression.

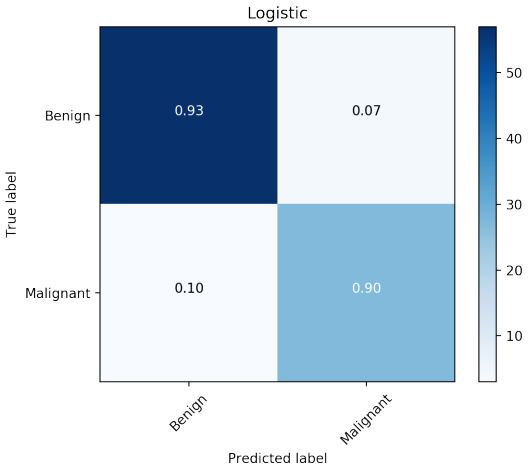

Figure 17: Confusion Matrix of Logistic Regression

4.5 Random Forest

Figure 18 depicts the testing of the optimal random forest algorithm, from the training data, on the testing data. The random forest algorithm was able to accurately predict the tumor to be benign 95% of the time and accurately predict the tumor to be malignant 87% of the time. These accuracies are worse than the accuracies from logistic regression as was expected from the results on the training data.

4.6 Bagging

Figure 19 depicts the testing of the optimal bagging algorithm, from the training data, on the testing data. The method was selected as the best method from the training data as described in section 4.1. Although this detection algorithm did not have the best accuracy on the testing data, the accuracy was the second best. The

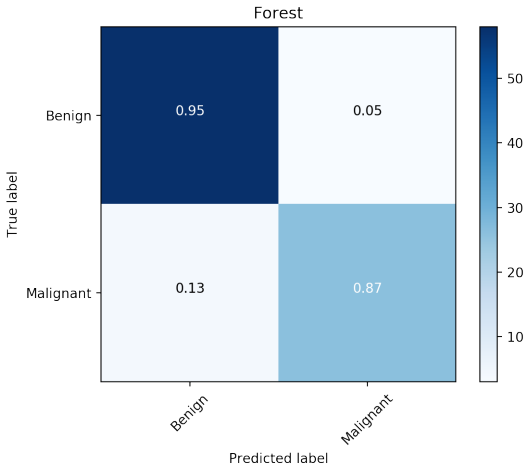

Figure 18: Confusion Matrix of Random Forest

bagging algorithm was able to accurately detect malignant tumors 93% of the time and benign tumors 98% of the time. The accuracy for predicting benign tumors is slightly higher than KNN, but the problem is more interested in accurately detecting malignant tumors. Since KNN performs better on malignant tumors, KNN is slightly better for the testing data set.

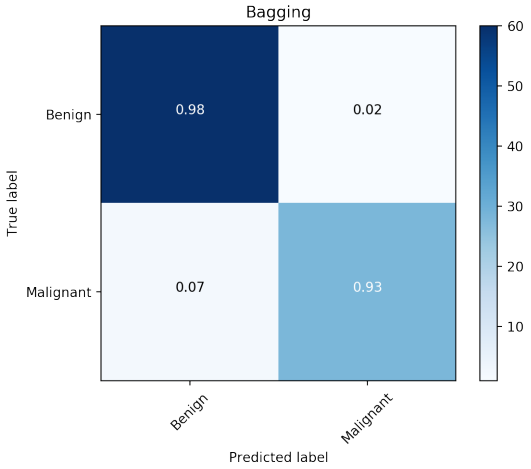

Figure 19: Confusion Matrix of Bagging

5 Conclusions

Although we were aware that model performances on training data, even with boot strapping and cross validation, would not be equavalent to the testing results, we were surprised to the extent that this was true. Specifically, logistic regression performed the second best by a very small margin, and ended up performing much worse on the testing data. It dropped from 98% to 92%. Additionally, KNN performed better than bagging in the testing data. Another observation we had from our analysis was that normalization before PCA is important. This is because the variances and scale of the different predictors may be different, and cause PCA to unduly favor them. This is data dependent.

We found that the intrinsic shape of the data for different classes had an observable effect on the different models. Specifically, the malignant data had a greater variance than the benign data and this impacted the precision and recall.

References

[1] K. P. Bennett and O. L. Mangasarian. Robust linear programming discrimination of two linearly inseparable sets. *Optimization Methods and Software*, pages 23–34, 1992.

[2] O. L. Mangasarian and W. H. Wolberg. Cancer diagnosis via linear programming. *SIAM News*, 23(5):1, 18, 1990.

[3] O. L. Mangasarian R. Setiono and W.H. Wolberg. Pattern recognition via linear programming: Theory and application to medical diagnosis. *SIAM Publications*, pages 22–30, 1990.

[4] William H. Wolberg and O.L. Mangasarian. Multisurface method of pattern separation for medical diagnosis applied to breast cytology. *Proceedings of the National Academy of Sciences*, 87:9193–9196, 1990.