# Using Machine Learning to Diagnose Breast Cancer

*Adam Northrup, Emilie Campos*

*December 14, 2018*

## Introduction

Fine needle aspiration (FNA) is a less invasive method of sampling patient tumor tissue than classic surgical removal. FNA may refer to the removal of small portions of a tumor or the visualization of cells on a microscope slide. We will refer to the latter for the remainder of this analysis. Diagnosis of malignant cancer from FNA is difficult relative to surgical resection due to the small number of cells captured and lack of tissue context.

Certain features of nucleus shape and size have been identified which may be indicative of malignancy by Street, et al (1993). These features include radius, perimeter, area, compactness, smoothness, concavity, fractal dimension, and texture.

We will attempt to identify features of importance and compare algorithms for predicting malignancy based on data from 556 subjects. We will compare classification trees, random forests, and support vector machines.

### Data

The data consists of 569 subjects, each with 32 variables. Missing data was indicated as zero, since measures of size and shape must be greater than zero. These missing values are not missing at random–they account for certain variables only. For the use of random forests, we removed all missing data, leaving 556 subjects.

The following variables were included in the data set:

- ID
- Diagnosis
- Radius mean/standard error/worst
- Texture mean/standard error/worst
- Perimeter mean/standard error/worst
- Area mean/standard error/worst
- Smoothness mean/standard error/worst
- Compactness mean/standard error/worst
- Concavity mean/standard error/worst
- Concave points mean/standard error/worst
- Symmetry mean/standard error/worst
- Fractal dimension mean/standard error/worst.

The Diagnosis variable is the true diagnosis used for training models, and ID refers to the subject identification, thus will be disregarded. "Worst" refers to the mean of the worst or highest measurements across all cells for a given variable.

From Figure 1, we note that some of the variables seem to have differing means so our classification methods should be able to pick that up.

## Methods

Data was divided into 75% training and 25% testing sets at random.

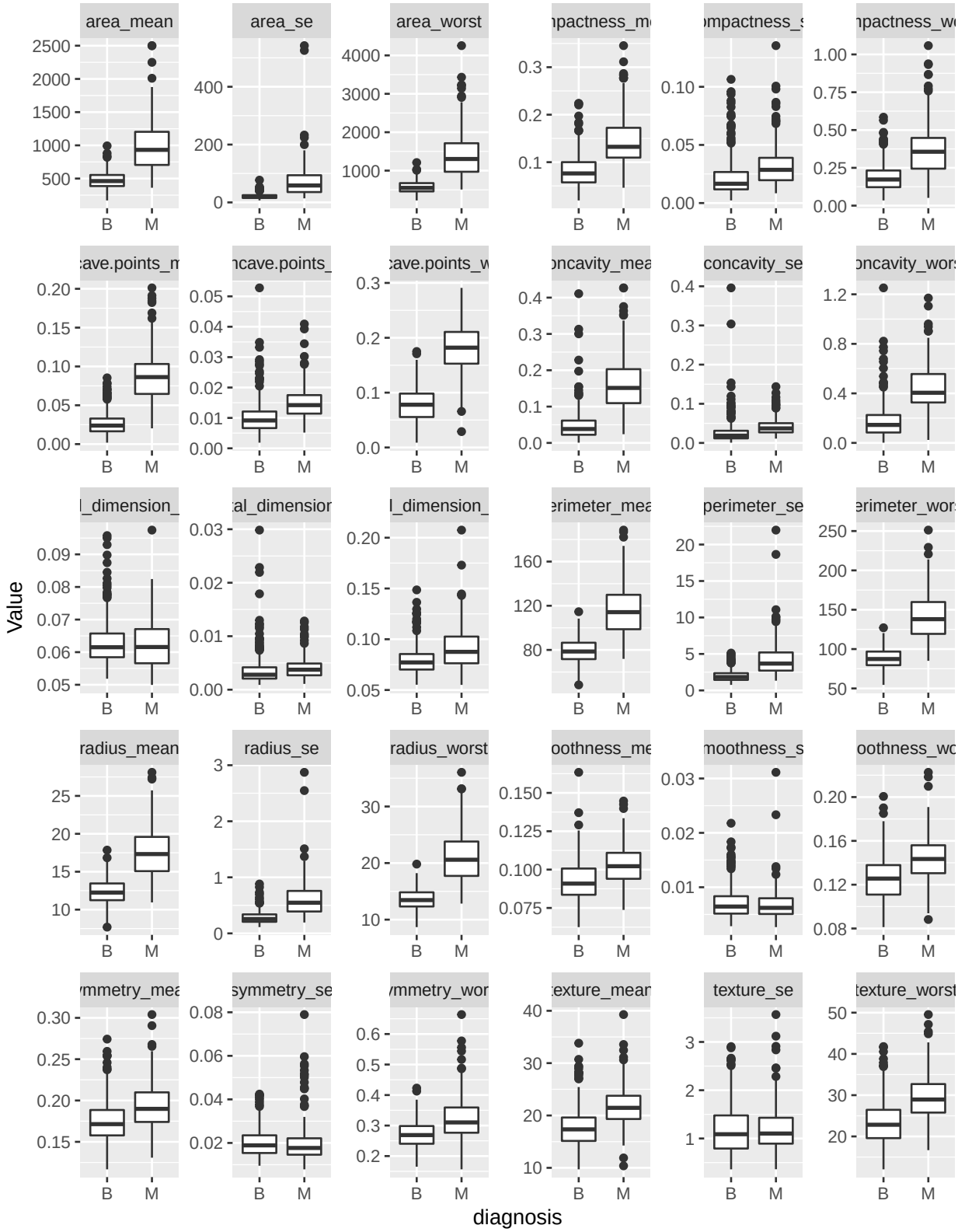


Figure 1: All Variables Boxplots

### Single Classification Tree

Using the "tree" package in R, we will create a single classification tree using all 30 remaining variables. We will assess the tree's misclassification error rate on the test set before pruning. Pruning will use the classification error to determine purity and remove the least important splits. The pruning process involves cross-validation to determine the node size and cost-complexity parameter that result in the lowest cross-validation error. We will use the node size determined to have the lowest error to select the best tree. The pruned tree will be compared to the unpruned using misclassification error on the test set.

### Random Forest

Using the tuneRF function from the "randomForest" package, we will create a random forest using all 30 variables. The "doBest" argument allows us to choose the optimal number of variables available from which to choose at each split ($m_{try}$) by minimizing the out of bag (OOB) error.

In order to determine variable importance, we will create 50 separate random forests and determine the importance rank for each variable based on the amount that the variable decreases the Gini index. Summing the ranks across the 50 trees will give a score, and the five variables with the highest scores will be considered the most important.

### Support Vector Machines

Although we would like to use the simplest model possible that gets the job done, we want to implement a support vector machine just for example. Using the package `e1071`, we can run the support vector machine algorithm on the variables found most important by the classification trees as well as on all of the variables. We'll use the `tune` function in order to find the best cost and gamma combination by misclassification error.

## Results

### Single Classification Tree

Figure 2 shows a classification tree grown from all 30 variables. This classification tree was used to predict the outcomes for subjects in the test set. The agreement between these predictions and the true diagnoses is shown in Table 1.

Table 1: Classification Tree Confusion Matrix

| | Truth | |
|---|---|---|
| **Predicted** | Benign | Malignant |
| **Benign** | 73 | 10 |
| **Malignant** | 2 | 54 |

The tree has a misclassification error rate of 0.086. See Table X for comparison of methods.

#### Tree Pruning

Tree pruning required use of cross-validation to determine the tree size with lowest error. Figure 3 shows cross-validation error as a function of tree size and of the cost-complexity parameter.

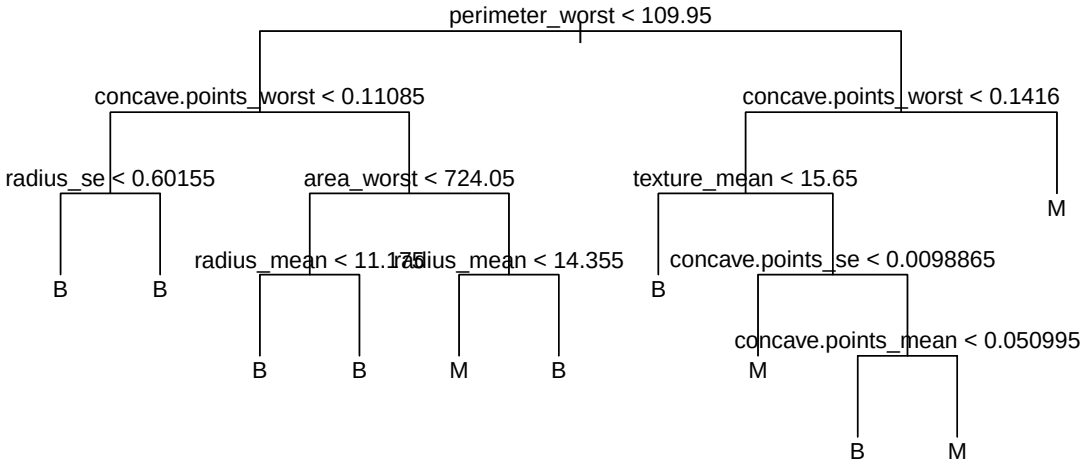


Figure 2: Classification Tree

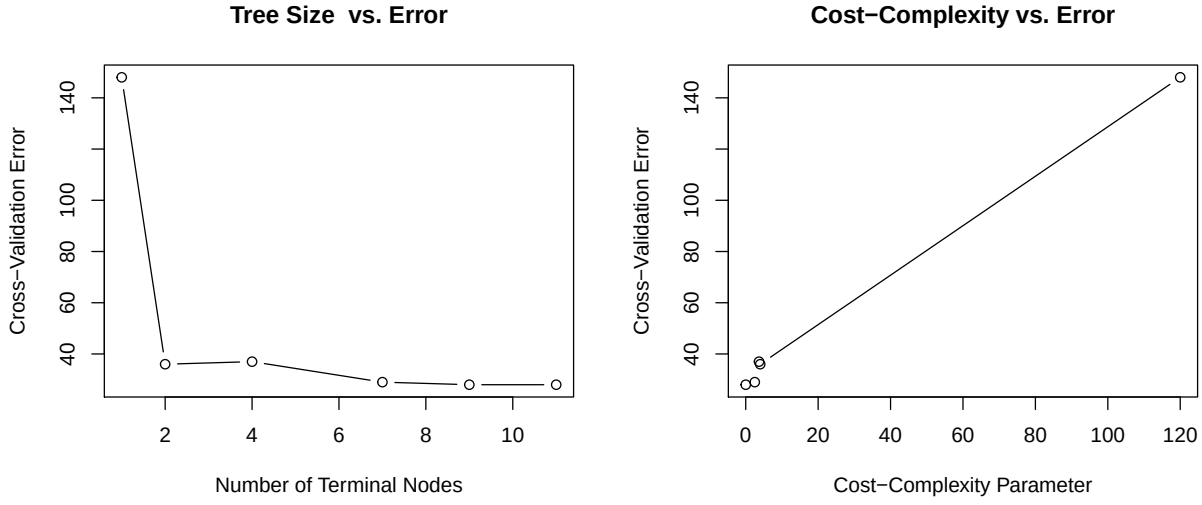


Figure 3: Error as Function of Tree Size and Cost-COmplexity

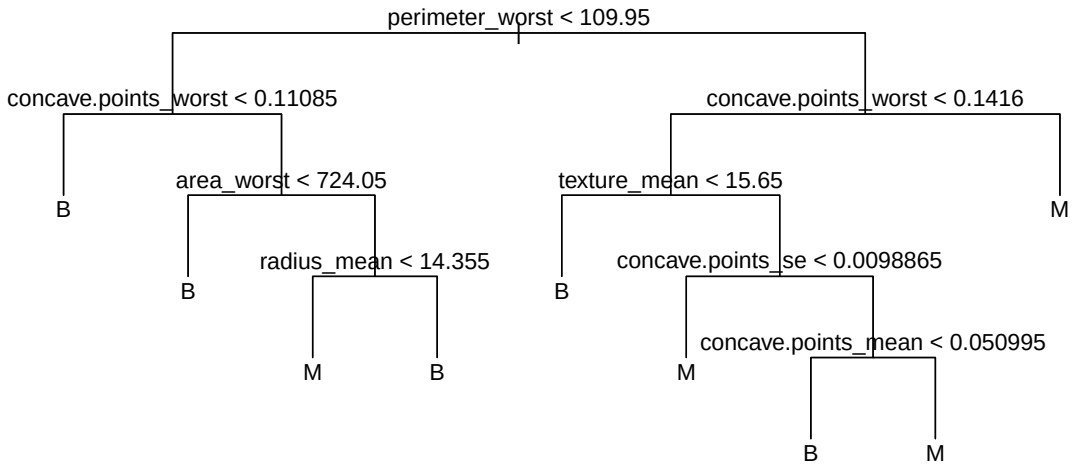


Figure 4: Pruned Tree

The pruned tree is shown in Figure 4, with a first split of worst perimeter at 109.95. Further splits are on worst concave points, worst area, radius mean, and texture mean.

The confusion matrix from comparing the predicted diagnoses from the pruned tree to the true diagnoses is shown in Table 2. The misclassification error rate for predictions from the pruned tree on the test data is 0.086.

Table 2: Pruned Tree Confusion Matrix

| | Truth | |
|---|---|---|
| **Predicted** | Benign | Malignant |
| **Benign** | 73 | 10 |
| **Malignant** | 2 | 54 |

## Random Forest

We chose a value for $m_{try}$ for our random forest that minimized the OOB error. This can be seen in Figure 4, as can the ranking of importance of variables in the model.

We chose $m_{try} = 3$ to produce our random forest. We used the forest to predict diagnoses on the test set and found a misclassification error rate of 0.05.

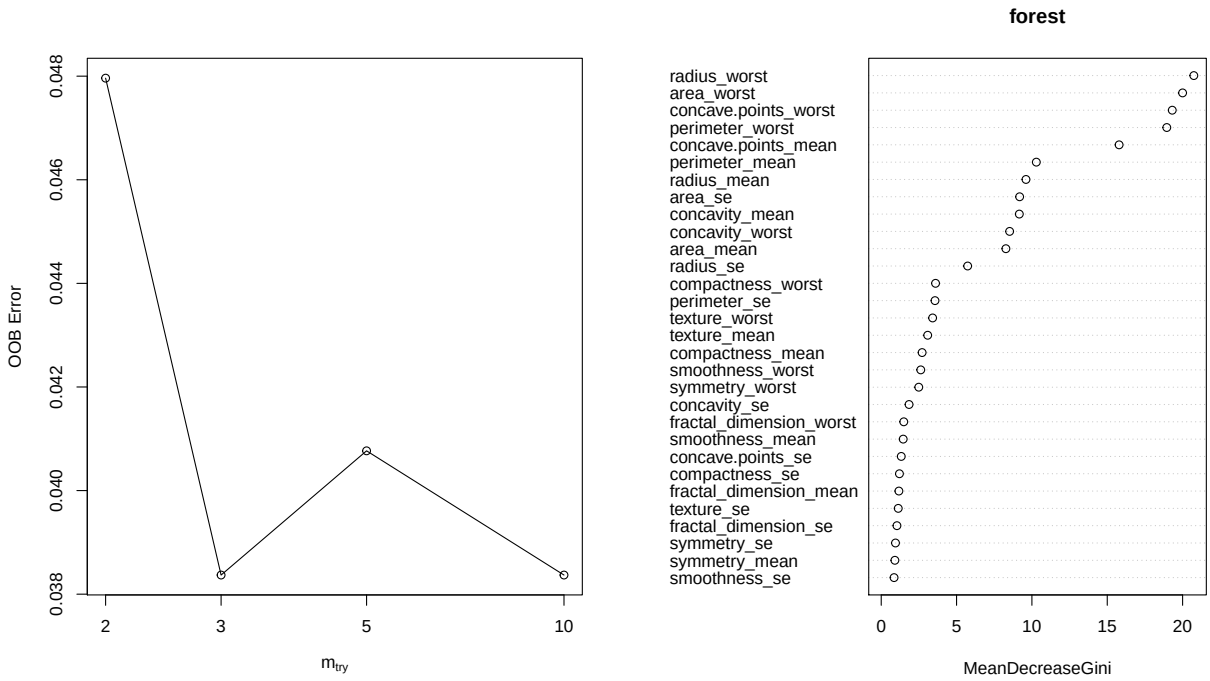


Figure 5: Random Forest

Table 3: Random Forest Confusion Matrix

| | Truth | |
|---|---|---|
| **Predicted** | Benign | Malignant |
| **Benign** | 73 | 5 |
| **Malignant** | 2 | 59 |

### Variable Importance

Variable importance scores were calculated from 50 random forests. The most important variables by score are as follows: worst perimeter, worst radius, worst area, concavity mean, and perimeter mean.

## Support Vector Machines

We used the top 5 most important variables found in the random forest as a variable selection procedure. From that, using the `best.tune` function, we find that the cost 1200 and gamma 0.007 has the lowest misclassification rate. The confusion matrix is given in Table 4.

In addtion, we also ran the support vector machine with all 30 variables for comparison. Using the `best.tune` function, we find that the cost 900 and gamma 0.005 has the lowest misclassification rate. The confusion matrix is given in Table 5.

Figure 6 has the ROC curves from each of the SVMs we ran. We can see that using only 5 of the variables gives tremendous results, almost identical to using all 30 variables.

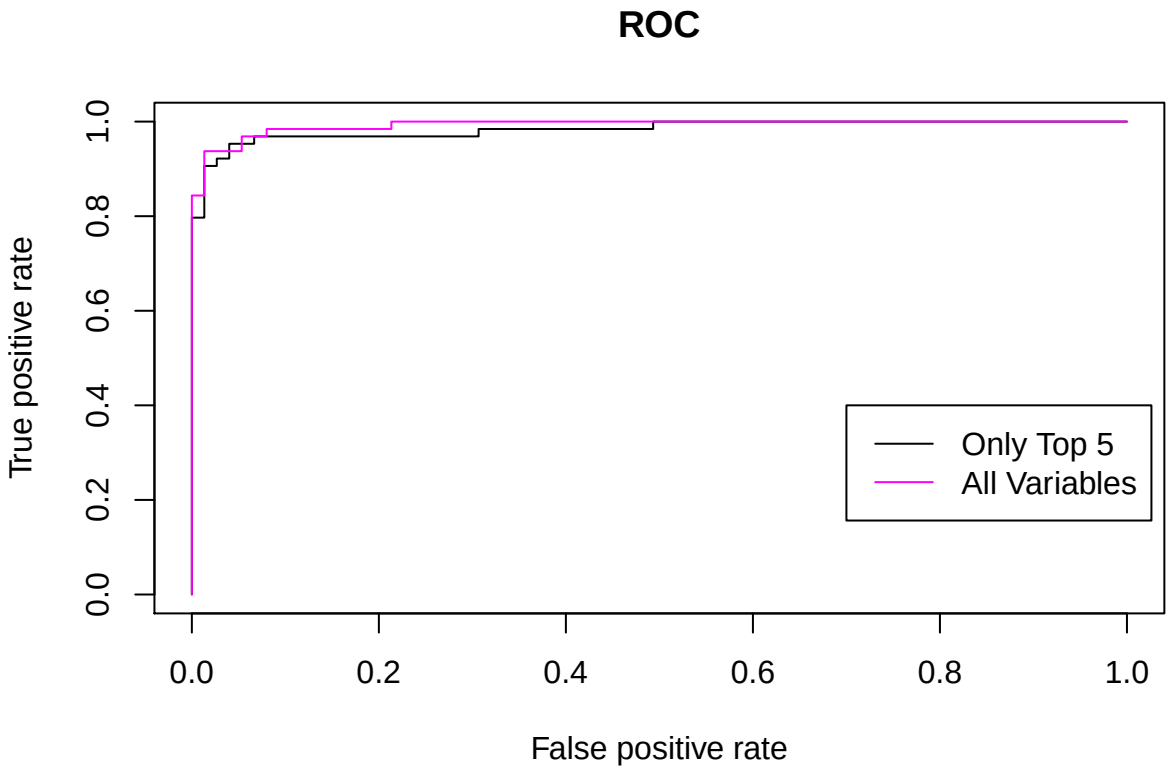


Figure 6: ROC Curve for SVMs

Table 4: SVM Top 5 Variables Confusion Matrix

| | Truth | |
|---|---|---|
| **Prediction** | Benign | Malignant |
| **Benign** | 72 | 5 |
| **Malignant** | 3 | 59 |

Table 5: SVM Confusion Matrix

| | Truth | |
|---|---|---|
| **Prediction** | Benign | Malignant |
| **Benign** | 71 | 4 |
| **Malignant** | 4 | 60 |

### Comparison of Algorithms

The unpruned and pruned classification trees each had a misclassification error rate of 0.09 on the test data. The random forest improved on this with an error rate of 0.05. The support vector machines each had a missclassificaion error rate of 0.058.

Table 6: Comparison of Algorithms

| Algorithm | Misclassification Error Rate |
|---|---|
| Unpruned Tree | 0.086 |
| Pruned Tree | 0.086 |
| Random Forest | 0.050 |
| SVM Top 5 | 0.058 |
| SVM All | 0.058 |

## Conclusions

Ideally, since we are diagnosing cancer, a truly horrible disease, we want the best classifier possible. It would be potentially devastating to give a false negative diagnosis. However, to keep costs low, it may be advantageous to find a subset of tumor characteristics that predict the diagnosis very well. Although our algorithms were not designed to avoid false negatives, we found that by using a random forest, a simpler and faster model than the support vector machines, we can diagnose cancer extremely accurately. Diagnosis through FNA using this algorithm would be recommended over surgical resection, and concerned physicians could test algorithm negatives more thoroughly.

## References

Street, N., Wolberg, W.H., & Mangasarian, O.L. (1993). Nuclear Feature Extraction for Breast Cancer Diagnosis. International Symposium on Electronic Imaging: Science and Technology: San Jose, CA. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.56.707&rep=rep1&type=pdf