Jacob Blizzard
Megan Eckstein
Jeffrey Griffiths
Tess Newkold

Breast Cancer

Malignant or Benign?

Abstract

This paper examines a set of tumor data from the University of Wisconsin. It was created through collaboration by their oncology and computer science departments. We used this data to classify the tumors as malignant or benign using Classification Trees and Support Vector Machines (SVMs). The classification was done with 9 predictor variables and 1 binary response variable, benign or malignant. The software of choice for running the classification was R, Python, and SPSS. Within R the “e1071” package was used and within Python the “Sklearn” library was used. The AUC of the ROC curves generated from the classifiers covered in this paper range between 0.956 and 1.00 with misclassification rates between 0.38% and 6.3%. The best prediction method in this situation is the support vector machine in python.

Introduction

More than 40,000 men and women die of breast cancer in the United States each year. Also, 1 in 8 women in the United States will be diagnosed with breast cancer in her lifetime. It starts when cells in the breast tissue start to grow at an uncontrolled rate. Since this is a prevalent disease, our group has decided to work with the Wisconsin Breast Cancer dataset. Throughout this project we used classification methods to determine if the tumor cells are malignant or benign based on the predictor variables. If the tumor is malignant, then the tumor cells are dangerous and are considered cancerous; benign means the tumor cells are relatively harmless. This information is critical for physicians to have, to trust, and to guide appropriate actions in subsequent medical decisions. Our objective was to accurately predict whether the tumor cells are malignant or benign using supervised learning techniques. We used a 75%/25% training/testing dataset split for our classifiers.

First, we used simple classification trees using R, Python, and SPSS. Running a classification tree in R was fairly straightforward, however building a tree in Python proved to be more of a challenge. In Python the parameters to automatically prune a tree are not built into the “Sklearn” library so we had to manually adjust parameters in a trial and error method to get the desired tree. Within SPSS the C&R node was used. Our classification trees yielded out of sample misclassification rates of 3.2%, 3.43%, and 6.79%, respectively.

We then constructed Support Vector Machine(SVM) classifiers using the same set of software. Again, we found success with low misclassification rates. Overall, the out of sample misclassification rates using SVMs were lower than the classification trees. We had testing sample MRs of 4.6%, 1.7%, and 3.7%, respectively. The Python SVM proved to be the best classifier for this data with a misclassification rate of 1.7%.

Data Details

The data set is from the UCI Machine Learning Repository. It was collected by Dr. William H. Wolberg during his clinical cases from January 1989 to November 1991 at the University of Wisconsin Hospitals. There are 10 attributes in this data set and 1 response variable. The different variables and their definitions that will be used in predicting malignancy are found in Table 1.

Attribute	Definition
ID Number	Unique identifier of case
Clump Thickness	Indicates the amount of layers of cell
Uniformity of Cell Size	How consistent the cells' size is
Uniformity of Cell Shape	How consistent the cells' shape is
Marginal Adhesion	Indicates how much the cells on the outside of the epithelial stick together
Single Epithelial Cell Size	Indicates size of epithelial cell relative to the cell uniformity
Bare Nuclei	Indicates ratio of cells not surrounded by cytoplasm to cells that are
Bland Chromatin	Describes a uniform "texture" of the nucleus seen in non-tumor cells
Normal Nucleoli	Describes size and visibility of nucleoli
Mitoses	Indicates level of mitotic cell activity
Class	Response variable

Table 1. Data attributes and definitions.

The types of data and the values they can take are in Table 2 below.

Attribute	Data Type	Values
ID Number	Char	-
Clump Thickness	Int	1-10
Uniformity of Cell Size	Int	1-10
Uniformity of Cell Shape	Int	1-10
Marginal Adhesion	Int	1-10
Single Epithelial Cell Size	Int	1-10
Bare Nuclei	Int	1-10
Bland Chromatin	Int	1-10
Normal Nucleoli	Int	1-10
Mitoses	Int	1-10
Class	Bin	0 = benign, 1 = malignant

Table 2. Data Values.

Exploratory Data Analysis

Summary statistics are shown in Table 3 for the breast cancer dataset. We chose to replace the 16 missing values in the column Bare Nuclei with the average of the column, in doing this it did not change the mean of the column and will allow us to do more analysis in the future. There are 699 observations and 11 variables, 34.5% are classified as malignant and 65.5% classified as benign seen in Figure 1. Figure 2 shows how benign and malignant diagnoses are distributed compared to each other in each respective variable.

	Mean	SD	%Missing
Clump Thickness	4.418	2.815	0
Uniformity Cell Size	3.134	3.051	0
Uniformity Cell Shape	3.207	2.97	0
Marginal Adhesion	2.807	2.85	0
Epithelial Cell Size	3.216	2.21	0
Bare Nuclei	3.545	3.601	16
Bland Chromatin	3.438	2.438	0
Normal Nucleoli	2.867	3.053	0
Mitoses	1.589	1.715	0

Table 3. Summary Statistics

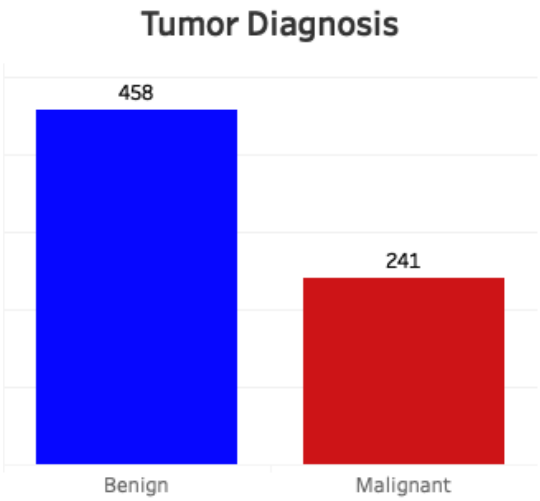

Figure 1. Chart of classification of benign or malignant tumors

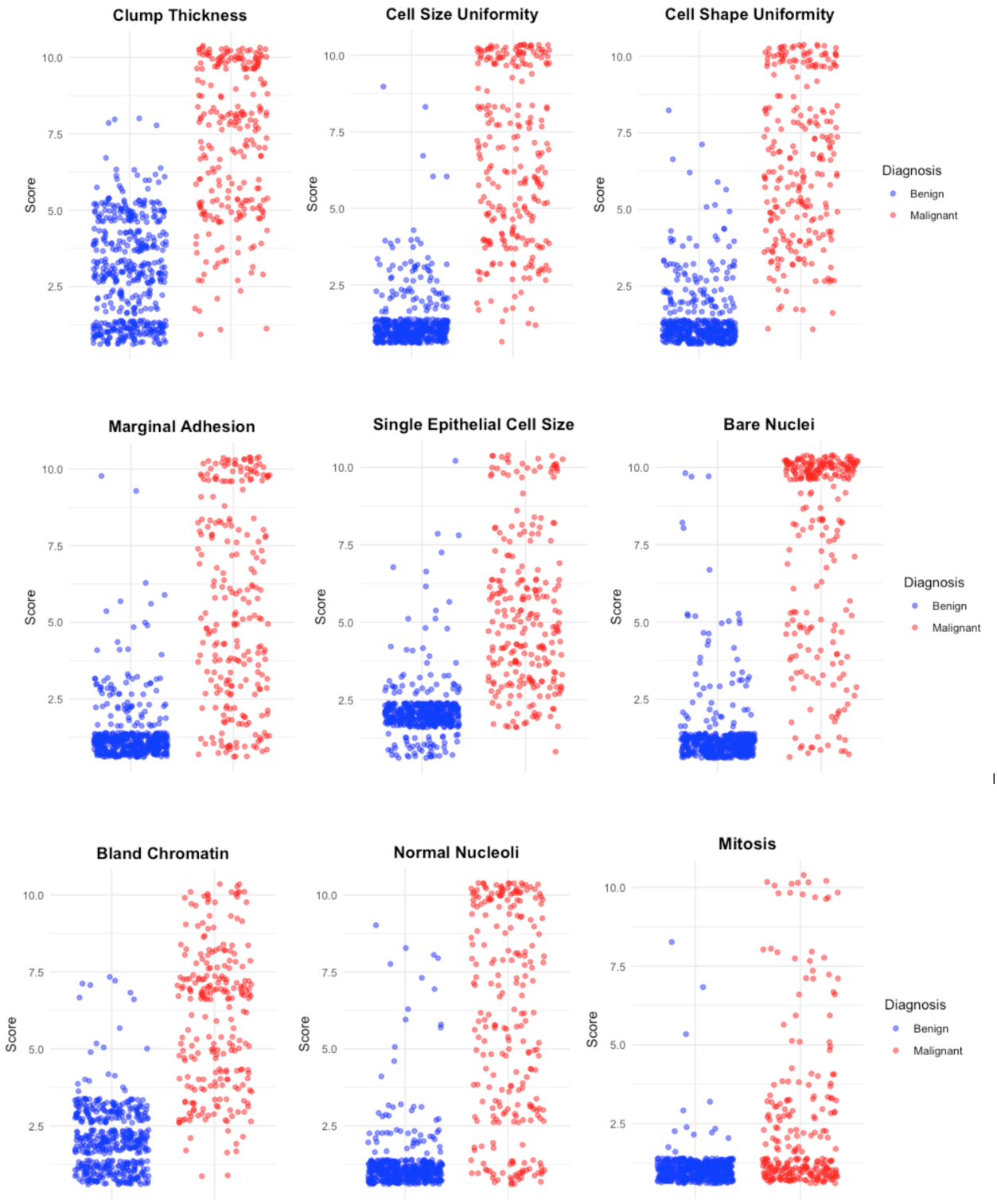

Figure 2. Scatter charts of the nine variables used in analysis.

Classification Tree

The following sections show the results of the breast cancer data analyzed with a classification tree in R, Python, and SPSS.

R

Using all of the predictor variables in the formula for the rpart function, corresponding classification tree is shown in Figure 3.

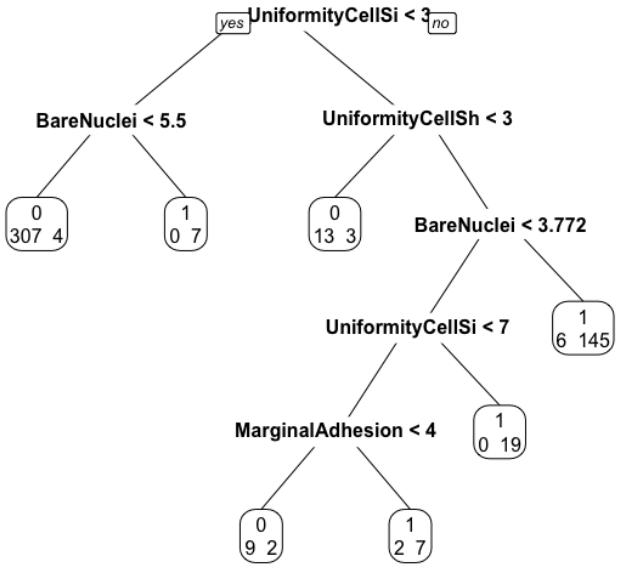

Figure 3. The classification tree uses the predictor variables Uniformity Cell Size, Bare Nuclei, Uniformity Cell Shape, and Marginal Adhesion.

The tree in Figure 3 is modeled on 75% of the original data - our training set. The remaining 25% makes up the testing set. For the training set, we obtain an AUC of 0.979 and a misclassification rate of 3.2%. The ROC curve depicting this AUC is found in Figure 4.

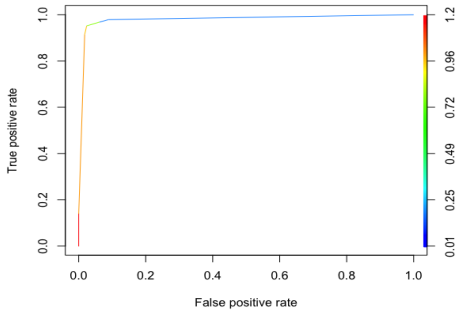

Figure 4. ROC curve for Classification Tree – Training set

Using the classification tree built on the training set, we want to test model performance on the testing set. On the testing set, we obtain an AUC of 0.967 corresponding to the ROC curve shown in Figure 5. The out-of-sample misclassification rate was 6.3%. While this is higher than in-sample, the model still performs very well on the testing set.

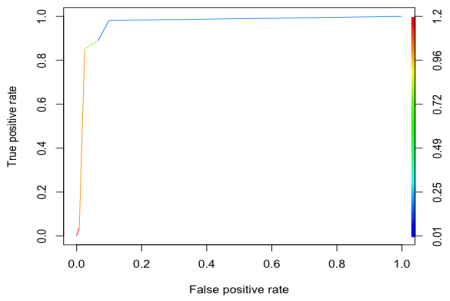

Figure 5. ROC curve for Classification Tree- Testing set

Python

Implementing a classification tree in Python is vastly different than implementing one in R. The library Sklearn was used to create the classification tree. The first test of a running a classification tree without setting any parameters results in a tree that fits perfectly to the training data as no parameters were set to limit the learning of the tree. This first tree is seen below.

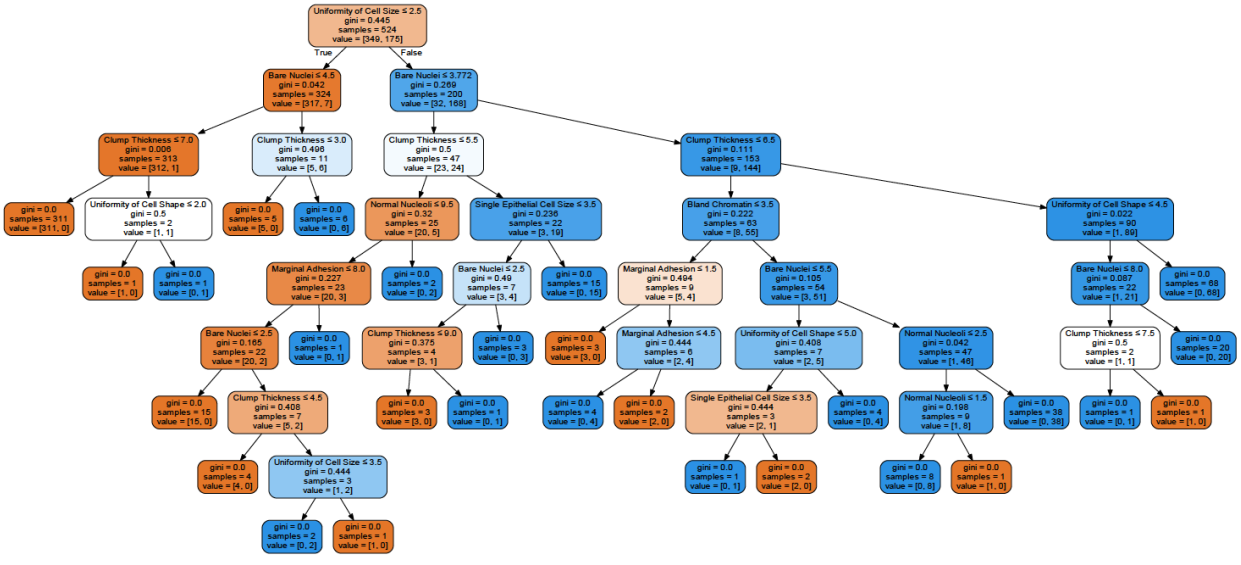

Figure 6. First classification tree in Python.

According the Sklearn website documentation pruning is not currently supported in the library. Therefore, to prune a tree it is a trial and error setting different parameters within the function. The next tree that was created limited the parameters of ‘max_leaf_nodes’ and ‘min_samples_leaf’ to 7 each. Max_leaf_nodes specifies the maximum number of terminal nodes that are allowed to be in the final tree. Min_samples_leaf specifies the minimum number of samples allowable in each leaf node. With these specifications the figure below shows the resulting tree.

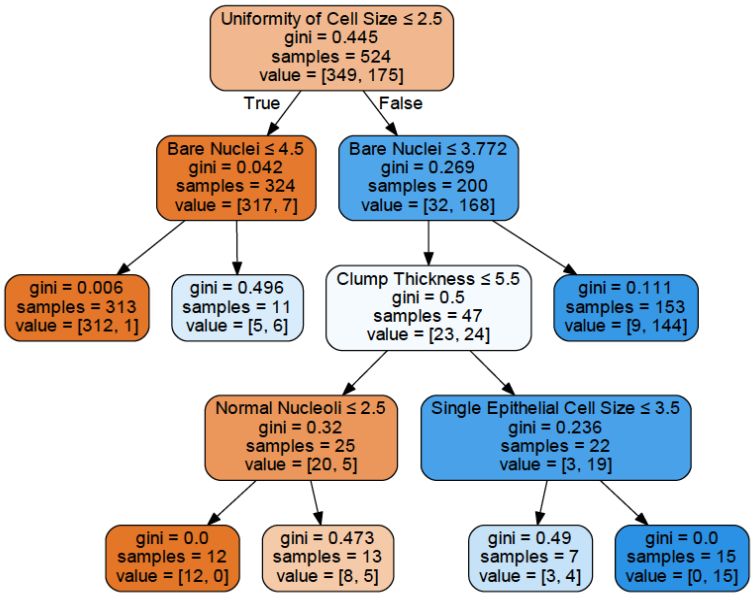

Figure 7. Second tree pruned with parameters of 7 max leafs and 7 minimum samples.

This tree has resulted with an in-sample AUC of 0.98 and a misclassification rate of 4.39%. This AUC corresponds to Figure 8.

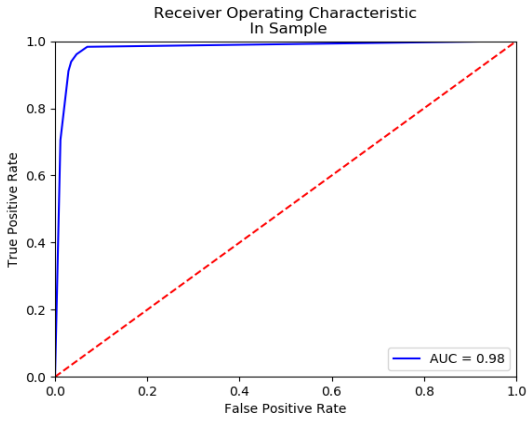

Figure 8. In Sample ROC Curve and AUC for the first pruned tree.

Using this tree on the testing sample of data has resulted in a AUC of 0.98 and a misclassification rate of 3.43% This AUC corresponds to Figure 9.

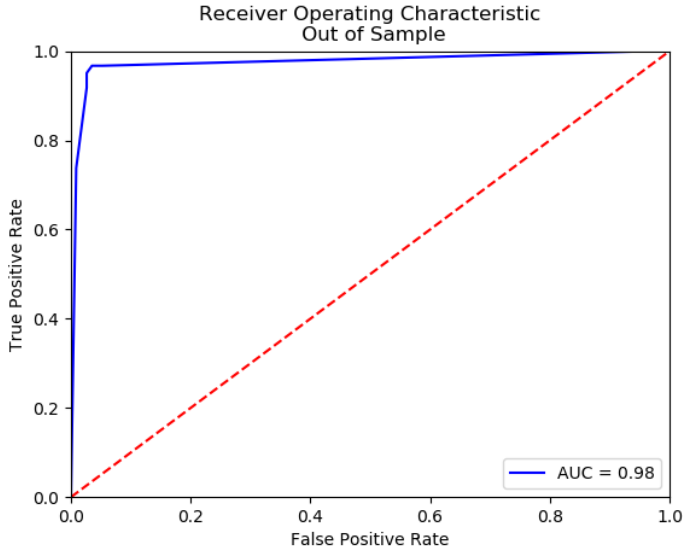

Figure 9. Out of Sample ROC Curve and AUC for the first pruned tree.

A second attempt at pruning the classification tree was performed. This time the parameters that were specified were ‘max_depth’, ‘max_leaf_nodes’, and ‘min_samples_leaf’ with values of 5, 10, and 2 respectively. The resulting tree is as follows in Figure 10 below.

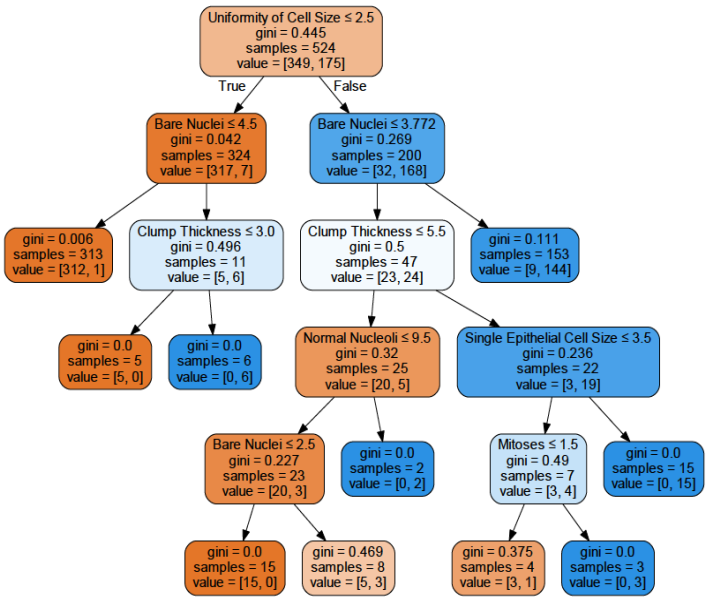

Figure 10. Second pruned tree in Python.

This tree has resulted with an in-sample AUC of 0.99 and a misclassification rate of 2.86%. This AUC corresponds to Figure 11.

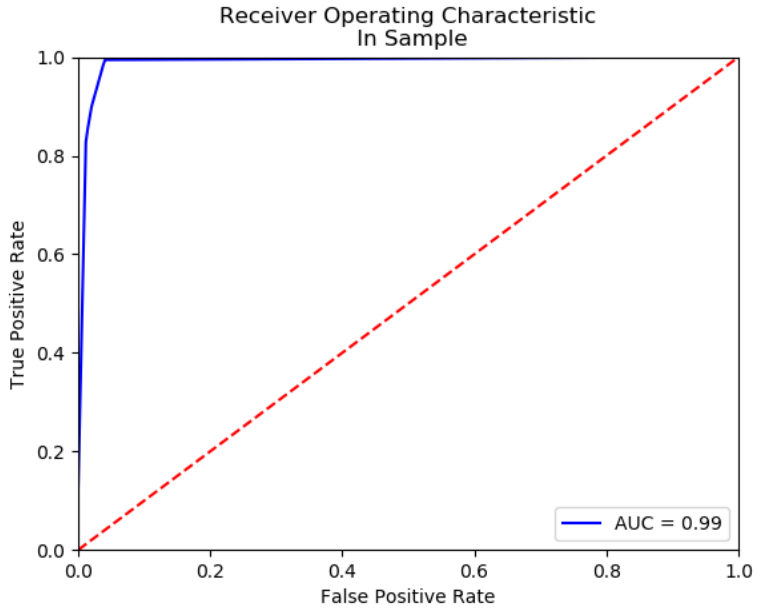

Figure 11. In Sample ROC Curve and AUC for the second pruned tree.

Using this tree on the testing sample of data has resulted in a AUC of 0.97 and a misclassification rate of 4.00% This AUC corresponds to Figure 12.

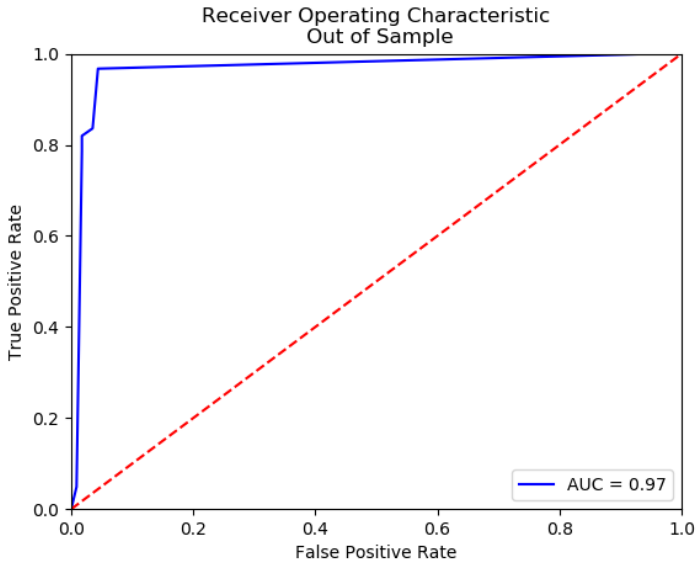

Figure 12. Out of Sample ROC Curve and AUC for the second pruned tree.

SPSS

All nine predictor variables and the one response variable are imported into SPSS and split into 75% training dataset and 25% testing dataset. To model this data with a classification tree in SPSS, the “C&R Tree” modeling node is used. The classification tree is shown below in Figure 13 using the training dataset.

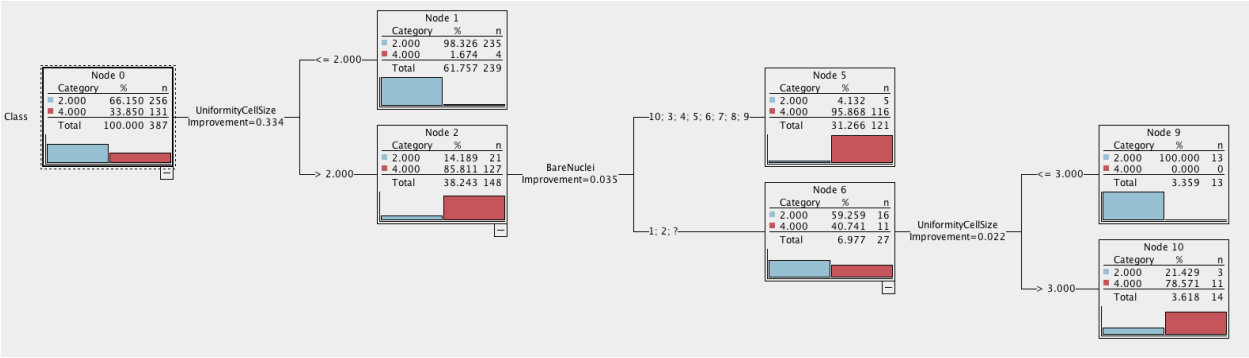

Figure 13. Classification tree in SPSS

The ROC curve for this same training set is shown in Figure 14 as well as the ROC curve for the corresponding testing dataset. The area under the curve for in-sample training set is 0.96 and the out-of-sample is 0.946. The misclassification rates for in-sample and out-of-sample are 3.72% and 6.79% respectively. These are both very good AUC’s and misclassification rates; however they are slightly not as good at predicting as Python or R.

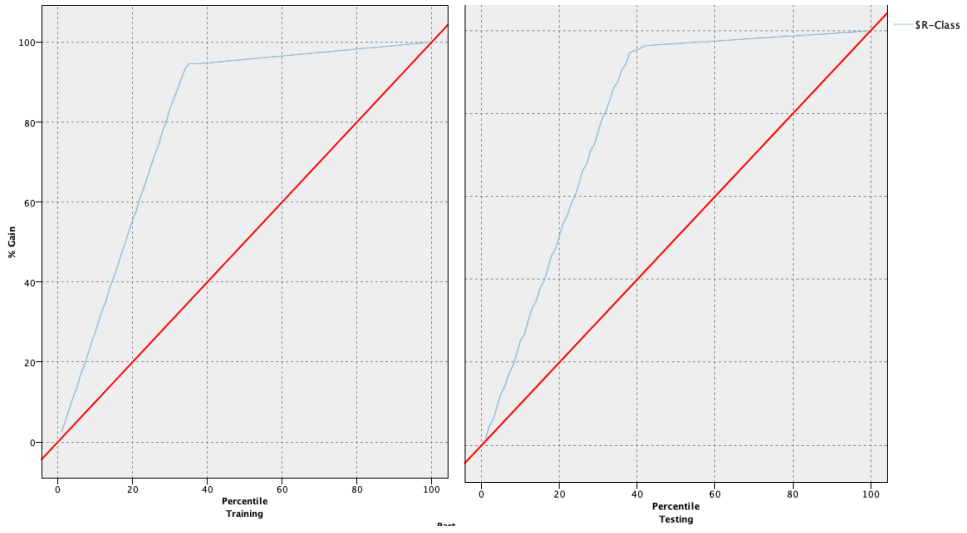

Figure 14. ROC curves for in-sample and out-of-sample datasets respectively in SPSS.

Support Vector Machine

The following sections show the results of the breast cancer data analyzed with support vector machine in R, Python, and SPSS.

R

Using the svm function within the e1071 package in R, we build a support vector machine model on the training set of the breast cancer data. The misclassification rate for this model was 2.3%. The ROC curve for the training set is shown in Figure 15. The AUC corresponding to this ROC curve is 0.976.

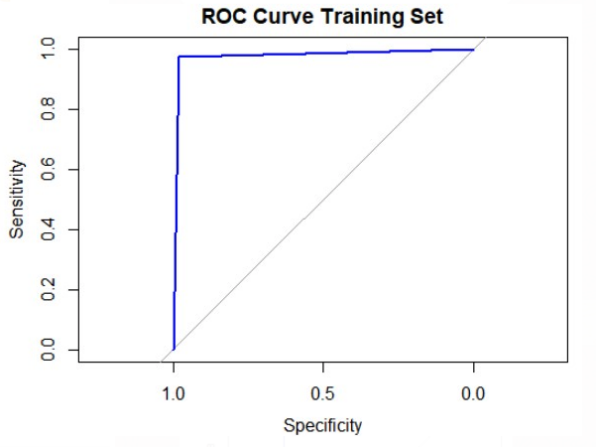

Figure 15. ROC curve for SVM – Training set

While misclassification rates and AUC can give us a good picture of model performance, we also choose to include a gains chart to compare the model's gains as opposed to a random model. This is shown in Figure 16.

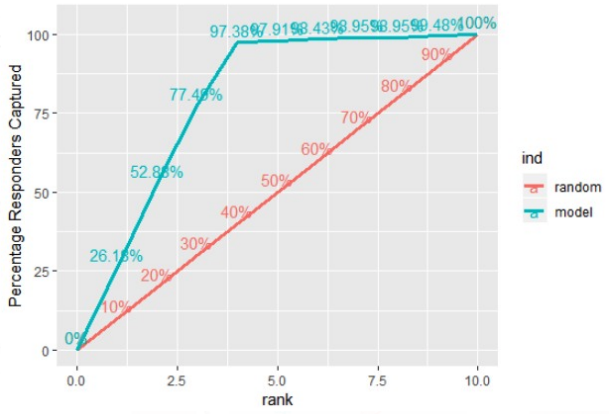

Figure 16. Gains chart for SVM – Training set

From the gains chart, we can see that the SVM model performs much better than a random model as seen in the great disparity between the SVM model curve versus the random model curve.

In order to assess the model's predictive power, we must test the SVM model on the testing data. It performs very well on the testing set with a misclassification rate of 4.6%. The ROC curve for the testing sample is shown in Figure 17 with an AUC of 0.956.

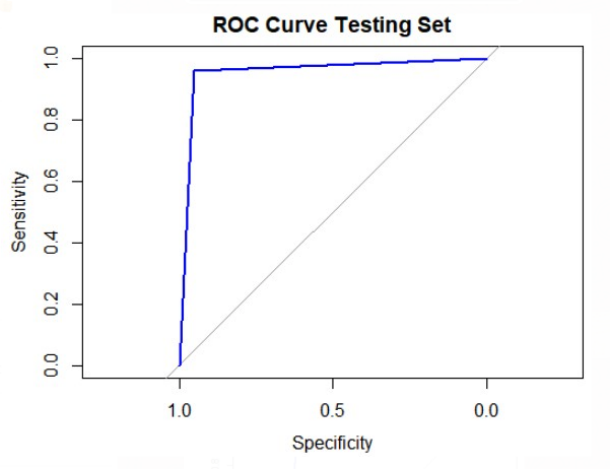

Figure 17. ROC curve for SVM – Testing set

While the testing sample performs worse than the training sample, the misclassification rate is still only 4.6% which demonstrates that the support vector machine model performs well even outside of data that it was built upon.

We also want to assess model performance by creating a gains chart and comparing the model performance to a random model. The gains chart is shown in Figure 18.

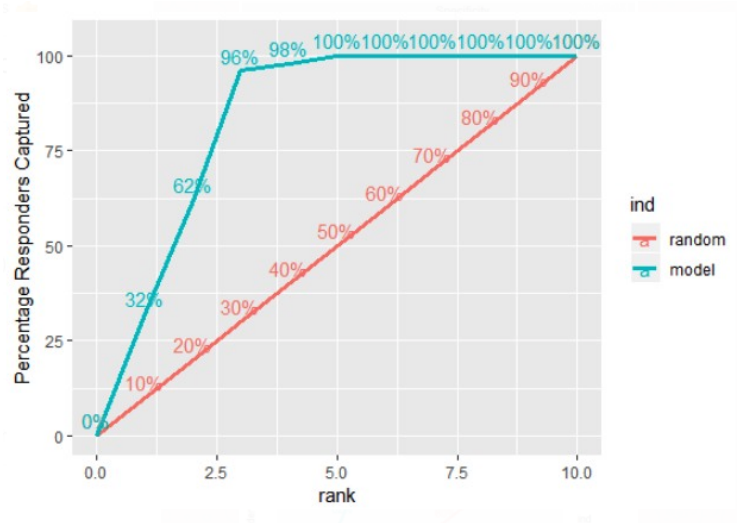

Figure 18. Gains chart for SVM – Training set

From the gains chart, it is quite visible that the SVM model performs much better than a random model. This is shown in the disparity in the area between the model curve and the random curve.

Python

The SVM created in python was created with the SVC (support vector classifier) function within the Sklearn library in python. After specifying the parameters to be the same as the SVM in R the results were still different. It has been noted in other cases that the results differ between R & Python. Valentin Kuznetsov has documented his findings to show that they differ when run with the same parameters.1 The ROC Curve of the SVM created is shown in Figure 19. It has an in-sample AUC of 0.99 and a misclassification rate of 0.38%. The out of sample performance of the SVM model is shown in Figure 20. The out of sample AUC is 1.00 and a misclassification rate of 1.7%.

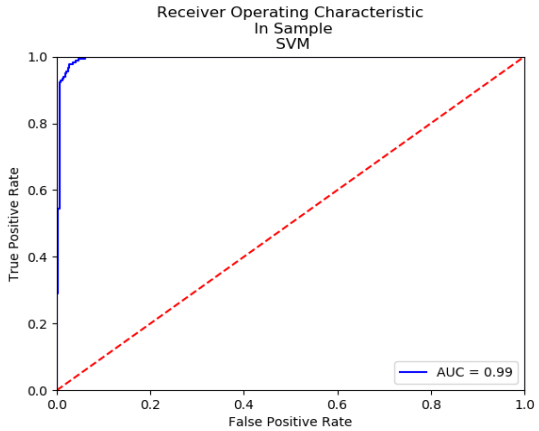

Figure 19. In sample ROC Curve for SVM.

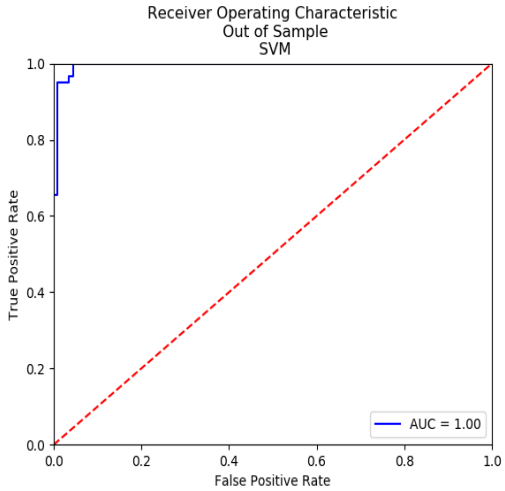

Figure 20. Out of sample ROC Curve for SVM.

SPSS

All nine predictor variables and the one response variable are imported into SPSS and split into 75% training dataset and 25% testing dataset. To model this data with the support vector machine in SPSS, the “SVM” modeling node is used.

The ROC curve for this same training set is shown in Figure 21 as well as the ROC curve for the corresponding testing dataset. The area under the curve for in-sample training set is 1.0 and the out-of-sample is 0.981. The misclassification rates for in-sample and out-of-sample are 0% and 3.7% respectively. These are both very good AUC’s and misclassification rates and they are much better than the classification tree in SPSS at predicting.

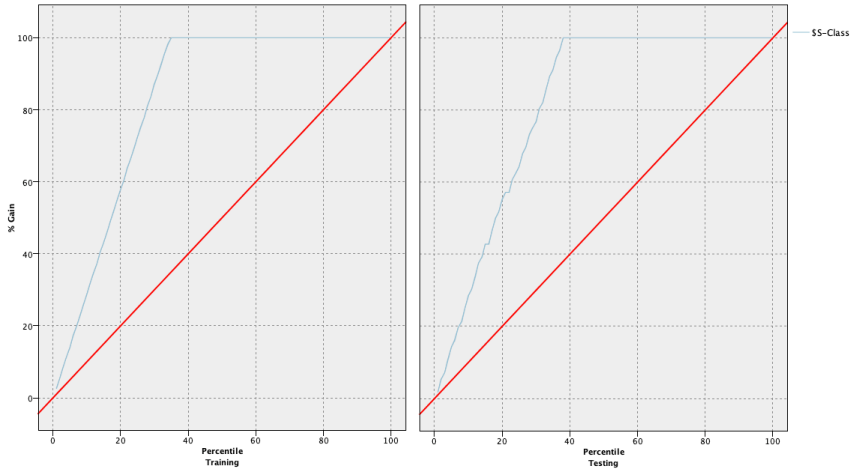

Figure 21. ROC curves for in-sample and out-of-sample datasets respectively for suport vector machine in SPSS.

Conclusions

Looking at one set of data, the Wisconsin Breast Cancer data set, in many different ways allows us to evaluate the performance of different techniques to see which does best at predicting malignancy in tumor cells. We used a classification tree and support vector machine each in three different programs, R, Python, and SPSS. Overall, both methods and all programs are good at predicting whether a tumor cell is benign or malignant. Support vector machines are however, better at predicting overall than a classification tree, and Python is better at predicting compared to R and SPSS. The best prediction in the end is the support vector machine in python with a misclassification rate of 1.7% in the testing set.

Bibliography

"CDC - Basic Information About Breast Cancer." *Centers for Disease Control and Prevention*, Centers for Disease Control and Prevention, www.cdc.gov/cancer/breast/basic_info/index.htm.

"Data Science Projects." *Inertia7,* https://www.inertia7.com/projects/3

Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani (2014). *An Introduction to Statistical Learning with Applications in R*. Springer New York, ISBN-13: 978-1-4614-7138-7

"Support Vector Machines in R". *Datacamp*. Course.
https://www.datacamp.com/courses/support-vector-machines-in-r

"Support Vector Machines in R". *Datacamp.* Tutorial by James Le.
https://www.datacamp.com/community/tutorials/support-vector-machines-r