**Objective:**

The primary focus of this work is to evaluate different techniques in machine learning. The course work is offered on UDEMY (<https://www.udemy.com/course/data-science-and-machine-learning-with-python-hands-on>). Data used in this course is from this course. The techniques are defined below, with a description of how each works in order.

1. Decision Tree
2. Random Forest
3. KNN
4. Naïve Bays
5. Support Vector Classification
6. Neural Network
7. Logistic Regression
8. Linear Probability

Utilizing a UDEMY course for different machine learning techhniques

**Data Section:**

Explanation of the data and the objective for the data.

**Model Method Overview:**

The report has a section for each model listed. Within each section a high level overview of the machine learning techniques are given. In most cases, each of the machine learning techniques have different parameters that can be utilized when fitting the models. These techniques will be documented at a high level for general understanding. In each machine learning classifier a model is fit using 60% of the data which is then used to predict the out of sample (40%). Models which have the best accuracy at predicting the outcome is the objective and reported.

**Model Iterations:**

This section will utilize the work off of the individual machine learning techniques. However, for model fit a method in python is utilized called Grid Search CV (GridSearchCV) which is an exhaustive approach to find the best performing model given parameter tuning. The suffix “CV” stands for cross validation and has a more robust sampling method than the 60/40 split in the Model Method Overview section.

**Conclusion:**

Concluding remarks and application of the modeling methods and comparison to the BI-RADS score which was in use.

**Data:**

For this project, I will use the data provided from the class on UDEMY which covers all of the machine learning techniques. The data used for this project is from the University of California’s Bren School of Information and Computer Science. The data is samples of Mammographic Masses collected from the Institute of Radiology at the University of Erlangen-Nuremberg between 2003 and 2006. There are a total of 961 observations in the data, with 516 observations classified as benign and the remaining being classified as malignant. The primary focus of the data is to understand if it is necessary to perform a biopsy or not. From the metadata in the data file used, it stated that “70% of mammograms lead to unnecessary biopsy.”

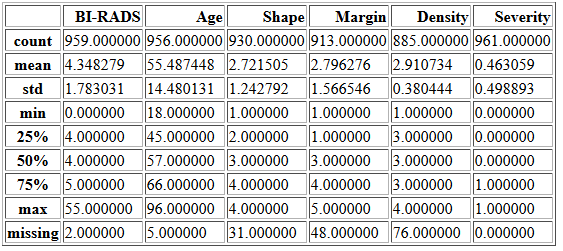
**Data attributes:**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Type** | **Usage** |
| BI-RADS | Ordinal | Non-Predictive |
| Age | Integer | Predictive |
| Shape of Mass | Nominal | Predictive |
| Margin | Nominal | Predictive |
| Density | Ordinal | Predictive |
| Severity | Binary | Outcome |

**Data definitions:**

|  |  |
| --- | --- |
| **Attribute** | **Definition** |
| BI-RADS | Breast Imaging Report and Database System Score. After a mammogram, doctors use this standard system to rate the mass between 1 and 5. 1 indicates benign and 5 highly likely it is malignant. This is assigned via a double review process by physicians. |
| Age | Females age in years |
| Shape | Shape of the mass: Round, Oval, Lobular, Irregular. Represented in data as 1 to 4 |
| Margin | Margin of the mass related to the normal cells surrounding it. Circumscribed (clearly defined mass), Microlobulated (angular, indistinct),Obscured, Ill-Defined (hard to determine the normal cells from mass), Spiculated (normally lines projecting from the mass). Values of each are represented in the data as 1 to 5 |
| Density | Density of the mass, High, ISO, Low and Fat Containing. Represented as 1-4 in the data |
| Severity | The tumor is benign or malignant, 0 or 1 in the data |

**Simple Statistics:**



**Simple Statistics by Severity:**

|  |  |
| --- | --- |
| **BI RADS** | **Age** |
|  |  |
| **Shape** | **Margin** |
|  |  |
| **Density** | **Missing** |
|  |  |

Here, BI RADS is reported, however not used in the modeling exercises. Clearly, there are illogical events as the represented data should be between 1 and 5, however we have values above 5 and below 1. Age is in-line with expectations, being the older you are the more likely it is that you will have malignant tumors. Shape of the tumors seems to be highly related to the malignant tumors, for women malignant tumors 75% of the sample shows the shape being lobular or irregular. The margins around the tumor suggest similar, having obscured, ill-defined or speculated covered 75% of the malignant tumors in the sample. Density observed a value of 3 for the interquartile range for both malignant and benign outcomes.

The data show higher levels of missing values for Margin and Density. It is good to note here that in the UDEMY course, these values were assumed to be randomly distrusted and dropped from the analysis. My work will impute zeros for Age, Shape, Margin and Density for the missing values. Additionally, when imputed observations occur they will be marked in the data for modeling purposes using a dummy variable. 30 observations were dropped, which contained two or more missing values for the predictors.

As stated previously, BI-RADS will not be used in the modeling exercises. However, it is good to note here, that BI-RADS is a variable which was used to assess the tumor for being malignant or benign, with 1 being highly likely benign and 5 being highly likely malignant. The process to assign the score is provided by a physican and goes through a double review process (two physicians agree upon the assigned value). From the simple statistics distributed by Severity, a high majority of the observations were classified as 4 or more in the data.

**Decision Tree Classifier**

For the purposes of this writing, we will focus on Decision Trees using classification. In general, a classification decision tree has an outcome variable which is discrete, this is applicable here as the outcome for our data is binary in nature (Severity=1/0). Decision trees use rules to split the data by the predictor variables (or referred to as features). In specific, the start of a decision tree splits the data two ways using one variable. Example, A, B, C, D are all predictor variables. The start of the tree can use variable A (binary) and create a path given A is 1 or 0. The tree would descend given the other predictors (B, C and D) and can use variable A in later splitting criterion depending on the depth of the tree.

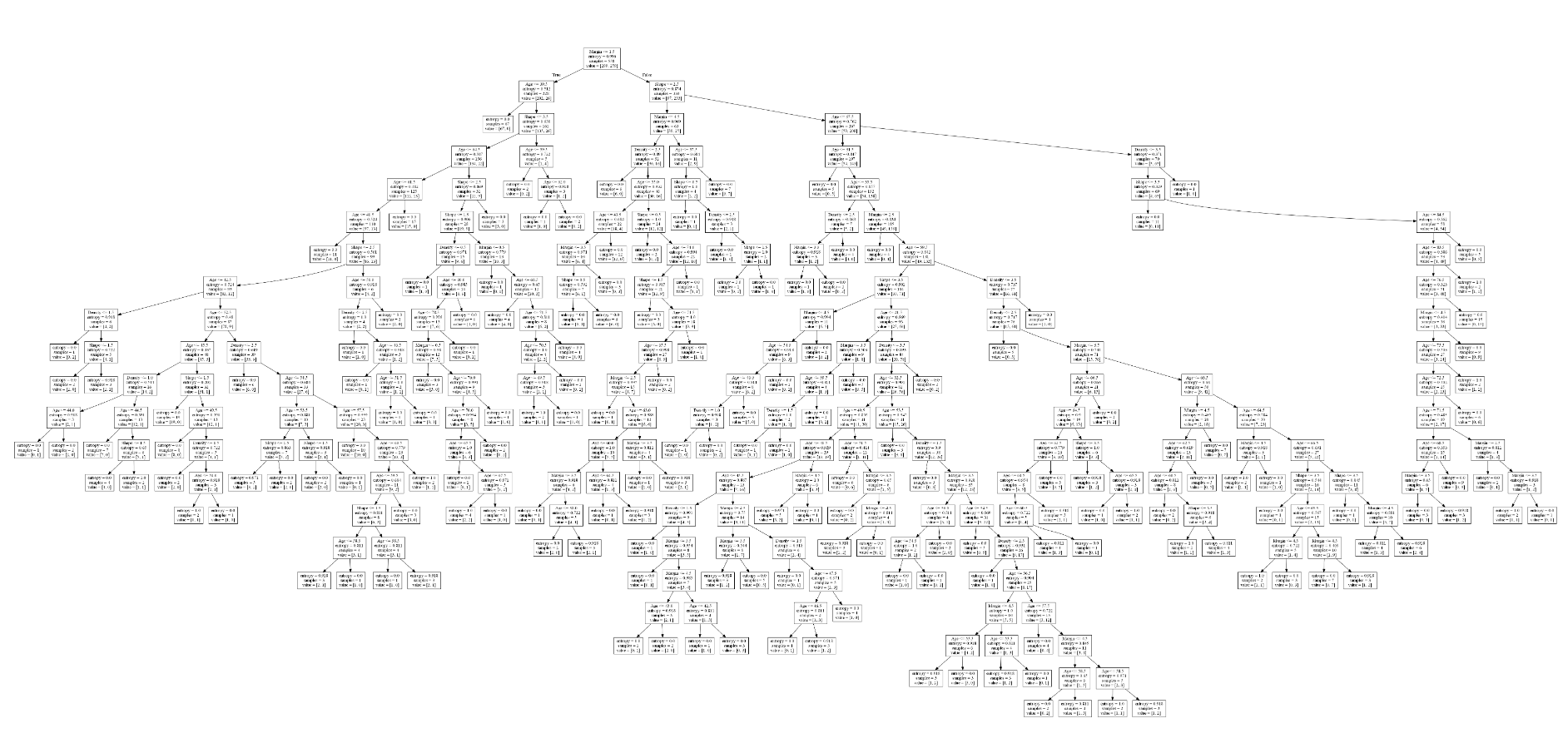
Decision Trees use the splitting method above to create what Leaves and Nodes. It is well known that Decision Trees will be overly complex (large nodes and leaf structure) which makes it subject to over fitting the data. In most cases, these extremely large trees will have very small populations which are used at the node to create another leaf. There is an automated way to prune the trees that uses what is known as a “cost complexity parameter.” This parameter is calculated using the number of terminal nodes and the misclassification rate of the terminal node. This method is known as the Minimal Cost-Complexity Pruning.

|  |  |
| --- | --- |
|  |  |
|  | Minimal Cost Complexity Function |

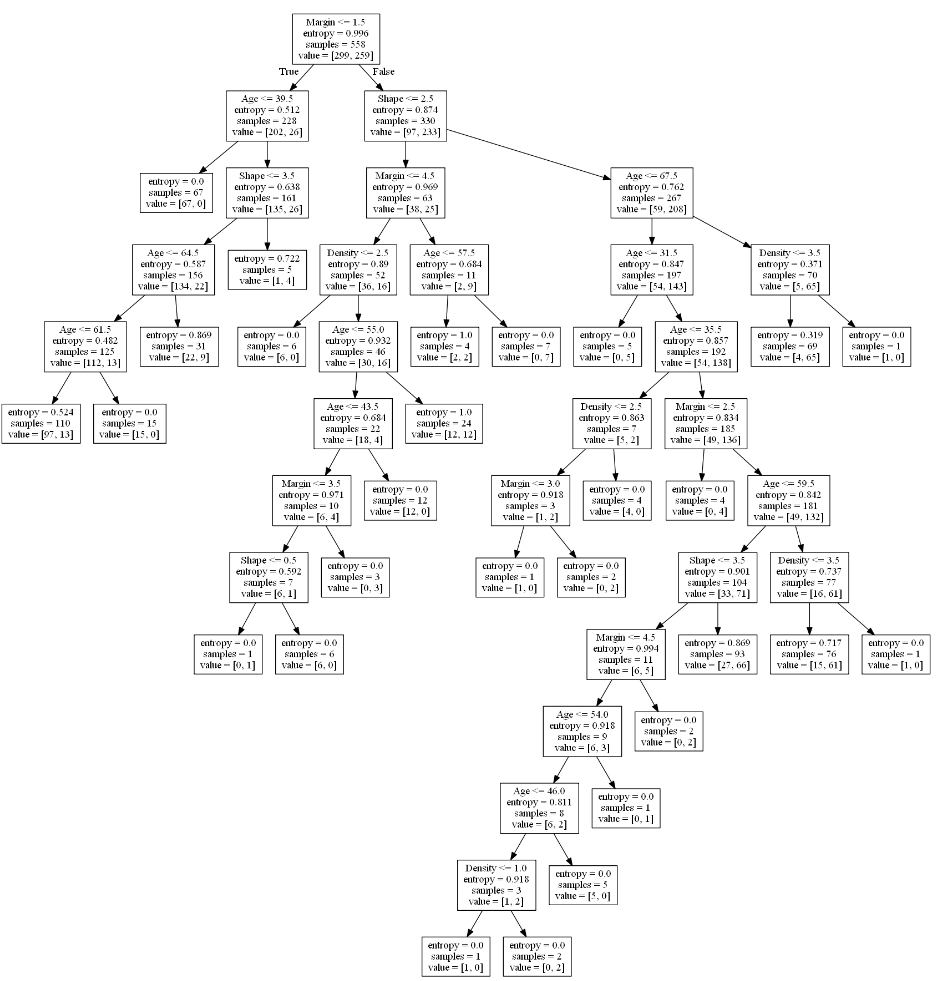
Where is the tree and is the number of terminal nodes in the tree, is the misclassification rate of the terminal nodes, is a single node. And is a measure of impurity of misclassification of the terminal nodes and its child nodes. is the pruning criterion, values of will cause the leaf node to be pruned.

Pictured, is the initial tree:

**Figure 1**, Decision Tree without prunning: (Entropy\_Tree.png)

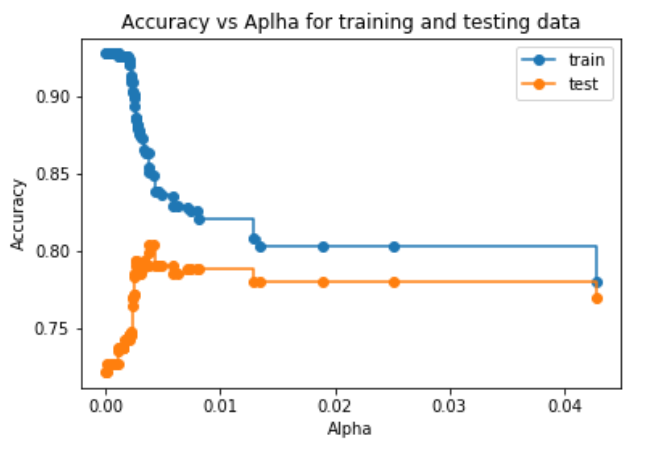


**Figure 2**, Decision Tree with pruning (Minimal Cost Complexity), (Entropy\_Tree\_Pruned.png)

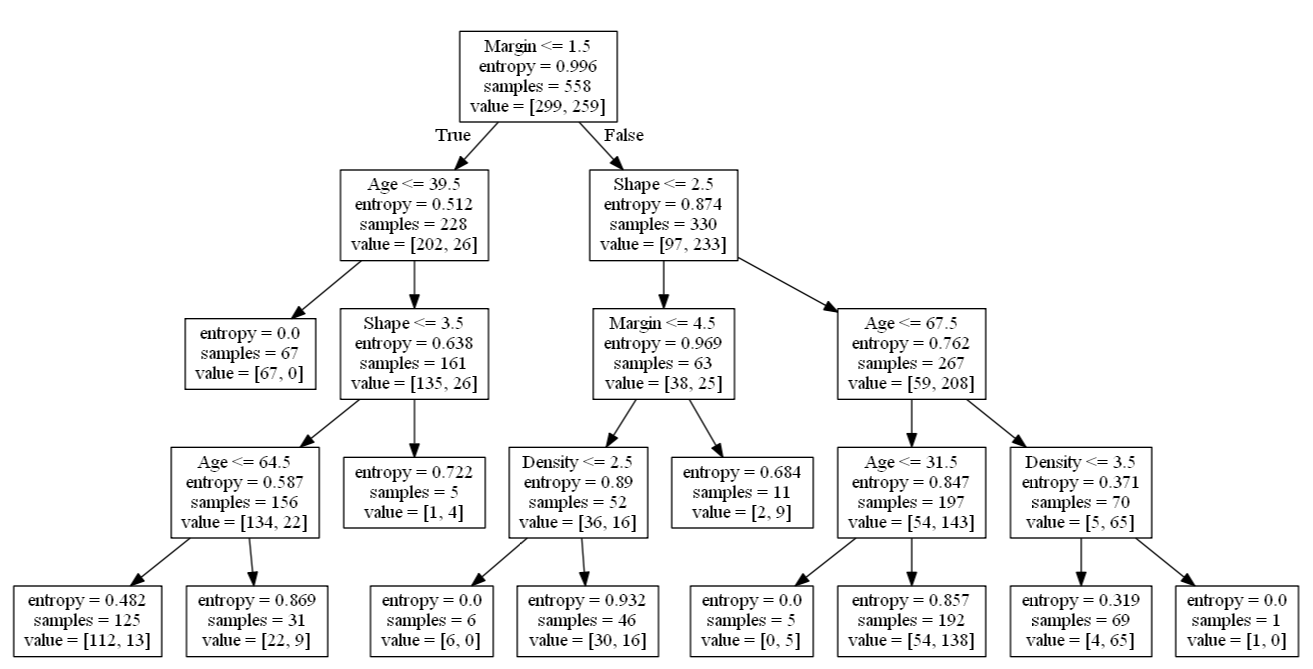


Clearly, the pruning algorithm reduced the number of nodes and the complexity of the tree. However, the nodes have very small sample sizes still. For benchmark purposes, a manually set parameter tree was generated which we can use to compare the accuracy of the non-pruned Tree and the pruned tree

Graphically, we can visualize the cost of decision trees with overfitting. As stated earlier, the decision tree model is highly susceptible to overfitting, normally by including too many leaves and nodes in the fitted tree. The below graph shows the minimal cost complexity function for the fitted model (train) and the out of sample (test) populations. For “Alpha” level 0 the fitted model had the best accuracy, however it performed the worst at predicting the out of sample population. As the “Alpha” level increases (tree pruning) the out of sample accuracy gets better and the model fit accuracy gets worse.



**Figure 3**, Manually Pruned Tree (Manual\_Tree.png)



For this tree, I manually specified the depth of the tree to be no greater than 4, additionally the minimum sample for a node to be split (number of observations) was 15.

Results of the three methods of the decision tree can be evaluated by the misclassification rate and the variable importance.

Table 1 depicts the accuracy of the three trees:

|  |  |
| --- | --- |
| **Decision Tree Method** | **Accuracy** |
| Normal Tree (no-pruning) | 72.1% |
| Pruned Tree | 80.4% |
| Manually Specified Tree | 78.5% |

Variable importance is the sum of the number of classification values in the variable and the ratio of the splits. This is known as the “Gini Impurity”

|  |  |
| --- | --- |
|  | (3) |
|  | (4) |

Equation (3) represents the Gini Impurity () for a node and the percent of its distribution (). From equation (3) we can calculate the feature importance, (4). Which is the Gini Impurity of the parent node less the sum of the Gini Impurity of the distribution of the child nodes denoted by .

With feature importance of:

|  |  |  |  |
| --- | --- | --- | --- |
| **Variable** | **Normal Tree** | **Pruned Tree** | **Manual Tree** |
| Age | 35.4% | 20.6% | 13.7% |
| Density | 7.4% | 5.6% | 3.2% |
| Margin | 44.2% | 60.5% | 69.3% |
| Shape | 13.1% | 13.3% | 13.8% |

From the three methods, the greatest accuracy comes from the pruned tree with second being the manually specified tree which was approximately 2% more error. Variable importance was similar in the pruned tree and manually specified, with Margin being the largest importance factor. Given the depth of the Pruned Tree which can be related to its complexity, the manually specified tree has desired attributes of larger split sizes on the node and is easier to interpret.

**Random Forest Classifier**

Random Forests are another means of machine learning which utilize the basis, the Decision Tree. However, a Random Forest is a collection of Decision Trees. This collection of trees greatly assists with over fitting and selection bias which occurs from a single Decision Tree. The reason being, is that a Random Forest uses multiple trees and utilizes a “committee” for the final response.

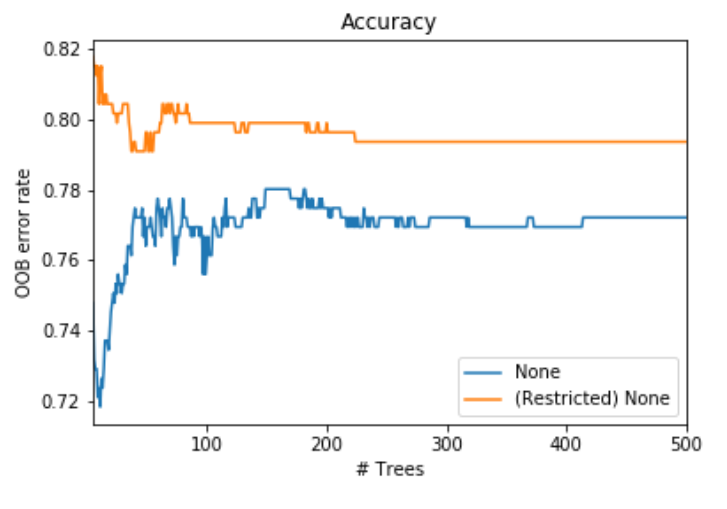
An example, say we have a binary outcome which is trying to be predicted, “Yes” and “No.” Using a single Decision Tree, one would cast one vote. Only having a single result will result in higher probability of bias and error which was outlined in the Decision Tree section. With the Random Forest, say for an example we create 10 trees we would then be able to cast 10 votes on the outcome. Even if one of the votes are incorrect in the Random Forest, the consensus should be able to overcome and present the correct classification.

Like the Decision Tree, the Random Forest has automated pruning techniques which is used to control error, over fitting and variance.

|  |  |
| --- | --- |
|  | is the total number of trees. is a vector of predictors and outcomes for binary case, and is an out of bag sample of the population. is the average vote of the Random forest |
|  | Standard deviation of the Random Forest |

The above equations represent the Bagging or Out of Bag sampling mechanism which can be used for the Random Forest specification. Simply put, if the total population is represented as a vector of predictors, and has a binary outcome, the Out of Bag sampling uses a random sample (with replacement) to build the forest. Using a sample to build many trees (in this case 500) helps to reduce the variance in the data, which is a known limitation of the Random Forest regressor, the restriction is that the sample draws when the forest is fit is uncorrelated to other draws.

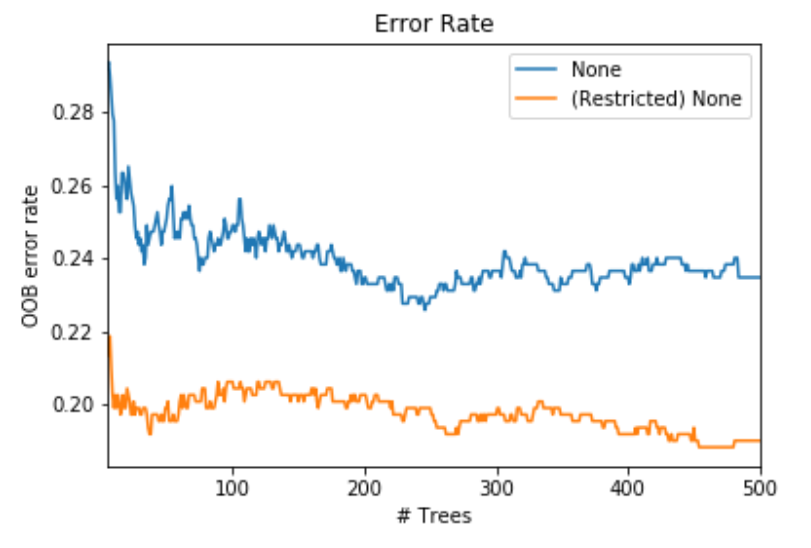
The following figures represent the Out of Bag accuracy and error for the Random Forest.



Like the Decision Tree, the accuracy for the out of sample population was measured using two Random Forests, first having no restrictions (None) and one which is restricted (Redistricted None). The restrictions here is to control the depth, no restrictions will allow the tree to have an unspecified depth, versus the restricted which is capped at four, additionally ensuring that the sample size for a leaf node split is more 15 or more.

Using the restriction, the accuracy is better with fewer trees feeding the Random Forest, then gradually decreases as more trees are included. Likewise, having no restrictions the accuracy increases to a certain point given the inclusion of more trees in the Random Forest. This is intuitive due to over fitting and as more trees are included in the non-restricted the average of the voting improves with additional trees.

For ease of interpretation, the Error Rate is shown below.



The below table represents the information which had the best accuracy along with the average accuracy of the Random Forest. In this table, a manually specified tree that does not include Out of Bag sampling was ran. The tree included 10 estimators using the max depth of the forest at 4 and minimum sample for a node split of 15.

|  |  |  |  |
| --- | --- | --- | --- |
| **Random Forest Method** | **Best Accuracy** | **Best (# of Trees)** | **Average Accuracy** |
| Out of Bag No Restrictions | 78.0% | 149 | 77.0% |
| Out of Bag Restricted | 81.8% | 5 | 80.0% |
| Manually Specified (No Out of Bag) | 75.6% | 10 | N/A |

With the following for variable importance. The variable importance of the Out of Bag methods is the average for all trees.

|  |  |  |  |
| --- | --- | --- | --- |
| **Variable** | **Out of Bag, No Restrictions** | **Out of Bag, Restricted** | **Manual Forest** |
| Age | 39.7% | 19.7% | 10% |
| Density | 18.9% | 26.0% | 0% |
| Margin | 35.4% | 51.2% | 20% |
| Shape | 6.1% | 3.1% | 70% |

**Nearest Neighbors**

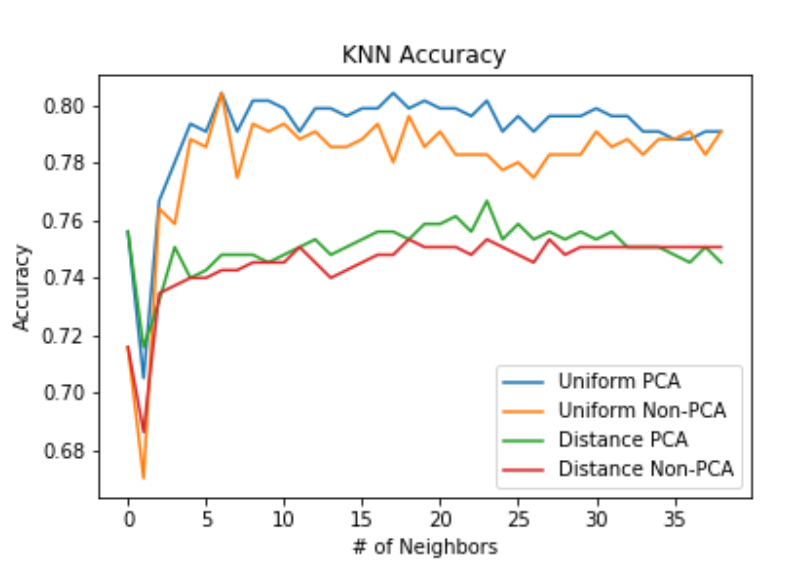
Nearest Neighbors uses a distance based modeling approach modeling method. In general, if the data is plotted a K Nearest Neighbor will group similar data points into bins, theory is that the grouped data is the same or representative. The number of data points considered to be in a group is used as a “hyper parameter” in the modeling exercise. This specific hyper parameters, called *k* which controls the number of data points used in the group. The distance between the points uses the Euclidean distance.

|  |  |
| --- | --- |
|  | Where and are two points in the Euclidean space. |
|  | Where is the current sample, is the mean of the sample and is the standard deviation |

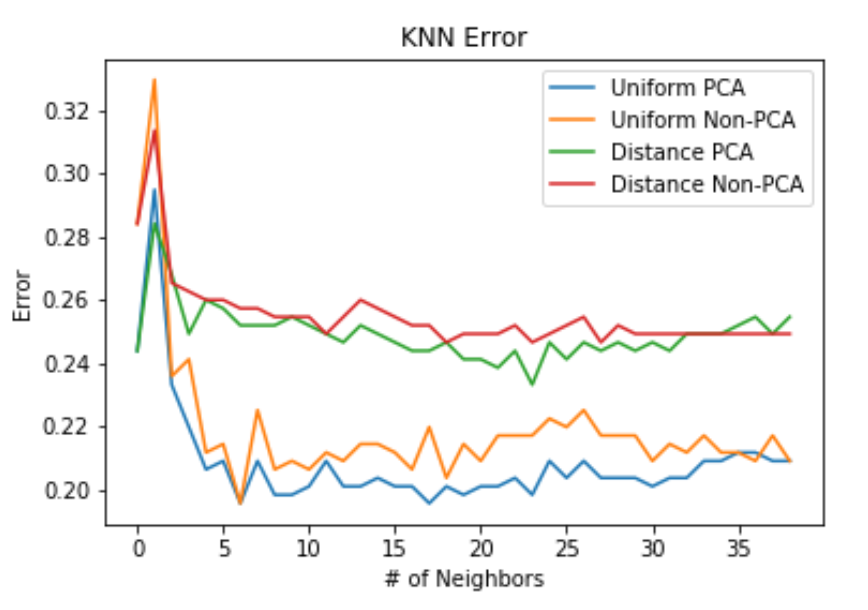
Given data that is discrete, such as what I am working with, the data needs to be centered on the mean in order to utilize the Euclidean distance, as it works better with a continuous series.

The dataset used has four dimensions (Age, Shape, Margin and Density) which is used in the modeling technique. The results provided for this modeling approach will be reduced using Principal Component Analysis (PCA). Which, PCA, in simple terms, uses an orthogonal transformation to reduce the number of dimensions. The reduction, or PCA dimension groups the data into a lower dimension which explains as much of the variance as possible.

In practice, it is often best to evaluate the hyper parameters, in this case, *k* and the Euclidean distance. Additionally, a PCA was used for both these hyper parameters. The change in the Euclidean distance takes two forms, “uniform” and “weighted.” Where the weighted applies a weight to the neighbors, closer neighbors would indicate that they are more in common and weighted higher. Uniform assumes that all groups have equal weights.

The accuracy for the out of sample population shows that the Uniform measure or equal weighting out performs the weighted distances. PCA performs marginally better, likely, PCA would perform better given we have more predictors, such as 10.

For the purposes of analysis, the error rates are plotted as well for the population.

Reported, is the *k* which produced the best accuracy:

|  |  |  |
| --- | --- | --- |
| **K Nearest Neighbors** | **Best Accuracy** | ***k*** |
| Distance (Weighted) Non-PCA | 75.3% | 19 |
| Distance (Weighted) PCA | 76.7% | 24 |
| Distance (Uniform) Non-PCA | 80.4% | 7 |
| Distance (Uniform) PCA | 80.4% | 7 |
| Manual (k=15, Uniform, Non-PCA) | 78.6% | 15 |

There is no specific module which calculates the feature importance of the predictors for the KNN model.

**Naive Bayes**

Naive Bayes classifier utilizes Bayes theorem to calculate the probability of an outcome given a vector of predictor variables. The primary classification is below, with the conditional independence assumption.

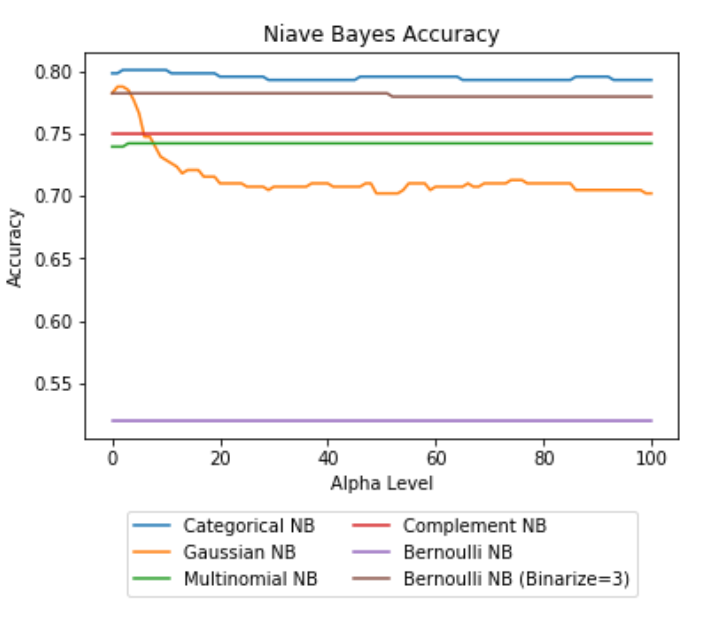
|  |  |
| --- | --- |
|  | Bayes Theorem |
|  | Conditional Independence Assumption |

Here, the conditional independence assumes that any predictor is conditionally independent of any other predictor, . Or, in simple terms, given we have two predictors to measure a single outcome, the first predictor provides no information on the second predictor would provide no information about the first for the outcome.

The distribution of has to be handled with care. In simple terms, if there are four different classifiers to handle the various attributes that can take. Each different distributions changes the likelihood function for maximization.

1. Gaussian Distribution, assumes that is continuous. (🡨 not the case for our data)
2. Multinomial Distribution, which is an extension of a binomial distribution (or discrete probability distribution). Decently suited for our data.
3. Categorical Distribution, used when takes discrete values (just like the data used in this analysis)
4. Complement Distribution, uses Multinomial Distribution, however designed for imbalanced data. Not really how the data used for this analysis is presented.
5. Bernoulli Distribution, assumes that is binary in nature. Does not fit our data. However, for the purposes of testing these methods I included this and will discuss later in the outcome of the results for the classifiers.

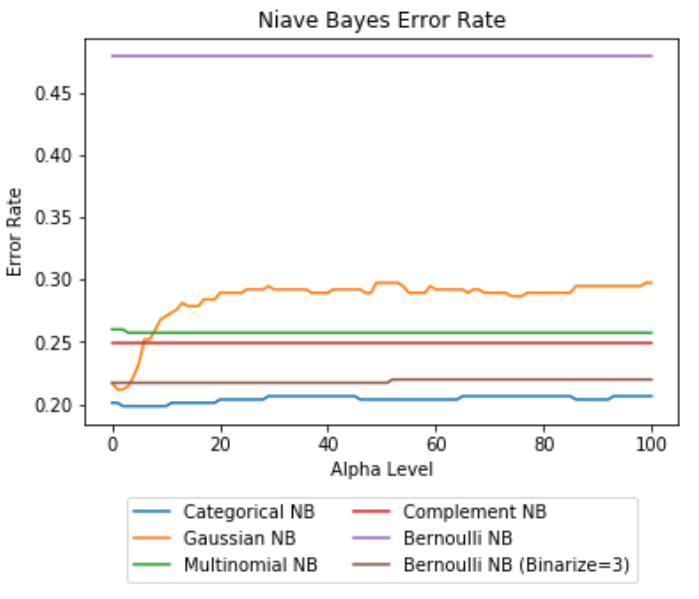
For the Naïve Bayes classifier, 0 probabilities in Bayes Theorem will raise issues. In order to handle this, there is a setting which uses Laplace smoothing for these events () or Lidstone (). Changing the has the ability to produce different results for each of the Distributions listed. The Accuracy is as follows:



The level of alpha has almost no effect for the Complement, Multinomial, Bernoulli Distributions. Suggesting that zero probability was most likely not encountered in the distributions. Strikingly enough, the Gaussian Distribution (for continuous variables, which is not present in our data) was only outperformed by the Categorical distribution in earlier levels of the alpha parameter. In higher levels of the parameter the accuracy deteriorates and does so drastically.

Lastly, given the Bernoulli Distribution is designed for binary predictors, there is a method “binarize” which can be used to automatically convert the discrete values in our data to binary points. For the purposes of the Naïve Bayes classifier, I manually set this to “3” to test the difference between no automatic binning of the discrete predictors (represented as Bernoulli NB in the plot) and comparing it to the automatic binning (Bernoulli NB (Binarize=3) clearly suggests that the data should be binned as the accuracy in the out of sample population increases from approximately 52% to 78%.

With corresponding Errors:



As with the other techniques, the classifier which produces the best accuracy is reported:

|  |  |  |
| --- | --- | --- |
| **Naïve Bayes** | **Best Accuracy** | **Alpha Level** |
| Categorical NB | 80.2% | 0.02 |
| Gaussian NB | 78.8% | 0.01 |
| Multinomial NB | 74.3% | 0.03 |
| Complement NB | 75.1% | 1e-10 |
| Bernoulli NB | 52.0% | 1e-10 |
| Bernoulli NB (Binarize=3) | 78.3% | 1e-10 |

**Support Vector Classification**

Support Vector Machine learning is best understood graphically. In simplest terms, this classifier plots data, then creates lines between the data points which maximizes (called support vectors) the differences between the data points. This classifier has kernels, like the others which can change the requested shape of the lines which are being maximized. For this work, three different kernels were used: linear, polynomial and Radial Basis Function. All SVM models have a tuning parameter “C” which can be utilized to control mis-classifications in the model. Higher values of “C” will allow for a higher margin of error in the model versus a lower value of “C” which allows for less margin of error. “C” controls the decision function, which is plotted below in the examples section.

Kernel functions defined mathematically:

|  |  |
| --- | --- |
|  | Linear, is a matrix |
|  | Polynomial, is degree (parameter to pass) and is a initial coefficient passed, or automatically calculated if not provided. |
|  | RBF. is restricted, must be greater than 0 |

For the purposes of simple explanation, the data was modeled using an SVC with both, linear and RBF kernels to show how this works. So that we can think in simplest terms, only two variables were utilized for this purpose, “Margin” and “Shape”, so that we can keep the plane in two dimensional, instead of four dimensions as the actual model was performed. The two variables used are plotted, Y axis is shape, X axis is margin.

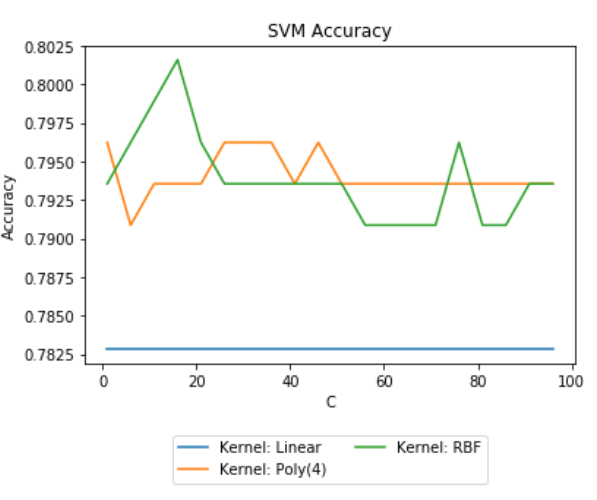
|  |  |
| --- | --- |
| **Linear Kernel** | **RBF Kernel** |
|  |  |

As implied, the linear kernel creates straight lines and carves the data. The shaded orange section identifies what the model would classify as severe which would need to be biopsied (severity=1). The blue shaded regions, the model classifies these as non-severe (severity=0). The dotted lines represent the significance level.

For the purposes of this explanation, you can clearly see that the RBF (polynomial 3rd degree) can better fit the data, as it uses curvature to bend around higher values of “shape” and ensures that higher values of “margin” are being captured. Recall from the simple statistics and other modeling techniques, “margin” is the most powerful predictor for severity.

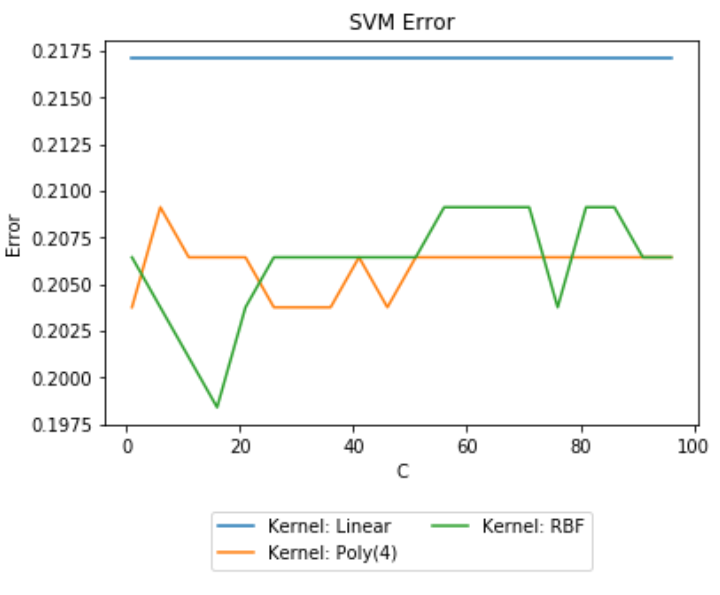
Often, when working with numeric data there is strong evidence that shows the model works best if the data is scaled, or mean centered. Given the characteristics of our data, being discrete this method is not shown, as it just adjusts data points by subtracting/adding its mean. Roughly speaking, mean centering for the purpose of this data would generate the same exact plots, just adjusting both axis’s.

The performance of the model was measured against three different kernels (linear, RBF and Polynomial 4th degree). For each of the models the “C” parameter was adjusted and took a range between 1 and 100, incrementing by 5. The plots below represent the accuracy on the held out population, or the 40% of the population which was not used to fit the SVM model.



Accuracy for out of sample population using the three kernels shows that RBF produced marginally better results than the Polynomial 4th degree. However, it did come at a cost of a higher value of “C” which allows for the model to accept higher rates of mis-classification. The linear kernel for this exercise was not sensitive to the “C” modifier.

Like the other modeling techniques, the Error is plotted as well:

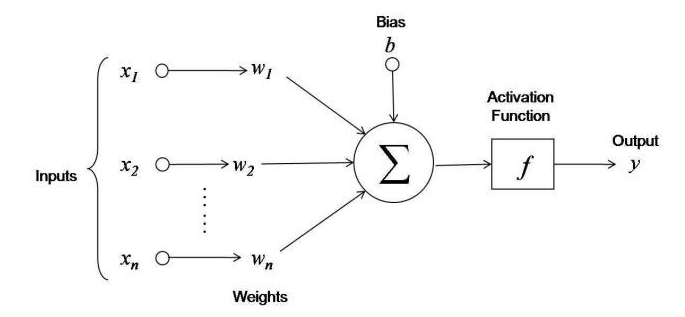


With the corresponding actual values table below:

|  |  |  |
| --- | --- | --- |
| **SVM Classifier** | **Best Accuracy** | **C Value** |
| Linear SVM | 78.3% | 1 |
| Polynomial (4th degree) | 79.6% | 1 |
| RBF (3rd degree) | 80.2% | 16 |

**Neural Network**

Neural Networks can be thought of as logistic regressions with enhanced features. Where, enhanced features have added layers/dimensions that the model creates. These enhanced features are called Multi-layer Perceptrons (MLP). Pictured is a graph that explains how the MLP works:



<https://towardsdatascience.com/first-neural-network-for-beginners-explained-with-code-4cfd37e06eaf>

The features, represented by are the actual values in the data. These features are then transformed to a weighted sum of the original features, . During model execution, the weights are set to some arbitrary number, then get adjusted to fine tune an objective. Here, the objective is to maximize accuracy. It is apparent that if the weights are set to larger numbers, the model inputs, have to be modified heavily, versus low weight values which would indicate that the original model inputs did not need to be adjusted. The weighted sum of the features are then evaluated for the misclassification of the prediction (the Bias, ). The model will iterate until the amount of bias is minimized.

In order to minimize the bias, the model uses different activation functions given the attributes of the objective, continuous, binary or discrete.

Table of activation functions for one hidden layer:

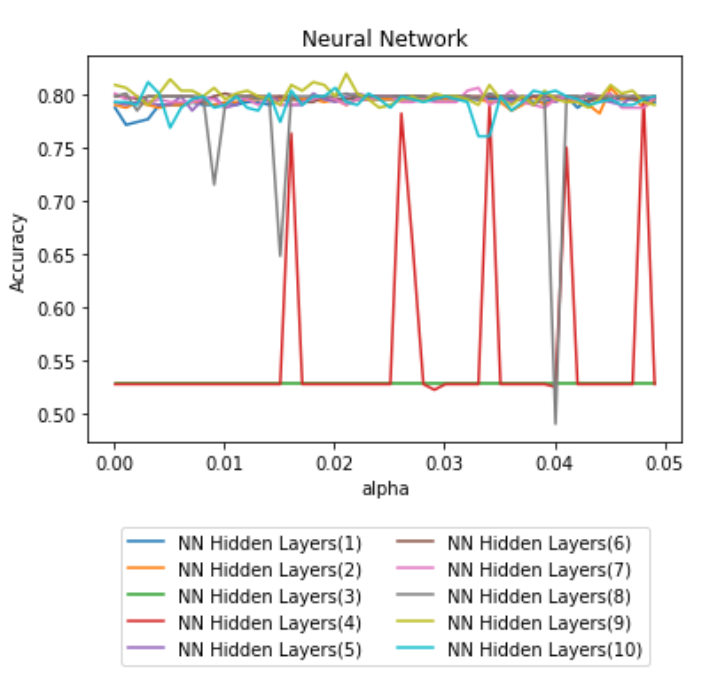
|  |  |
| --- | --- |
|  | Continuous Outcome |
|  | Binary Outcome (Logistic Function) |
|  | Softmax (discrete outcome). represents a distinct outcome, is the number of outcomes |

The activation functions of the hidden layer are then passed to a loss function which is used to evaluate the overall accuracy of the model. Like the activation function, this iterates until there is minimal change in the loss functions is found.

For continuous and binary outcomes the loss functions are as follows:

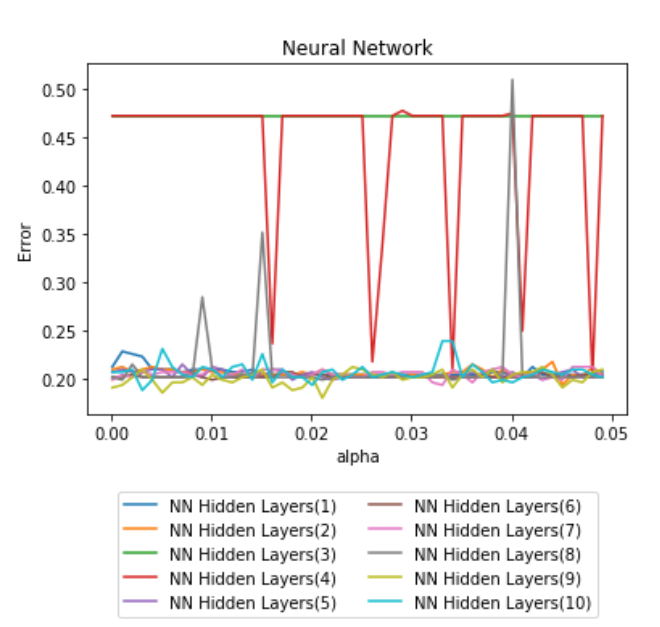
|  |  |
| --- | --- |
|  | Binary Outcome, represents the weights from the hidden layer in the network. penalizes the complexity of the model |
|  | Continuous Outcome, represents the weights from the hidden layer in the network. penalizes the complexity of the model |

For the purpose of this analysis, the optimization function for the matrix of weights is set to “lbfgs” which is an optimizer of the quasi-Netwon family. The number of hidden layers the model will be using range between 1 and 10 and the alpha (penalty in the loss function) parameter is incremented from 0.0001 to 0.05 by 0.001. As usual, the model was fit on the training data, then scored using the out of sample population (40% of the overall data).



The neural network performed the worst using 3 hidden layers, which the model predicted correctly on the out of sample population less than 55% of the time. Potentially, this is due to the model not converging which is a reasonable assumption given the flat line. When 4 hidden layers was used the model observed odd spikes in the accuracy, the rational for this is likely that the model did not converge on other levels of alpha. Using 9 hidden layers in the neural network seemed to produce the best accuracy.

Accompanying Error Rate:



The below table reports the level of alpha that produced the highest accuracy for each of the neural networks.

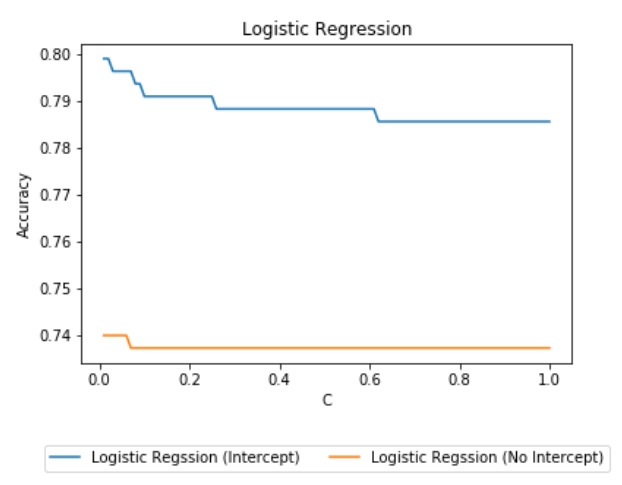
|  |  |  |
| --- | --- | --- |
| **Neural Network** | **Best Accuracy** | **Alpha** |
| Hidden Layers (1) | 79.9% | 0.0411 |
| Hidden Layers (2) | 80.7% | 0.0451 |
| Hidden Layers (3) | 52.8% | 0.0001 |
| Hidden Layers (4) | 79.9% | 0.0481 |
| Hidden Layers (5) | 80.2% | 0.0181 |
| Hidden Layers (6) | 80.2% | 0.0101 |
| Hidden Layers (7) | 80.7% | 0.0331 |
| Hidden Layers (8) | 80.2% | 0.0011 |
| Hidden Layers (9) | 82.0% | 0.0211 |
| Hidden Layers (10) | 81.2% | 0.0031 |

**Logistic Regression**

Classic technique…

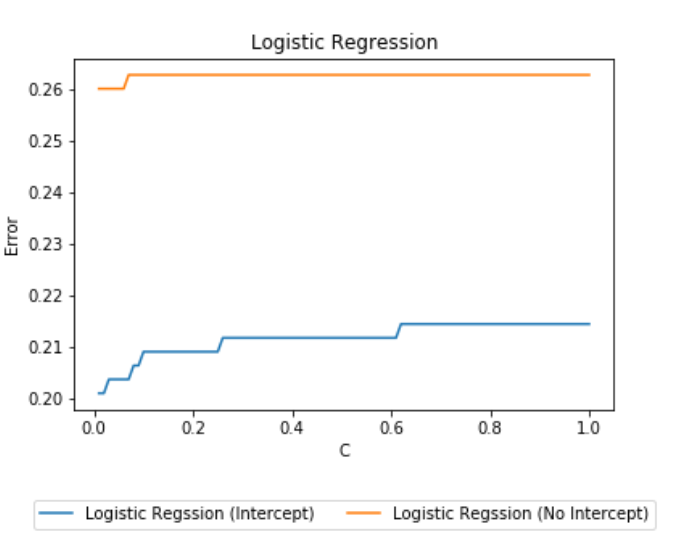
For this portion, a logistic model was fit with and without intercept. In addition, within python there is a attribute called “C” which is in the regularization and controls overfitting. Generally speaking, this parameter is used for small amounts of data with a lot of predictors. The introduction of this parameter is strictly referenced for informational benefits. Given the high level definition of the “C” parameter, it is expected that the model performs best at the lowest value given the data for this exercise. Which is intuitive, as the data only has four predictors with roughly 900 observations.

The accuracy of the Logistic Regression is plotted below.



The model performs better with inclusion of an intercept.

Accompanying Error Rate:



|  |  |  |
| --- | --- | --- |
| **Logistic Regression** | **Best Accuracy** | **C** |
| Intercept | 79.9% | 0.01 |
| No Intercept | 74.0% | 0.01 |

**Linear Probability**

Given the ease of interpretation of the coefficients a linear probability model was fit. The accuracy is reported for a OLS model with an without intercept.

|  |  |
| --- | --- |
| **Linear Probability** | **Best Accuracy** |
| Intercept | 78.3% |
| No Intercept | 77.7% |

**Exhaustive Grid Search**

Utilizing functionality from Scikit-Learn Python module Grid Search each of the modeling methods were fit by tuning the hyper parameters for the modeling technique. Given the different modeling methods have different hyper parameters and different number of hyper parameters to modify the time it takes to find the best performing model is dramatic.

Each of the iterations for a model goes through a stratified K-Fold cross validation. This technique splits the data by the outcome (Severity) and fits the model, utilizing the hold out to measurer the accuracy of the fitted model. As an example, given a Predictor and Outcome we have:

|  |  |  |
| --- | --- | --- |
| **Observation #** | **Outcome** | **Predictor** |
| 1 | 0 | 0.1 |
| 2 | 0 | 0.2 |
| 3 | 0 | 0.3 |
| 4 | 1 | 0.4 |
| 5 | 1 | 0.5 |
| 6 | 1 | 0.6 |

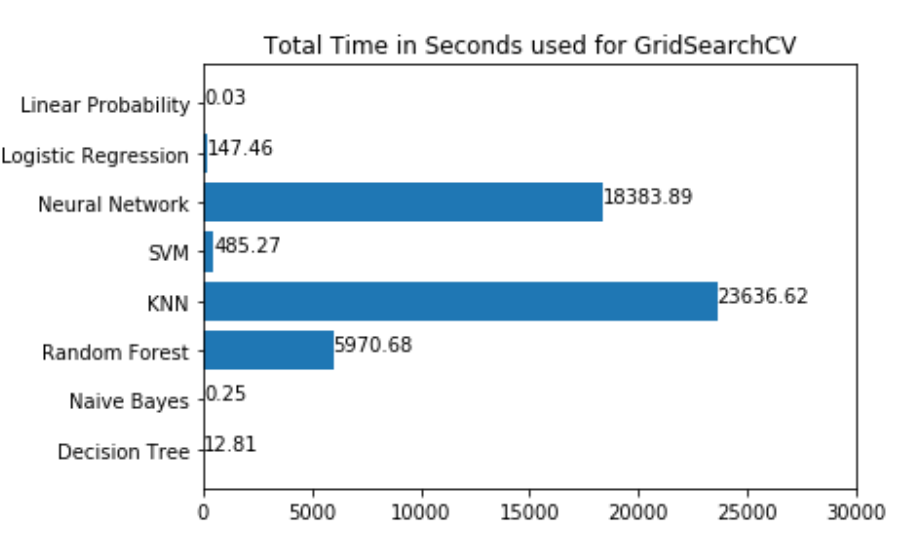
Utilizing stratified K-Folds splitting the data into 3 folds to fit the data the model would be fit three times then validated three times as follows:

|  |  |  |
| --- | --- | --- |
| **Fold #** | **Fit on Observation #** | **Scored on Observation #** |
| 1 | 2,3,5,6 | 1,4 |
| 2 | 1,3,4,6 | 2,5 |
| 3 | 1,2,4,5 | 3,6 |

The average accuracy is reported from the three folds. A model which has the best accuracy is utilized for reporting purposes. For the purposes of the exhaustive Grid Search in model selection, a 5 fold method was chosen for all model classifiers.

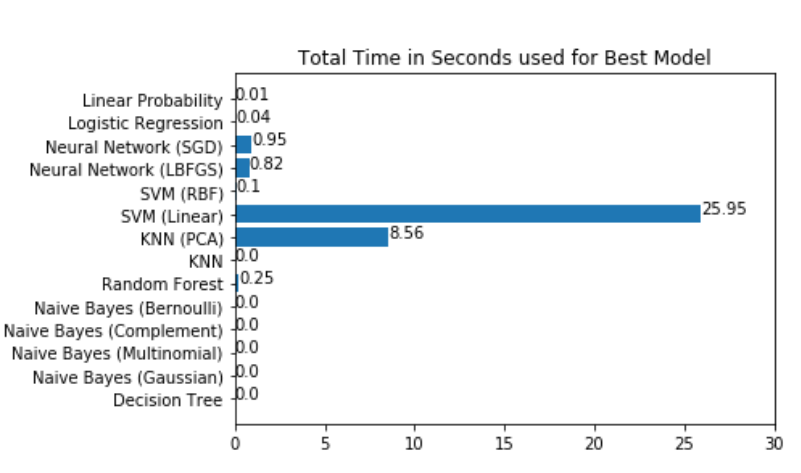
In order to ensure the comparison between modeling methods is comparable, each method was ran using total processing power of the custom built machine. Which was performed using a 10 core, 20 thread xeon intel server process (xeon E5-2640 @ 2.4GHz) with 32GB of DDR3 memory.

The total time observed (in seconds) to fit the modeling technique is pictured below. Note, there were multiple methods for particular modeling techniques, these methods are lumped together for the total time. Example, and SVM can have multiple kernels for fitting the model, here both the RBF and Linear kernel fit times were added together and reported.



As expected, the Naïve Bayes and Linear Probability models were the fastest to fit. The Neural Network took approximately 5 hours. K Nearest Neighbors took approximately 6.5 hours which was mainly attributed to the KNN model using Principal Component Analysis. The Random Forest took approximately 1.7 hours. These three modeling techniques took longer primarily due to the amount of hyper parameters which can be tuned.

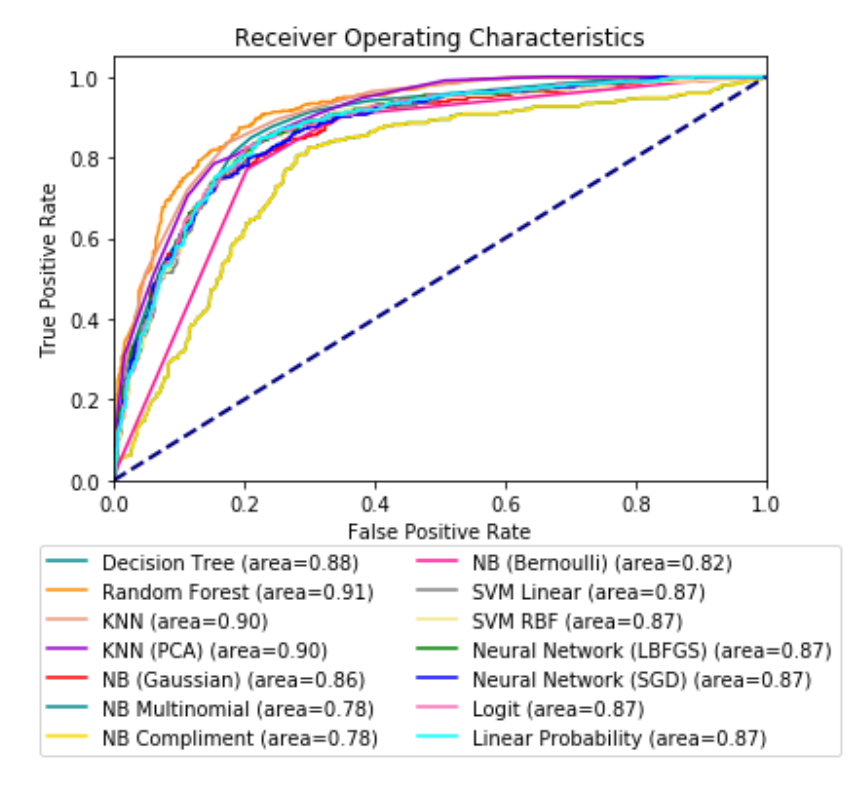
Post the training portion, the Grid Search returns the model with the appropriate parameters set which produced the best out of sample accuracy. The next graph is the time it took to for the best model from the Grid Search Results. This plot shows a single model for each of the classifiers and how long it took to fit.



The accuracy closely aligns to the model classification section. The model which produces the best accuracy is the Random Forest, with K Nearest Neighbors second. Naïve Bayes models performed slightly worse, otherwise all other modeling techniques results in similar accuracy, which is approximately 80% correct classification.

|  |  |  |
| --- | --- | --- |
| **Model** | **Best Accuracy** | **AUC (Area Under Curve)** |
| Decision Tree | 81.7% | 88% |
| Random Forest | 83.4% | 91% |
| K Nearest Neighbors | 83.0% | 90% |
| K Nearest Neighbors (Principal Component Analysis) | 81.0% | 90% |
| Naïve Bayes (Gaussian) | 78.6% | 86% |
| Naïve Bayes (Multinomial) | 74.9% | 78% |
| Naïve Bayes (Complement) | 75.4% | 78% |
| Naïve Bayes (Bernoulli) | 78.4% | 82% |
| Support Vector Classification (Linear) | 80.0% | 87% |
| Support Vector Classification (RBF) | 80.0% | 87% |
| Neural Network (LBFGS) | 80.6% | 87% |
| Neural Network (SGD) | 79.6% | 87% |
| Logistic | 80.7% | 87% |
| Linear Probability | 80.8% | 87% |

A common practice is to obtain the AUC (Area Under the Curve) in a ROC (Receiver Operating Curve) plot. Unlike accuracy, which basically states evaluates did you or did you not predict the correct binary outcome, the ROC shows how strongly you predicted. That is, if the model gives a predicted probability of 51%, this would be classified as a likely cancerous tumor (Severity=1). However, 51% is not exactly a strong probability. Likewise, fi the predicted probability is 90%, an accuracy score would still be correct, however the models predicted probability is much stronger. In general, a model which has better accuracy normally trends in similar direction for on a ROC plot.

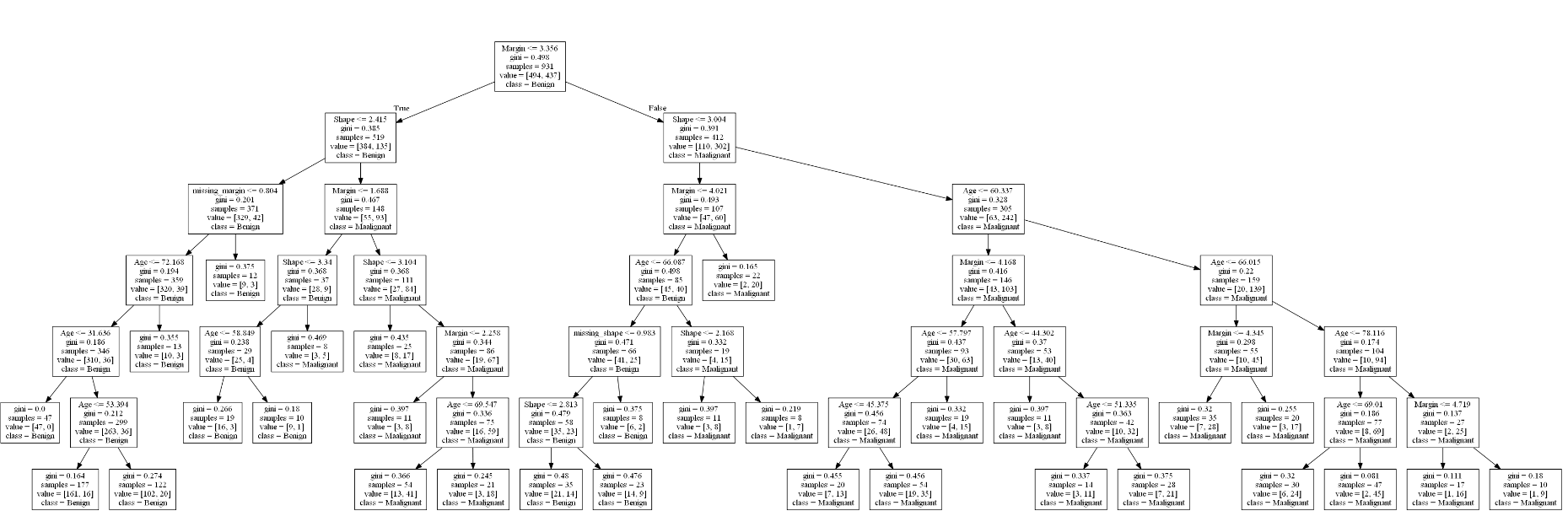


**Conclusion:**

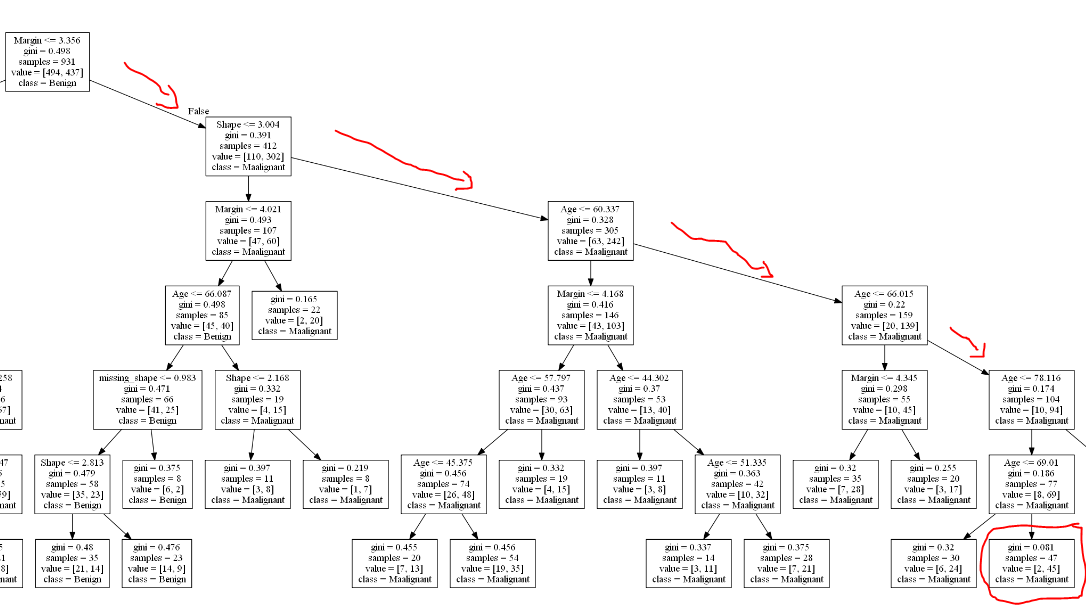
Given the objective, determine if a tumorous mass needs to be biopsied or not. It is assumed, that anytime a person is being evaluated for some kind of medical need (especially cancer), the person being evaluated would want information regarding the what and why explanation. Meaning, if some needs to be evaluated, likely, they would want to know why a doctor chose or made an informed suggestion. The suggestion a doctor makes given an informed decision from a model would potentially alter or have adverse effects on the patient.

With that in mind, my personal preference would be the ability to explain why a given outcome was chosen. For ease of interpretation, there is only three techniques that would be interpretable, the Logistic Regression, Decision Tree and the Linear Probability Model. The other model techniques, Random Forest, SVC, Neural Network, K Nearest Neighbors could be used as support, however the predictions offered would not be able to be explained in a coherent fashion.

Case in point, for a Decision Tree, a doctor could show the nodes and leaves of the tree. Explanation would be the path the person takes given the characteristics (predictors) provided.



Blown up path for explanation purposes:



The above path could easily be explained to a patient and could take the form of the following:

Using a single observation from the data

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Value** | **Definition** |
| Age | 70 | Implied |
| Shape | 4 | Irregular |
| Margin | 4 | Ill-Defined |
| Density | 3 | Low Density |

*Age, Shape and Margin measures which can be used to predict the need for a biopsy. In your case, the size of the Margin is ill-defined with an irregular shape and you are 70. Given this, a biopsy is highly suggestive as other patients with the same attributes as yours had a cancerous tumor 98% (45/47) of the time and is highly suggestive to obtain a biopsy.*

Linear Probability is has ease of interpretation as well. Given the parameter estimates are ratios and the predictors are discrete we can simply multiply the parameter estimate by the distinct values of the patients attributes to give them a probability of the tumor being cancerous.

The Linear Probability Model parameter estimates are:

|  |  |
| --- | --- |
| **Parameter** | **Estimate** |
| *Intercept* | -0.558 |
| Age | 0.008 |
| Shape | 0.104 |
| Margin | 0.093 |
| Density | 0.018 |

*\*\* Missing values are withheld for interpretation*

Here,an explanation could be stated to the patient as follows:

*Margins are ill defined and the shape is irregular which both have a higher probability of being cancerous and approximately 87% of other patients with these same attributes had cancerous tumors. Given your age, it is strongly recommended to have a biopsy.*

Logistic Regression can be interpreted as well, here a likely best case would be to inform the patient of the probability. The estimates for the best Logistic model is:

|  |  |
| --- | --- |
| **Parameter** | **Estimate** |
| *Intercept* | -6.060 |
| Age | 0.051 |
| Shape | 0.565 |
| Margin | 0.493 |
| Density | 0.217 |

The same statements as the Decision Tree and Linear Probability model could be utilized, here the probability from the Logistic model was 88%. However, in this case we could use marginal effects (at the median). The marginal effects givens an idea of how a unit change in one variable effects the probability, while holding all other variables at their median value. Unfortunately, the scikit learn modeling package does not really emphasis significance (such as p-values). However, another package in python “statsmodels.api” provides more traditional techniques that are widely used today. The results from the statsmodels.api marginal effects are as follows:

|  |  |  |
| --- | --- | --- |
| **Variable** | **Marginal Effect (at median)** | **P>|z|** |
| Age | 0.014 | 0.000 |
| Shape | 0.143 | 0.000 |
| Margin | 0.121 | 0.000 |
| Density | 0.010 | 0.865 |

Shape and Margin are the most power indicators for malignant tumors. Density is not significant, and could be due to the fact that the values were imputed (set to 0 for missing) and the attribute of density is ordinal in nature.

Which can be presented as:

*Given the irregular shape of the tumor alone and holding everything else constant (at the median value) the probability of the tumor being malignant is 57% more likely than benign.*

Not to say that the other machine learning techniques are not valid in the classification. If for example the data used in this analysis was geared towards something less severe, such as a movie recommendation. Someone reviewing recommended shows on Amazon Prime would not really give second thought to a recommended movie which they do not have interest in. In cases such as this, there is not really a desire to understand why a particular movie was recommended and the best performing model (Random Forest) at prediction could be used.

Lastly, we can compare the modeling techniques to the BI-RADS values assigned by the physicians. Recall, that the BI-RADS score is assigned a value of 1 to 5 and is done so by way of a double review process (2 physicians agreed). Values of 1 in a BI-RADS score would indicate highly benign, while 5 would indicate highly malignant.

Given the simple statistics of the data, it was noted that a vast majority of the population had a BI-RADS score of 4 and 5 and would indicate that physicians tend to lean on higher level of malignant tumors which in turn would require a biopsy. Note here, that the average value of BI-RADS for the data was 4.35, while the average for Severity=0 (biopsy shows it was not malignant) the BI-RADS average value was 3.97, Severity=1 (biopsy shows that it was in fact malignant) the average BI-RADS score was 4.79.

The distribution of BI-RADS to severity is:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Severity** | **BI-RADS** | | | | | **Total** |
| **1** | **2** | **3** | **4** | **5** |
| **0** | 3 | 12 | 27 | 412 | 40 | 494 |
| **1** | 3 | 1 | 4 | 120 | 309 | 437 |
| **Total** | 6 | 13 | 31 | 532 | 349 | 931 |

Given the distribution of BI-RADS, it is clear that the physicians leaned towards malignant tumors in evaluation. Potentially this was could be due to a “better safe than sorry” evaluation.

Using the predicted probabilities of the modeling techniques (only reporting the chosen, Logistic, Linear Probability and Decision Tree) a quantile can be obtained for comparison to the BI-RADS.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Severity** | **Decision Tree Quantile** | | | | | ***Total*** |
| **1** | **2** | **3** | **4** | **5** |
| **0** | 335 | 60 | 0 | 86 | 13 | *494* |
| **1** | 40 | 31 | 0 | 234 | 132 | *437* |
| ***Total*** | *375* | *91* | *0* | *320* | *145* | *931* |
|  |  |  |  |  |  |  |
| **Severity** | **Logistic Quantile** | | | | | ***Total*** |
| **1** | **2** | **3** | **4** | **5** |
| **0** | 293 | 64 | 55 | 52 | 30 | *494* |
| **1** | 31 | 25 | 49 | 146 | 186 | *437* |
| ***Total*** | *324* | *89* | *104* | *198* | *216* | *931* |
|  |  |  |  |  |  |  |
| **Severity** | **Linear Probability Quantile** | | | | | ***Total*** |
| **1** | **2** | **3** | **4** | **5** |
| **0** | 246 | 106 | 66 | 56 | 20 | *494* |
| **1** | 20 | 31 | 60 | 172 | 154 | *437* |
| ***Total*** | *266* | *137* | *126* | *228* | *174* | *931* |

For all intensive purposes, lets assume on the conservative side, being that any value of the quantile greater than 2 should be biopsied.

The distribution of biopsies would be greatly reduced using the modeling probability, ranging between 50 and 57% of the population would NOT require a biopsy versus the BI-RADS which was 98% of the population. The catch is the false positive rates in bins 1 and 2 which should be explained to the patient. For all intensive purposes, the Linear Probability had the lowest false negative rate for bins 1 and 2 at 10.3% (44/427).

**References:**

|  |  |
| --- | --- |
| Minimal Cost Complexity | https://scikit-learn.org/stable/modules/tree.html#id2 |
| Gini Impurity and Importance | https://stats.stackexchange.com/questions/92419/relative-importance-of-a-set-of-predictors-in-a-random-forests-classification-in |
| Out of Bag Sampling | https://en.wikipedia.org/wiki/Random\_forest |
| Euclidean Distance | https://en.wikipedia.org/wiki/K-nearest\_neighbors\_algorithm |
| Mean Centering | https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html |
| Principal Component Analysis | https://en.wikipedia.org/wiki/Principal\_component\_analysis |
| Conditional Independence | https://en.wikipedia.org/wiki/Conditional\_independence |
| Naïve Bayes Slides for GB | http://www.cs.cmu.edu/~tom/10601\_sp09/lectures/NBayes-1-28-2009-ann.pdf |
| Naïve Bayes Classifiers (Distributions) | https://scikit-learn.org/stable/modules/naive\_bayes.html |
| SVM (Parameter Tuning) | https://scikit-learn.org/stable/auto\_examples/svm/plot\_rbf\_parameters.html#sphx-glr-auto-examples-svm-plot-rbf-parameters-py |
| SVM Modeling Techniques | https://scikit-learn.org/stable/modules/svm.html#svm-kernels |
| SVM Theory | https://towardsdatascience.com/https-medium-com-pupalerushikesh-svm-f4b42800e989 |
| Neural Network Methods | https://scikit-learn.org/stable/modules/neural\_networks\_supervised.html |
| Neural Network Interpretation | https://deepai.org/machine-learning-glossary-and-terms/weight-artificial-neural-network |
| Neural Network Parameters | https://scikit-learn.org/stable/modules/generated/sklearn.neural\_network.MLPClassifier.html |
| Regularization (Logit) | https://en.wikipedia.org/wiki/Regularization\_%28mathematics%29#Regularization\_in\_statistics\_and\_machine\_learning |
| Stratified K-Fold | https://scikit-learn.org/stable/modules/generated/sklearn.model\_selection.StratifiedKFold.html |
| ROC Curve | https://scikit-learn.org/stable/auto\_examples/model\_selection/plot\_roc.html |