Tumor Stage Prediction

Yun Lin, Yupan Wang, Sangyu Baek, and Hongkai Lou

Abstract

We are studying the prediction of tumor stage of breast cancer using machine learning (ML). Through the data collected from 2509 patients, multiple machine learning algorithms are applied for the tumor stage prediction of the breast cancer by considering the most influential histopathology parameters, aiming to help doctors to better decide the procedure and treatment for that patient. Random Forest was utilized for feature selection and neural network is mainly applied for predicting model. We found that the algorithm has an accuracy of approximately 88.6%.

1 Introduction

According to Centers for Disease Control and Prevention, Breast cancer is the most prevalent kind of cancer in women in the US, aside from skin cancer. Although the number of breast cancer deaths has decreased over time, it is still the second most common malignancy among all women and the most common among Hispanic women. About 264,000 women and 2,400 men are given the diagnosis of breast cancer each year in the US. The United States loses 500 men and 42,000 women to breast cancer each year. Compared to White women, Black women have a greater mortality rate from breast cancer.

Breast Cancer is usually diagnosed through methods like breast ultrasound, diagnostic mammogram, MRI, biopsy, etc. Once it is diagnosed, it is necessary to stage the cancer to see if it has spread. However, accurately identify the cancer cells is hard to achieve in practice. Thus, predicting the stage of breast cancer through results from other tests is essential.

Since physical conditions are mostly quantifiable, and the quantity of patients is numerous, machine learning algorithms are the most suitable tool to handle the prediction.

1.1 Organization

We organize the paper as follows: In section two we will introduce the mathematical models we decide to use, and talk more about their mathematical concepts, formulas, and intuitions. These are the fundamentals for section four. Section three talks about our data cleaning, imputation, visualizations, and initial feature selections. We include plots in visualizations as many as possible to help our readers better understand the relationship between our feature and tumor stage. In section four we will apply all models described in section two to our cleaned dataset. Based on each model's performance, we will select the best predicting variables and models. In section five we will make conclusions and discuss about certain limitations and future work.

2 Mathematical Formulation

2.1 Linear Regression

$$y(x_i, w) = w_0 + \sum_{i=1}^{I} w_i x_i + \epsilon_i$$

The Error function we used is given by the sum of the squares of the errors between the predictions $y(x_i, w)$ for each data point x_i and the corresponding target values t_i, so that we minimize, where the factor of 1/2 is included for later convenience.

$$E(w) = \frac{1}{2} \sum_{i=1}^{n} (y(x_i, w) - t_i)^2$$

with Lasso, we minimize the Error function with the lasso regularizer, which we minimize the following equation. M = 17

$$E(w) = \frac{1}{2} \sum_{i=1}^{n} (y(x_i, w) - t_i)^2 + \frac{\lambda}{2} \sum_{m=1}^{M} |w_m|$$

with Ridge, we minimize the Error function with the Ridge regularizer, which we minimize the following equation. M = 17

$$E(w) = \frac{1}{2}\sum_{i=1}^{n}(y(x_i, w) - t_i)^2 + \frac{\lambda}{2}\sum_{m=1}^{M} w_m^2$$

2.2 Neural Network

All the activation function $h^{(i)}, i = 1, 2, 3$ we use is softmax function: $h(x) = max(0, x)$. More details about number of predictors are on section 4.

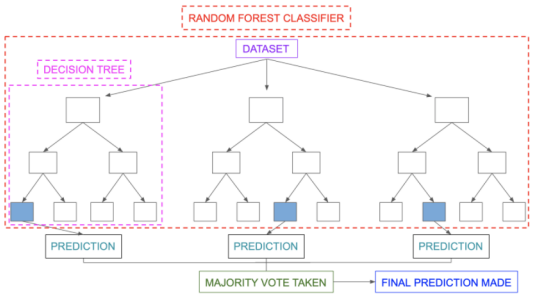
With one hidden layer: J = 17, I = 4
Output (t_n): Tumor Stage, $\forall\, t \in [1, 2, 3, 4]$

$$y_n(x, w_1, w_2) = h^{(2)}(\sum_{i=0}^{I} w_{ni}^{(2)} h^{(1)}(\sum_{j=0}^{J} w_{ij}^{(1)} x_j))$$

With two hidden layer: K = 17, J $\in [5, 16]$, I = 4
Output (t_n): Tumor Stage, $\forall\, t \in [1, 2, 3, 4]$

$$y_n(x, w_1, w_2, w_3)$$
$$= h^{(3)}(\sum_{i=0}^{I} w_{ni}^{(3)} h^{(2)}(\sum_{j=0}^{J} w_{ij}^{(2)} h^{(1)}(\sum_{k=0}^{K} w_{jk}^{(1)} x_k)))$$

2.3 Random Forest



Random forests use the Bootstrap aggregating method as known as bagging method. It creates a subset of the original dataset. Random forest randomly selects observations, builds a decision tree and the average result is taken. Therefore it doesn't use any set of formulas. The final output for classification case is based on majority vote, and for regression case the output is using the mean or median of all output. Hence the problem of overfitting is taken care of.

2.4 Support Vector Machine

$$max_{\{a_i\}} g(\{\lambda_n\}, \{\alpha_n\})$$

$$= \sum_n \alpha_n + \frac{1}{2}\sum_n \alpha_m \alpha_n y_m y_n k(x_m, x_n)$$

$$\alpha_n, \lambda_n \geq 0, \forall n; \sum_n \alpha_n y_n = 0; C - \alpha_n - \lambda_n = 0$$

3 Applications or Experiments

To build our model and make predictions ion on cancer stage, we first have to conduct data cleaning and imputations to deal with NAs and unusable features. We divide our section into three parts: Data cleaning and imputations; Visualizations and the first round of feature selections; Model training.

3.1 Data cleaning and imputations

Again, our Dataset consists of 2509 patients' information, with 33 features and our Y variable tumor stage. As mentioned above, Doctors attempt to diagnose tumor stage by using X-rays, MRI, or other Videographic information. However, currently, precisely diagnosing the tumor stage usually requires tumor samples that are cut and preserved during operations by surgeons. Thus, if we are predicting tumor stage before operations, we need to clear out information that cannot be obtained after certain treatment. In the end, these are the features that we decided to delete: Type of Breast Surgery, Vital Status, Relapse Free Status and Month, Radiotherapy, and Overall Survival Status and Month.

To deal with missing values, we have to conduct imputations. We start with numeric continuous variables. Luckily, among three numeric continuous variables(Age, Tumor Size, Nottingham prognostic index), only age has eleven missing values. We fill it in with the mean age of diagnosis from all other samples, which is 61 years old. This value is similar to the current world mean age of diagnosis for breast cancer, which is 62 years old.

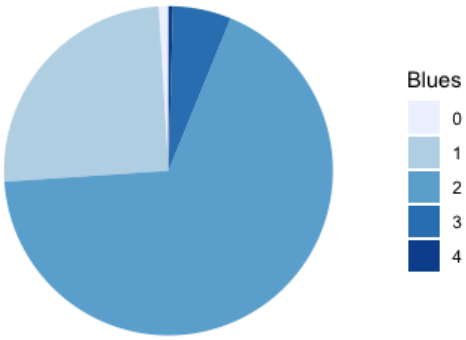
We then deal with numeric discrete variables and categorical variables. If some features have missing values of more than 600, we would have to remove them, since they aren't convincing. For those features that have missing values less than 600, we created a table of all existing samples that count each category's frequency. We then fill in the missing values with the category corresponding to the highest frequency. If, for example, we have two categories that have equal high frequency, such as 30 percent and 31 percent, we will fill in the missing values with these two categories. In the end, only 21 features are left.

For the remaining columns with missing values, we first quantify the value to integers. Take the column "HER2.status.measured.by.SNP6" as an example, the values under this type of data result in neutral, loss or gain. We set neutral = 0, loss = -1, and gain = 1. Then calculate the mean of existing values which is 0.170632911. To achieve the least mean shift, we then set all the missing values to neutral. As a result, we have dealt with all the missing values in such a column and remain the least

shift from the original data. Similarly, we applied the method of imputation to the rest of the data and therefore we get a preliminary cleaned dataset.

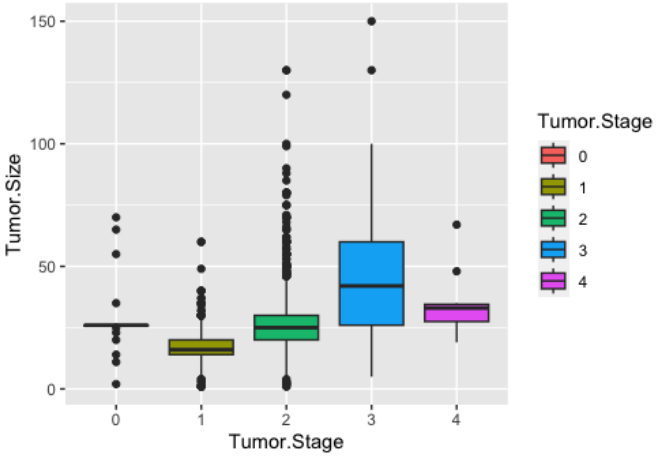
3.2 Visualization

To better help our reader to understand some possible correlations between features and the Tumor Stage, we would like to provide some visualization in the following sections. We start by looking at the distribution of the tumor stage. Here is a pie plot.

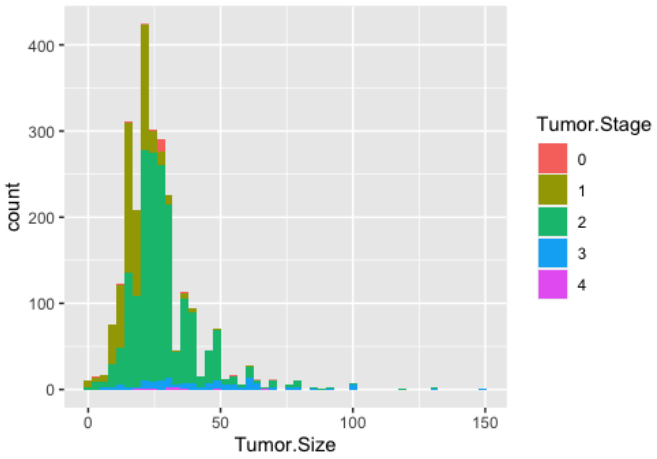


For those concerning the specific distribution of tumor stage, the dataset consists of 24 stage 0 tumor, 630 stage 1 tumor, 1700 stage 2 tumor, 144 stage 3 tumor, and 11 stage 4 tumor. Thus, tumor stage distributions are highly unbalanced, posting challenges to our classification test. Next, We can look at some possible correlations between feature and tumor stage below.

Age At Diagnosis vs Tumor Stage Boxplot

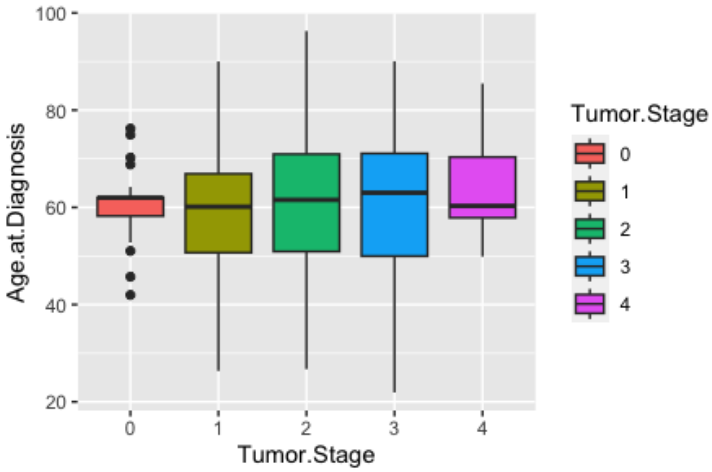


Age At Diagnosis vs Tumor Stage Histogram



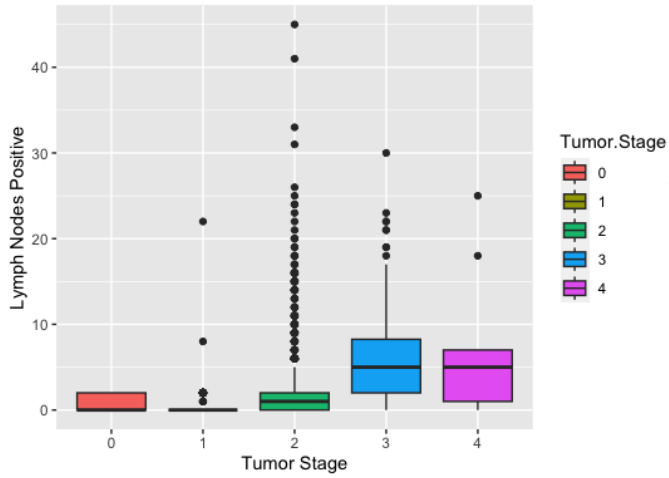
Our first two plots visualize the relationship between tumor size and tumor stage. Intuitively, they are positively correlated: Bigger tumor size indicates its longer existence and thus should correspond to a higher tumor stage. From both the side-by-side boxplot and histogram, tumor stage 1 toward tumor stage 3 show the pattern: A higher tumor stage indicates a larger mean and median value for tumor size. This is not the case for tumor stage 0 and 4: there is no pattern. We speculate that it might be due to the tiny sample size for these two stages.

Age At Diagnosis vs Tumor Stage



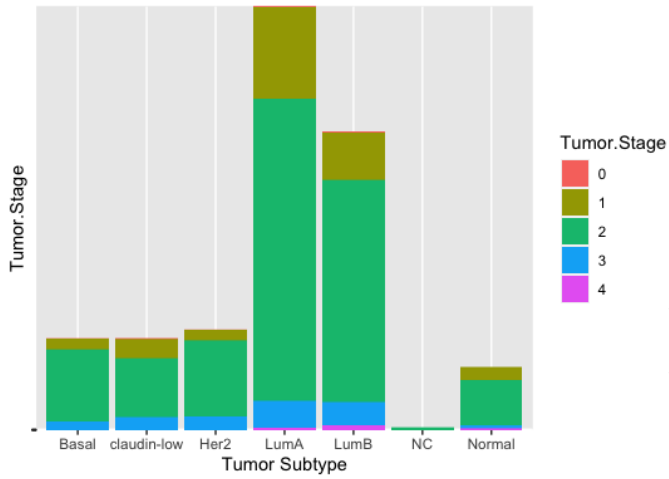
Surprisingly, there is no obvious pattern between age at diagnosis and tumor stage. One possible explanation is that the development speed of the tumor stage varies from human to human since the speed that breast cancer destroys genes might be different. Furthermore, the subtype of breast cancer plays an important role: Tumor subtype LumA and LumB develop much slower than tumor Her2 Positive. Thus, Age may not be an important indicator of tumor Stage. An interesting fact is that Tumor stage 1, 2, and 3 all have approximately normal distributions with single mode.

Positive Cancer Cell in Lymph Nodes vs Tumor Stage



Besides tumor size, the lymph node is also an important factor in breast cancer, or actually in all types of cancer. Usually, to identify if a tumor is benign or malignant, besides the tumor stage, positive cancer cell in the lymph node conveys bad news. Thus, we look at the possible correlation between positive lymph nodes and the tumor stage. From the plot, it seems as the tumor stage increases, the mean and median values of positive lymph node increase. Tumor stage 2 clearly has the largest variance due to its sample size. We can't detect patterns for tumor stage 0 and 4, possibly again due to its tiny sample size.

Tumor Subtype vs Tumor Stage



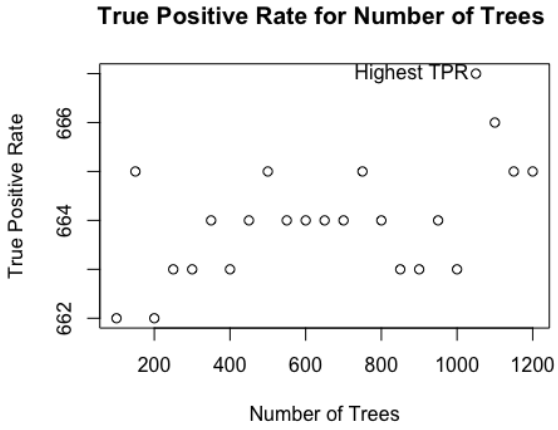
In the last visualization let's look at the importance of tumor subtype in determining or affecting tumor stage. From a medical standpoint, the subtype of a tumor is closely related to the development of the tumor stage. A simple concept to introduce their relationship is the grade and level of students. We can think of the tumor stage as, for example, freshman, sophomore, junior, and senior in high school. In this example, if a student accumulates enough knowledge, he or she can rise to the upper-level quicker, like 5 or 6 months. The subtype of the tumor will decide if the tumor is benign or malignant, which in this student's example, straightly decides a student's ability to absorb knowledge. If the tumor's subtype is Her2 positive, for example, in the third category of the tumor subtype, then the tumor will likely be malignant, and the student will learn the knowledge very fast. At this point, he or she will upgrade to the upper level much quicker. Thus, the tumor stage will evolve quickly. If the tumor's subtype is LumA or LumB, then the tumor will more likely be benign, and the upgrade speed will be much slower. The tumor stage in this case will evolve much slower. From the stacked bar plot, we can see there is likely to be some correlation between tumor subtype and tumor stage. For example, the proportion of tumor stage 1 in LumA is much higher than the proportion of tumor stage 1 in LumB and Her2. As the tumor subtype differs, there seem to exhibit different patterns for the distribution of tumor stages. However, we are not making any statistical inference at this point; we are just squeezing out information from the dataset that further helps us to build our model.

3.3 Feature Selection

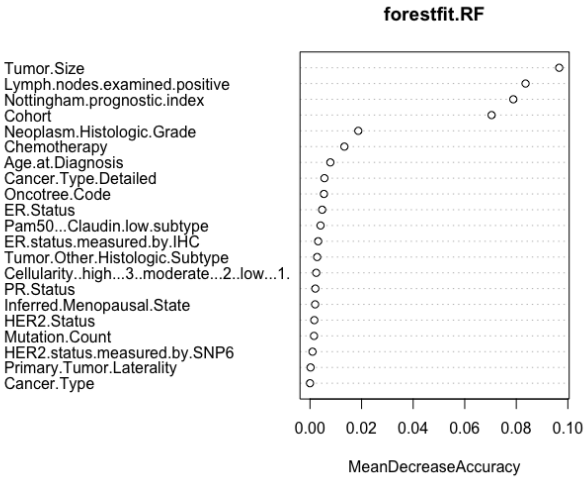
We have now 21 features left, and we will delete some unuseful features. First, we did consider the use of Principal Component Analysis (PCA), or Kernal PCA. Just a brief introduction, PCA is a popular technique used for dimension reduction or feature extraction. Usually, the goal of PCA is to project the original data points with dimension D, to a space with dimension M such that M«D. When projecting the data into the new feature space, PCA will maximize the variance of the projected data so that the data points in lower dimensional space explained more variation of the original space. Beyond maximizing variance, PCA also tries to mean the squared distance between data points and their projections. So intuitively, PCA tries to project from higher dimensions to lower dimensions when maximizing its variance and minimizing its mean squared distance with the original point. However, as we touched on earlier, among the 21 features left, only three features, which are age, Nottingham index, and tumor size. Moreover, PCA is usually used to solve the problem of colinearity, while here, as we will see in the correlation plot in the following part, they are not highly correlated. Applying PCA to categorical or discrete features will not be helpful. Thus, we ignore PCA in our model.

We next look at the random forest classifier algorithm. Random Forest basically combines a large number of decision trees together and will make decisions based on the majority of decision trees. We would like to use the random forest to help us start with feature selection. We use the function *randomForest* from the library *randomForest*. We specify the

number of variables randomly sampled as candidates at each split to be 4, and we used a train test split with a ratio of 7 to 3 to determine the optimal number of trees to use for our data. We used the number of the True positive rate to determine the best number of trees. From the following plot, we can see that using 1050 decision trees, the true positive is the highest.



Thus, we decide to use 1050 decision trees to build the model and graph out the most important features. We use the function *varImpPlot* to graph out the features.



As we can see, besides Tumor Size, Positive Lymph nodes, Nottingham index, and Cohort, all other features have pretty similar Mean Decrease Accuracy in terms of the model accuracy. This provides us with some possible unimportant features for models. Third, we look into the correlation table below to observe the correlation between all the predictors and the tumor stage. We decided to remove all the predictors that has closed to zero correlation with our response, which we have 16 features left.

3.4 Encode Categorical data

Last, before we started training the models is we encoded the categorical predictors by using the LabelEncoder from sklearn package.

4 Model Selection

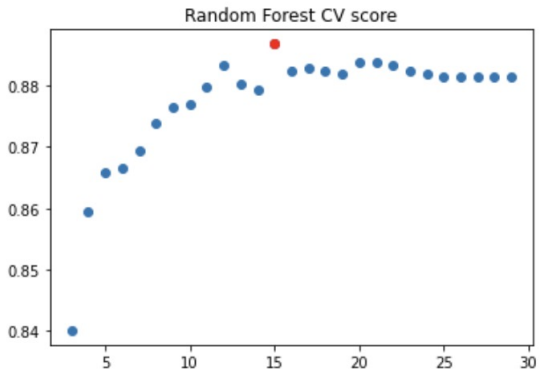
4.1 Train-Test split

First, we split the data into 80% training set and 20% test set. For us to choose the right model to get deep into, we decided to apply different models with default setting first to see which model perform best with the default setting.

Since our desire output contains 5 classes, for the regression models we created decision boundaries in order to categorize the float values as desired output.

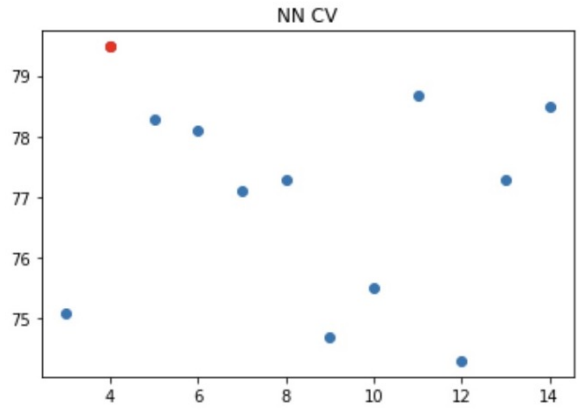
Model	CV training score
Linear regression	76%
Linear regression w Lasso	71.3%
Linear regression w Ridge	73.1 %
Neural Network w 1 hidden layer	77.09%
Neural Network w 2 hidden layer	76.69%
Random Forest	87.48%
SVM	74.7%

The table above shown each model with their cross validation score with the training set respectively. Therefore, we decided to get depth into the Neural Network and random forest model.

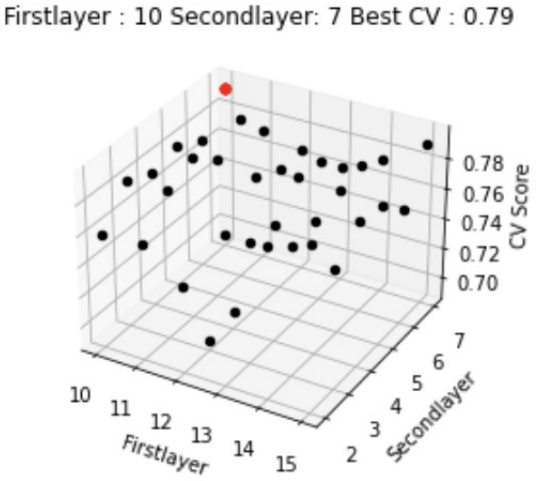


Once the regression models are ruled out, we did a cross validation test on Random Forest model. Random Forest gave the highest cv scores among all models we tried. However, although the accuracy of the model with the training set was very high, it had a serious problem. Since our dataset is highly unbalanced, the Random Forest model was able to score very high on the cross validation check without assigning any input to tumor stage 4. This was due to the fact that tumor stage 4 patients consist of only a very small portion of all patients. Furthermore, we used 16 features for modeling and creating a Random Forest model with depth 15(the depth with the highest CV score) would rise the concern of overfitting. Thus, we concluded that it would be logical to rule out the Random Forest from our options.

```
Best CV scroe for NN at 79.48 %
```



The graph above shows the cross validation scores of our Nerual Network model with one hidden layer with different layer sizes. The best CV score we obtained from this test was hidden layer of size 4 when only one layer is chosen to be used for the model, which satisfies one of the rules of thumb for number of hidden neuron $= sqrt(input_{layernode} * output_{layernode})$. The accuracy of with the hidden layer size with the best CV score was 79.48 percent. This is a descent result. However, we investigated further to see if increasing the number of layers would increase the accuracy.



The 3D plot above shows the CV scores with different number of hidden neurons on each layer with two hidden layers. Same rules were applied in determining the size of each layer. However, as shown above, increasing the number of layers from one to two gave the accuracy of 79 percent with the layer sizes with the highest CV score. This is slightly less than what we got from the cross validation check with one layer. Thus, it is logical to decide to use one hidden layer because it is both more accurate and efficient.

5 Conclusions

This project was conducted to build a model that predicts a patient's tumor stage with a set of patient information. In medical practice, a doctor would have to physically take the sameple of the tumor in order for them to be able to determine which stage the tumor is in. Thus, this model has a possible use that can be very practical in medical fields since it is more desirable to be able to determine the tumor stage without actually going through an operation. In order to achieve the goal, we searched for the model that gives the most accurate and relevant predictions on tumor stage. The dataset we used to build this model contains numerous different kinds of information from breast cancer patients. However, the original dataset required some cleaning and feature selection because there were some rows of data that could not be utiliezed for modeling and some data columns that do not show much correlation with the tumor stage data. Through this process, we were able to reduce the number of data columns to 16. Once data cleaning and feature selection were done, we built a few regression models, a Random Forest model, an SVM model and a Neural Network model. Regression and SVM models were ruled out of the possible options first due to their low accuracy on the prediction. Also, using a linear regression model with 16 predictors raised the concern of overfitting. Furthermore, although the Random Forest model gave the highest accuracy on its prediction, it was not cho-

sen since it failed to identify any stage 4 tumors and had a possibility of overfiting. After these thorough considerations, the Nerual Network was chosen for this project and through cross validation test, we were able to decide to use a Neural Network model with hidden layer with 4 hidden neurons. Although there is still a room for some improvements, the final model gives quite accurate and relevant predictions, which shows that this model can be a quite promising alternative to physical examination of tumors for determining a breast cancer patient's tumor stage.

References

"Comp.ai.neural-Nets FAQ, Part 3 of 7: Generalizationsection - How Many Hidden Units Should I Use?" Faqs.org, http://www.faqs.org/faqs/ai-faq/neural-nets/part3/section-10.html.

"Introduction to Random Forest in Machine Learning." Section, www.section.io/engineering-education/introduction-to-random-forest-in-machine-learning/.

Bishop, Christopher M. Pattern Recognition and Machine Learning. New York :Springer, 2006.