Analysis of Breast Cancer Data (Selena and Vicky)

**Problem**

This report is a reflection of the analysis on the conditions of the breast cancer tumors (benign or malignant). The data is gathered by Dr. William H. Wolberg, W. Nick Street, and Olvi L. Mangasarian from the University of Wisconsin and donated online by Nick Street in 1995. The data consists of 569 subjects with binary classification (357 benign and 212 malignant), ten attributes (radius, texture, perimeter, area, smoothness, compactness, concavity, concave points, symmetry and fractal dimension) and 30 features, derived from digitalized images of fine needle aspirate (FNA) of breast mass. The features include means, standard errors and worsts for each of the attributes. No missing data is identified in the dataset.

**Procedure**

Based on the binary characteristics of the dataset, we used three different classification approaches: K-Nearest Neighbors, Logistic Regression and Random Forest. K-Nearest Neighbors classifies the condition of the breast cancer tumor based on the number of training data closest in distance to the breast cancer tumor. The input of the classifier consists of k closest training examples and the output of the classifier is the class membership of the examined data. The majority membership of the data’s k nearest neighbors determines the condition of the examined data point. For example, if there were five data points identified as class “M” near data point A and three data points identified as class “B”, data point A should be classified as “M” due to pure number comparisons.

In our study, we used sklearn KNeighborClassifier for our K-Nearest Neighbor classifier. First, we needed to tune the parameter k (how many neighbors to be considered) to obtain optimal classification results. We decided to use k-fold cross-validation to choose the optimal k. K-fold cross-validation is the process where the entire dataset is divided into k clusters of data. One subset of data is used as the cross-validation dataset while the remaining (k-1) subsets of data are used as train set. Repeat the process k times until each subset of data has been utilized as the cross-validation set. Obtain the output parameter, in this case accuracy, through averaging over k different accuracies from the k processes. To perform k-fold cross-validation, we decided to use GridSearchCV from sklearn, which is a function that inputs the number of neighbors, ranging from 1 to 100, the number of cross-validation sets and output variables and outputs the mean scores for each possible number of neighbors. The graph for the relationship between k (number of neighbors) and accuracy can be seen in Figure 1. We can see from Figure 1. that the number of nearest neighbor associated with the highest accuracy is 13, therefore our K-Nearest Neighbor classifier has k of 13.

Logistic Regression classifies data based on the probability of a certain binary result from multiple predictors or features. It is a further development from linear regression, which finds the optimal linear relationship between the independent variables and the dependent variables. From linear regression, logistic regression assumes the conditional distribution as [Bernoulli distribution](https://en.wikipedia.org/wiki/Bernoulli_distribution) rather than a [Gaussian distribution](https://en.wikipedia.org/wiki/Gaussian_distribution) due to the binary characteristics of the output. Besides, the outputs of the logistic regression are confined between zero and one due to the fact that the outputs are probabilities of data points being assigned to certain categories. The probabilities are estimated using logistic function, which converts the hypothesis function into a binary output. If the direct output of the hypothesis function is bigger than 0.5, the data point will be classified as positive; if the direct output is smaller than 0.5, the data point will be classified as negative.

In our study, we used LogisticRegressionCV to implement our logistic regression classifier, which fits binary and one-vs-all logistic regression with built-in regularization. Instead of tuning the learning rate, LogisticRegressionCV uses cross-validation to optimize C, a regularization parameter. Therefore no regularization was involved in building the logistic regression classifier.

Random Forest classifies data points by constructs multiple decision trees and finding the mode or the mean predictions of the decision trees. Decision trees predict the memberships of the outputs based on multiple input variables and are essentially representations of the classifications based on multiple features. Based on one feature, a decision tree can split the datasets into two subsets. Within the two subsets, another feature can be used to divide the remaining datasets. Multiple features are utilized recursively in the process until the subset of a node obtains the same membership or no additional values are added with further divisions. Given the multiple features in the data, various decision trees can be constructed based on the different selections of features. Therefore multiple classifications can be obtained through the different decision trees constructed. Random Forests increase the validity and reliability of the classification by averaging over the results of multiple decision trees or by selecting the membership with greater presence.

In our study, we used RandomForestClassifier to build our random forest classifier. We used cross-validation to tune the parameters (the number of trees and the number of features) with GridSearchCV. In our random forests, our optimal number of trees was 21 and our optimal number of features was 10. From Figure 2., we can see the distribution of mean cross-validation scores given the number of features and the number of trees.

**Results**

For the three classifiers: K-Nearest Neighbor, Logistic Regression and Random Forests, we consistently looked at the recall, precision, learning curves, confusion matrix, and cross-validation scores for each of the classifiers. Recall equals to true positive divided by the sum of true positives and false negatives. True positives exist when predicted positives match with actual positive outputs. False positives exist when predicted positives are matched with actual negative outputs. True negatives exist when predicted negatives match with actual negative outcomes. False negatives exist when predicted negatives are matched with actual positive outcomes. Recall essentially assesses the proportion of true positives among actual positive populations or the fraction of positive hits among all that are retrieved. Precision, on the other hand, assesses the proportion of true positives among the predicted positive or the fraction of retrieved positives among all positives. Precision equals to true positives divided by the sum of true positives and false positives. In the process of tumor identification, the consequences for false identification of actual positives are quite extreme given that the patients would possibly lose chances of survival. Therefore in our analysis, we tried to decrease the number of false negatives as much as possible. Since recall equals to true positive divided by the sum of true positives and false negatives, the best classifier should present the largest recall. Besides, the best classifier should present the smallest false negatives in the confusion matrix.

Due to the binary characteristics of our outputs, our learn curves are graphic relationships between accuracy instead of the cost function and the size of the training examples. Accuracy equals to the sum of true positives and true negatives divided by the total population. It assesses how accurately the classifiers categorize the output, and essentially is the proportion of successfully identified data points among all data points. The learning curves for both the training set and the cross-validation sets are presented and compared in the graph. (See Figure 2.) We can see that for all three classifiers, the learning curves for both training set and cross-validation set are above 90%, which means that the classifiers are functioning well and the bias and variances for each of the classifiers are low.

Cross-validation scores are generated using cross\_val\_score in sklearn, which uses k-fold cross-validation method to test the reliability and the applicability of the classifiers. Like mentioned before, k-fold cross-validation method divides the total sample into k subsamples and uses one subsample as cross-validation set and the remaining dataset as test set. We preferred the k-fold cross-validation assessment of the classifier over manually separating the total sample into 60% training set, 20% cross-validation set and 20% test set and checking the applicability of the classifiers because of our limited sample, 569. The k-fold cross-validation solves the problem of limited sample size using randomization, therefore is a preferable method in out analysis. We can see from Table 1. and Table 2. that across most possible measurements, recall, the false negatives and the learning curves, Random Forest was our best performer.

**Discussion**

From our extensive analysis of the three different classifiers, we can see that in terms of the learning curves and minimal number of false negatives, Random Forest classifier is a the best performer. The classifiers are generally well performing with an average of over 90% in accuracy and aver 95% in cross-validation scores. The generally satisfying performance of the classifiers can be attributed to the extensive use of cross-validation in our study both during parameter tuning and model assessment. The use of cross-validation significantly decreased bias and variances and undermined over-fitting and under-fitting problems usually present in machine learning classifiers. From the comparisons of confusion matrix and precisions, we can see that a minimal presence of false negatives is maintained among the three classifiers, which ensures that patients who have malign tumors can get assess to treatment as early as possible. Even with a generally well performing classifiers, we believed that the analysis results could improved significantly with a larger sample size. Five-hundred sample size did not allow us to divide the total sample into training, cross-validation and test set, which would have been a useful method to avoid over-fitting.

Table 1. *Comparison statistics for three classifiers*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Classifiers** | **Precision** | **Recall** | **False Negative** | **Accuracy** |
| K-Nearest Neighbor | 0.83 | 0.96 | 5.00 | 0.95 |
| Logistic Regression | 0.96 | 0.93 | 7.00 | 0.93 |
| Random Forest | 0.93 | 1.00 | 2.00 | 0.96 |

*Note.* Precision: assesses the proportion of true positives among the predicted positive.

Recall: proportion of true positives among actual positive populations.

False Negative: predicted negatives are matched with actual positive outcomes.

Accuracy equals to the sum of true positives and true negatives divided by the total population.

Table 2. *Learning Curves for three classifiers*

|  |  |
| --- | --- |
| Classifiers |  |
| K-Nearest Neighbor |  |
| Logistic Regression |  |
| Random Forest |  |

Figure 1.

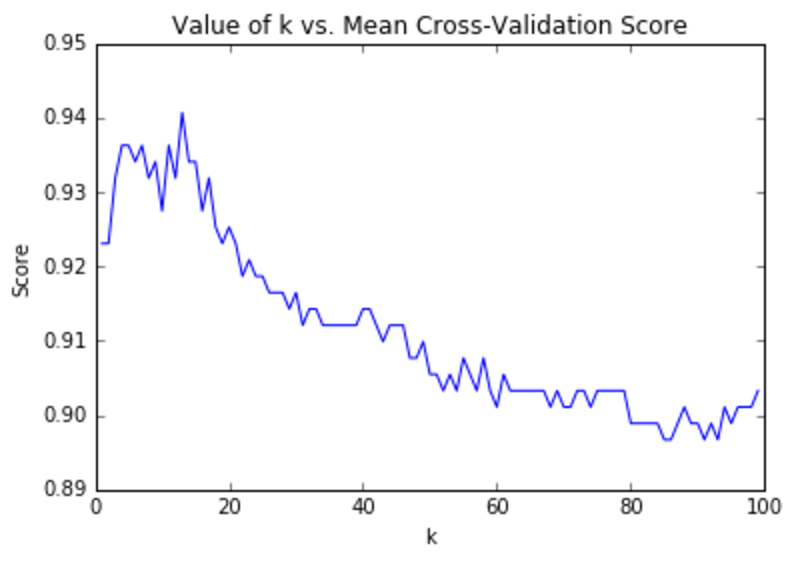


Figure 2.

