Computational Molecular Medicine Project

Introduction

Each year, almost 71,000 new cases of bladder cancer are diagnosed in the States. About 90% of the people with bladder cancer are above 55 years old, and the average age people are diagnosed with bladder cancer is 73. Bladder cancer is usually highly treatable when it is diagnosed and treated in the early stages. Stage T1 bladder cancer can be treated with cytoscopic surgery without removing the bladder, whereas a more lethal, muscle-invasive stage (T2 – T4) requires a more radical treatment (removal of the entire bladder). The objective of this project is to develop a predictor for distinguishing patients with superficial T1 cancer who progress later on from those who don't based on gene expression (Information on sex and age is not incorporated in this project).

There are three datasets containing gene expression values and corresponding phenotype labels (Y = 0 for "no progression" and Y = 1 for "progression") from two lab platforms. All patients had superficial T1 cancers at the time of data collection, some progressed. The gene expression values are normalized to z-score by distracting the global mean and dividing the global standard deviation for each gene. This project will only use the information on the 9,584 genes that all three datasets have in common. Those common genes are extracted from three datasets and ordered alphabetically, then combined into one new dataset for analysis. Wilcoxon-rank sum test is used to select the top N most differentially expressed genes from the training set.

Methods

k-fold cross validation

3-fold cross validation is used for assessing the performance of the classifiers due to the limited number of progression cases (49 out of 321 cases). On every round of cross validation, the dataset is partitioned into complementary subsets. The classifiers learn from the training set and validate on the testing set, and validation results are averaged over the rounds to give an estimate of the classifier's performance (sensitivity, specificity and auroc).

K-Nearest Neighbors

K-nearest neighbors is a non-parametric method used for classification. The principle behind is to find k number of training data in the feature space closest in distance to the new data point and predict the class membership from these by vote. The distance, in this project, is standard Euclidean distance. The discriminant function ("score") *g(x)* is the number of the k nearest neighbors that vote for class 1 (i.e. progression).

Logistic Regression

Logistic regression uses a logic function to model a binary dependent variable (in this case, progression vs. no progression). A linear relationship between the predictor variables and log-odds of the event Y = 1 is assumed. This linear relationship can be written in the following mathematical form:

$$\iota = log\frac{P(Y=1)}{1-P\,(Y=1)} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_N x_N$$

Where ι is the log odds of Y = 1, β_i is the coefficients of the predictor variable (features) x_i

Naïve Bayes

Naïve Bayes classifier is a family of simple probabilistic classifiers based on applying Bayes Theorem with strong independence assumptions between the features (all features in X are mutually independent, conditional on the category C_k), which is normally untrue for biological systems.

Abstractly, naïve Bayes is a conditional probability model:

$$p(C_k|X) = \frac{p(C_k)\, p(X|C_k)}{p(X)} \;\; or\; psterior\; probability = \frac{prior * likelihood}{marginal\; distribution}$$

Where C_k are possible class membership and X = (x_1, x_2, …, x_N) representing some N features

Naïve Bayes classifier picks the most probable class membership using, for instance, maximum a posterior (MAP). It assigns a class k as follows:

$$\hat{y} = arg \max_{k \in \{1,\ldots,K\}} p(C_k) \prod_{i=1}^{n} p(x_i|C_k)$$

Random Forests

Random forests are an ensemble method for classification by constructing a multitude of decision trees at training and outputting the class membership that is the combination of the classes predicted by the individual trees. It applies the general technique of bootstrap aggregating (or "bagging"). Trees in the ensemble are trained on bootstrap samples of the entire dataset. In addition, a random subset of total N features selected using the Wilcoxon-rank sum test is used at each decision split, rather than using all N. The discriminant function is the number of trees that vote for class 1.

Results

k-nearest neighbors (k = 30)

Given the staircase nature of the ROC curve in Matlab, exact point with sensitivity equal to .8 sometimes cannot be obtained, therefore, threshold t_{80} is set to be the first point on the ROC curve where its sensitivity(t) is greater than .80, and spec(t_{80}) is the corresponding specificity at that threshold. Performances for k-nearest neighbors vary among three cross validation subgroups. Results are summarized in Table 1 below. Testing set 1 has a specificity of about 35% compared to a mere 10% for testing test 2.

Table 1. Performance metrics of k nearest neighbors (cv = 3) at threshold t_{80}

	sensitivity	specificity, spec(t_{80})	auroc
test 1	.8125	.3516	.5117
test 2	.8824	.1000	.4817
test 3	.8125	.2308	.5264
average	.8358	.2275	.5066

Though it is consistent with the fact that k-nearest neighbors generally does not perform well on imbalanced and noisy dataset. The dataset worked on in this project has only 49 progression cases out of 321 patients. For binary classification, if the training set contains mostly class 0, the model will ultimately give preference to class 0, thus getting the less common class 1 wrongly classified.

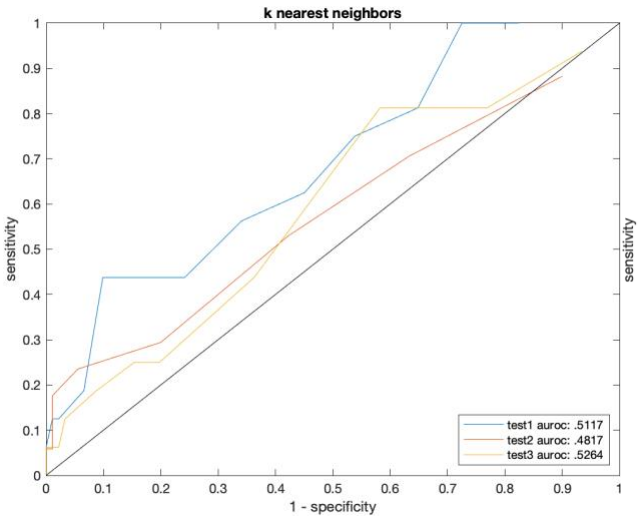

Figure 1. Standard ROC curve for k nearest neighbors

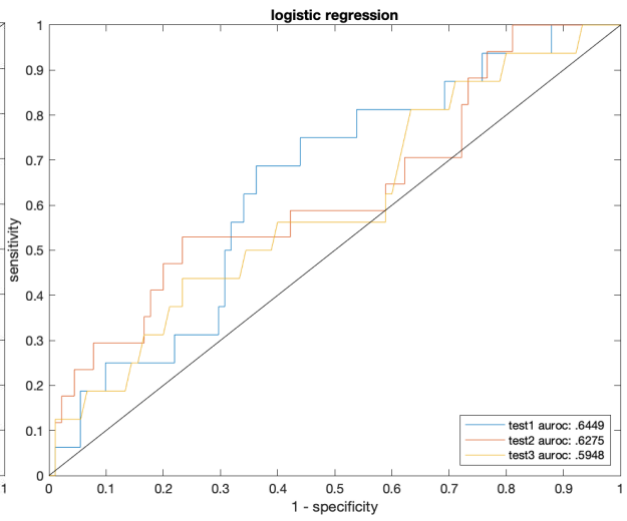

Figure 2. Standard ROC curve for logistic regression

Logistic regression

In contrast, logistic regression generates much more consistent results than k-nearest neighbors does. Results are tabulated in Table 2. Refer to Figure 2 above for the standard ROC curve. On average, it has a spec(t_{80}) around 37% and the area under the ROC curve is .62, both of which are of significant improvement over k nearest neighbors (spec(t_{80}) = 23% and auroc = .51).

Table 2. Performance metrics of logistic regression (cv = 3) at threshold t_{80}

	sensitivity	specificity, spec(t_{80})	auroc
test 1	.8125	.4615	.6449
test 2	.8235	.2778	.6275
test 3	.8125	.3846	.5928
average	.8162	.3746	.6217

Naïve Bayes

Naïve Bayes classifier has slightly lower spec(t_{80}) values compared to logistic regression model. Results are tabulated in Table 3. Refer to Figure 3 below for the standard ROC curve. On average, it has a specificity around 30% and the area under the ROC curve is .61. Partly due to the fact the classifier has a strong independence assumption that all the features (in this case, gene expressions) are mutually independent from each other, conditional on the class membership. This is normally far from true in reality, which might introduce substantial bias into the model that cannot be ignored.

Table 3. Performance metrics of t_{80}Naïve Bayes (cv = 3) at threshold t_{80}

	sensitivity	specificity, spec(t_{80})	auroc
test 1	.8125	.2857	.6442
test 2	.8235	.2333	.5895
test 3	.8125	.3956	.6106
average	.8162	.3049	.6148

Quadratic discriminant analysis (QDA) and support vector machine (SVM) were also attempted. QDA performs the worst among all, achieving results that are barely better than random guessing, and SVM performs roughly the same as logistic regression, as illustrated in Figure 4 below.

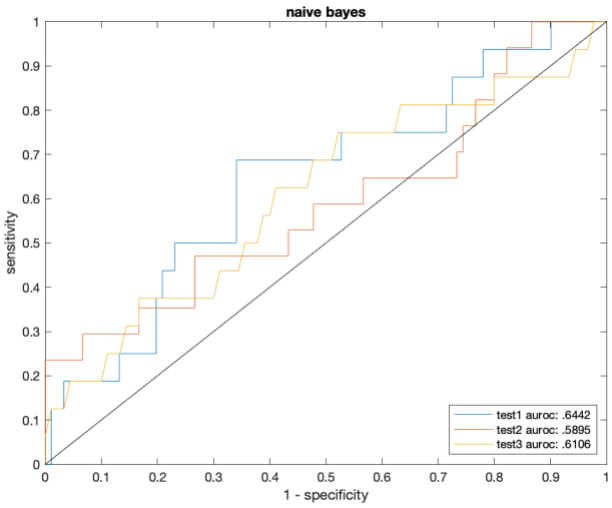

Figure 2. Standard ROC curve for naïve Bayes

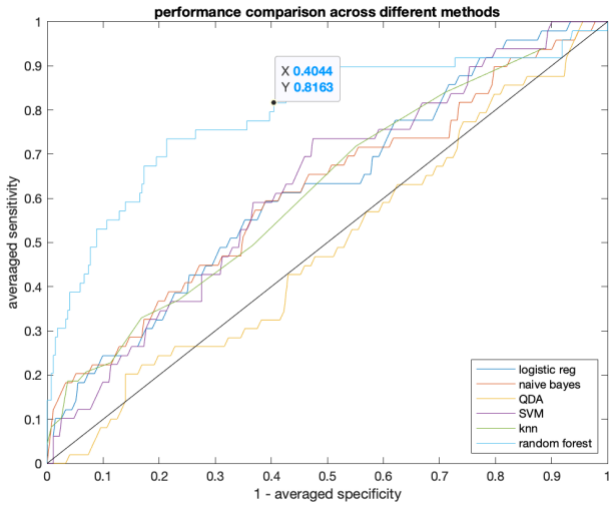

Figure 4. Performance comparison among classifiers

Performance of random forests is significantly better compared to all other classifiers. The point displayed on Figure 4 are the average sensitivity and (1 – specificity) values on threshold t_{80} for random forest. The following section will discuss it in details.

Random Forest

Individual decision trees tend to overfit. Random forests correct for decision trees' habit of overfitting to the training set. Figure 5 (left) below shows how the ensemble error change with accumulation of trees. Out-of-bag error flattens out after about 400 trees. Figure 5 (right) illustrates a measure of importance for each predictor variable.

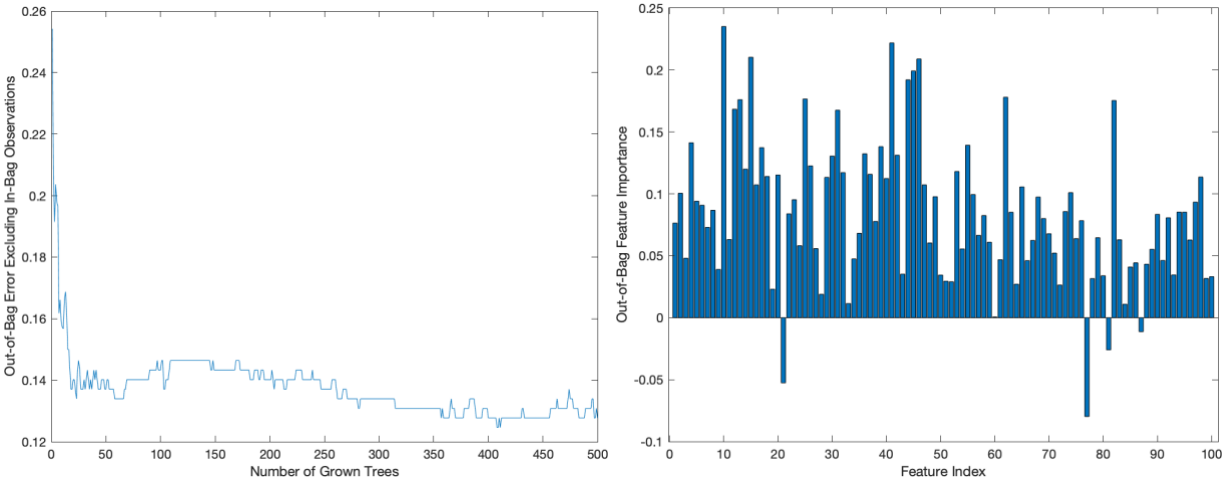

Figure 3. Out-of-Bag Error vs. Accumulation of Trees (left) and Out-of-Bag Feature Importance (right)

For any feature, the measure of importance is the increase in prediction error if the values of that variable are permuted across the out-of-bag observations. This measure is computed for every tree, then averaged over the entire ensemble and divided by the standard deviation over the entire ensemble. Select the features yielding importance greater than .11 (an arbitrary threshold), further narrowing down the number of features used at each decision split.

Use the newly chosen features (29 out of initial 100) to train another random forest (500 trees) classifier, and the new ensemble error is plotted against accumulation of tree in Figure 6. In addition to classification error, the average classification margin is also monitored (as shown in Figure 7). For each observation, the margin is defined as the difference between the score for the true class and the maximal score for other classes predicted by this tree. If the training is successful, a gradual increase in the mean classification margin is expected.

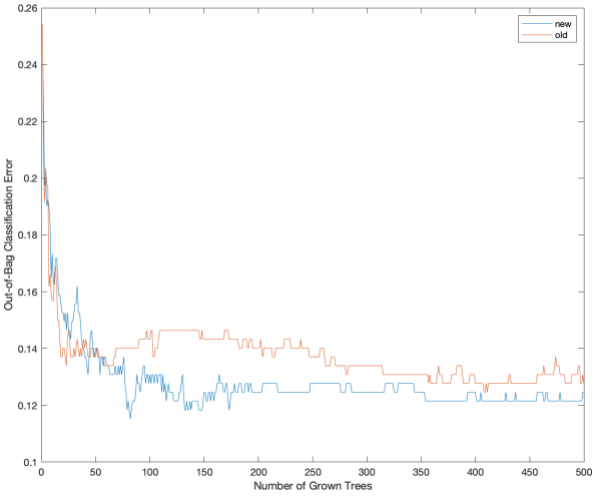

Figure 4. Ensemble Errors b/w Two Random Forests

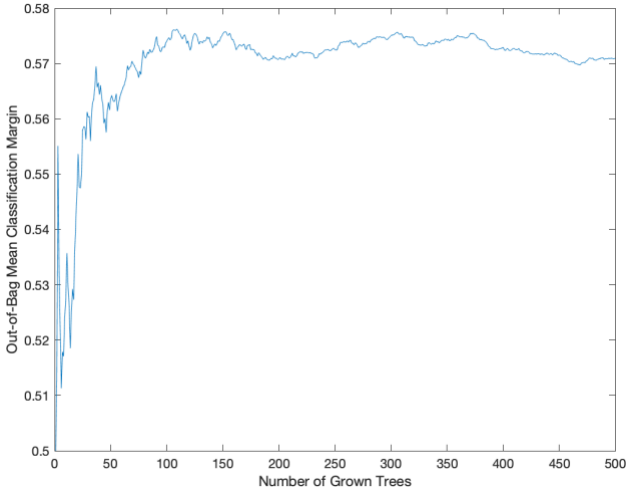

Figure 7. Average Classification Margin for New RF Model

Random forests perfoms significantly better compared to all other classifiers aforementioned, with an auroc of roughly .8 and average specificity approaching 60%. Performance metrics are tabulated in Table 4.

Table 4. Average performance metrics of random forest at threshold t_{80}

	sensitivity	specificity	auroc
average RF	.8163	.5956	.7918

Apply multidimensional scaling. A rather clean boundary separating two classes can be seen in Figure 8 below. A standard ROC curve (average sensitivity vs. 1 – specificity) is plotted for random forests and corresponding performance metrics are tabulated in Table 4 above.

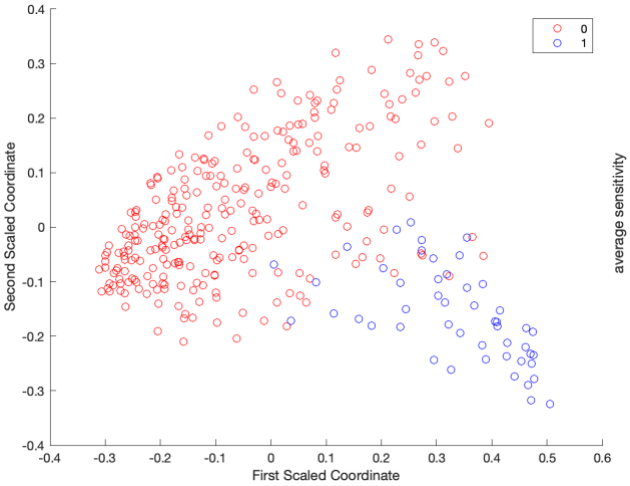

Figure 8. Multidimensional scaling

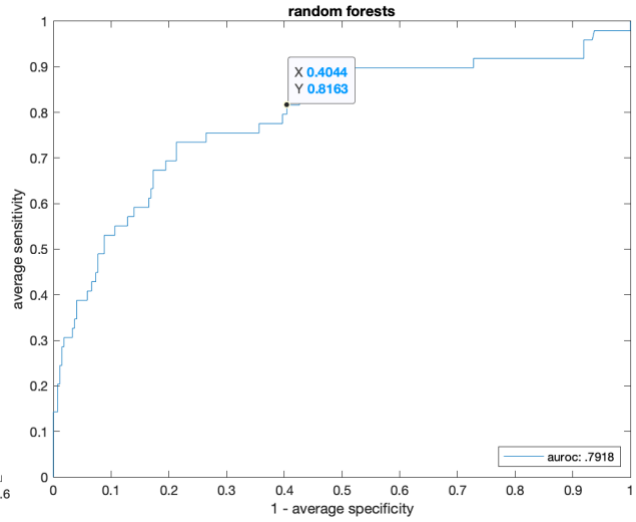

Figure 9. Standard ROC curve for random forests

Conclusions

A handful of classifiers were tested in this project in attempt to learn a predictor for distinguishing between "progression" and "no progression" bladder cancer patients. The goal is to achieve as high specificity as possible while maintain at least 80% sensitivity. On average, most classifiers (naïve Bayes, logistic regression and SVM) have a spec(t_{80}) value in range of 30% to 40%. Due to imbalanced and high dimensional nature of the dataset, k nearest neighbors performs slightly worse compared to the three mentioned before, having a mid 20% spec(t_{80}). QDA does the worst among all, doing slightly better than random guessing. The best performance achieved is random forests, with an average spec(t_{80}) of roughly 60% and auroc of nearly .80, far above all the others. Random forests are a way of averaging multiple deep decision trees, trained on bootstrap samples from the dataset, with the goal of reducing the variance and improving generalization. This comes at the expense of a small increase in the bias (but better bias/variance trade-offs overall) and some loss of interpretability, but generally greatly boosts the performance in the final model.