STAT 432 Final Project

Ruining Tao (rtao6), Elsie Wang (erqianw2), Zhan Shi (zhans5)

Dec 10, 2021

Contents

Project description and summary

The World Health Organization's International Agency for Research on Cancer (IARC) released the latest global cancer data, showing that in terms of cancer distribution types, the number of new cases of breast cancer in 2020 will reach 2.26 million, surpassing lung cancer (2.2 million) for the first time, becoming "the world's number one cancer". Thus, breast cancer classification is a problem worth solving. In the project, we applied what we learned in STAT 432 to analyze a breast cancer dataset.

The dataset `BRCA Multi-Omics (TCGA)` can be found on Kaggle. For the project, we read literature, applied data preprocessing, predicted outcomes `PR.Status` and `histological.type` using five different models. At last, we selected 50 most important input variables to make predictions of all four outcomes.

In data preprocessing, we removed null values, addressed multicollinearity, and transformed and normalized certain variables for modeling. We used SVM and random forest to predict `PR.Status`. For `histological.type` we used logistic regression, and K-means clustering first. Then because of the poor performance of the clustering, we adopted decision trees to do the prediction. Finally, we selected important variables based on literature and random forest and built four cross-validation random forests to obtain the mean AUC value. We concluded that the 50 variables' performance was satisfactory enough, though the AUC was not as high as the best value we obtained in the previous section.

Literature Review

Supervised Risk Predictor of Breast Cancer Based on Intrinsic Subtypes

A rigorous examination of 817 breast tumor samples confirms invasive lobular carcinoma as a molecularly separate illness with discrete genetic traits, offering critical information for patient classification that may allow for better informed clinical follow-up.

We concluded from this paper that three ILC transcriptional subtypes linked with survival differences were identified using proliferation and immune-related markers. Cases with mixed IDC/ILC were molecularly classed as ILC-like or IDC-like, with no actual hybrid traits revealed. This multimodal molecular atlas throws fresh insight on the genetic foundations of ILC and suggests potential therapeutic treatments. This research reveals several genetic mutations that distinguish ILC from IDC, proving at the molecular level that ILC is an unique breast cancer subtype and giving new insights into ILC tumor biology and treatment alternatives.

Comprehensive Molecular Portraits of Invasive Lobular Breast Cancer

Using microarray and quantitative reverse transcriptase polymerase chain reaction data from 189 prototype samples, a 50-gene subtype predictor was constructed. Prognosis was determined using test sets from 761 individuals, and pathologic complete response (pCR) after a taxane plus anthracycline regimen was determined using test sets from 133 patients.

We noticed that this part of Sample Subtype Prediction is particularly valuable for our project. The reliability of categorization across three centroid-based prediction approaches was tested for the 50 gene set: Prediction Analysis of Microarray, a basic nearest centroid, and Classification of Nearest Centroid. The subtype categorization is always determined based on the closest of the five centroids. The final approach comprised of centroids built as stated for the PAM algorithm and distances calculated using Spearman's rank correlation due to its reliability in subtype categorization.

Tumor characteristics and patient outcomes are similar between invasive lobular and mixed invasive ductal/lobular breast cancers but differ from pure invasive ductal breast cancers

Our first relevant paper has a total of 4,336 individuals with IDC, ILC, and mixed breast tumors were detected between 1996 and 2006. The Kaplan-Meier method was used extensively in this paper, and survival curves were constructed using it. Chi-square tests and Fisher's exact tests were used to compare clinical variables. The correlations between patient and tumor variables were summarized using contingency tables and investigated using Fisher's exact test as among three histologic groups. Patients with ILC and mixed breast cancers were more probable as IDC patients to have tumors that were estrogen receptor and progesterone receptor positive ($P < 0.001$ and $P < 0.05$, correspondingly).

After having read, we can conclude the following from the paper: first, despite being identified at lower clinical stages of infection, patients with IDC had the poorest long-term survival; second, individuals with ILC and "mixed" malignancies had a better prognosis than patients with IDC, despite having more advanced cancer. We were also motivated to utilize the log-rank test to estimate P values if necessary.

Infiltrating lobular carcinoma of the breast: tumor characteristics and clinical outcome

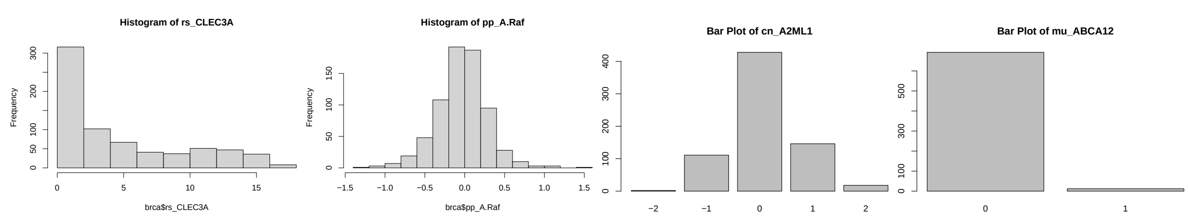
From the second paper, we summarize that these patients do not have improved clinical outcomes as IDC patients when ignoring the fact that ILC has a positive biologic pattern. Consequently, management decisions should be made based on the patient's and tumor's biologic characteristics, that instead of lobular histology. About statistical methods, the clinical and biologic features of lobular and ductal carcinoma were compared by contingency tables, Chi-square tests and Fisher's exact tests, which is similar with the method using in the first paper. To see if ILC was an independent predictive predictor for recurrence and death, researchers used multivariate analysis and Cox regression models. Tumor size, number of affected nodes, age, ER status, PgR status, DNA ploidy, S-phase, and histologic type were all considered in these analyses.

The findings of this huge dataset have shown that ILC and IDC are distinct entities with distinctive clinical histories and biologic features, yet there are no clinically important variations in survival. At the present, both kinds of breast cancer should be treated identically, and histologic subtype (lobular or ductal) should not be regarded a determinant in therapeutic decision-making or an essential prognostic or predictive factor at diagnosis. Emerging technologies such as high throughput genome mapping and microchip cDNA expression arrays may help to uncover molecular distinctions between these different types of breast cancer.

Summary Statistics and Data Preprocessing

Data Overview

The dataset has 705 observations and 1941 features (1936 predictors and 5 outcomes). There are four different kinds of predictors: `rs` (gene expression), `cn` (copy number variations), `mu` (mutations), and `pp` (protein levels). Among them, `rs` and `pp` are continuous variables, and `cn` and `mu` are categorical variables.



Remove Missing Values

According to the instruction, we dropped `vital.status`, and we only considered each response variable as a binary variable. Therefore, we treated the observations that had other outcomes as missing values and removed them from our dataset.

Then the dataset `sub` had 507 observations and 1940 features.

```
dim(sub)
```

```
## [1]  507 1940
```

Deal with Multicollinearity

One of the noticeable characteristics of the data is its high dimensionality. There are 1936 predictors, almost four times as many as there are observations. Therefore, it is essential to check correlation.

Since there are four kinds of predictors, it is unlikely that two variables coming from different kinds would be highly correlated. Also, to reduce the computational cost, we split the data into four subsets: `rs`, `cn`, `mu`, and `pp`, each of which contained only ond kind of predictors.

Then, we created the correlation matrix for each subset, and extracted variables that are highly-correlated with at least one other variable. Take `rs` as an example. The dataframe `idx` stored all matrix indices of highly-correlated variables and the corresponding correlation coefficients. If the i-th variable is highly-correlated with the j-th variable, then we only need one of them. Thus, we removed all variables with indices `i`. For `rs`, 94 predictors were removed. We applied the same process to the other three subsets. In total, 882 predictors were removed. There are 1059 predictors remained.

```
names(idx) = c("i", "j", "corr")
idx[1:3,]
```

```
##   i  j corr
## 2 3  4 0.94
## 3 5 56 0.84
## 4 9 10 0.95
```

```
# remove highly-correlated variables
rmv = unique(idx[,1])
length(rmv)
```

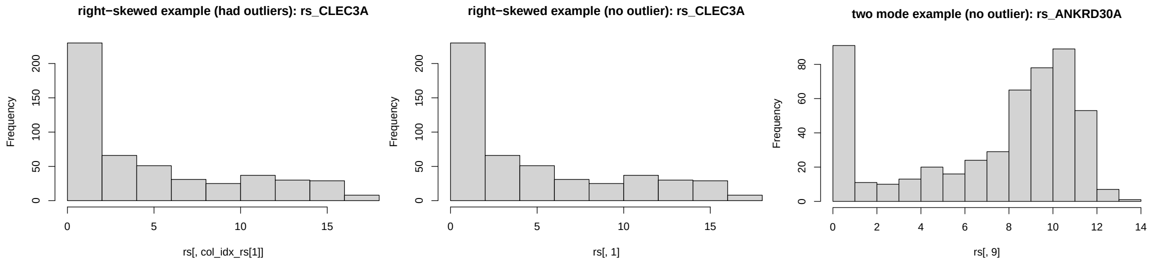
```
## [1] 94
```

```
rs = rs[,-rmv]
```

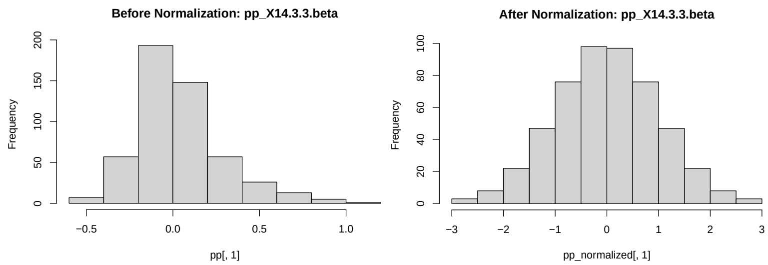
Continuous Variables

As mentioned before, `rs` and `pp` are continuous variables, so we should examine if there are outliers. We first normalized the variable, and stored row and column indices if the data point was three standard deviations away from the mean. For the two subsets, `rs` had 100 outliers, and `pp` had no outlier.

We further looked into `rs` predictors that included outliers, and we found the vast majority of them had a long tail, mostly right and some left. In addition, a number of `rs` predictors that did not contain outliers also had a non-standard distribution. As a result, a log transformation of `rs` predictors would be beneficial.



Unlike `rs`, `pp` variables were distributed quite normally. However, many of the variables would contain outliers without normalization. Therefore, we normalized `pp` variables.



Categorical Variables

All four outcomes were more or less imbalanced, among which `histological.type` was the most imbalanced. Only 10% of the responses were ILC. In some cases, such imbalance would be problematic since models would learn nothing from the minority class. If our data also suffer from problems like this, we should resolve the imbalance by techniques like undersampling. However, if our models perform well enough on current data, no further action needs to be taken.

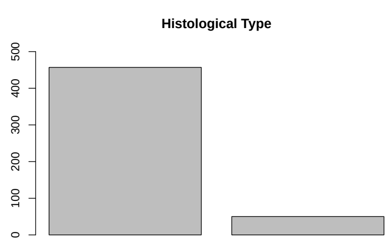
Fortunately, we trained our models first, and found the models obtained high enough accuracy and AUC scores. Therefore, we decided to not address imbalanced outcomes.

```
##
## Negative Positive
##  0.34714  0.65286
```

```
##
##  Negative  Positive
## 0.2445759 0.7554241
```

```
##
##  infiltrating ductal carcinoma infiltrating lobular carcinoma
##                     0.90138067                     0.09861933
```

```
##
##  Negative  Positive
## 0.8382643 0.1617357
```



Modeling `PR Status`

Before modeling, we split the train and test datasets. We used 25% of the samples (126) for testing and 75% of the samples (381) for training.

Support Vector Machine (SVM)

The goal of the project was to make classifications. Plus, we needed to alleviate "the curse of dimensionality". Therefore, we should choose classification models that perform well on high-dimensional data. Support vector machines are famous for its capability in high-dimensional spaces, so we first fitted a basic linear SVM, with the default `cost = 1` to see how it worked.

As the confusion matrix showed, the in-sample accuracy was 1.0, which implied that we might prefer the linear kernel to the radial kernel.

```
##            actual
## predicted  Negative Positive
##   Negative      133        0
##   Positive        0      248
```

Then we constructed two grids of tuning parameters for both linear and radial kernels, and we used 5-fold cross-validation to tune the parameters and to determine which kernel was better.

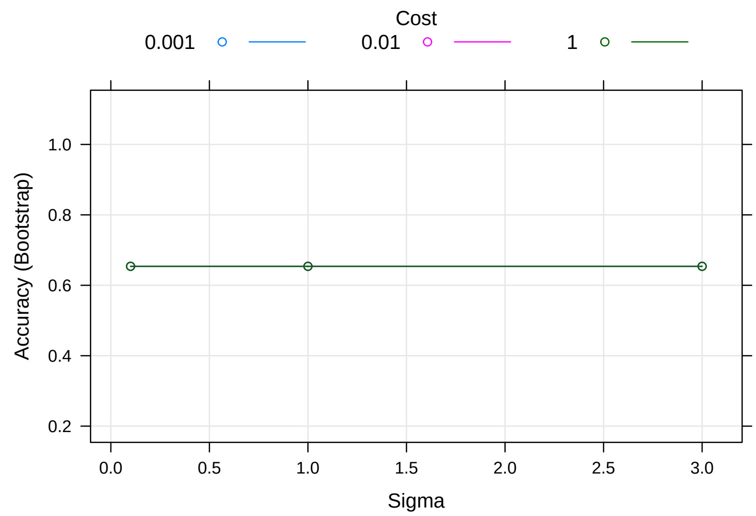
For the linear kernel, the best `C` was 0.001 with an in-sample accuracy of 0.8324.

```
##       C
## 1 0.001
```

```
##       C  Accuracy     Kappa AccuracySD    KappaSD
## 1 0.001 0.8336029 0.6058770 0.02730027 0.06541415
## 2 0.010 0.7999388 0.5464954 0.02526173 0.06257166
## 3 0.050 0.8014261 0.5503720 0.02442031 0.06150767
## 4 0.100 0.8014261 0.5503720 0.02442031 0.06150767
## 5 0.500 0.8014261 0.5503720 0.02442031 0.06150767
## 6 1.000 0.8014261 0.5503720 0.02442031 0.06150767
```

For the radial kernel, the best `C` was 0.001 and the best `sigma` was 3. However, the figure showed that the radial SVM fitted poorly, since the accuracies remained the same and were merely 0.6677. The fact verified our hypothesis that a linear kernel would work better. Thus, we picked the linear SVM with `C` equals 0.001 to make classifications.

```
##   sigma     C
## 3     3 0.001
```



We made predictions for the test data, and printed the confusion table below. The accuracy was 0.9048. Thus, the linear SVM performed quite well.

```
##            actual
## predicted  Negative Positive
##   Negative       33        2
##   Positive       10       81
```

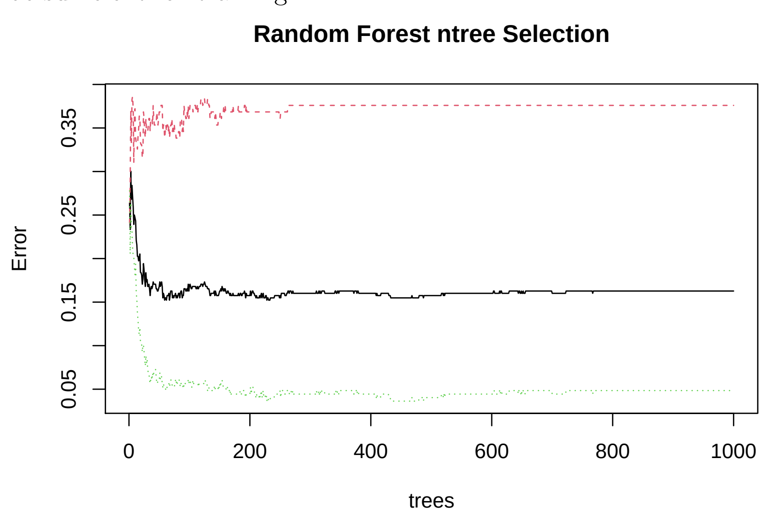
```
## [1] 0.9047619
```

Random Forest

Random forests are another classification model that is less vulnerable to "the curse of dimensionality". Since there were many parameters that needed to be tuned, we again utilized `caret` package to cross validate the model.

Before applying cross-validation, we should first determine `num.trees`, because it could not be included in the grid of parameters. Therefore, we first fitted a random forest using the `randomForest()` method with `ntree = 1000`, and plotted the error against trees.

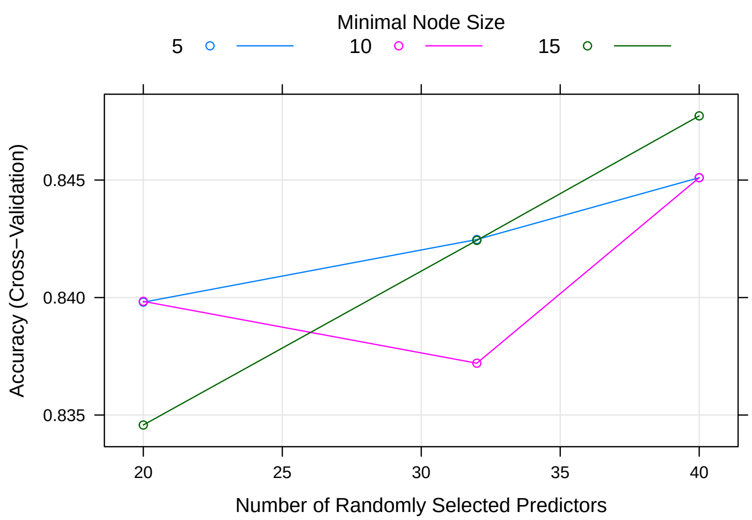
As the plot demonstrated, the error stopped decreasing after the number of trees reached around 500. Thus, `ntree = 500` should be sufficient for training.



In a 5-fold cross-validation, we tuned `mtry` and `min.node.size`. The output showed that the best parameters were `mtry = 40` and `min.node.size = 15` when `num.trees = 500` according to the test accuracy.

```
##   mtry splitrule min.node.size
## 9   40      gini            15
```

```
##   mtry splitrule min.node.size  Accuracy     Kappa AccuracySD    KappaSD
## 1   20      gini             5 0.8398027 0.6199996 0.01816714 0.04744967
## 2   20      gini            10 0.8398378 0.6191721 0.01181501 0.02631224
## 3   20      gini            15 0.8345746 0.6097325 0.01575637 0.03413845
## 4   32      gini             5 0.8424694 0.6272972 0.01375275 0.03444365
## 5   32      gini            10 0.8372062 0.6150392 0.01262152 0.02909492
## 6   32      gini            15 0.8424343 0.6268844 0.01948896 0.05058240
## 7   40      gini             5 0.8451009 0.6341821 0.01487847 0.03698468
## 8   40      gini            10 0.8451009 0.6356210 0.02177987 0.05139481
## 9   40      gini            15 0.8477325 0.6408985 0.01796807 0.04555969
```



The confusion matrix and the highest test accuracy, 0.9048, were shown here.

```
##           actual
## predicted  Negative Positive
##   Negative       33        2
##   Positive       10       81
```

```
## [1] 0.9047619
```

Modeling `Histological Type`

To establish the Modeling for `histological.type`, we observed that the response variable `Histological Type` had only two levels "infiltrating lobular carcinoma" and "infiltrating ductal carcinoma". Thus, logistic model became a good choice. To peform a logistic model, we first split the data into train and test data. 25% of the samples were randomly selected to be the test data.

Logistic Regression

When we fitted a logistic model, `R` outputed a warning saying "algorithm did not converge", which means that a complete sepration occured. As the confusion matrix shows below, our model has 100% accuracy and

AUC with 1.

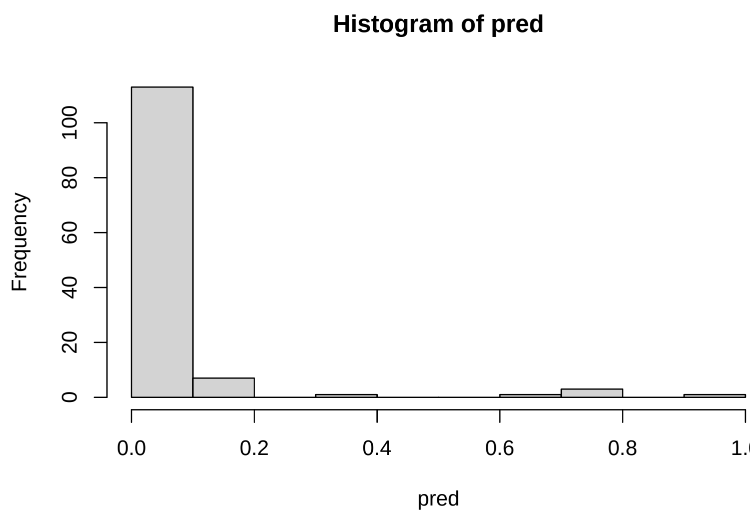
```
## Warning: glm.fit: algorithm did not converge
```

```
## AUC = 1
```

```
##
##           0   1
##   FALSE 457   0
##   TRUE    0  50
```

Logistic Regression with Ridge Penalty and Cross Validation

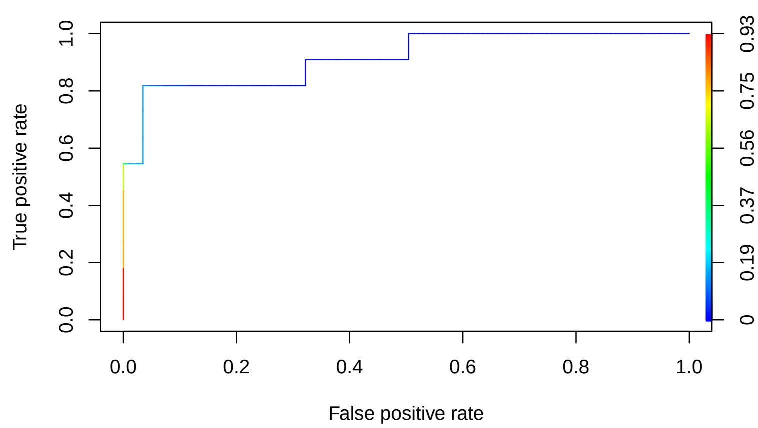
Thus, we decided to use penalized regression and cross validation. We performed a logistic model with ridge penalty and 10-fold cross validation. Since the logistic regression did not return values of 0 and 1, we determined the cut-off value based on the distribution of the predict result. Therefore, we plotted the histogram of the prediction result.



Based on the graph shows above, we set the cut-off value to be 0.5. The the prediction result gave 121 `FALSE` and 5 `TRUE`.

```
##
## FALSE  TRUE
##   121     5
```

The AUC and ROC curve were shown below.



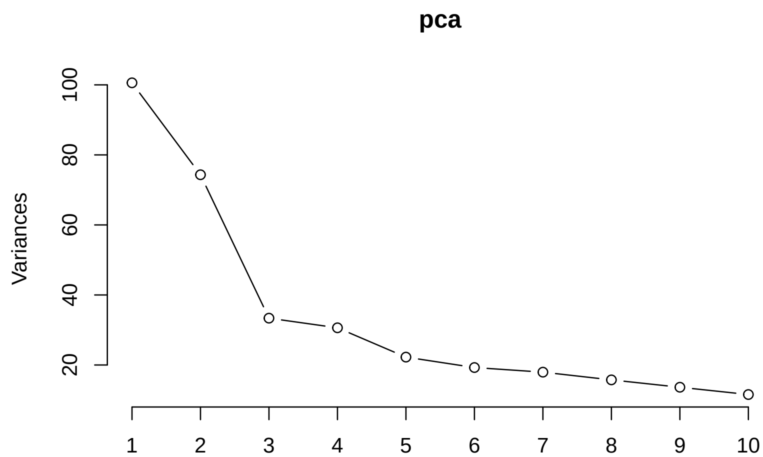
```
## AUC = 0.915415
```

```
##        ytest
##           0   1
##   FALSE 115   6
##   TRUE    0   5
```

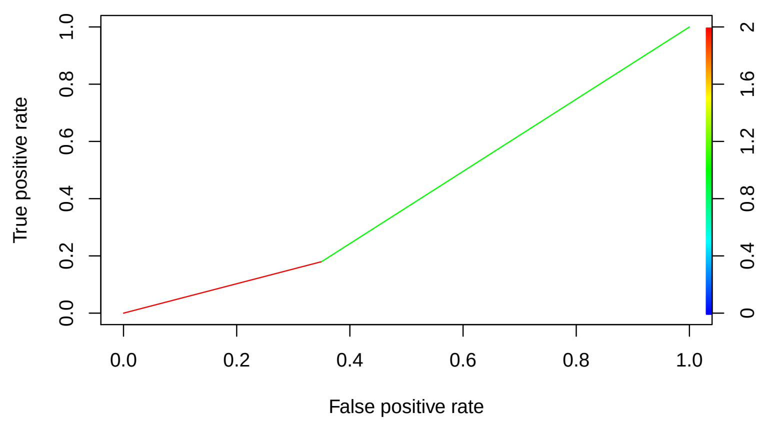
We obtained an AUC of 0.915415, which was a good result.

K mean Clustering with Principal Components Analysis

We experimented with k-means clustering to see how it worked. Since clustering algorithms could not handle high dimensional data well, we lowered the dimensionality by applying PCA.



It is essential to scale the data before PCA to avoid variables with large variances dominanting the first principal component. Then we should determine the number of components which can represent our data. By using the elbow method and based one the above graph, we picked the first three components.

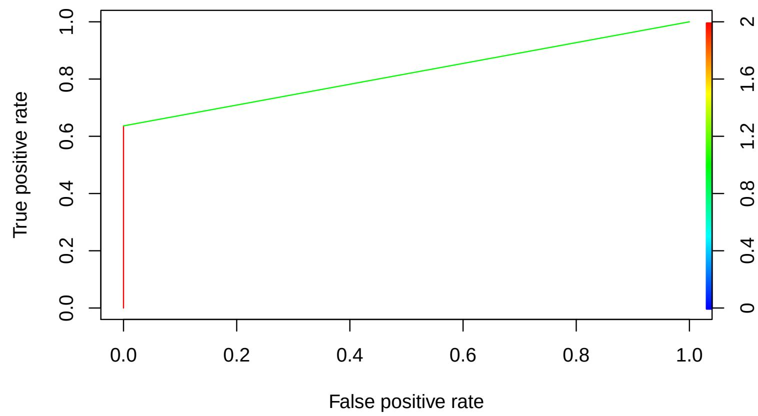



```
## [1] 0.4149453
```

However, K-means did not give us a good result. The value of AUC is 0.4149453, which was worse than a random predicion. Therefore, K-means was not suitable for our data, and we wanted to find another model that performed better.

Classficiation Tree

Like random forests, decision trees also suffers less severely from high dimensionality. Thus, we chose classification tree as our model, and used 10-fold cross-validation to tune parameters `control`.




```
## [1] 0.8181818
```

The AUC was 0.8181818, which was a good result.

Variable Selection for All Outcomes

We used both literature and random forest to determine the 50 variables.

Literature

Based on "Comprehensive Molecular Portraits of Invasive Lobular Breast Cancer", we selected 7 variables: `pp_PTEN`, `mu_PTEN`, `mu_TBX3`, `cn_FOXA1`, `mu_FOXA1`, `rs_FOXA1`, and `pp_E.Cadherin`.

Fit Random Forest to Select Variables

Then, we fitted Random Forest using `randomForest()` to select the other 43 important variables and make predictions with the total 50 variables of all outcomes.

First, we fitted a random forest with best parameters chosen by cross-validation previously. Then we sorted the importances of variables and picked the most important 43 variables. Combined with the 7 variables from the paper, we constructed our input matrix.

Generate Fold ID

Using the provided code, we generate the fold ID for all 705 observations. Since we removed null values in data preprocessing, we would use only the first 507 IDs.

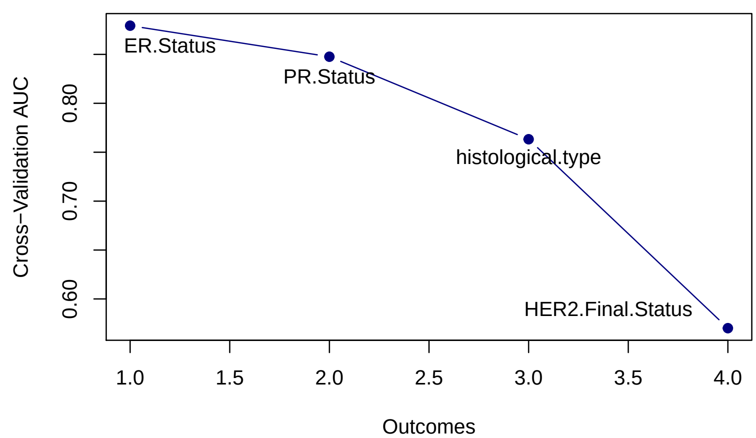
Predict All Four Outcomes

To perform 3-fold cross-validation, we wrote a for loop. For each ID $\in \{1, 2, 3\}$, we created test indices and split train and test datasets. We stored the three AUC scores in a vector and took its mean after the for loop ended. Below is an example for `ER.Status`. The same process was applied for the other three outcomes.

```r
#fit for `ER.Status`
AUC = c()
set.seed(651978735)
for (i in 1:3) {
  tst_idx = which(fold_id == i)
  Xtrain = sub4[-tst_idx, 1:50]
  Xtest = sub4[tst_idx, 1:50]
  ytrain = sub4$ER.Status[-tst_idx]
  ytest = sub4$ER.Status[tst_idx]
  rf.fit = randomForest(Xtrain, ytrain,
                      ntree=500,
                      mtry=40,
                      nodesize=15,
                      samplesize=400,
                      importance=TRUE)
  pred = predict(rf.fit, Xtest)
  roc = prediction(as.numeric(pred), ytest)
  auc = performance(roc, measure = "auc")@y.values[[1]]
  AUC = c(AUC, auc)
}
mean_1=mean(AUC)
mean_1
```

```
## [1] 0.8793647
```

As shown in the graph belwo. The cross-validation AUC of `ER.Status`. `PR.Status`, `histological.type` were satisfactory. However, CV AUC of `HER2.Final.Status` was the lowest (0.5584). Since we did not model for `HER2.Final.Status` in previous prediction, it is possible that the model did not work well on the outcome.




```r
# average the cross-validated AUC of all four outcomes
mean(c(mean_1,mean_2,mean_3,mean_4))
```

```
## [1] 0.7650967
```

At last, we took the average of the four cross-validation AUC. The final AUC value we obtained was 0.7650967. Based on our experience and conclusion drawn from relevent papers and analysis of our upper results, a small set of biomarkers could predict four outcomes well, but the performance was not as good as the prediction with all variables.