Received: Added at production | Revised: Added at production | Accepted: Added at production

DOI: xxx/xxxx

STAT 432 FINAL PROJECT

Applications of Statistical Learning in Medicine; The Classification of Breast Cancer Histological Types

Leo Liu[1] | Lucas Nelson[2] | Paul Wang[3]

[1]University of Illinois at Urbana-Champaign, Senior in Department of Statistics (B.S. Statistics)
[2]University of Illinois at Urbana-Champaign, Senior in Department of Statistics (B.S. Statistics), Department of Economics (B.S. Econometrics and Quantitative Economics)
[3]University of Illinois at Urbana-Champaign, Senior in Department of Statistics (B.S. Statistics)

In this paper, we construct a collection of classification models that make use of various subsets of 1837 genes across four omics (gene expressions, copy number variations, mutations, and protein levels) to accurately predict the presence of cancer subtypes in 705 patients. Referencing established papers published in the field, we integrated researched methods into the array of classification models we built, testing and comparing their performance across previous findings to better suit our models. In building these models, we focused on models whose methods were similar but whose unique properties varied significantly, including penalized general linear models (e.g. logistic regression), supervised learning algorithms (k-nearest neighbors), and classification algorithms consisting of decision trees (e.g. random forests). Following the generation of our models, we derived results confirming those found across multiple papers, further solidifying the current body of research that exists in the medical field of breast cancer: classification algorithms with greater control over tuning parameters show GATA3, CDH1, and PTEN are significant determinants of breast cancer and lead to greater accuracy in predicting cancer-related responses.

KEYWORDS:
Breast cancer; Statistical learning; Classification algorithms;

1 | INTRODUCTION

Breast cancer is one of the most common cancers worldwide, estimated to affect upwards of 1,500,000 individuals annually, 500,000 of which will result in death. Although a cure for the disease has not yet been discovered, an overwhelming amount of research has been performed to understand more about the array of causes and treatments for impacted patients. One such effort that breast cancer research heavily focuses on is the ability to classify a patient's breast cancer status given an array of information related to a given patient (demographics, genetic sequences, etc.) with the purpose of providing the best possible treatment. Given our current knowledge of cancer cell biology, histological types are classified depending on the presence of specific receptors (estrogen (ER), progesterone (PR), and human epidermal growth factor (HER2)). Following our review of established literature in this field, we focus on constructing a number of models that make use of 1837 genetic expressions across 705 patients' to accurately predict the presence of specific receptors and the presence of cancer.

2 | LITERATURE REVIEW

Of the studies we referred to in preparation for this project, we noticed some commonalities among the results and some advancements in the statistical learning techniques used to derive those results. Although nothing new, supervised and unsupervised clustering algorithms continue to populate various papers' methods sections, specifically classification algorithms. These methods assist analysts in locating and highlighting the significance of patterns that exist within the data using majority voting as a way of grouping the data. Considering not only the topic of our data but also the vast size of the data presented to us for this project, clustering algorithms presented themselves as a reasonably sound approach for analyzing the inherent relationships among the covariates as they communally explain the response variables.

2.1 | Comprehensive molecular portraits of human breast tumours (Cancer Genome Atlas Network)

Published in 2012 by The Cancer Genome Atlas Network, this paper was the first TCGA breast to report the impact specific genes have in classifying the status of breast cancer in a patient. Using "supervised clustering of mRNA expression data," the group responsible for this study provided two subsets of genes that provided them with significant results compared to other genetic subsets they tested. The first of two groups identifies genes "previously implicated in breast cancer", consisting of genes like TP53, GATA3, CDH1, and more. The second group the research deemed significant was of "novel significantly mutated genes", including TBX3, PIK3R1, PTPRD, and others. Given the explanatory power these genes showed in their study, we integrated these two subsets - as well as the union of these two subsets - into our `PR.Status` models in the section below. Although it will be mentioned again there, we found results that agreed with results from another paper we were required to read (*"Comprehensive Molecular Portraits of Invasive Lobular Breast Cancer"*, G. Ciriello et. al.). Although Ciriello's paper is heavily influenced by this one here, further proof of the results using statistical learning shows the importance of the genes we've selected.

2.2 | Differences between invasive lobular and invasive ductal carcinoma of the breast: results and therapeutic implications (Romualdo Barroso-Sousa and Otto Metzger-Filho)

In the paper "Differences between invasive lobular and invasive ductal carcinoma of the breast: results and therapeutic implications" by Romualdo Barroso-Sousa and Otto Metzger-Filho, the authors compared the differences between invasive lobular carcinoma (ILC) and invasive ductal carcinoma (IDC) in their responsive to various treatment and their genomic profiles. It is learned that ILC is the second most common histological type of breast cancer, and makes up approximately 10% of the cases. Some key features of ILC described were: lack of E-cadherin protein expression, and this is driven by alterations that target its CDH1 gene. This helped confirm the result of variable selection from my LASSO and tree model when modeling histological types.

The paper mentions that ILC typically displays positive estrogen receptor (ER) expression and absence of HER2 protein over-expression. and low Ki-67 expression, and that those features are associated with good prognosis. This really helped me understand some of the terms and variables encountered in this project. Furthermore, ILCs typically shows more activation of AKT, and mutations in TBX3 and FOXA1 genes compared to IDC. The paper investigated different treatment approaches, such as chemotherapy and uses of drugs like trastuzumab for both breast cancer types. For instance, it is thought that since most ILC patients have ER positive tumors, they are unlikely to benefit much from adjuvant or neoadjuvant chemotherapy. The paper discussed various findings of the experiments and limitations. Although medical papers can be difficult to read for people outside the field, the paper, along with the Stat432 project really showed the benefits that data science can bring in developing new treatment for diseases.

3 | SUMMARY STATISTICS AND DATA PROCESSING

The BRCA Multi-Omics (TCGA) dataset contains the genetic information and cancer status of 705 individual patients all diagnosed with breast cancer, either infultrating lobular carcinoma or infultrating ductal carcinoma. The genetic information is stored across four subsets in the data, referred to as omics (gene expressions (`rs`), copy number variations (`cn`), mutations (`mu`), and protein levels (`pp`)), that total to 1936 columns of the 1941 in the dataset. These columns serve the purpose of being the origin

of each model's explanatory variabels, although some models will not make use of all variables. Across all omics groups, the only transformation we made to the raw dataset was the deletion of 99 columns in the `cn` subgroup, as these 99 columns were duplicates of at least one other column in the `cn` subgroup. This leaves the final number of explanatory variables as 1837.

	Observations	Covariates	Deletions
Gene Expressions (rs)	705	604	0
Copy Numer Variations (cn)	705	761	99
Mutations (mu)	705	249	0
Protein Levels (pp	705	223	0

The remaining five covariates from the raw dataset are response variables indicating the presence of receptors (`ER.Status`, `PR.Status`, `HER2.Final.Status`), the presence of cancer (`histological.type`), and - for the sake of repetition - the presence of life (`vital.status`); however, `vital.status` is not considered in this project. Receptors play an important role in breast cancer by controlling the growth of proteins (like estrogen) within the lobular or ductal carcinoma of a breast, which factors into the treatment prescribed to a breast cancer patient. These four response variables will be treated as binary categorical variables, with "Positive" indicating the presence of a receptor and "Negative" indicates the lack of a receptor. (For the purpose of the visualization below, given `histological.type`, "Positive" indicates "infultrating lobular carcinoma" and "Negative" indicates "infultrating ductal carcinoma").

	Positive	Negative	Missing
Estrogen	414	135	156
Progesterone	353	193	159
HER2 Protein	86	457	162
Histological	131	574	0

Before inspecting our data more closely, we made a strong decision not to transform any of the 1837 remaining covariates for any of the 705 observations. We felt that the distributions were unique to the data and would not have served a greater purpose had it been transformed. Furthermore, we did not feel well-suited to defend the deletion of one gene over another given our lack of prior knowledge in the field, making the case for deleting any columns, if there were any we felt compelled to delete, difficult to support. When addressing the other dimension of the dataset, we did not alter any of the 705 observed observations. Although we found two rows to be identical across all `cn` and `mu` columns (rows 418 and 670), this information was more of a quirk of the dataset than a severe problem that would impact our modeling abilities.

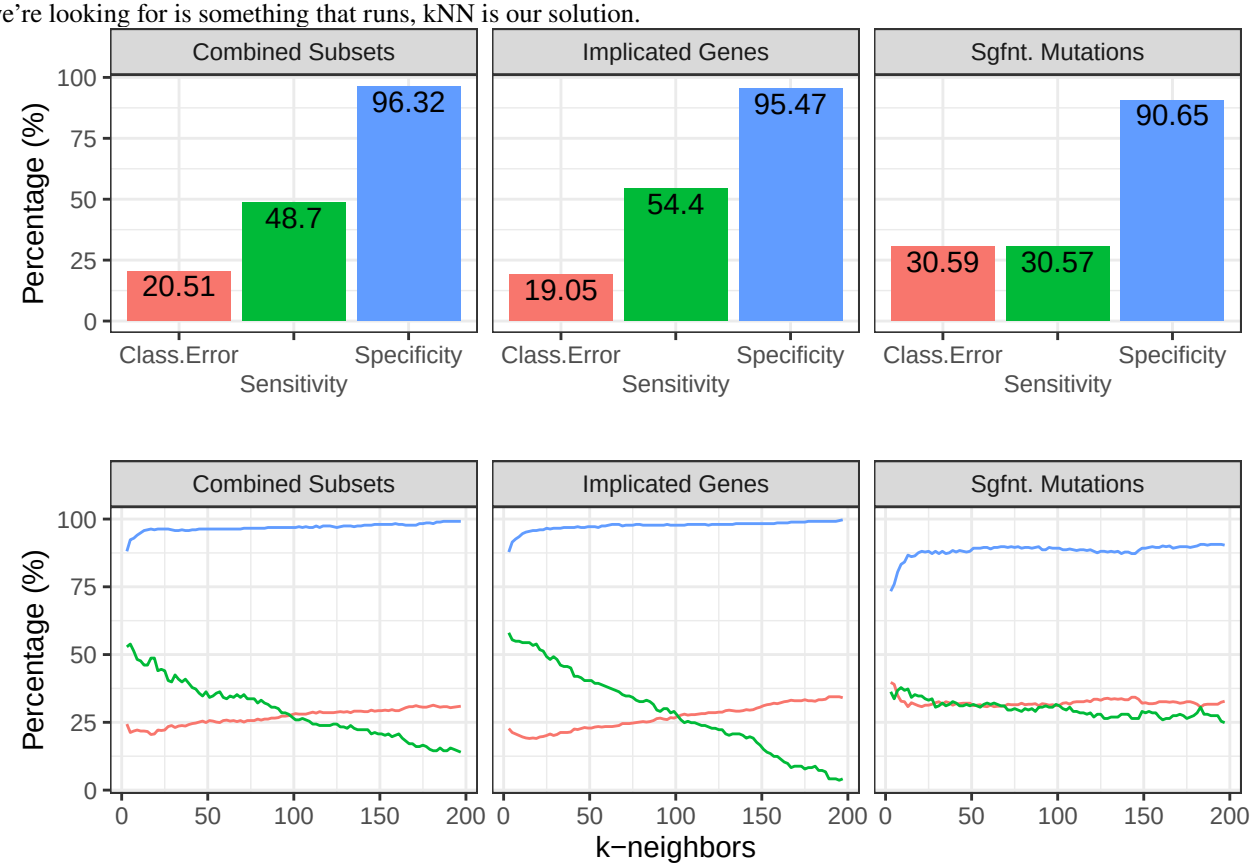
All this information together left us with a final data frame of the original 705 observations across 1837 columns, where the 99 deleted columns were duplicates of one of the remaining `cn` columns. This data served as the core of the analyses that follow below, ranging from models that make use of the entire dataset to models that are limited to using just 50 of the possibly 1833 explanatory variables to explain the remaining four response variables.

4 | MODELING `PR.STATUS`

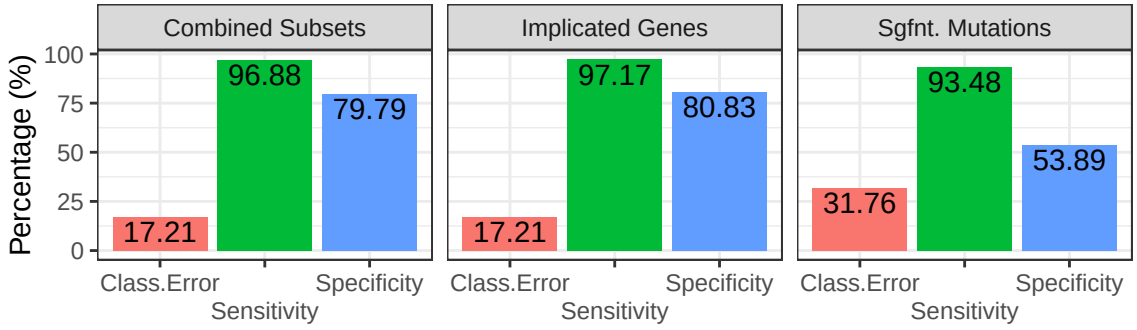
Progesterone (or PR) is a receptor commonly found in breast cancer cells that affects the rate at which progesterone proteins develop and grow in the breast. As with other receptors, diagnosis and treatment of a patient's breast cancer is largely determined by the presence and severity of receptors like the progesterone receptor, making classification algorithms essential for assisting a doctor's procedures and patient's well-being. In this section, we consider two models that attempt to accurately classify whether or not a given patient's cancer cells contain a progesterone receptor (PR+) or does not contain a progesterone receptor (PR-). The first model is a simple k-nearest neighbors (kNN) algorithm. As a fundamental classification algorithm, kNN is favorable for its simplicity, both in its method of classification and, hence, its interpretability. The second model is constructed using a random forest, another statistical learning method that makes use of majority voting (like kNN) which an array of customizable parameters to better tune the selection scheme in the background.

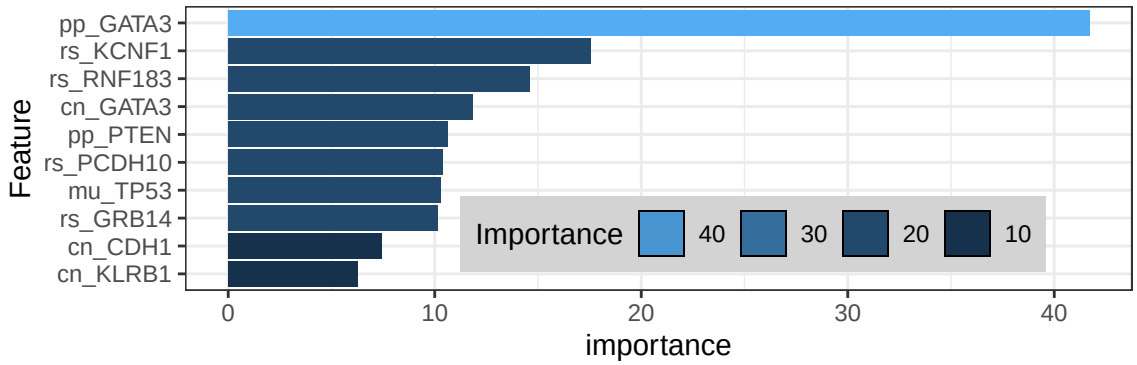
Rather than using a random subset of the explanatory variables (or, better yet, the entire set of explanatory variables), we identified an array of genes mentioned in the paper we reviewing by The Cancer Genome Atlas Network; specifically, two array of genes: one for "previous indicators" of breast cancer, the other for "signficantly mutated" genes that correlated with breast cancer status. The first array contains the genes PIK3CA, PTEN, AKT1, TP53, GATA3, CDH1, RB1, MLL3, MAP3K1 and CDKN1B, and was stored in the data frame `id.rs`. The second array contains the genes TBX3, RUNX1, CBFB, AFF2, PIK3R1, PTPN22, PTPRD, NF1, SF3B1 and CCND3, and was stored in the data frame `id.mu`. Finally, we aggregated the two data frames into one, referred to by `id.pr` in this section. These three data frames were then used to fit their own cross-validated kNN models and random forest models, and their performances are discussed below.

As a common pitstop for most statistical learning students, kNN offers a simple, yet effective, way of locating clusters in your data. Using an array of odd-numbered k-values (odd to allow for majority voting to always exist), the `FNN` package applies a fast alternative to the standard kNN approach in `R`. Taking this methodology, we pass the three datasets mentioned above and discover two fascinating details about the kNN algorithm. First, kNN is a cheap, yet efficient, solution for classification and regression tasks. With as many as 19 neighbors, kNN can predict the `PR.Status` of a patient with about 80% accuracy. If what we're looking for is something that runs, kNN is our solution.



However, kNN suffers a pitfall that makes it impractical in the grand scheme of statistical learning: kNN suffers in high-dimensions. In other words, as the data becomes more complex, kNN results suffer (as shown by the increase in `Class.Error` and decrease in `Sensitivity` from the graphs above). This phenomenon is known as the curse of dimensionality, and considering our data set contains 1837, we will need a more robust solution if we want a more accurate model. Here is where we introduce random forests, a behemoth of a classification algorithm that can find underlying patterns in virtually any dataset, no matter the complexity. Additionally, random forests allow for more tuning than kNN to control the range of results you could possibly receive. (Specific tuning can be seen in `project.functions.glossary.R` file.)





After successfully building these models, we can view their output in a similar to kNN. From this output, we almost immediately notice the decrease in `Class.Error`, or classification error, that kNN does not enjoy the benefits of boasting. In the application of the dataset, we are able to more correctly diagnose the presence of progesterone in patients and, as a result, address treatment in a more appropriate manner. Additionally, random forests provide a metric known as feature importance, or, in other words, how important a specific gene is in relation to the entire model. As we see here, the GATA3 gene is significantly important in predicting progesterone status, and this agrees with findings claiming "GATA3 [is a] key regulator" of PR activity. Although random forests outperform their kNN counterparts, it does come at the cost of computational time. Further consideration may weight these two algorithms accordingly, but the extra computation time could save lives if this model were guiding doctors' decisions.

5 | MODELING `HISTOLOGICAL.TYPE`

The data contains now rows with NA values. However, there are groups of columns within the "rs" (gene expression) and "cn" copy number columns that have correlation of 1. The following code reads the data and identifies correlations among columns using the cor() function. The groups with perfect correlations are identified and from each group, identical columns are deleted so that there is only one column left.

Since the goal of this section is to model histological.type, columns for other responses ('PR.Status','ER.Status','HER2.Final.Status') are removed.

For using glmnet(), categorical columns need to be encoded using dummy variables via the model.matrix() function. In the following code, I identified the categorical columns (those that starts with "cn" and "mu") and their column numbers, and converted them to factors. Then I used to columns to build the model matrix (for categorical variables only).

Then, I created a dataframe for the numeric columns, removed the response column, and column binded it to the model matrix for the categorical variables.

For the response vector, I encoded "infiltrating lobular carcinoma" as 1, otherwise it is 0.

For the training/testing split, I want to get 80% of both response types for training data, and the remaining for testing data. The following code achieves this.

I used number of fold = 4, family = "binomial", and type.measure = "auc" for my cv.glmnet() call. The best lambda is then used to retrain the model.

The nonzero coefficients are shown below.

```
##  [1] "(Intercept)"     "rs_GP2"          "rs_CIDEA"        "rs_MMP1"
##  [5] "rs_FAM3B"        "rs_ALDH1L1"      "rs_LOC100271831" "rs_TMPRSS3"
##  [9] "rs_LOC642587"    "rs_TSIX"         "rs_PLCH1"        "rs_HPX"
## [13] "rs_DLX2"         "rs_HP"           "rs_ANKRD43"      "rs_PITX1"
## [17] "rs_OSR1"         "rs_C2orf82"      "rs_CST2"         "rs_TNNT3"
## [21] "rs_CADM3"        "pp_AR"           "pp_Dvl3"         "pp_E.Cadherin"
## [25] "pp_ERCC1"        "pp_Fibronectin"  "pp_Jak2"         "pp_TFRC"
## [29] "pp_beta.Catenin" "pp_eEF2"         "cn_ROBO20"       "cn_EREG1"
## [33] "cn_CCDC1581"     "cn_PART11"       "cn_ABCB50"       "cn_FAM181B.1"
## [37] "cn_A2ML1.1"      "cn_GJB22"        "cn_SEZ6L22"      "cn_NOL42"
## [41] "cn_LRG1.1"       "cn_PLIN4.1"      "cn_PLIN5.1"      "cn_CPAMD8.1"
## [45] "mu_CDH11"        "mu_APOBR1"       "mu_TP531"        "mu_CROCC1"
## [49] "mu_GON4L1"
```

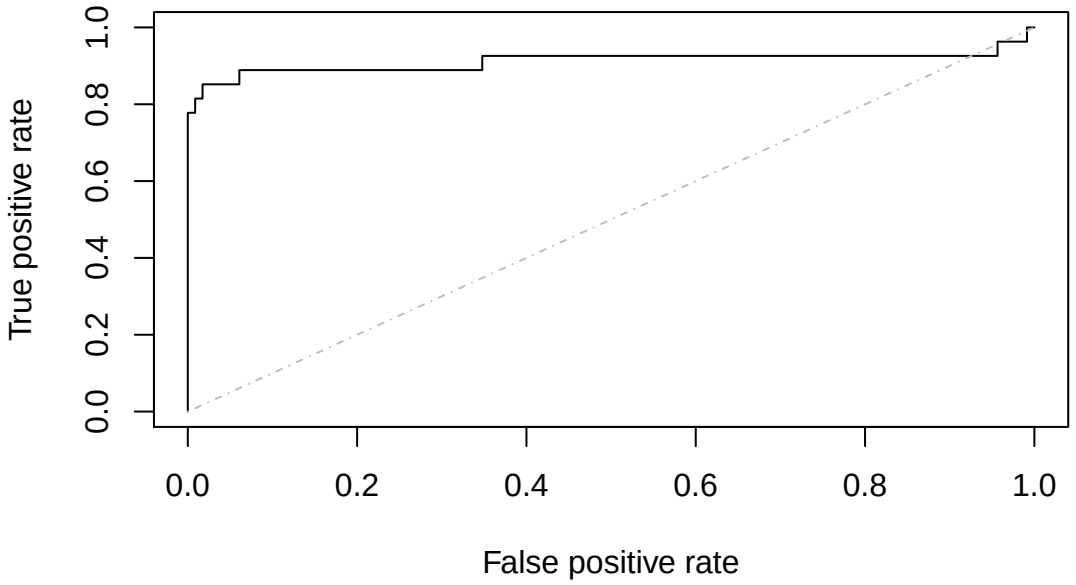
From the non-zero coefficients above we see that pp_E.Cadherin and mu_CDH11 are in the list. This corresponds to information in the paper "Differences between invasive lobular and invasive ductal carcinoma of the breast: results and therapeutic implications", where it was mentioned that "approximately 90% of ILCs lack E-cadherin protein expression - and this feature has thus become a hallmark of ILC.

It was also mentioned that "ILCs exhibit a high frequency of CHD1 mutations.

If we predict each observation as "infiltrating lobular carcinoma" if the output probability is above 0.5, mean accuracy is decent at 93.6%.

```
## [1] 0.9366197
```

The ROC curve looks great.



AUC is 0.9117, which is good.

```
## [1] "AUC: 0.911755233494363"
```

5.1 | Decision Tree

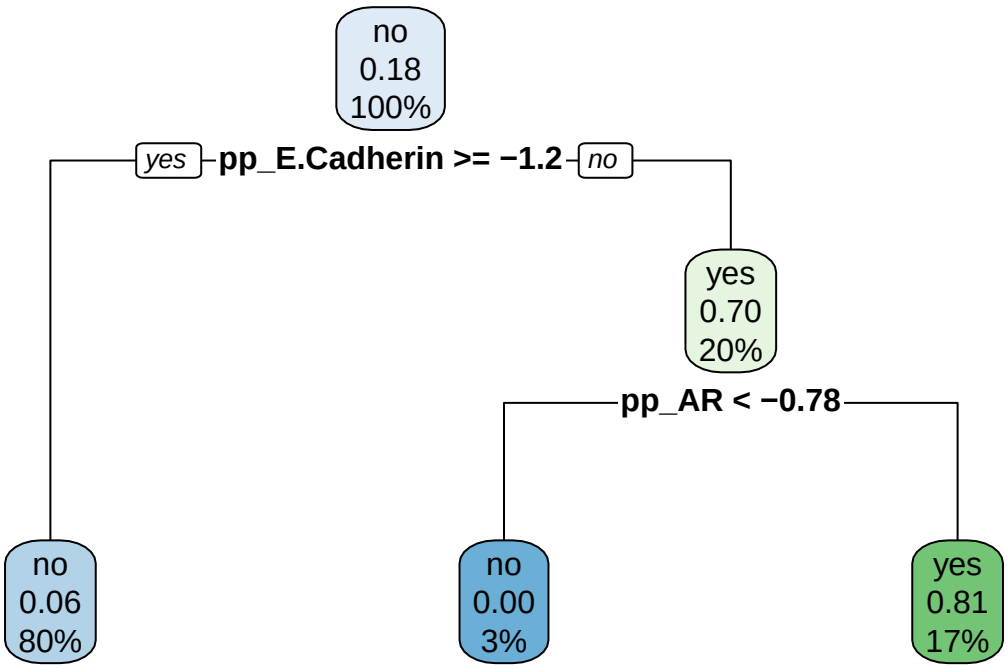
For the decision tree, I had to encode the response as "yes" and "no", where "yes" corresponds to "infiltrating lobular carcinoma". In the traincontrol() function, one has to set summaryFunction=twoClassSummary and classProbs=TRUE in order to generate AUC.

Again, I wanted to have 80% of both classes for training.

The following code trains the tree, using 'ROC' as the metric. We see below that the best cp is fairly small at 0.048 and the AUC is 0.845.

```
## CART
##
##  563 samples
## 1864 predictors
##    2 classes: 'no', 'yes'
##
## No pre-processing
## Resampling: Cross-Validated (4 fold)
## Summary of sample sizes: 422, 423, 422, 422
## Resampling results across tuning parameters:
##
##   cp          ROC        Sens       Spec
##   0.04807692  0.8455326  0.9586003  0.6442308
##   0.15384615  0.7680162  0.9302632  0.6057692
##   0.43269231  0.7680162  0.9302632  0.6057692
##
## ROC was used to select the optimal model using the largest value.
## The final value used for the model was cp = 0.04807692.
```

A visual representation of the best tree:



The most important variables are shown below. We see that pp_E.Cadherin, mu_CDH11 are most important.

```
##                  Overall
## mu_CDH11         84.08169
## pp_AR            24.35365
## pp_beta.Catenin  58.08271
## pp_E.Cadherin   100.00000
## pp_EPPK1         15.38874
```

```
## pp_ER.alpha      19.78918
## pp_INPP4B        16.18329
## rs_ANKRD43       40.38245
## rs_CASP14        16.67195
## rs_FXYD1         30.02180
## rs_CLEC3A         0.00000
## rs_CPB1           0.00000
## rs_SCGB2A2        0.00000
## rs_SCGB1D2        0.00000
## rs_TFF1           0.00000
```

Decision trees are less robust than random forest in that the tree structure is heavily dependent on the sampled data.

The following shows a confusion matrix when used to predict on test data, with an overall accuracy of 0.936.

```
##
## predicted  no yes
##       No  113   7
##       Yes   2  20
```

The tree method allows one to see the relative importance of variables, despite its lower AUC compared to the LASSO method.

6 | VARIABLE SELECTION FOR ALL OUTCOMES

6.1 | LASSO Approach

For this section, I attempted to build logistic regression model via LASSO for other responses (Er.Status, HER2.Final.Status, PR.Status) similar to my regression model for histological.type.

I reread the data, performed similar data cleaning and pre-processing, the I created 3 dataframes corresponding to (Er.Status, HER2.Final.Status, PR.Status). In each respective dataframe, only the response of interest is kept.

In addition, only rows with "Positive" and "Negative" responses are kept.

In the code below, I built model.matrix for categorical variables in each of the dataframes, and then combined it with the numeric columns. The method is similar to the one I used when modeling histological.type.

The following results shows the non-zero coefficients for modeling PR.Status. There are 18 non-zero coefficients (not counting the intercept).

```
##  [1] "(Intercept)" "rs_CYP2B7P1" "rs_VSTM2A"   "rs_FABP7"    "rs_PGR"
##  [6] "rs_A2ML1"    "rs_NAT1"     "rs_PPP1R14C" "rs_TUBA3E"   "rs_SBSN"
## [11] "rs_SLC6A11"  "rs_RASAL1"   "pp_CDK1"     "pp_ER.alpha" "pp_JNK2"
## [16] "pp_PR"       "pp_p53"      "cn_EDIL3.1"  "cn_DEGS22"
```

The following results shows the non-zero coefficients for modeling ER.Status. There are 3 non-zero coefficients (not counting the intercept).

```
## [1] "(Intercept)" "rs_AGR3"     "rs_ESR1"     "pp_ER.alpha"
```

The results below shows non-zero coefficients for modeling HER2.Final.Status. There are 44 non-zero coefficients (not counting the intercept).

```
##  [1] "(Intercept)"               "rs_CLEC3A"
##  [3] "rs_GSTM1"                  "rs_DIO1"
##  [5] "rs_SLC7A4"                 "rs_PAX7"
##  [7] "rs_TMPRSS3"                "rs_CHGB"
##  [9] "rs_SCRG1"                  "rs_S100A1"
```

```
## [11] "pp_X53BP1"                "pp_Acetyl.a.Tubulin.Lys40"
## [13] "pp_Claudin.7"             "pp_HER2"
## [15] "cn_NPFFR22"               "cn_ALB2"
## [17] "cn_CCDC1582"              "cn_FBLL12"
## [19] "cn_FOXI12"                "cn_ENPP3.1"
## [21] "cn_GPR126.1"              "cn_ESR12"
## [23] "cn_CNTNAP2.1"             "cn_CDKN2A.1"
## [25] "cn_SYNPO2L.1"             "cn_ELF51"
## [27] "cn_PCDH202"               "cn_GAS2L22"
## [29] "cn_PPP1R1B2"              "cn_IKZF32"
## [31] "cn_KRT311"                "cn_KRT151"
## [33] "cn_WNK42"                 "cn_RND22"
## [35] "cn_MEOX12"                "cn_TMEM1002"
## [37] "cn_ABCA102"               "cn_B3GNT32"
## [39] "cn_SEC14L21"              "mu_PLCE11"
## [41] "mu_KIF26B1"               "mu_ATM1"
## [43] "mu_MAP21"                 "mu_ALMS11"
```

It is difficult to select 50 variables that have high predictive power for all four responses, since we cannot conveniently understand the relative importance of the variables. Random forest may be better suited for this task.

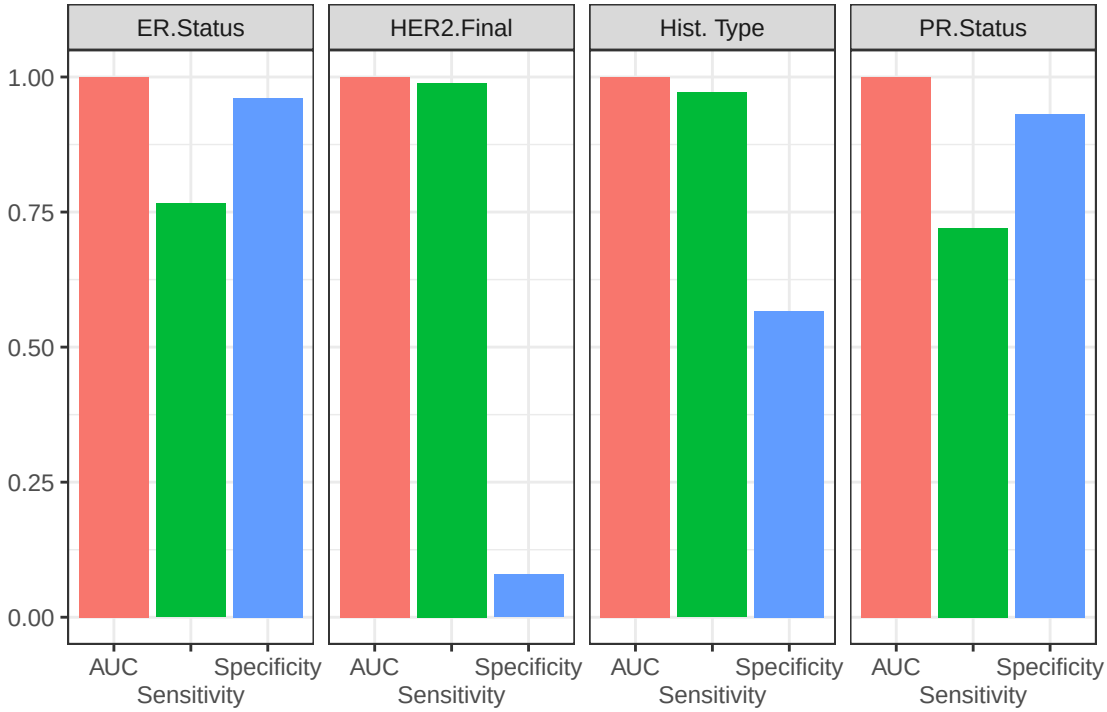
6.2 | Random Forest Approach

Using an approach similar to the one utilized in the "Model `PR.Status`" section earlier, random forests can provide us with the importance of each feature as it relates to the underlying decision schema in the model. Since we want to select a subset of 50 covariates to explain all four response variables accurately, we want to be methodical about or selection. Before feeding all covariates into a random forest model and hoping for the best, we establish a selection criteria in advance: we seek covariates that are significant in at least three of the four models, and if that is not satisfactory, we randomly sample from the covariates significant in at least two of the four models. (This came after observing `rs_LOC100271831` as the only covariate significant in all four models.)

Feeding all 1837 predictors into a random forest model (see `generate.geq.covariates()` in `project.functions.glossary.R` for more details), we found 31 covariates significant in at least three models, and an additional 43 covariates significant in at least two models. In order to select 50 covariates from the 74 on offer, we automatically assigned the 31 covariates significant in three models to our ultimate array of 50 covariates, and randomly sampled the remaining 19 from the second subset of significant covariates so as to eliminate any selection bias that might have been present otherwise.

```
## [1] "Selected covariates:"
```

```
##  [1] "rs_TFF1"          "rs_AGR3"          "rs_C1orf64"       "rs_TFF3"
##  [5] "rs_PGR"           "rs_MMP1"          "rs_MSLN"          "rs_WNK4"
##  [9] "rs_ABCC8"         "rs_CAPN8"         "rs_ERBB4"         "rs_LOC100271831"
## [13] "rs_NAT1"          "rs_FLJ45983"      "rs_TPRG1"         "rs_TSIX"
## [17] "rs_AR"            "rs_PLCH1"         "rs_SPDEF"         "rs_ANKRD43"
## [21] "rs_TTC36"         "rs_LOC389033"     "rs_PITX1"         "rs_DEGS2"
## [25] "rs_TNNT3"         "rs_CLSTN2"        "rs_DACH1"         "pp_Bcl.2"
## [29] "pp_E.Cadherin"    "pp_GATA3"         "pp_PR"            "rs_SERPINA5"
## [33] "rs_RGS22"         "rs_PPP1R14C"      "rs_COL11A1"       "rs_PLIN4"
## [37] "rs_A2ML1"         "rs_LPPR3"         "rs_TUBA3D"        "rs_GABBR2"
## [41] "rs_FREM2"         "rs_FXYD1"         "rs_DLX2"          "pp_p53"
## [45] "rs_FOSB"          "rs_ANKRD30A"      "rs_SERPINA11"     "rs_SLC7A2"
## [49] "rs_ANKRD30B"      "rs_RASAL1"
```



After running our model per the directions in the requirements, we obtain the following results. Though this is guilty to in-sample fitting, our AUC is a consistent 1.00 across all four models, showing that for all 705 patients, these 50 covariates selected above have accurately predicted the labels of all four response variables. Further improvements could have been made to this model with the addition of testing data and domain knowledge as we could have had more information to better suit us for selecting the 50 covariates that would explain our responses.

In retrospect, this project not only reinforced our learning of statistical learning, but also opened us up to the applications of this increasingly expanding field. Although confusing at moments given our lack of knowledge in the field of breast cancer, we appreciated taking on the challenge of learning more about a foreign subject and using familiar technologies to hypothesize, model, and test an array of theories we had about our understanding of the field, as well as solidifying our current approaches to the methods behind these models.

□