Classifying Breast Cancer Images Using Machine Learning Methods

Yuxuan Chen | Yuan Meng | Paula Wu

Introduction & Data Preprocessing

Breast cancer is only of the most common invasive cancer in women in the Unied States, second only to skin cancer [1]. Starting from different parts of the breast, breast cancer is usually marked by lumpiness or swells in the breasts and surrounding tissues. However, it is important to distinguish benign breast tumors from malignant ones, as non-cancer breast tumors are usually abnormal growths that do not spread outside of the breast. They are not life-threatening, even though some types of benign breast lumps can increase a women's risk of getting breast cancer [2]. On the other hand, malignant tumors are aggressive and deadly. Fortunately, the prognosis of the disease has been greatly improved once it is detected. Therefore, it's important to have diseased tissue accurately diagnosed, as misdiagnosis may leads to delayed intervention or may cause unnecessary stress on a patient.

Data Overview

The original dataset contains 569 observations and 33 columns [3]. Among the 569 observations, there are 212 malignant observations and 357 benign observations. Thirty out of the 33 columns are predictors regarding imaging features, computed from a digitized image of a fine needle aspirate (FNA) of a breast mass [4]. These features describe the characteristics of the cell nuclei present in the image. Ten real-valued features are computed for each cell nucleus:

- radius: mean of distances from the center to points on the perimeter
- texture: standard deviation of gray-scale values
- perimeter: perimeter of the tumor image
- area
- smoothness: local variation in radius lengths
- compactness: computed as "perimeter^2 / area - 1.0"
- concavity: severity of concave portions of the contour
- concave points: number of concave portions of the contour
- symmetry
- fractal dimension: comuted as "coastline approximation - 1"

For each variable, its mean, standard error, and "worst" or largest (mean of the three largest values) were computed for each image, resulting in 30 total predictors. In this project, our goal is trying to determine which features contribute to a better diagnosis of breast tissues (M = malignant, B = benign).

Data preprocessing

By the time we got the dataset, it is quite clean and complete. Therefore major data preprocessing of this dataset only entails factorizing outcome variables and removing the ID column.

Exploratory Data Analysis

Correlation Plots

After data preprocessing, we plot the correlation plot (**Figure 1**) of all the predictors. We can tell from the graph that there is a high correlation across the predictors. This is because our predictors include mean, standard deviation, and the largest values of the distributions of 10 features, which means that some predictors can be calculated from others. The plot shows that the size of the core tumor has a strong correlation with tumor perimeter, radius, and area in the breast. It also shows tumor compactness is most correlated with its concavity and symmetry. Though the dataset has high multicollinearity, we choose not to trim any predictor given that

Feature Plots

According to the density featurePlot visualization of each feature's mean (**Figure 2**), we can see that the distributions of benign versus malignant tumors are not that overlapping, suggesting a relatively clear distinction. More specifically, patients with malignant tumors tend to have higher mean values on "compactness", "radius", "concavity", "perimeter", "area", and "concave points" of the tumor images.

K-mean

We use K-mean clustering to partition the observation into 2 clusters. By utilizing the `fviz_nbclust()` function, we can visualize and determine the optimal number of clusters equals to 2 (**Figure 3**), which corresponds to the number of categories of the response variable (M = malignant vs. B = benign). The **K-mean** plot shows the distribution of clusters on two principal components - PC1 and PC2. With each response labeled by their true categories, we can see that the malignant cases are mainly in cluster 1, while the benign cases are mainly in cluster 2.

Models

All 30 continuous predictors are included to predict the outcome variable - "diagnosis". A 70:30 test-train split was applied and yielded 399 rows of training data and 170 rows of testing data. The 30 predictors are the "mean", "standard error", and "worst" or largest of 10 real-valued features. A total of 8 models, including penalized logistic regression, KNN, MARS, linear discriminant analysis, CART, random forest, boosting, and support vector machine (both linear and radial kernels), were used to fit the data. Data are centered and scaled during model training using `caret`, if necessary. Considering the computation cost, we use 10-fold cross-validation to find the optimal values for models involving tuning parameters. The process to select tuning parameters is shown in **Figure 4**. Last but not the least, we use the resampling method to compare the fitted models and use an unseen test dataset to examine the model fit.

Penalized Logistic regression

Due to the nature of the binary outcome of the dataset, we fit a penalized logistic regression to assess the performance of linear decision boundaries. All 30 predictors are included to fit the model. The best-tuned model has $\alpha = 0.45$ and $\lambda = 0.004$ as hyperparameters.

KNN

KNN is a non-linear relation model to predict class labels among k neighbors. It is also a black-box model that is easy to implement and has good classification performance. The best tuning parameter is when the number of neighbors equals 28.

MARS

MARS model can automatically model non-linearities and interactions between variables. This model is well-fit for high-dimensional problems. There are two tuning parameters associated with this model: degree of interaction and number of retained terms. So, we performed a grid search and cross-validation to identify the optimal combination of these hyper-parameters.

Linear Discriminant Analysis (LDA)

As its name suggests, LDA is a linear method for classification. This model projects a feature space onto a smaller subspace and classifies data points based on the nearest centroid; therefore does not involve any hyperparameters. It also assumes that the features follow Gaussian distribution [5]. While one advantage of LDA is its robustness to data that has two or more response classes, it is not the case in this breast cancer study that has binary outcomes.

Naive Bayes

With mechanisms similar to LDA, Naive Bayes becomes handy when there is a large number of predictors. However, this model does require feature independence, an assumption that can be easily violated here - as shown in EDA. Therefore, this method will only be briefly discussed in this section, and no model is fitted.

Classification and Regression Tree (CART)

CART is a tree-based method that grows branches in a top-down greedy fashion through recursive binary splitting and aims to partition the predictor space into several axis-parallel regions. When building the model, we chose to use the Gini index, a measure of node purity, as the stopping criteria for splitting. We tune the hyperparameter `Cp` through cross-validation and find that the optimal `Cp` equals 0.0072. As a single tree, it does have some drawbacks such as instability and high variance. Nevertheless, this model is included as a baseline for comparison with more complex tree-based models presented next.

Random Forest

As an ensemble model, the random forest is flexible and has a better chance to capture a complex underlying true model by adopting the "wisdom of crowds" idea [6]. The model contains multiple trees that are built on bootstrapped random observations, and for each split within each tree, a fresh subset of predictors will be selected. Therefore, the random forest is able to decorrelate the trees and thus reduce the variance compared to a single tree. Similar to CART, we adopt the Gini index to control splitting. There are 2 hyperparameters to tune in the random forest model: `mtry` and `minimun node size`. According to the `bestTune` object of the result, the model achieved its highest AUC when `mtry` = 1 and `min.node.size` = 2, indicating a small tree for each tree built.

Boosting

Boosting is a similar approach to the random forest, except that the trees are grown sequentially - each tree is grown using information from previously grown trees. Three hyperparameters, namely the total number of trees B, numbers of splits d, and shrinkage λ, are tuned. We use grid search, together with cross-validation, to find the optimal value for each tuning parameter. The optimal model that has the largest AUC is achieved when there are 4 degrees of interaction, 5000 trees fit, and 0.002 shrinkages.

Support Vector Machine (SVM with Linear and Radial Kernels)

SVM works by mapping data to a high-dimensional feature space so that data points can be categorized by the separating hyperplanes. The separating hyperplane that makes the biggest gap or margin between the two classes is selected. The decision function is fully specified by the support vectors (SVs), data points that lie closest to the decision surface, and the elements in the training set that would change the position of the hyperplane if removed [7]. For the linear kernel, support vector classifiers with one tuning parameter `cost` (tuned to be 0.969) are used to build a linear boundary. Since most real data sets will not be fully separable by a linear boundary, the radial kernel is used. At its core, the radial kernel indicates that support vector machines can construct classification boundaries that are nonlinear in shape. When tuning for the radial kernel, we add another tuning parameter - sigma. After cross-validation, the optimal `cost` is found to be 4.365 while `sigma` is 0.0286.

Results

After fitting the models, we would like to compare them and select the best one. We are also curious about what are the most important variables for models that have a relatively good fit. Details of model comparison and variable importance are shown below.

Conclusion & Discussion

Appendix

```
## Warning: package ’ggplot2’ was built under R version 4.1.2
```

```
## Warning: package ’tibble’ was built under R version 4.1.2
```

```
## Warning: package ’tidyr’ was built under R version 4.1.2
```

```
## Warning: package ’readr’ was built under R version 4.1.2
```

```
## Warning: package ’dplyr’ was built under R version 4.1.2
```

```
## [1] M B
## Levels: B M
```

Figure 1: Feature Plots

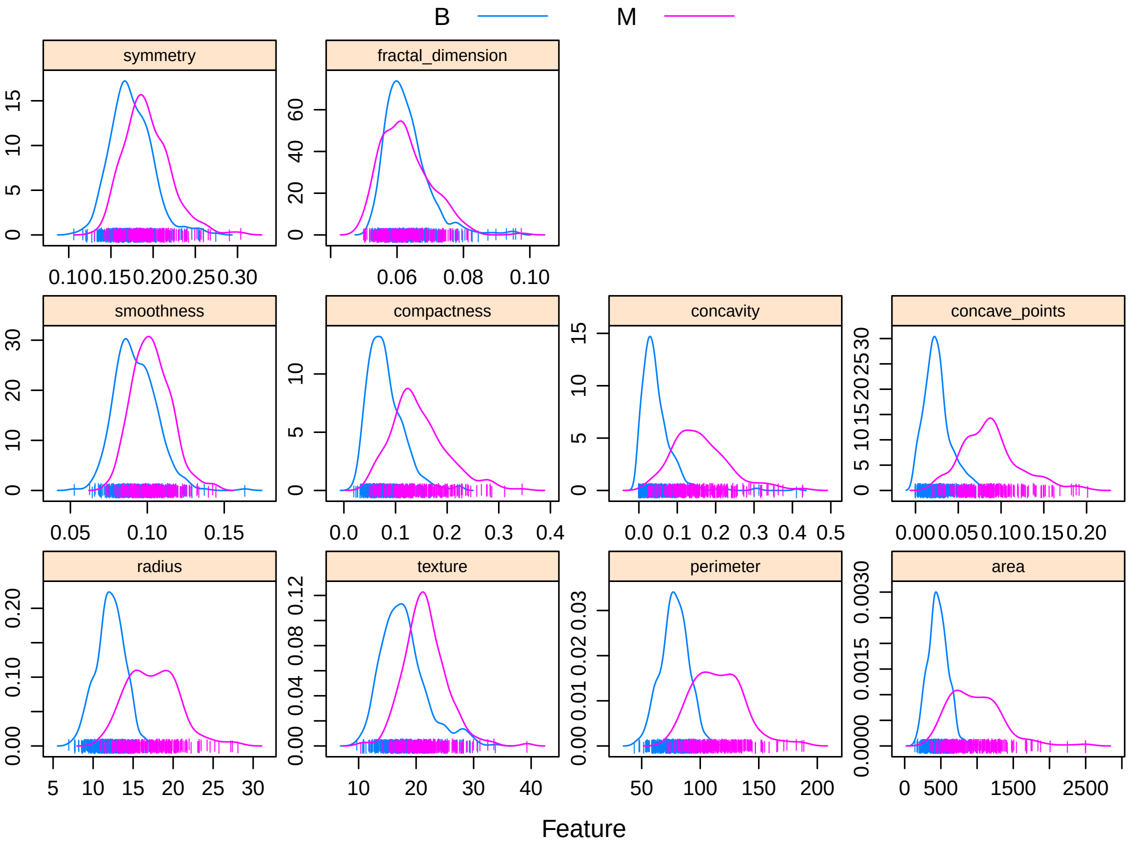

Figure 1: Feature Plots

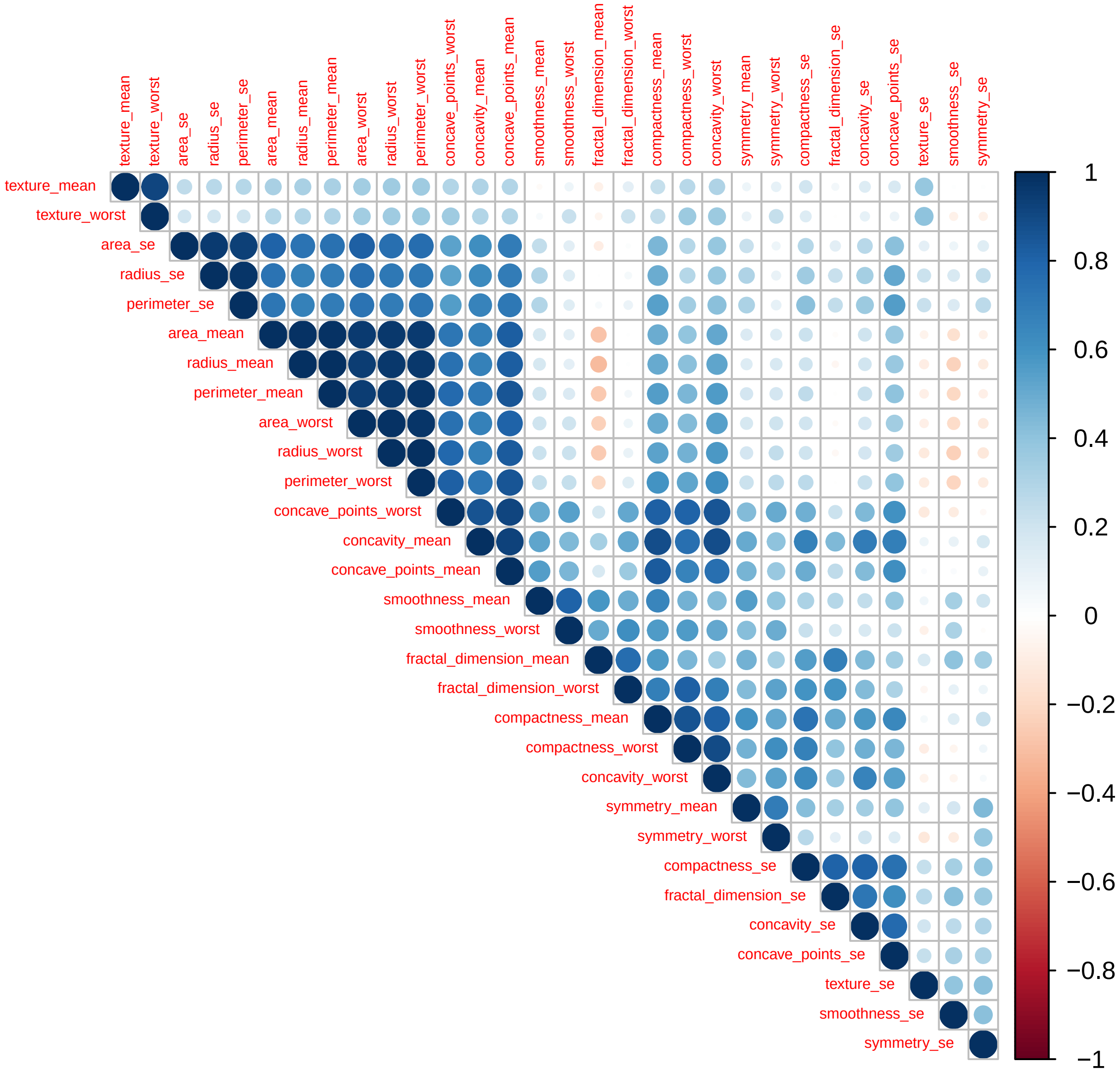

Figure 2: Correlation Plot

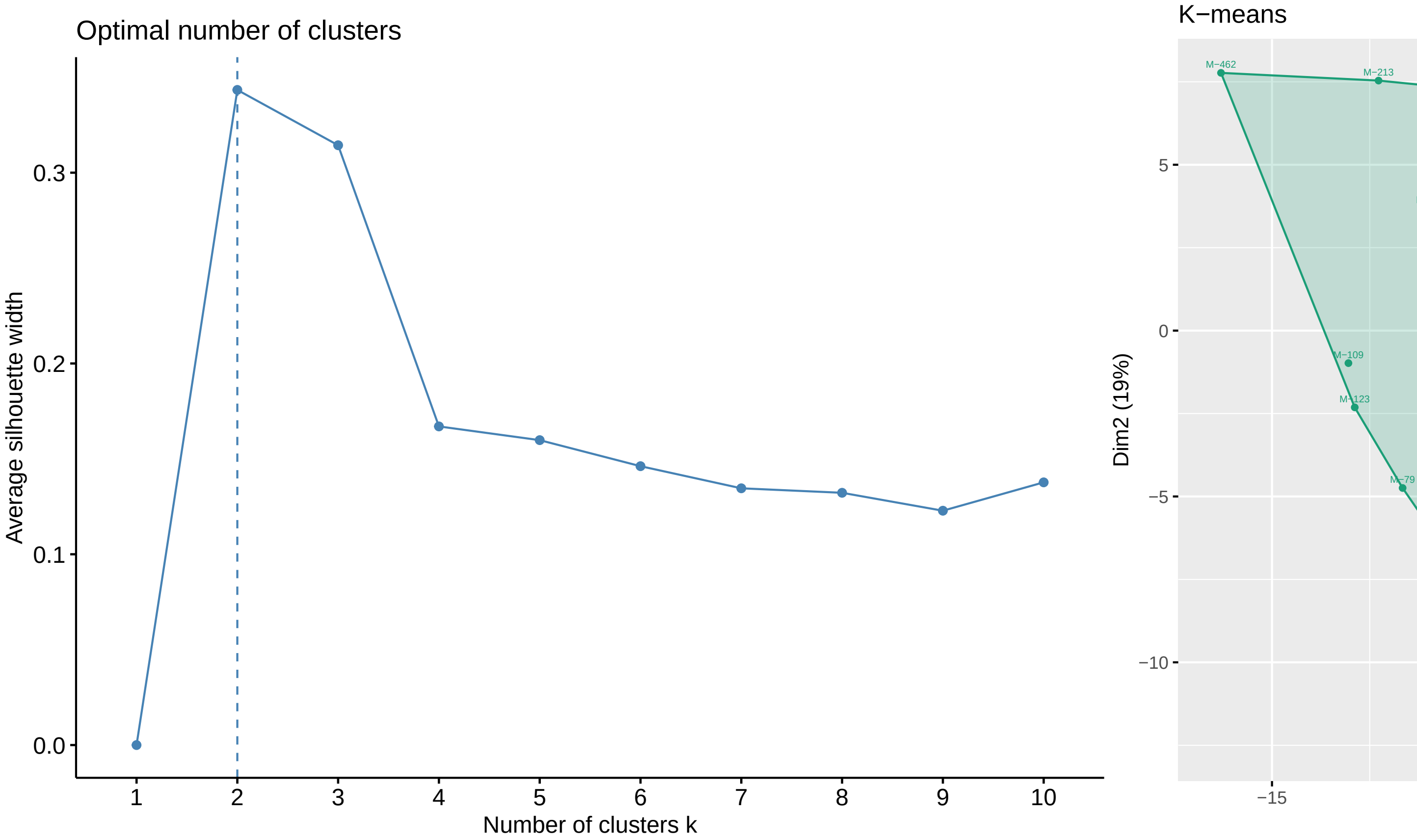

Figure 3: K-mean

```
##   nprune degree
## 7      8      1

##                      (Intercept)             h(1299-area_worst)
##                     1.682024e+01                  -3.260643e-02
## h(0.07911-concave_points_worst)            h(17.68-radius_mean)
##                     1.132215e+02                   2.046438e+00
##  h(0.07507-concave_points_mean)          h(35.64-texture_worst)
##                    -1.660836e+02                  -3.391145e-01
##        h(perimeter_worst-124.3)        h(perimeter_worst-98.27)
##                    -3.899007e-03                  -3.267962e-02

##     k
## 28 28

##    n.trees interaction.depth shrinkage n.minobsinnode
## 64    5000                 4     0.002              1

## maximum number of iterations reached 0.0001344337 0.0001316066maximum number of iterations reached 0
```

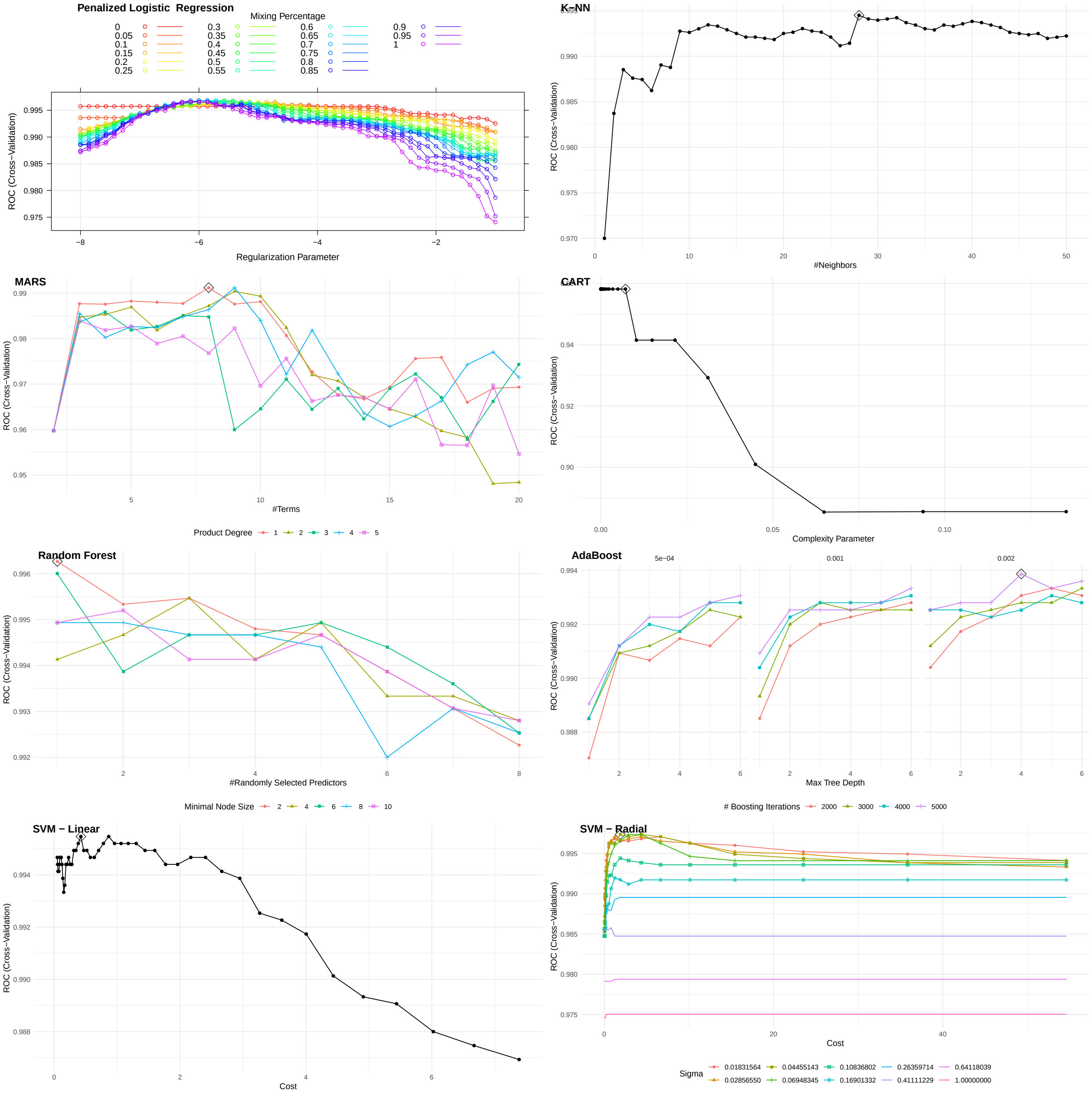

Figure 4: Model tuning

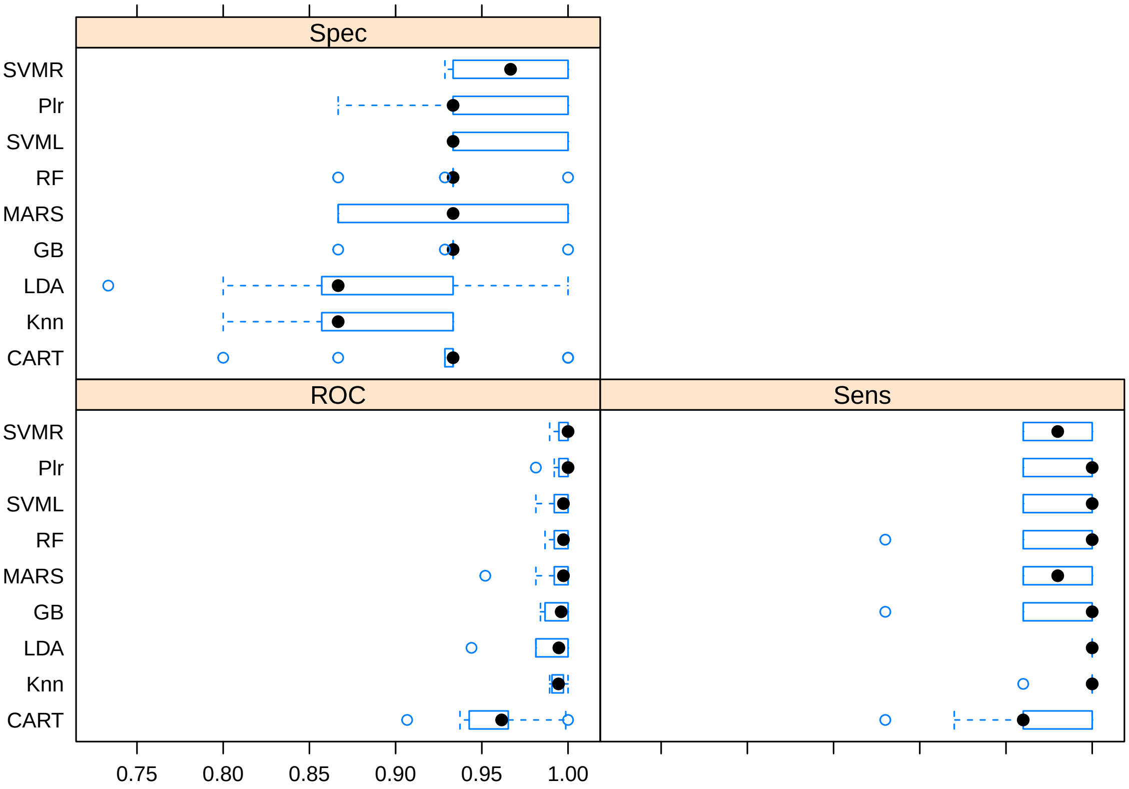

Figure 5: Resampling Results

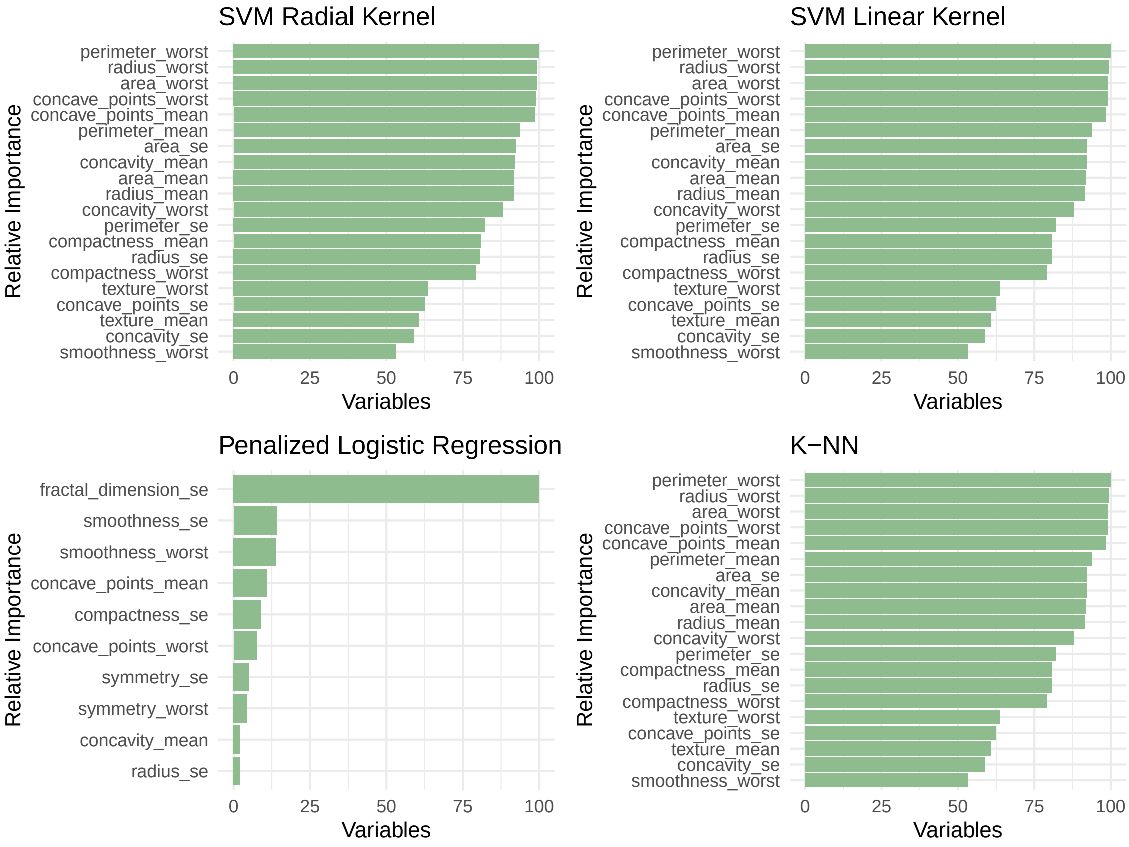

Figure 6: Variable Importance

Table 1: testAUC scores

model	testAUC
SVML	0.9974781
Plr	0.9958463
LDA	0.9958463
MARS	0.9951046
SVMR	0.9949562
KNN	0.9913959
GB	0.9884290
RF	0.9872422
CART	0.9456312

Table 2: Coefficients of Penalized Logistic Regression Model

	coef
(Intercept)	-25.5059446
radius_mean	0.0557381
texture_mean	0.1050963
perimeter_mean	0.0072075
area_mean	0.0004757
concavity_mean	4.2732321
concave_points_mean	21.0406499
radius_se	3.9013407
perimeter_se	0.2671455
area_se	0.0103224
smoothness_se	-27.1800544
compactness_se	-17.2343871
symmetry_se	-9.6302843
fractal_dimension_se	-193.9228525
radius_worst	0.2049761
texture_worst	0.1229049
perimeter_worst	0.0245657
area_worst	0.0013846
smoothness_worst	27.1684565
concavity_worst	2.1546786
concave_points_worst	14.5823135
symmetry_worst	8.6031276
fractal_dimension_worst	1.0808845

References

[1] Breast cancer basic information.

[2] Lecture Note 7 p2-3

[3] 3

[4] 4

[5] 5