Glass Type Prediction on Basis of the Chemical Analysis

Xinlei Chen, Xinyi Lin, Jiawei Ye

INTRODUCTION

Glass fragments are often found on the clothes of suspects for burglary or other crimes. These fragments are routinely collected and analysed at forensic laboratories. It is often of interest to determine whether the glass is from a broken window, or from other sources such as broken containers, tablewares or headlamps.

The aim of this project is to predict the glass type based on the chemical analysis. Our dataset comes from the package `mlbench`. This dataset was originally collected by Home Office Forensic Science Laboratory, Birmingham, UK. It contains the refractive index (RI) and weight percentage of 8 element components (sodium, magnesium, aluminum, silicon, potassium, calcium, barium and iron) of 214 sample glass fragments. Out of 214 total observations, 163 are window glasses and 51 are non-window glasses with no missing values. The response variable is the type of the glass. The seven types are:

1: building windows which was preprocessed by floating molten glass on molten tin

2: building windows which was not preprocessed by floating molten glass on molten tin

3: vehicle windows which was preprocessed by floating molten glass on molten tin

4: vehicle windows which was not preprocessed by floating molten glass on molten tin

5: containers

6: tableware

7: headlamps

To simplify the problem, we decided to use dichotomous classification responses and classify the observations as either "window glass" (1-4) or "non-window glass" (5-7), which is also acceptable from a forensic point of view.

EXPLORATORY DATA ANALYSIS

To obtain an general understanding of the dataset, the density plot [fig.1] was used to summarize the distribution of the data. The result showed that `Al`, `Na`, `RI` and `Si` are almost normally distributed and clusters are detected on the `Mg`. Next, we further improved the density plot by separating each attribute by their class value for the observation [fig.2]. For most predictors, there were different patterns of distribution between different glass types (window glass and non-window glass). The distributions of `Mg` between 2 classes were well separated so it might have a strong association with the response variable.

Then, correlation plot [fig.3] was used to check the correlation between predictors. If the correlation between predictors equals to 1, it means that these together won't add anything new to the model and such collinearity should be removed. According to the correlation plot, the highest correlation between variables came from `Ca` and `RI` (0.81) and it was considered acceptable.

METHOD

We investigated the binary classification problem (window vs. non- window) by randomly splitting glass identification dataset into training set (75%) and test set (25%) using all variables. We then built classifiers on training dataset using `caret` package and evaluated each of classifiers using an independent test set (not model selection).The selected tuning parameters were listed in Table.1. Besides, we compared

the performance of several models that are built on training glass dataset using cross-validation. The model building methods we used are listed as below:

Linear Methods

1. Logistic Regression

   Logistic regression models the probability that response variable belongs to a particular category, which can be represented as $Pr(response = 1|predictors)$. The value of $Pr(response = 1|predictors)$ ranges between 0 and 1. Then for any given value of each predictors, a prediction can be made. The logistic model takes the form of $log\left(\frac{\pi_1}{1-\pi_1}\right) = X^T\beta$, where $\pi_1 + \pi_0 = 1$. $\pi_1$is the probability that the glass type belongs to the window glass category, and π_0 is the probability that the glass type belongs to the non-window glass category. The coefficients β are estimated by maximum likelihood approach.

2. Regularized Logistic Regression

   Regularized Logistic Regression is a method of fitting a generalized linear model via penalized maximum likelihood. The regularization path is computed for the lasso or elasticnet penalty at a grid of values for the regularization parameter lambda.

3. Linear Discriminant Analysis

   The LDA model assumes that the input variables $X = (X_1, X_2, ...X_p)$ are drawn from a multivariate normal distribution (which assumes that each individual predictor follows a one-dimensional normal distribution with some correlation between each pair of predictors), with a class-specific mean vector and a common covariance matrix. That is, it assumes that an observation from the kth class is of the form $X \sim N(\mu_k, \Sigma)$, where Σ is a covariance matrix for the kth class. The observation $X = x$ is assigned to the class for which

   $$\delta_k(x) = x^T\Sigma^{-1}\mu_k - \frac{1}{2}\mu_k^T\Sigma^{-1}\mu_k + log_{\pi_k}$$

   is largest (π_k denotes the prior probability that an observation belongs to the kth class). Then we get,

   $$log\left[\frac{P(Y=1|X=x)}{P(Y=0|X=x)}\right] = log\left[\frac{\pi_1}{\pi_0}\right] - \frac{1}{2}\mu_1^T\Sigma^T\mu_1 + \frac{1}{2}\mu_0^T\Sigma^T\mu_0 + (\mu_1^T\Sigma^{-1} - \mu_0^T\Sigma^{-1})x = \gamma_0 + \gamma^T x$$

   where $\gamma_0 = log\left[\frac{\pi_1}{\pi_0}\right] - \frac{1}{2}\mu_1^T\Sigma^T\mu_1 + \frac{1}{2}\mu_0^T\Sigma^T\mu_0$ and $\gamma = \Sigma^{-1}(\mu_1 - \mu_0)$. μ_1 and μ_0 are prior probabilities of being in window class and in non-window class repectively. Thus the decision boundary is $\gamma_0 + \gamma^T x = 0$. If $\gamma_0 + \gamma^T x > 0$, the the subject belongs to the window class; otherwise, the subject belongs to the non-window class. Parameters are estimated by the maximum-likelihood estimation.

Non-Linear Methods

1. QDA

   The QDA classifier assumes that the observations from each class are drawn from a Gaussian distribution, and plugging estimates for the parameters into Bayes' theorem in order to perform prediction. QDA assumes that each class has its own covariance matrix. In other words, it assumes that an observation from the kth class is of the form $X \sim N(\mu_k, \Sigma_k)$, where Σ_k is a covariance matrix for the kth class. The observation $X = x$ is assigned to the class for which

   $$\delta_k(x) = -\frac{1}{2}x^T\Sigma_k^{-1}x + x^T\Sigma_k^{-1}\mu_k - \frac{1}{2}\mu_k^T\Sigma_k^{-1}\mu_k - \frac{1}{2}log|\Sigma_k| + log\pi_k$$

   is largest. Then we get the log odds of window versus non-window:

   $$log\left[\frac{\pi_1}{\pi_0}\right] - \frac{1}{2}log|\Sigma_1| + \frac{1}{2}log|\Sigma_2| - \frac{1}{2}\mu_1^T\Sigma_1^T\mu_1 + \frac{1}{2}\mu_0^T\Sigma_0^T\mu_0 + (\mu_1^T\Sigma_1^{-1} - \mu_0^T\Sigma_0^{-1})x - \frac{1}{2}x^T(\Sigma_1^{-1} - \Sigma_0^{-1})x$$

which is a quadratic function of x. When it equals to 0, we obtain the decision boundary.

2. Naive Bayes

   The naive Bayes classifier is an approximation to the Bayes classifier, in which we assume that the features are conditionally independent given the class instead of modeling their full conditional distribution given the class. Given a sample X, the classifier will predict that X belongs to the class having the highest a posteriori probability, conditioned on X. That is X is predicted to belong to the class C_i if and only if

   $$P(C_i|X) > P(C_j|X), \quad for\ 1 \leq j \leq m,\ j \neq i.$$

3. K-Nearest Neighbors

   Given a positive integer K and a test observation x_0, the KNN classifier first identifies the K points in the training data that are closest to x_0, represented by N_0. It then estimates the conditional probability for class j as the fraction of points in N_0 whose response values equal j:

   $$Pr(Y = j|X = x_0) = \frac{1}{K} \sum_{i \in N_0} I(y_i = j)$$

   KNN applies Bayes rule and classifies the test observation x_0 to the class with the largest probability.

Classification Trees and Ensemble Methods

1. Classification Tree

   Classification tree method predicts that each observation belongs to the most commonly occurring class of training observations in the region to which it belongs. Recursive binary splitting is used to grow a classification tree and Gini index or the entropy are typically used to evaluate the quality of a particular split.

2. Random Forests

   Random forests is an improvement method over bagged trees by way of a small tweak that decorrelates the trees. In bagging, we build a number of decision trees on bootstrapped training samples. However, random forests adds additional randomness to the model while growing the trees. Instead of searching for the most important feature while splitting a node, it searches for the best feature among a random subset of features. Therefore, in random forests, only a random subset of the features is taken into consideration by the algorithm for splitting a node.

3. Boosting

   Boosting is another approach for improving the predictions resulting from a decision tree. Boosting works in a way that the trees are grown sequentially: each tree is grown using information from previously grown trees, each tree is fit on a modified version of the original data set.

Support Vector Machine

A support-vector machine constructs a hyperplane or set of hyperplanes in a high- or infinite-dimensional space, which can be used for classification. A good separation is achieved by the hyperplane that has the largest distance to the nearest training-data point of any class, since in general the larger the margin, the lower the generalization error of the classifier.

RESULT

Fitted Models

Among linear methods, logistic regression produces the highest AUC for ROC curve (0.989). Three out of 10 coefficients (including intercept) for logistic regression are significant at 0.05 level. The significant coefficients are intercept, coefficients for `RI` and `Al`. The best tuned quadratic discriminant model produces the highest AUC (0.99) among the 3 non-linear models (QDA, Naive bayes and k-nearest neighbours).

For classification trees and ensemble methods, we fit 3 models which were classification tree, random forests and boosting. The result of classification tree showed that the optimal tree size was 3 with misclassification error rate of 9.62%. Both random forests and boosting provided the variable importance. In random forests, the top 5 important perdictors were `Mg`, `Al`, `K`, `Na` and `Ba`; in boosting, the top 5 important perdictors were `Mg`, `Al`, `RI`, `Fe` and `Na`. The misclassification error rate of random forests was 1.92% and that of boosting was 3.85%.

We fit 2 SVM models using linear kernal and radial kernal respectively. The AUC of linear kernal was 0.96 and the test error rate was 1.92%; the AUC of radial kernal was 0.99 and the test error rate was 5.77%. Accoridng to the resample result, we found that the support vector machine model with radial kernel had higher medians in accuracy and kappa. It also had higher upper bound in kappa.

Model Selection and Interpretation

	Cross-Validation Training AUC
logistic	0.9583974
regularized logistic regression	0.9683974
LDA	0.9710256
QDA	0.9327297
naive bayes	0.9690491
KNN	0.9725000
classification tree	0.9369872
random forests	0.9818590
boosting	0.9702724
svm linear	0.9615705
svm radial	0.9866346

Model selection was conducted based on cross-validation training AUC [Fig.4]. Among all models, the model showed the best performance was SVM with radial kernal with cross-validation training AUC of 0.987. The training error rate of SVM with radial kernal was 1.23% and the test error rate was 5.77%. According to the confusion matrix, it had near 1 sensitivity and about 0.9 specificity. The kappa was 0.85 and the accuracy was 0.94. The PPV was about 0.8 and the NPV was near 1.

Based on cross-validation training AUC, LDA was the best-performing linear model with cross-validation training AUC of 0.971; KNN was the best-performing non-linear model with cross-validation training AUC of 0.973; random forests was the best-performing tree model with cross-validation training AUC of 0.982. Among all the models, the wrost-performing one was QDA with AUC of 0.933.

It's worth mentioning that most of the model we built had better test AUC than the cross-validation training AUC [Fig.4]. Among linear models, logistic model had the highest test AUC; among non-linear models, QDA had the highest test AUC; among tree models, boosting and random forests had the highest test AUC; SVM with linear kernal had higher test AUC than SVM with radial kernal. Overall, boosting and random forest had best test performance.

CONCLUSION

Based on cross-validation training AUC, SVM with radial kernal was the best-performing model among all, and random forests, KNN and LDA were the best-performing models in their own categories respectively. All fitted models had good AUC (>0.9) and this met our expectation, since from the density plot [Fig.2] we could see that the distributions of most predictors under each catergories were well-separated. The variable importance provided by random forests and boosting both suggested that `Mg` was the most importance perdictor in glass type classification. This result also met our expectation as in exploratory analysis we observed clusters.

We considered the inadequate sample size as a limitation. There were only 214 observations in this dataset and after splitting the training (75%) and test set (25%), there were only 53 observations left in test set. This might also explain why our test AUC result was not very consistant with our model selection result.

REFERENCE

[1] Evett, I.W. and Spiehler, E.J., 1987. Rule induction in forensic science. KBS in Goverment, pp.107-118.

[2] Goswami, S. and Wegman, E.J., 2016. Comparison of different classification methods on glass identification for forensic research. J Stat Sci Appl, 4(03-04), pp.65-84.

APPENDIX

Figure 1: Density plot of the dataset

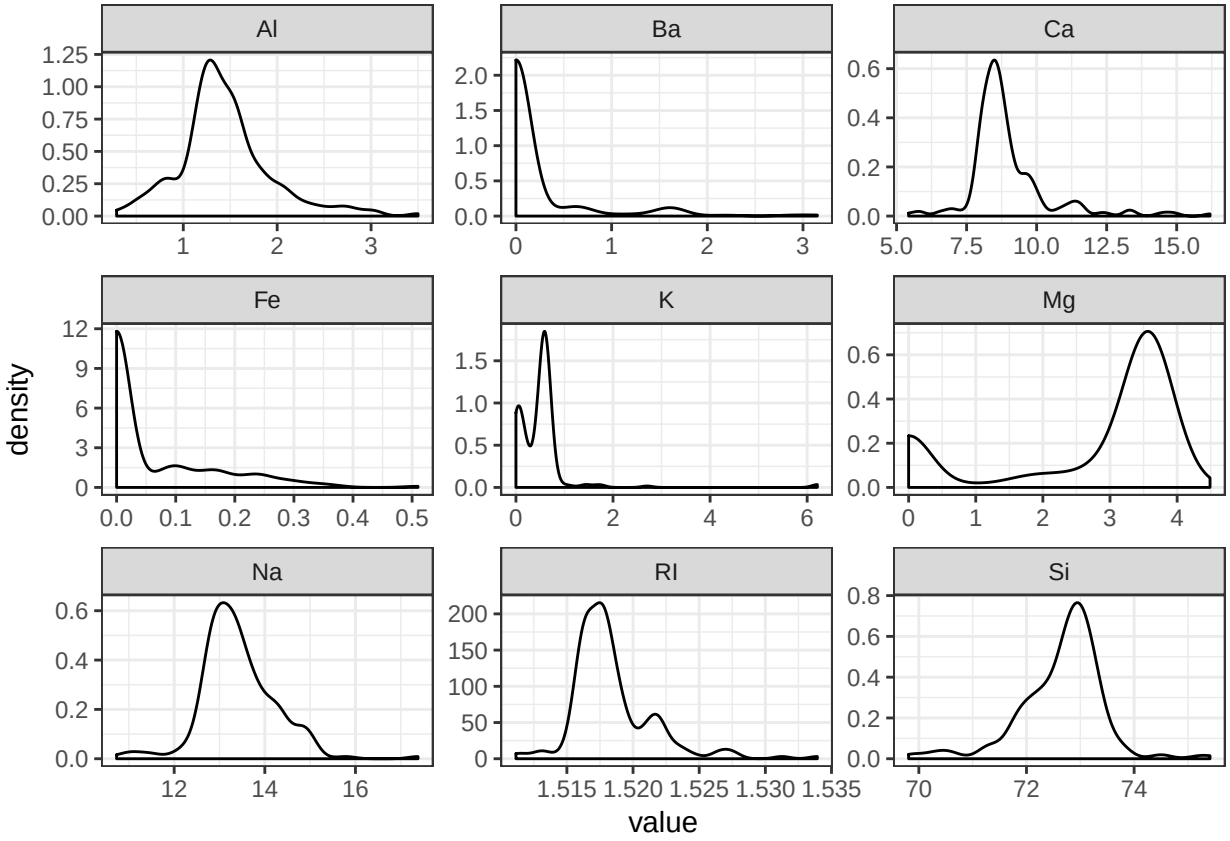

Figure 2: Density plot of the dataset for each response group

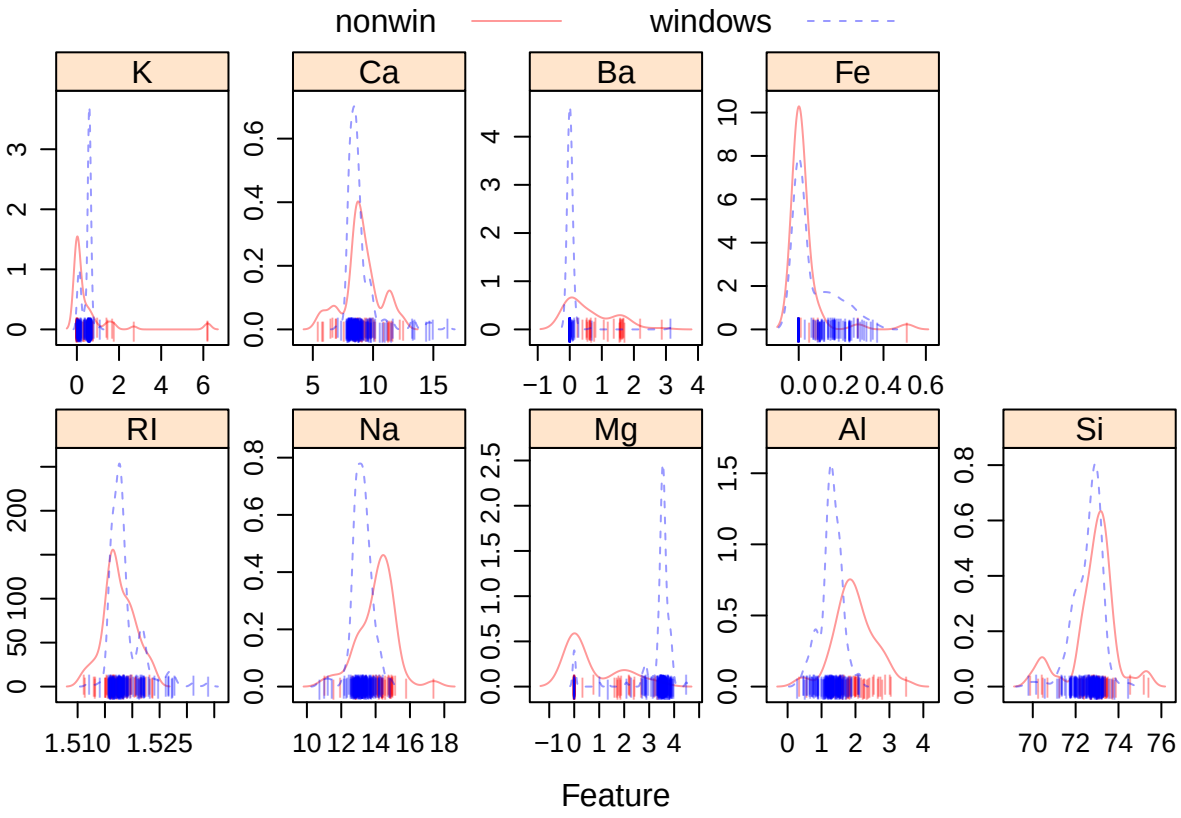

Figure 3: Correlation plot of the dataset

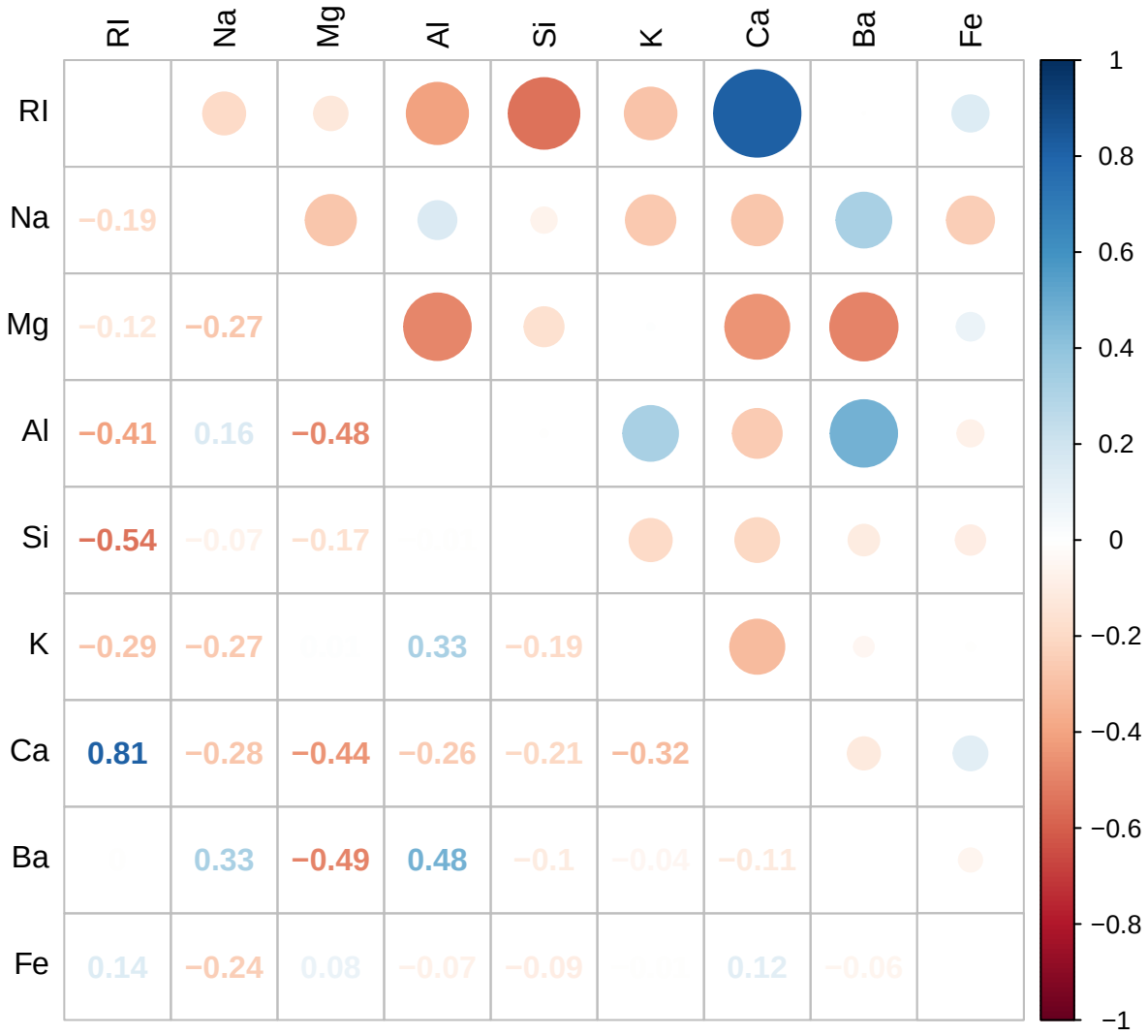

Figure 4: Model selection: cross-validation AUC (The blue points in this plot stand for the test set AUC)

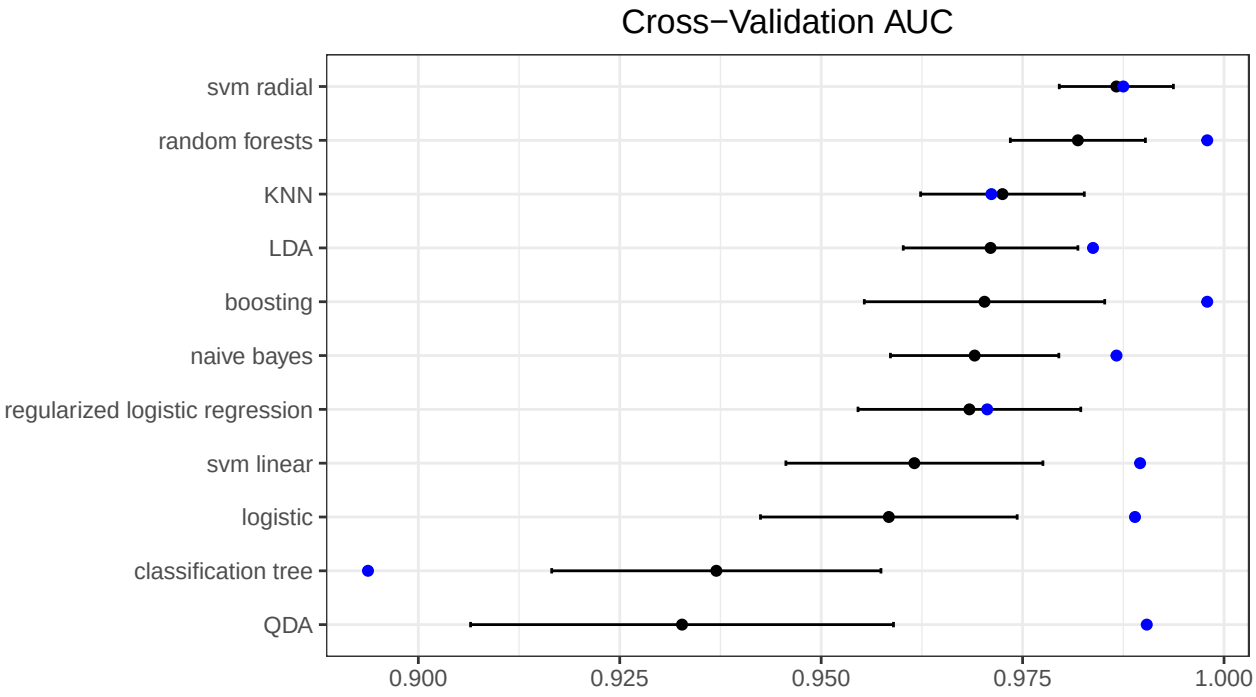

Table 1: Tuning parameters and selected values

Model	Tuning.Parameter	Selected.Value
Logistic	None	
Regularized Logistic	Alpha	0.25
	Lambda	0.4308026
LDA	None	
QDA	None	
Naive Bayes	Kernel	Non-parametric
	LaPlace Smoother	1
	Adjustment	1.5
KNN	k	51
Classification Tree	Cp	0.02634798
Random Forest	Mtry	1
	Split Rule	Gini
	Minimum Node Size	10
Boosting	Number of trees	801
	Interaction depth	3
	Shrinkage (Learning Rate)	0.02
	Minimum observation in node	1
SVM Linear	Cost	1.982206
	Kernel	Linear
SVM Radial	Sigma	0.3678794
	Cost	0.3678794
	Kernel	Radial