Project 3: Classification

36-600

The goal of this project is to use both logistic regression and machine learning models to predict whether a given wine is either `BAD` or `GOOD`.

Your `HTML` file (generated by knitting an `R Markdown` source file) should be uploaded to Canvas by Friday, November 22nd, at 11:59 PM. As was the case for the first two projects, there is no limit on the length here, but concision is a virtue.

Background

The data were collected as part of a study of Portuguese wines. Please see the paper for more information.

Data

You will examine the dataset `wineQuality.csv`, which is linked to on the Project 3 `Canvas` module page.

The response variable is `label` (either `BAD` or `GOOD`... by default, `BAD` is Class 0 due to alphabetical order). Your goal is prediction! We don't care about that multicollinearity "stuff" here.

predictors: 6497 x 11

name	description
fix.acid	fixed acidity (in grams of tartaric acid per decimeter cubed)
vol.acid	volatile acidity (in grams of tartaric acid per decimeter cubed)
citric	citric acid (in grams per decimeter cubed)
sugar	residual sugar (in grams per decimeter cubed)
chlorides	chlorides (in grams of sodium chloride per decimeter cubed)
free.sd	free sulfur dioxide (milligrams per decimeter cubed)
total.sd	total sulfur dioxide (milligrams per decimeter cubed)
density	1 (= <0.99 g/dm^3), 2 (= [0.99,1] g/dm^3), or 3 (= >1 g/dm^3)
pH	wine acidity
sulphates	grams of potassium sulphate per decimeter cubed
alcohol	percentage of the volume

response:

name	description
label	GOOD or BAD wine

Expectations

Your report should include the following elements:

- A description of the data (sample size, number of variables), and a list of removed variables (if there are any).

- Concise EDA, including the identification and removal of proposed anomalies (if there are any). You can, if you wish, transform any predictor variables that are highly skew, but this is not required.
- A description of how you split the data into training and test sets.
- An analysis of the full dataset with logistic regression. Since we will assume that classifying exoplanets as confirmed or false positive is equally important, create a ROC curve and utilize Youden's J to determine the optimal probability threshold for splitting the test data into classes. Show the ROC curve, give the area under curve (AUC), and show the confusion matrix and the misclassification rate.
- You can attempt best-subset selection but it may run very slowly (given that we have 11 predictor variables).
- (Also, to be clear: do not assess multicollinearity, compute vif values, or utilize PCA... unless you simply want to see if we can reduce the dimensionality of the problem. Saying "we would retain the first six PCs" is a conclusion in and of itself and you need not got beyond that. Again, this is not required.)
- Repeat the above with as many ML models as you wish to use. Possibilities include (pruned) trees, random forest, XGBoost, K-nearest neighbors, and the various flavors of SVM. Note: SVM may run too slowly to be worth exploring. If you try SVM, try the linear kernel first to get a sense of computation speed, before proceeding (or not proceeding) to the polynomial and radial kernel models.
- You should include a table that has the model name, the AUC, and the MCR for the Youden's J probability threshold, for all the models you code. Which model would you declare is the best model? For that model, determine the optimal probability threshold and show the confusion matrix given that threshold.
- EXTRA: if you can, overlay all the ROC curves on one plot! It looks cool, but if it "isn't happening" when you try to code it, don't worry... just move on.