Project 3: Classification

36-600

Fall 2022

The goal of this project is to learn classifiers to predict wine quality as a function of wine properties. (All while remembering an important life lesson: if you like cheap wine, great! You can buy more of it.)

Your `HTML` file (generated by knitting an `R Markdown` source file) should be uploaded to Canvas by Tuesday, November 29th, at 11:59 PM.

Data

You will examine the dataset `wineQuality.csv`, which you will find in the `DATA` directory on `Canvas`.

The response variable is `label`. This is a factor variable with levels `BAD` and `GOOD`. When you read in the data, be sure to set `stringsAsFactors=TRUE`!

The 11 predictor variables are

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

All of the predictor variables are quantitative, meaning you can use all classifiers at your disposal. (Except maybe deep learning. See below.)

Note that SVM may be very slow; to test how long SVM will take, perhaps try linear SVM while testing a single value of the cost parameter…then extrapolate. For instance, if SVM with a single cost value takes 2 CPU minutes to run, then testing a grid of 10 cost values will take 20 CPU minutes. Determine if the amount of time you face is too much.

Also note that deep learning will not yield interesting results here because of the small sample size, but if you really feel like trying it out…

There are no missing data.

Expectations

Your report should include the following elements:

- A description of the data (sample size, number of variables).
- Concise EDA, including the identification and removal of proposed outliers (if there are any) and, potentially, the transformations (some) predictor variables that are highly skew. (Note: always try transformations first before identifying and removing outliers…that lonely data point in a skew distribution may look fine after a transformation is performed.) Also, create a correlation plot and comment on the possibility of multicollinearity in the predictor variables (while mentally noting that your goal here is prediction, so multicollinearity is OK).
- (Don't do any PCA and such here. You know how to use it now. In theory.)
- A description of how you split the data into training and test sets.
- An analysis of the dataset with logistic regression, etc., up through and perhaps including KNN and SVM. The classes are unbalanced, so do not assume a class-separation threshold of 0.5!
  - Output Class 1 probabilities for each model.
  - Create ROC curves for each model.
  - Determine the AUC for each model. Tabulate the values; Google how to format basic tables in R Markdown.
  - Pick the model with the highest AUC value.
  - For that model, determine the optimal class-separation threshold by maximizing the Youden's J statistic (senstivity + specificity - 1) and finding the threshold associated with that maximum value.
  - Given the threshold, make class predictions for your best model.
  - Create confusion matrix and compute the misclassification rate. Comment on how good the model is. Done.
- …however, do remember to include bestglm() as one of your models.