**Question 16.1**

# *Using the GermanCredit data set germancredit.txt from* [*http://archive.ics.uci.edu/ml/machine-learning-databases/statlog/german*](http://archive.ics.uci.edu/ml/machine-learning-databases/statlog/german) */ (description at* [*http://archive.ics.uci.edu/ml/datasets/Statlog+%28German+Credit+Data%29*](http://archive.ics.uci.edu/ml/datasets/Statlog+%28German+Credit+Data%29) *), use the xgboost gradient boosting algorithm to find a good classifier. Use R’s xgboost function and set the objective to “binary:logistic.”*

* *Unlike glm, xgboost will not create categorical columns for you. One way to approach this is to use dummy\_cols from the fastDummies package, which will take all your categorical columns and one-hot-encode them.*
* *xgboost requires the response variable to be 0s and 1s. The data set uses 1s and 2s instead, so after reading in the data, your R code will need to make that change.*

Here’s one possible solution. Please note that a good solution doesn’t have to try all of the possibilities in the code; they’re shown to help you learn, but they’re not necessary.

The file solution 16.1.R shows a possible approach with 100 rounds of gradient boosting.

Here’s the ROC curve.

Chart, scatter chart

Description automatically generated

1. *Compare your xgboost classifier to your logistic regression classifier from Question 10.3. How does model performance vary over key metrics (accuracy, sensitivity, specificity, etc.)?*

# The table below compares the performance of xgboost and the performance of the logistic regression. The output of solution 16.1.R includes more rows of xgboost performance; for this table, we took some rows of the logistic regression output showing specificity and sensitivity and compared each one with whatever row of xgboost output had just slightly less specificity.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **LOGISTIC REGRESSION** | | | | **GRADIENT BOOSTING** | | | |
| **Threshold** | **Specificity** | **Sensitivity** | **Accuracy** | **Threshold** | **Specificity** | **Sensitivity** | **Accuracy** |
| 0.6967 | **0.990** | **0.196** | 0.75 | 0.9331 | **0.985** | **0.075** | 0.68 |
| 0.6145 | **0.962** | **0.348** | 0.77 | 0.8076 | **0.955** | **0.179** | 0.70 |
| 0.5516 | **0.940** | **0.457** | 0.77 | 0.6496 | **0.910** | **0.285** | 0.70 |
| 0.4686 | **0.851** | **0.533** | 0.75 | 0.4605 | **0.850** | **0.343** | 0.68 |
| 0.3774 | **0.798** | **0.630** | 0.75 | 0.3315 | **0.797** | **0.465** | 0.69 |
| 0.3199 | **0.740** | **0.717** | 0.73 | 0.2616 | **0.737** | **0.537** | 0.67 |
| 0.2699 | **0.678** | **0.793** | 0.71 | 0.1965 | **0.677** | **0.567** | 0.64 |
| 0.2221 | **0.606** | **0.848** | 0.68 | 0.1247 | **0.602** | **0.657** | 0.62 |
| 0.1675 | **0.529** | **0.891** | 0.64 | 0.0965 | **0.526** | **0.687** | 0.58 |
| 0.1375 | **0.442** | **0.913** | 0.59 | 0.0516 | **0.436** | **0.791** | 0.56 |
| 0.1165 | **0.365** | **0.957** | 0.55 | 0.0370 | **0.361** | **0.851** | 0.53 |
| 0.0869 | **0.274** | **0.967** | 0.49 | 0.0183 | **0.271** | **0.910** | 0.49 |
| 0.0626 | **0.188** | **0.989** | 0.43 | 0.0105 | **0.181** | **0.955** | 0.44 |
| 0.0420 | **0.096** | **1.000** | 0.37 | 0.0060 | **0.090** | **0.955** | 0.38 |

# The table shows that for any of these thresholds used in gradient boosting, there is a threshold to use in logistic regression that achieves a better specificity *and* a better sensitivity. So, for this specific question, we would probably be better off using a logistic regression model instead of a gradient boosting model. (However, that’s not the case for every question; there are also instances where gradient boosting does better than logistic regression. In general, you might want to try multiple types of models and see which one works best.)