Generalized Boosted Regression Model for Income Level Prediction

Cody Minns

May 21, 2023

This project looks at census data from the University of California Irvine machine learning repository. The goal is to predict whether income exceeds $50,000 per year using a variety of predictors.

The final model that I used to make predictions about income was a generalized boosted regression model (GBM). The model has a learning rate of 0.075 which was determined by maximizing the training accuracy. The model also has 2369 trees which was determined using 10-fold cross-validation. The final model uses all predictors from the dataset, except for `id`.

Several different types of models were fit along the way to the final model. I started by fitting neural networks while varying number of layers and dropout rate. Excluding the predictors `capital_gain` and `capital_loss` greatly improved test accuracy.I also considered excluding or somehow incorporating `fnlwgt` into the model as a weight, but this did not improve test accuracy. In addition to neural networks, I also tried fitting a support vector machine using a radial kernel. None of these models had a greater test accuracy than the GBM. The excluded predictors were included in the GBM because they improved test accuracy.

The code for the final model is included in the appendix on the next page.

Appendix

```r
# load generalized boosted model library
library(gbm)
#read in and format data
adult = read.csv('adult.csv')
test = read.csv('adult_test.csv')
for (i in 1:length(adult$income)) {
  if (adult$income[i] == "<=50K") {
    adult$income[i] = 0
  }
  else {
    adult$income[i] = 1
  }
}
for (i in 1:(ncol(adult))) {
  if (class(adult[,i]) == "character") {
    adult[,i] = factor(adult[,i])
  }
}
for (i in 1:(ncol(test))) {
  if (class(test[,i]) == "character") {
    test[,i] = factor(test[,i])
  }
}
adult$income = as.numeric(as.character(adult$income))

# create and fit model
boosted = gbm(income ~ . - id, data = adult,
              distribution = "bernoulli", n.trees = 2369, shrinkage = 0.075)

#make predictions and write to file
preds = predict(boosted, newdata = test, type = 'response')
preds = round(preds)
for (i in 1:length(preds)) {
  if (preds[i] == 0) {
    preds[i] = '<=50K'
  }
  else {
    preds[i] = '>50K'
  }
}
mydf = data.frame(id = test$id, income = preds)
write.csv(mydf, 'predictions.csv', row.names = FALSE)
```