Group Discussion

Daniel Dulaney

October 15, 2020

```r
library(tidyverse)
library(ISLR)
library(tidymodels)
library(gbm)
```

(a)

```r
caravan <- Caravan %>%
  mutate(Purchase = ifelse(Purchase == "Yes", 1, 0))

train <- caravan %>%
  slice(1:1000)

test <- caravan %>%
  slice(1001:nrow(caravan))
```

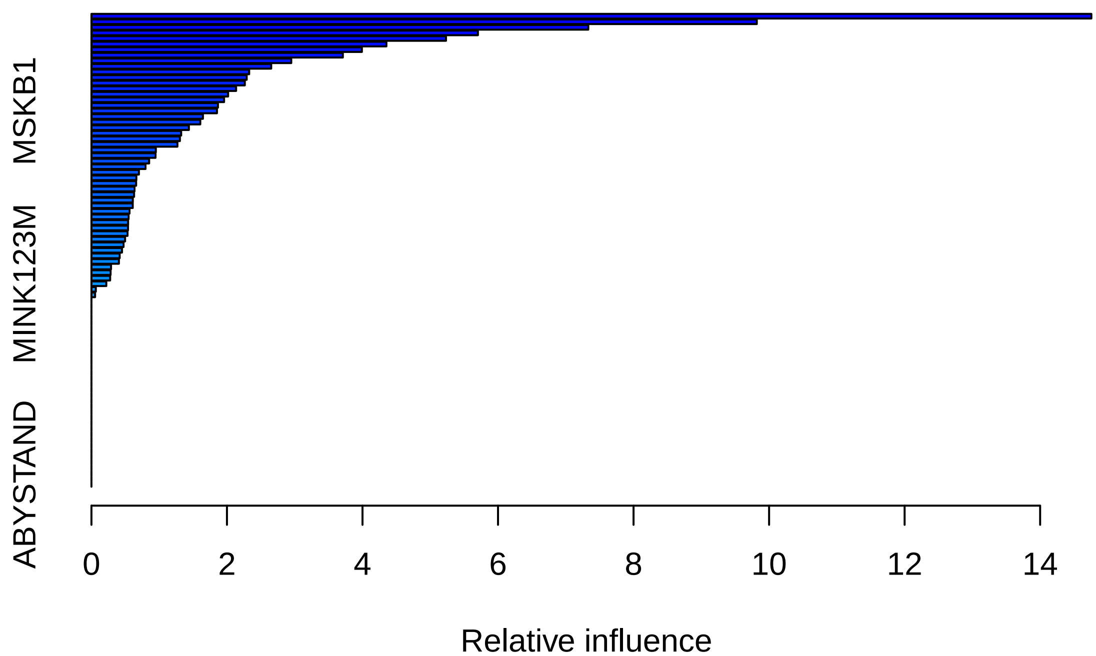
(b)

```r
boost_caravan <- gbm(Purchase ~ ., data = train, n.trees = 1000, shrinkage = .01, distribution = "berno
```

```
## Warning in gbm.fit(x = x, y = y, offset = offset, distribution = distribution, :
## variable 50: PVRAAUT has no variation.

## Warning in gbm.fit(x = x, y = y, offset = offset, distribution = distribution, :
## variable 71: AVRAAUT has no variation.
```

```r
summary(boost_caravan)
```




```
##               var     rel.inf
## PPERSAUT PPERSAUT 14.75586171
## MKOOPKLA MKOOPKLA  9.81854600
## MOPLHOOG MOPLHOOG  7.33165051
## MBERMIDD MBERMIDD  5.70324627
## PBRAND     PBRAND  5.23215171
## MGODGE     MGODGE  4.35266314
## ABRAND     ABRAND  3.99122001
## MINK3045 MINK3045  3.71011323
## MOSTYPE   MOSTYPE  2.94852098
## MAUT1       MAUT1  2.65262797
## MSKC         MSKC  2.32756497
## MSKA         MSKA  2.29116709
## MAUT2       MAUT2  2.26370146
## PWAPART   PWAPART  2.13492616
## MBERARBG MBERARBG  2.01671373
## MGODPR     MGODPR  1.95726550
## MBERHOOG MBERHOOG  1.86681551
## MSKB1       MSKB1  1.85236143
## PBYSTAND PBYSTAND  1.64522058
## MINKGEM   MINKGEM  1.60639623
## MGODOV     MGODOV  1.43804624
## MFWEKIND MFWEKIND  1.32422173
## MINK7512 MINK7512  1.30545520
## MRELGE     MRELGE  1.26887415
## MHHUUR     MHHUUR  0.95043464
```

```
## MINKM30   MINKM30  0.94576678
## MRELOV     MRELOV  0.85151831
## MGODRK     MGODRK  0.79756588
## MOPLMIDD MOPLMIDD  0.70056043
## MINK4575 MINK4575  0.66266903
## MFGEKIND MFGEKIND  0.65948270
## APERSAUT APERSAUT  0.63551453
## MBERARBO MBERARBO  0.62940844
## MZPART     MZPART  0.61053076
## MAUT0       MAUT0  0.60952882
## MOSHOOFD MOSHOOFD  0.56206572
## MGEMLEEF MGEMLEEF  0.54820951
## MSKD         MSKD  0.54114341
## MGEMOMV   MGEMOMV  0.53974509
## MHKOOP     MHKOOP  0.53239061
## MBERBOER MBERBOER  0.49674310
## MZFONDS   MZFONDS  0.47388474
## PLEVEN     PLEVEN  0.45055546
## MRELSA     MRELSA  0.41685648
## PMOTSCO   PMOTSCO  0.40529143
## MBERZELF MBERZELF  0.28925328
## MFALLEEN MFALLEEN  0.28235662
## MSKB2       MSKB2  0.27573515
## MINK123M MINK123M  0.21910896
## MOPLLAAG MOPLLAAG  0.06439662
## MAANTHUI MAANTHUI  0.05395201
## PWABEDR   PWABEDR  0.00000000
## PWALAND   PWALAND  0.00000000
## PBESAUT   PBESAUT  0.00000000
## PVRAAUT   PVRAAUT  0.00000000
## PAANHANG PAANHANG  0.00000000
## PTRACTOR PTRACTOR  0.00000000
## PWERKT     PWERKT  0.00000000
## PBROM       PBROM  0.00000000
## PPERSONG PPERSONG  0.00000000
## PGEZONG   PGEZONG  0.00000000
## PWAOREG   PWAOREG  0.00000000
## PZEILPL   PZEILPL  0.00000000
## PPLEZIER PPLEZIER  0.00000000
## PFIETS     PFIETS  0.00000000
## PINBOED   PINBOED  0.00000000
## AWAPART   AWAPART  0.00000000
## AWABEDR   AWABEDR  0.00000000
## AWALAND   AWALAND  0.00000000
## ABESAUT   ABESAUT  0.00000000
## AMOTSCO   AMOTSCO  0.00000000
## AVRAAUT   AVRAAUT  0.00000000
## AAANHANG AAANHANG  0.00000000
## ATRACTOR ATRACTOR  0.00000000
## AWERKT     AWERKT  0.00000000
## ABROM       ABROM  0.00000000
## ALEVEN     ALEVEN  0.00000000
## APERSONG APERSONG  0.00000000
## AGEZONG   AGEZONG  0.00000000
```

```
## AWAOREG   AWAOREG  0.00000000
## AZEILPL   AZEILPL  0.00000000
## APLEZIER APLEZIER  0.00000000
## AFIETS     AFIETS  0.00000000
## AINBOED   AINBOED  0.00000000
## ABYSTAND ABYSTAND  0.00000000
```

The 5 most important variables, in order, are:

1. PPERSAUT
2. MKOOPKLA
3. MOPLHOOG
4. MBERMIDD
5. ABRAND

(c)

```
boost_probs <- predict(boost_caravan, test, n.trees = 1000, type = "response")
```

```
test <- test %>%
  mutate(prob_Purchase = boost_probs,
         pred_Purchase = ifelse(prob_Purchase >= .20, 1, 0),
         pred_Purchase = as.factor(pred_Purchase))

table(test$Purchase, test$pred_Purchase)
```

```
##
##        0    1
##   0 4418  115
##   1  255   34
```

```
33/(33+111)
```

```
## [1] 0.2291667
```

About 23% of those predicted to make a purchase actually made the purchase

```
logistic <- glm(Purchase ~ ., data = train, family = "binomial")
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
log_probs <- predict(logistic, test, type = "response")
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading
```

```
test <- test %>%
  mutate(prob_Purchase_log = log_probs,
         pred_Purchase_log = ifelse(prob_Purchase_log >= .20, 1, 0),
         pred_Purchase_log = as.factor(pred_Purchase_log))

table(test$Purchase, test$pred_Purchase_log)
```

```
##
##         0    1
##   0 4183  350
##   1  231   58
```

```
58 / (58 + 350)
```

```
## [1] 0.1421569
```

Compared to the boosted tree model, the logisitc regression is worse when predicting a purchase. 14% of those predicted to make a purchase actually did.