Week4 Project

Kevin McBeth

February 12, 2018

Introduction and Data Cleaning

This week focuses on trees, random forest, and boosting. For this project, a random forest will be trained to predict whether or not a user will sign up for a term deposit with a bank. In order to create a model to identify individuals who should be called, the duration feature was removed. Other data points were converted to factors or numerics for ease of prediction. The days variable, due to it's spread, was converted to a three part factor based on time of the month, either early, mid, or late.

Following this data cleaning, a random subset was selected for a train set and test set. Looking at the correlation plot below, we see that none of the variables are strongly correlated to the output. This indicates that we are interested in looking at variable combinations, hinting that random forests might be a good approach in creating a model.

```
bank.dt <- fread(filename)
str(bank.dt)
```

```
## Classes 'data.table' and 'data.frame':   4521 obs. of  17 variables:
##  $ age      : int  30 33 35 30 59 35 36 39 41 43 ...
##  $ job      : chr  "unemployed" "services" "management" "management" ...
##  $ marital  : chr  "married" "married" "single" "married" ...
##  $ education: chr  "primary" "secondary" "tertiary" "tertiary" ...
##  $ default  : chr  "no" "no" "no" "no" ...
##  $ balance  : int  1787 4789 1350 1476 0 747 307 147 221 -88 ...
##  $ housing  : chr  "no" "yes" "yes" "yes" ...
##  $ loan     : chr  "no" "yes" "no" "yes" ...
##  $ contact  : chr  "cellular" "cellular" "cellular" "unknown" ...
##  $ day      : int  19 11 16 3 5 23 14 6 14 17 ...
##  $ month    : chr  "oct" "may" "apr" "jun" ...
##  $ duration : int  79 220 185 199 226 141 341 151 57 313 ...
##  $ campaign : int  1 1 1 4 1 2 1 2 2 1 ...
##  $ pdays    : int  -1 339 330 -1 -1 176 330 -1 -1 147 ...
##  $ previous : int  0 4 1 0 0 3 2 0 0 2 ...
##  $ poutcome : chr  "unknown" "failure" "failure" "unknown" ...
##  $ y        : chr  "no" "no" "no" "no" ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

```
summary(bank.dt)
```

```
##       age            job              marital           education
##  Min.   :19.00   Length:4521        Length:4521        Length:4521
##  1st Qu.:33.00   Class :character   Class :character   Class :character
##  Median :39.00   Mode  :character   Mode  :character   Mode  :character
##  Mean   :41.17
##  3rd Qu.:49.00
##  Max.   :87.00
##    default             balance        housing             loan
##  Length:4521        Min.   :-3313   Length:4521        Length:4521
##  Class :character   1st Qu.:   69   Class :character   Class :character
##  Mode  :character   Median :  444   Mode  :character   Mode  :character
```

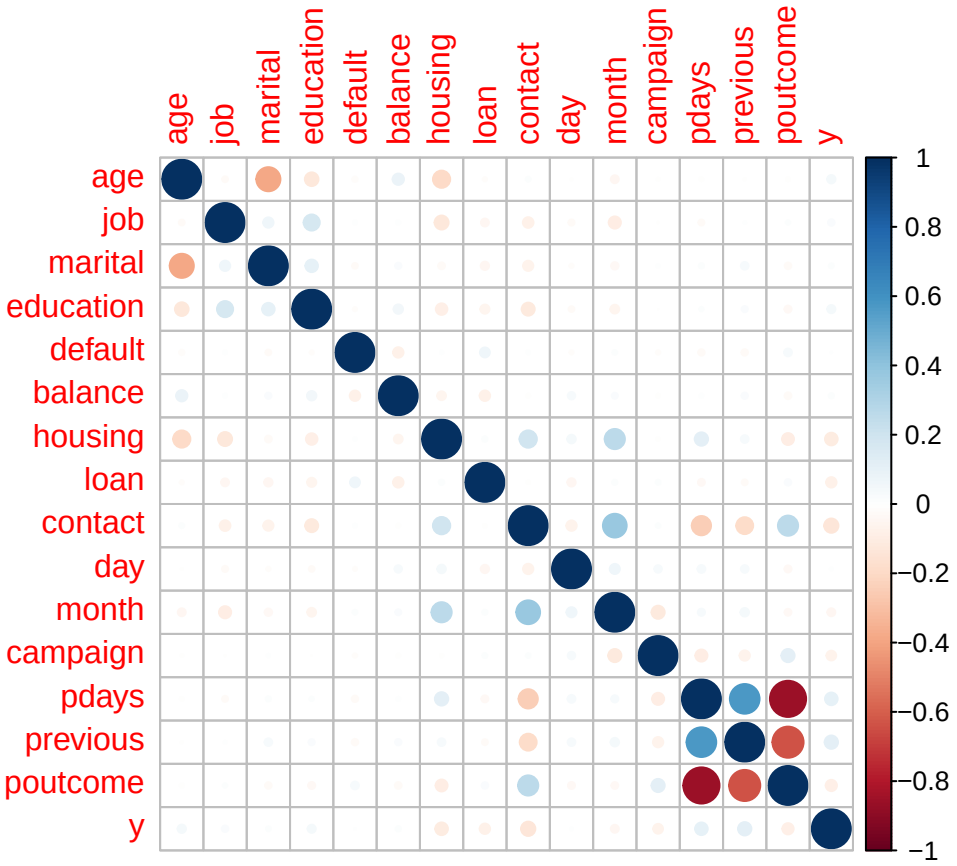
```
##                     Mean   : 1423
##                     3rd Qu.: 1480
##                     Max.   :71188
##    contact               day           month              duration
##  Length:4521        Min.   : 1.00   Length:4521        Min.   :   4
##  Class :character   1st Qu.: 9.00   Class :character   1st Qu.: 104
##  Mode  :character   Median :16.00   Mode  :character   Median : 185
##                     Mean   :15.92                      Mean   : 264
##                     3rd Qu.:21.00                      3rd Qu.: 329
##                     Max.   :31.00                      Max.   :3025
##     campaign          pdays           previous         poutcome
##  Min.   : 1.000   Min.   : -1.00   Min.   : 0.0000   Length:4521
##  1st Qu.: 1.000   1st Qu.: -1.00   1st Qu.: 0.0000   Class :character
##  Median : 2.000   Median : -1.00   Median : 0.0000   Mode  :character
##  Mean   : 2.794   Mean   : 39.77   Mean   : 0.5426
##  3rd Qu.: 3.000   3rd Qu.: -1.00   3rd Qu.: 0.0000
##  Max.   :50.000   Max.   :871.00   Max.   :25.0000
##       y
##  Length:4521
##  Class :character
##  Mode  :character
##
##
##
```

```
bank.cleanDataTypes <- bank.dt
bank.cleanDataTypes$age <- as.numeric(bank.dt$age)
bank.cleanDataTypes$balance <- as.numeric(bank.dt$balance)
bank.cleanDataTypes$duration <- as.numeric(bank.dt$duration)

bank.cleanDataTypes$job <- as.factor(bank.dt$job)
bank.cleanDataTypes$marital <- as.factor(bank.dt$marital)
bank.cleanDataTypes$education <- as.factor(bank.dt$education)
bank.cleanDataTypes$default <- as.factor(bank.dt$default)
bank.cleanDataTypes$housing <- as.factor(bank.dt$housing)
bank.cleanDataTypes$loan <- as.factor(bank.dt$loan)
bank.cleanDataTypes$contact <- as.factor(bank.dt$contact)
bank.cleanDataTypes$y <- as.factor(bank.dt$y)
bank.cleanDataTypes$poutcome <- as.factor(bank.dt$poutcome)
bank.cleanDataTypes$campaign <- as.factor(bank.dt$campaign)
bank.cleanDataTypes$pdays <- as.numeric(bank.dt$pdays)
bank.cleanDataTypes$month <- as.factor(bank.dt$month)
bank.cleanDataTypes$previous <- as.numeric(bank.dt$previous)


day <- character(nrow(bank.cleanDataTypes))


day[which(bank.cleanDataTypes$day <= 10)] <- "early"
day[which(bank.cleanDataTypes$day <= 20 & bank.cleanDataTypes$day > 10)] <- "mid"
day[which(bank.cleanDataTypes$day > 20)] <- "late"
day <- as.factor(day)
bank.cleanDataTypes$day <- day
bank.cleanDataTypes$duration <- NULL
```

```
bank.dt.numeric <- bank.cleanDataTypes[,lapply(.SD, as.numeric)]
str(bank.dt.numeric)
```

```
## Classes 'data.table' and 'data.frame':   4521 obs. of  16 variables:
##  $ age      : num  30 33 35 30 59 35 36 39 41 43 ...
##  $ job      : num  11 8 5 5 2 5 7 10 3 8 ...
##  $ marital  : num  2 2 3 2 2 3 2 2 2 2 ...
##  $ education: num  1 2 3 3 2 3 3 2 3 1 ...
##  $ default  : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ balance  : num  1787 4789 1350 1476 0 ...
##  $ housing  : num  1 2 2 2 2 1 2 2 2 2 ...
##  $ loan     : num  1 2 1 2 1 1 1 1 1 2 ...
##  $ contact  : num  1 1 1 3 3 1 1 1 3 1 ...
##  $ day      : num  3 3 3 1 1 2 3 1 3 3 ...
##  $ month    : num  11 9 1 7 9 4 9 9 9 1 ...
##  $ campaign : num  1 1 1 4 1 2 1 2 2 1 ...
##  $ pdays    : num  -1 339 330 -1 -1 176 330 -1 -1 147 ...
##  $ previous : num  0 4 1 0 0 3 2 0 0 2 ...
##  $ poutcome : num  4 1 1 4 4 1 2 4 4 1 ...
##  $ y        : num  1 1 1 1 1 1 1 1 1 1 ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

```
corrplot(cor(bank.dt.numeric))
```




```
tr.idxs <- createDataPartition(bank.cleanDataTypes$y, p = 0.8)$Resample1
train <- bank.cleanDataTypes[tr.idxs]
```

```
test <- bank.cleanDataTypes[-tr.idxs]
```

Model Training and Prediction Accuracy

Looking at our model's performance, we see that the best value for the tuned hyperparameter is mtry = 12. While this is useful, it would be even more useful if caret could tune things such as the nodesize, which is helpful in affecting overfitting. Too small of leaf nodes can result in overfitting, which is unfavorable for our model. Looking at the performance, we note that a node size of around 8 items yields a training accuracy, or r2 value of .92, and a test accuracy of .89. This indicates that our model is performing appropriately, and not under nor overfitting, as evidenced by similar r2 values.

```
trControl <- trainControl(method = 'repeatedcv',
                          number = 3,
                          repeats = 4)

rf.model <- train(y ~ .,
                  data = train,
                  method = 'rf',
                  trControl = trControl,
                  ntree = 250,
                  nodesize = 8,

                  tuneGrid = expand.grid(mtry = c(4, 8, 12)))


rf.model
```

```
## Random Forest
##
## 3617 samples
##   15 predictor
##    2 classes: 'no', 'yes'
##
## No pre-processing
## Resampling: Cross-Validated (3 fold, repeated 4 times)
## Summary of sample sizes: 2411, 2412, 2411, 2411, 2412, 2411, ...
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa
##    4    0.8855405  0.03359145
##    8    0.8885124  0.14706767
##   12    0.8889965  0.17996020
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 12.
```

```
tr.preds <- predict(rf.model, train)
postResample(tr.preds, train$y)
```

```
##  Accuracy     Kappa
## 0.9236937 0.4793485
```

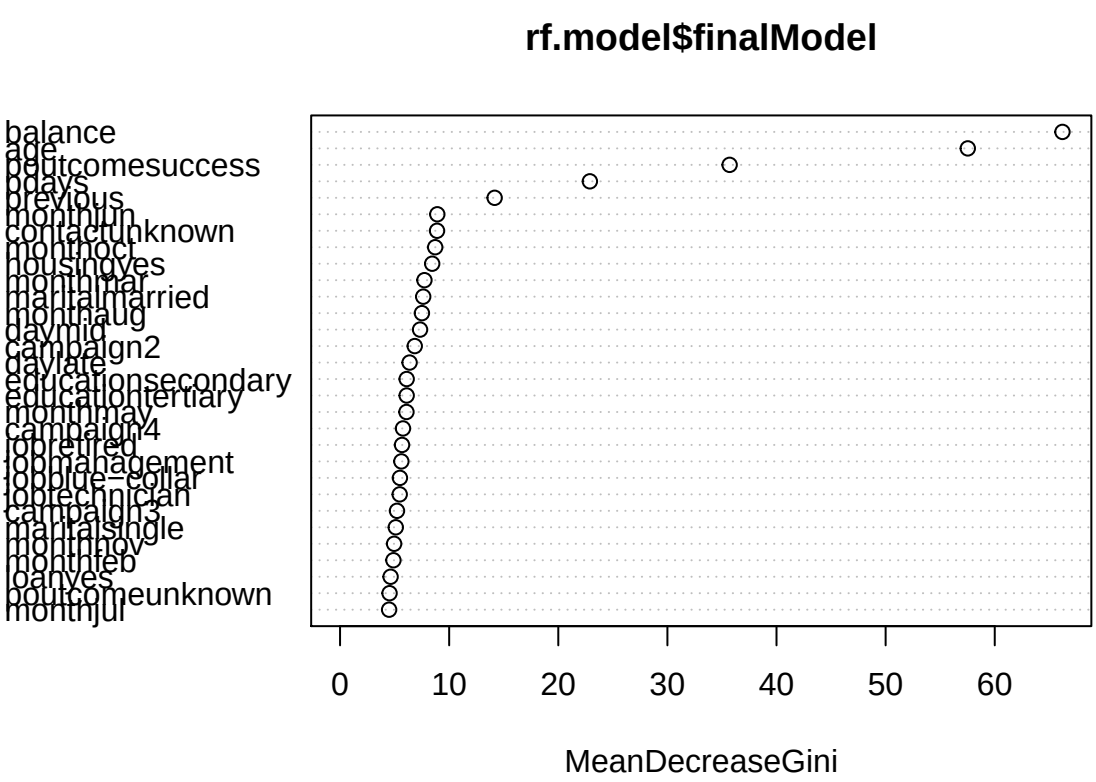
```
te.preds <- predict(rf.model, test)
postResample(te.preds, test$y)
```

```
##  Accuracy     Kappa
## 0.8938053 0.2354102
```

Importance

Before removing our phone duration value, which was strongly correlated to the output variable, the importance of the duration value was significantly outweighing the importance of the other factors. Here we see that the most important parameters in driving a user towards choosing yes or no is the individuals balance, age, outcome of the previous marketing campaign, and the number of days since the client was previously contacted.

```
## rf variable importance
##
##   only 20 most important variables shown (out of 72)
##
##                     Overall
## balance              66.206
## age                  57.530
## poutcomesuccess      35.702
## pdays                22.894
## previous             14.167
## monthjun              8.913
## contactunknown        8.888
## monthoct              8.719
## housingyes            8.443
## monthmar              7.731
## maritalmarried        7.620
## monthaug              7.503
## daymid                7.331
## campaign2             6.846
## daylate               6.372
## educationsecondary    6.104
## educationtertiary     6.104
## monthmay              6.084
## campaign4             5.766
## jobretired            5.679
```


rf.model$finalModel
balance
age
poutcomesuccess
pdays
previous
monthjun
contactunknown
monthoct
housingyes
monthmar
maritalmarried
monthaug
daymid
campaign2
daylate
educationsecondary
educationtertiary
monthmay
campaign4
jobretired
jobmanagement
jobblue−collar
jobtechnician
campaign3
maritalsingle
monthnov
monthfeb
loanyes
poutcomeunknown
monthjul
0
10
20
30
40
50
60
MeanDecreaseGini