BART Model

Saptarshi Maiti (A20447671)

12/4/2020

#Load Data

```
filepath <- '/Users/saptarshimaiti/Desktop/Statistical Learning/Project/Project/Data/'
df_train <- read.csv(file = paste0(filepath, "H1.csv"),sep=",",stringsAsFactors = FALSE, na.strings = "
df_test <- read.csv(file = paste0(filepath, "H2.csv"),sep=",",stringsAsFactors = FALSE, na.strings = "NU
```

#Data Transformation for training

```
leadtime<-as.numeric(df_train$LeadTime)
country<-as.numeric(factor(df_train$Country))
marketsegment<-as.numeric(factor(df_train$MarketSegment))
deposittype<-as.numeric(factor(df_train$DepositType))
customertype<-as.numeric(factor(df_train$CustomerType))
rcps<-as.numeric(df_train$RequiredCarParkingSpaces)
week<-as.numeric(df_train$ArrivalDateWeekNumber)
IsCanceled<-as.numeric(factor(df_train$IsCanceled))

IsCanceled[IsCanceled == "1"] <- "0"
IsCanceled[IsCanceled == "2"] <- "1"

df_train <- data.frame(leadtime,country,marketsegment,deposittype,customertype,rcps,week,IsCanceled)
```

#Training and Validation Split

```
library(caret)
```

```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

```
trainIndex = createDataPartition(df_train$IsCanceled, p = .8, list = FALSE)
val <- df_train[-trainIndex, -length(df_train)]
IsCanceled_val <- df_train[-trainIndex, length(df_train)]
train <- df_train[trainIndex, -length(df_train)]
IsCanceled_train <- df_train[trainIndex, length(df_train)]
```

Bart Model Training

```
options(java.parameters = "-Xmx40g")
library(bartMachine)
```

```
## Loading required package: rJava
```

```
## Loading required package: bartMachineJARs
```

```
## Loading required package: randomForest
## randomForest 4.6-14
## Type rfNews() to see new features/changes/bug fixes.
##
## Attaching package: 'randomForest'
## The following object is masked from 'package:ggplot2':
##
##     margin
## Loading required package: missForest
## Loading required package: foreach
## Loading required package: itertools
## Loading required package: iterators
## Welcome to bartMachine v1.2.5.1! You have 38.18GB memory available.
##
## If you run out of memory, restart R, and use e.g.
## 'options(java.parameters = "-Xmx5g")' for 5GB of RAM before you call
## 'library(bartMachine)'.
set_bart_machine_num_cores(4)
## bartMachine now using 4 cores.
bart_machine <- bartMachine(X = train, y = IsCanceled_train, use_missing_data = T, num_iterations_after
## bartMachine initializing with 50 trees...
## bartMachine vars checked...
## bartMachine java init...
## bartMachine factors created...
## bartMachine before preprocess...
## bartMachine after preprocess... 8 total features...
## bartMachine sigsq estimated...
## bartMachine training data finalized...
## Now building bartMachine for classification where "0" is considered the target level...Missing data
## evaluating in sample data...done
summary(bart_machine)
## bartMachine v1.2.5.1 for classification
##
## Missing data feature ON
## training data n = 32049 and p = 7
## built in 51 secs on 4 cores, 50 trees, 100 burn-in and 500 post. samples
##
## confusion matrix:
##
##            predicted 0 predicted 1 model errors
## actual 0     21336.000    1815.000        0.078
## actual 1      3207.000    5691.000        0.360
## use errors       0.131       0.242        0.157
```

Bart Model Validation

```
prediction_val <- predict(bart_machine, val, type = 'class')
```

Validation Missclassification Rate

```
val_missclassification <-sum(prediction_val!=IsCanceled_val)/length(prediction_val)
val_missclassification
```

```
## [1] 0.1607789
```

Validation Confusion Matrix

```
confusionMatrix(as.factor(prediction_val), as.factor(IsCanceled_val), positive="1")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 5302  803
##          1  485 1421
##
##                Accuracy : 0.8392
##                  95% CI : (0.831, 0.8472)
##     No Information Rate : 0.7224
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.5807
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.6389
##             Specificity : 0.9162
##          Pos Pred Value : 0.7455
##          Neg Pred Value : 0.8685
##              Prevalence : 0.2776
##          Detection Rate : 0.1774
##    Detection Prevalence : 0.2379
##       Balanced Accuracy : 0.7776
##
##        'Positive' Class : 1
##
```

Testing

Data Transformation

```
leadtime<-as.numeric(df_test$LeadTime)
country<-as.numeric(factor(df_test$Country))
marketsegment<-as.numeric(factor(df_test$MarketSegment))
deposittype<-as.numeric(factor(df_test$DepositType))
customertype<-as.numeric(factor(df_test$CustomerType))
```

```
rcps<-as.numeric(df_test$RequiredCarParkingSpaces)
week<-as.numeric(df_test$ArrivalDateWeekNumber)
IsCanceled_test<-as.numeric(factor(df_test$IsCanceled))

IsCanceled_test[IsCanceled_test == "1"] <- "0"
IsCanceled_test[IsCanceled_test == "2"] <- "1"

df_test <- data.frame(leadtime,country,marketsegment,deposittype,customertype,rcps,week)
```

Prediction

```
prediction_test <- predict(bart_machine, df_test, type = "class")
```

Missclassification Rate

```
pred_missclassification_test <-sum(prediction_test!=IsCanceled_test)/length(prediction_test)
pred_missclassification_test
```

```
## [1] 0.2692424
```

```
nrow(df_test)
```

```
## [1] 79330
```

```
nrow(df_train)
```

```
## [1] 40060
```

Test Confusion Matrix

```
confusionMatrix(as.factor(prediction_test), as.factor(IsCanceled_test), positive="1")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     0     1
##          0 43216 18347
##          1  3012 14755
##
##                Accuracy : 0.7308
##                  95% CI : (0.7277, 0.7338)
##     No Information Rate : 0.5827
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.4074
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.4457
##             Specificity : 0.9348
##          Pos Pred Value : 0.8305
##          Neg Pred Value : 0.7020
##              Prevalence : 0.4173
##          Detection Rate : 0.1860
##    Detection Prevalence : 0.2240
##       Balanced Accuracy : 0.6903
```

```
##
##        'Positive' Class : 1
##
```