Problemset_Module_7

Matt He

2022-11-09

Reading Data

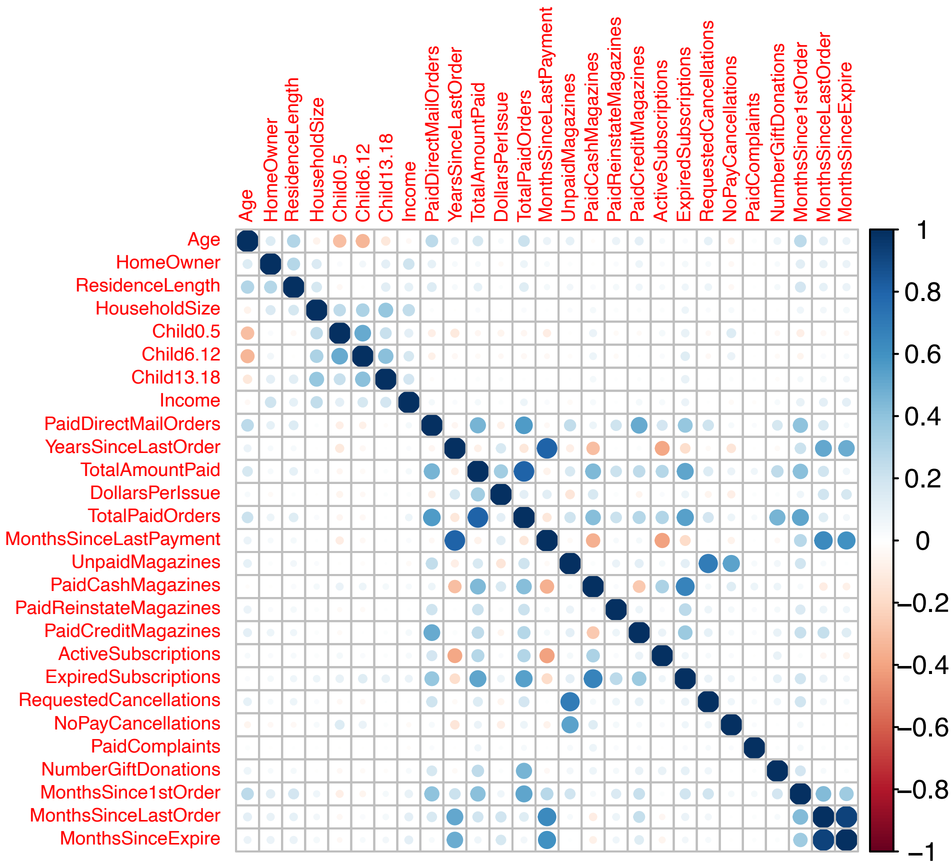
##second

```
grey <- read_csv('~/Downloads/Grey.csv')
```

```
## Rows: 42077 Columns: 38
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (9): DwellingType, Gender, Marital, ChildPresent, Occupation, MagazineS...
## dbl (29): CustomerID, Age, HomeOwner, ResidenceLength, HouseholdSize, Child0...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
grey <- grey%>%
  mutate_at(vars(DwellingType, Gender, Marital, ChildPresent,
                 Occupation, HomeValue, MagazineStatus, LastPaymentType,
                 GiftDonor), .funs = factor) %>%
  mutate(Renewal = factor(ifelse(Renewal=="Yes", 1, 0)))
```

```
grey %>%
  select(-CustomerID) %>%
  keep(is.numeric) %>%
  cor() %>%
  corrplot::corrplot(tl.cex = 0.6)
```



By looking at this corrplot, we can discover that there are some variables that relationships are too strong to each other. For example, `MonthsSinceLastPayment` and `YearsSinceLastOrder` have too strong correlation with each other. According to the name, we can also find out that these two are similar things but measured in different period. `TotalPaidOrders` and `TotalAmountPaid` are also quite close. `UnpaidMagaines` and `PaidCashMagazines` are too strong.If you paid with cash, that means you it is paid, the opposite of unpaid. `MonthSinceLastOrder` and `MonthSinceExpire` have an extremely strong relationship. Last but not least, `RequestedCancellations` and `UnpaidMagazines` have strong correlation, I am assuming if is because the custmor wants to cancel so the next magazine is not paid.

Therefore, we should take off these variables.

```
grey = grey %>%
  select(-TotalPaidOrders, -YearsSinceLastOrder, -MonthsSinceLastPayment,
         -MonthsSinceExpire, -UnpaidMagazines, -ChildPresent, -MonthsSince1stOrder)
```

##2. 2. Experimented with various classification methods (random forest, gradient boosting machine) and compared performance using proper evaluation methods, proper use of accuracy measures and visualizations with ROC curves, Precision-Recall curves, and a Lift chart. You still need to select a final model to recommend for the purpose of identifying customers who will respond to the targeted marketing .

First, we need to do some data partitioning.

```
set.seed(596)
samp = createDataPartition(grey$Renewal,
                           p = 0.7,
                           list = FALSE)
train = grey[samp, ]
test = grey[-samp, ]
```

We can see here the number of positive samples and negative samples are not balanced

```
summary(train$Renewal)
```

```
##     0     1
## 28829   626
```

So we need to balance `Renewal`:

```
library(performanceEstimation)
set.seed(1122)
train.bal = smote(Renewal ~ .,
                  data = train,
                  perc.over = 2,
                  perc.under = 1.5)
summary(train.bal$Renewal)
```

```
##    0    1
## 1878 1878
```

Train the model:

```
rf_model = readRDS("~/Downloads/GCC_rf_model.rds")
gbm_model = readRDS("~/Downloads/GCC_gbm_model.rds")
```

Make prediction:

```
rf_pred = predict(rf_model, newdata = test, type = "raw")
gbm_pred= predict(gbm_model, newdata = test, type = "raw")

rf_pred_prob = predict(rf_model, newdata = test, type = "prob")[,2]
gbm_pred_prob = predict(gbm_model, newdata = test, type = "prob")[,2]
```

Set up confusion matrices for both models:

```
rf_cm = confusionMatrix(rf_pred, test$Renewal, positive = "1")
gbm_cm = confusionMatrix(gbm_pred, test$Renewal, positive = "1")
```

Let's take look at the confusion matrices.

```
rf_cm
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     0     1
##          0 11699    86
##          1   655   182
```

```
##
##                Accuracy : 0.9413
##                  95% CI : (0.937, 0.9453)
##     No Information Rate : 0.9788
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.3071
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.67910
##             Specificity : 0.94698
##          Pos Pred Value : 0.21744
##          Neg Pred Value : 0.99270
##              Prevalence : 0.02123
##          Detection Rate : 0.01442
##    Detection Prevalence : 0.06631
##       Balanced Accuracy : 0.81304
##
##        'Positive' Class : 1
##
```

```
gbm_cm
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     0     1
##          0 11897    98
##          1   457   170
##
##                Accuracy : 0.956
##                  95% CI : (0.9523, 0.9595)
##     No Information Rate : 0.9788
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.3609
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.63433
##             Specificity : 0.96301
##          Pos Pred Value : 0.27113
##          Neg Pred Value : 0.99183
##              Prevalence : 0.02123
##          Detection Rate : 0.01347
##    Detection Prevalence : 0.04968
##       Balanced Accuracy : 0.79867
##
##        'Positive' Class : 1
##
```