Santander Customer Transaction Prediction

Ravi Shankar Vats

Problem Overview

- This is a project given by Edwisor as a part of Assignment

- 1.5 years of customers behavior data from Santander bank to predict what new products customers will purchase.
- Columns 1 - 24 are customer information; 25 - 48 are products purchased
- The test and train sets are split by time, and public and private leaderboard sets are split randomly.
- Training data over 13 million rows; Test data about 1M
- **GOAL: predict which products existing customers will buy but here we have considered product prediction based on features of existing customers.**

Problem - Solving Strategy

Exploratory Analysis (John)

- Market Basket Analysis of training data (20K rows)
- Concept: use arules package in R to use apriori algorithm for association rules (example: {item 1, item 2} ➔ {item 3})

Prediction (Sri)

- Use boosting to predict product purchases
- In R, we use XGBoost package, which is good for product recommendations (has won many Kaggle competitions before)

Exploratory Analysis (Market Basket Analysis)

Savings Account	Guarantees	Current Accounts	Derivada	Payroll Account	Junior	Mas Particular
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	0	1	0	0	0	0
0	0	1	0	0	0	0
0	0	0	0	0	0	0
0	0	1	0	0	0	0

	V1	V2	V3	V4
2245	Current Accounts			
2246	Current Accounts			
2247	Current Accounts			
2248	Current Accounts	e-account	Direct Debit	
2249	Current Accounts			
2250	Current Accounts			
2251	Current Accounts			
2252	Current Accounts			
2253	Current Accounts			

Step 1: Convert Data Frame into a Sparse Matrix (I did this in Python). This allows the arules package to work

Exploratory Analysis (Market Basket Analysis)

Bank2 <- read.transactions("santander3.csv",sep=",")

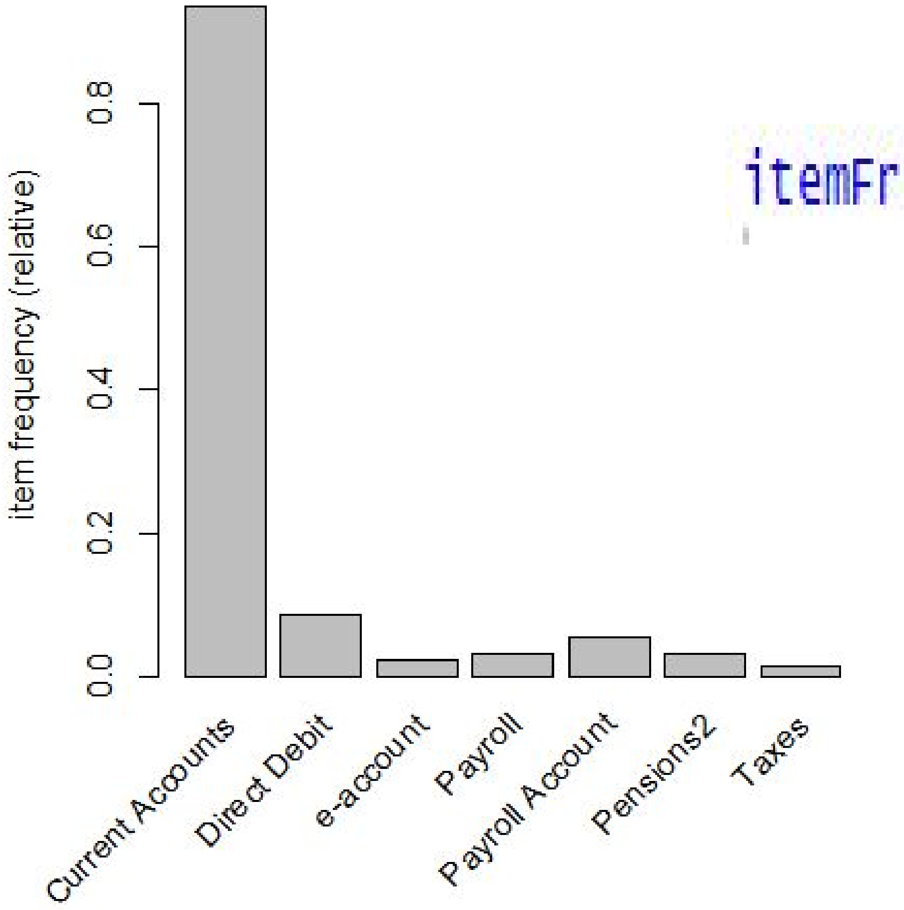
summary(Bank2)

```
transactions as itemMatrix in sparse format with
 16785 rows (elements/itemsets/transactions) and
 16 columns (items) and a density of 0.07491436

most frequent items:
Current Accounts     Direct Debit  Payroll Account         Pensions2          Payroll          (Other)
           15674             1475              924               546              520              980

element (itemset/transaction) length distribution:
sizes
    1     2     3     4     5     6     7
14855  1174   328   257   126    41     4

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.000   1.000   1.000   1.199   1.000   7.000
```

Exploratory Analysis (Market Basket Analysis)

```
itemFrequency(Bank2[,1:6])
   Credit Card Current Accounts        Derivada     Direct Debit       e-account           Funds
   0.008459934      0.933809949     0.000119154      0.087876080     0.022222222     0.001608579
```

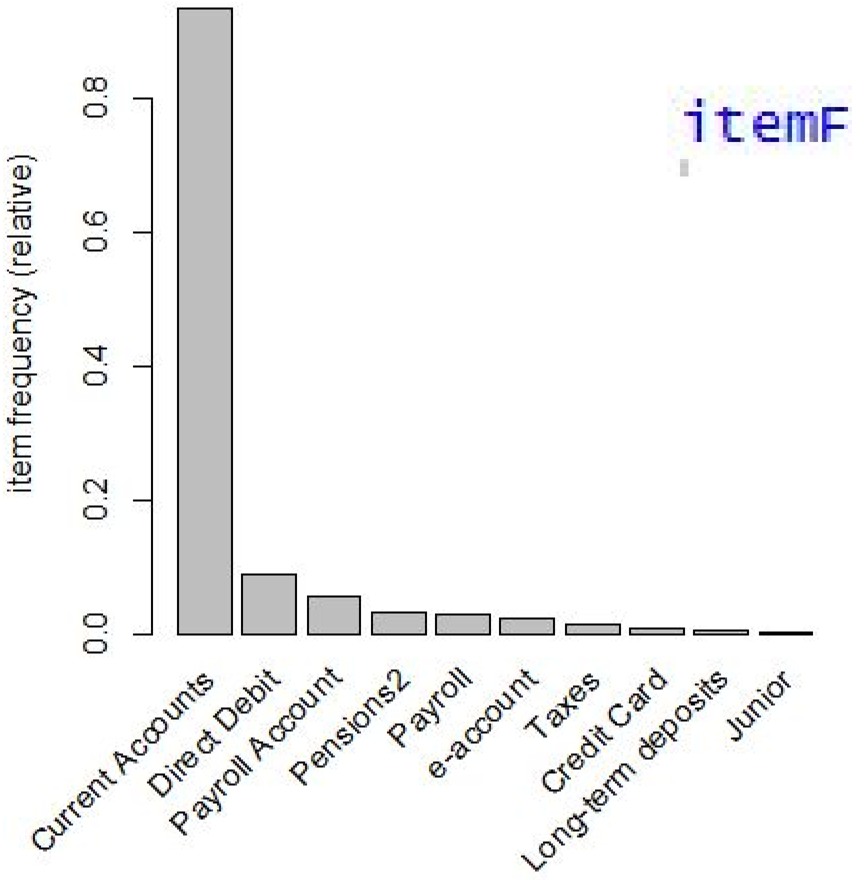
Exploratory Analysis (Market Basket Analysis)




```
itemFrequencyPlot(Bank2,support=.01)
```

- This means plot items with support >= 0.01

Exploratory Analysis (Market Basket Analysis)




```
itemFrequencyPlot(Bank2, topN=10)
```

Exploratory Analysis (Market Basket Analysis)

```r
m1 <- apriori(Bank2,parameter=list(support=0.003,confidence=0.9,minlen=2))
inspect(sort(m1,by="lift")[1:10]) #top 10 rules by lift
```

```
     lhs                                                    rhs         support     confidence lift
[1]  {Direct Debit,e-account,Pensions2}                  => {Payroll}   0.003932082 0.9850746  31.79707
[2]  {Direct Debit,e-account,Payroll Account,Pensions2}  => {Payroll}   0.003693774 0.9841270  31.76648
[3]  {Direct Debit,Pensions2}                            => {Payroll}   0.019898719 0.9709302  31.34051
[4]  {Direct Debit,Payroll Account,Pensions2}            => {Payroll}   0.019124218 0.9697885  31.30365
[5]  {e-account,Pensions2}                               => {Payroll}   0.005064045 0.9659091  31.17843
[6]  {Direct Debit,Pensions2,Taxes}                      => {Payroll}   0.003157581 0.9636364  31.10507
[7]  {e-account,Payroll Account,Pensions2}               => {Payroll}   0.004706583 0.9634146  31.09791
[8]  {Payroll}                                           => {Pensions2} 0.030980042 1.0000000  30.74176
[9]  {Pensions2}                                         => {Payroll}   0.030980042 0.9523810  30.74176
[10] {Credit Card,Payroll}                               => {Pensions2} 0.003336312 1.0000000  30.74176
```

Support: the proportion of item(s) in the dataset.

Equation: number of occurences of item X / number of items in dataset

Confidence: the likelihood of an item Z is purchased given items X,Y purchased

Equation: conf (X -> Y) = support (x U y) / support (X)

Lift: how much more likely an item is to be purchased with these other items than by itself

Exploratory Analysis (Market Basket Analysis)

```
#subsetting--looking for specific items in rules
e_account_rules <- subset(m1, items %in% "e-account")
inspect(e_account_rules)
```

```
     lhs                                                    rhs               support     confidence lift
[1]  {e-account,Payroll}                                 => {Pensions2}       0.005064045 1.0000000  30.74176
[2]  {e-account,Pensions2}                               => {Payroll}         0.005064045 0.9659091  31.17843
[3]  {e-account,Payroll}                                 => {Payroll Account} 0.004706583 0.9294118  16.88331
[4]  {e-account,Pensions2}                               => {Payroll Account} 0.004885314 0.9318182  16.92702
[5]  {e-account,Payroll,Pensions2}                       => {Payroll Account} 0.004706583 0.9294118  16.88331
[6]  {e-account,Payroll,Payroll Account}                 => {Pensions2}       0.004706583 1.0000000  30.74176
[7]  {e-account,Payroll Account,Pensions2}               => {Payroll}         0.004706583 0.9634146  31.09791
[8]  {Direct Debit,e-account,Payroll}                    => {Pensions2}       0.003932082 1.0000000  30.74176
[9]  {Direct Debit,e-account,Pensions2}                  => {Payroll}         0.003932082 0.9850746  31.79707
[10] {Direct Debit,e-account,Payroll}                    => {Payroll Account} 0.003693774 0.9393939  17.06464
[11] {Direct Debit,e-account,Pensions2}                  => {Payroll Account} 0.003753351 0.9402985  17.08107
[12] {Direct Debit,e-account,Payroll,Pensions2}          => {Payroll Account} 0.003693774 0.9393939  17.06464
[13] {Direct Debit,e-account,Payroll,Payroll Account}    => {Pensions2}       0.003693774 1.0000000  30.74176
[14] {Direct Debit,e-account,Payroll Account,Pensions2}  => {Payroll}         0.003693774 0.9841270  31.76648
```

Prediction with Xgboost - eXtreme Gradient Boosting

- Supervised Learning Technique
- Prediction based on set of weak ensemble models to get a strong model.
- Tries to optimize the differentiable loss function(cost associated with each misclassification)

Objective: Prediction of a customer opening a Current Account based on the existing customer features (Test 10K rows and Train 10K rows)

Features:
Sex,Age,RelAtBegMOnth,ResIndex,ForIndex,Channel,ProvinceCode,ActivityIndex,Income,Segment

Dependent Variable: CurrentAccount

XGBoost on Santander Data

1. **Data Cleanup & Convert all the features into a numeric matrix**

   santanderTrain$Sex <- as.numeric(santanderTrain$Sex)

2. **Tuning Parameters**

```
#setting the parameter for Cross validation and XGBoost
param       = list("objective" = "binary:logistic", # binary classification
                   "num_class"= 2 ,     # Number of classes in the dependent variable.
                   "eval_metric" = "mlogloss",     # evaluation metric
                   "nthread" = 8,          # number of threads to be used
                   "max_depth" = 2,        # maximum depth of tree
                   "eta" = 0.3,           # step size shrinkage
                   "gamma" = 0,           # minimum loss reduction
                   "subsample" = 0.7,         # part of data instances to grow tree
                   "colsample_bytree" = 1,     # subsample ratio of columns when constructing each tree
                   "min_child_weight" = 12       # minimum sum of instance weight needed in a child
)
```

3. Predictor Selection & Numeric Label

```
#Identify the predictors and the dependent variable
predictors <- colnames(santanderTrain[-ncol(santanderTrain)])
label <- as.numeric(santanderTrain[,ncol(santanderTrain)])
print(table(label))
length(label)
```

4. Used Cross Validation to identify the best minimal loss

```
cv.nround = 200;

bst.cv = xgb.cv(
  param=param,
  data = as.matrix(santanderTrain[,predictors]),
  label = label,
  nfold = 3,
  nrounds=cv.nround,
  prediction=T)

################ Get the minimum logloss####

min.loss.idx = which.min(bst.cv$dt[,test.mlogloss.mean])
cat ("Minimum logloss occurred in round : ", min.loss.idx, "\n")

print(bst.cv$dt[min.loss.idx,])
```

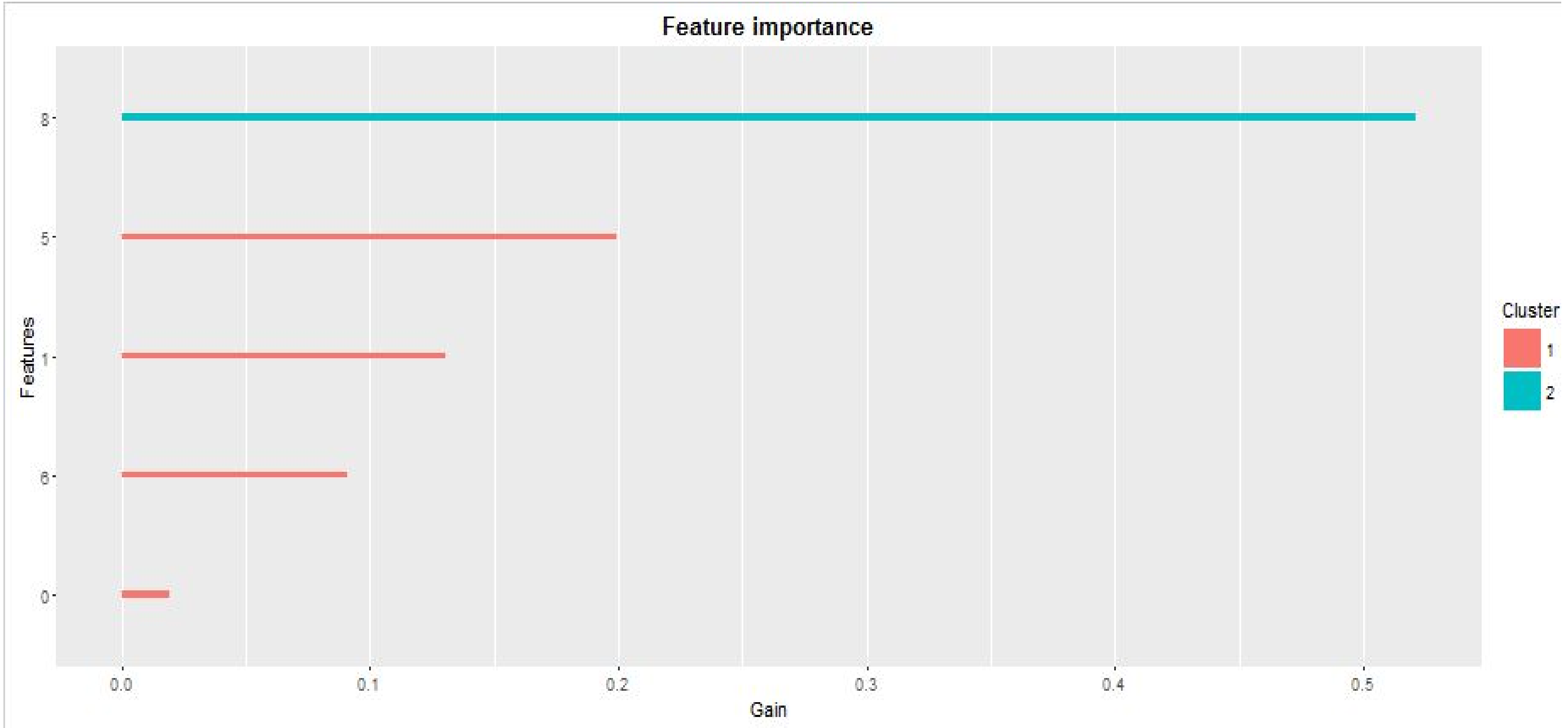
```
> min.loss.idx
[1] 18
> bst.cv$dt[min.loss.idx,]
   train.mlogloss.mean train.mlogloss.std test.mlogloss.mean test.mlogloss.std
1:            0.274907           0.004019           0.275241          0.008179
```

5. Train the model

```
> bst = xgboost(
+    param=param,
+    data =as.matrix(santanderTrain[,predictors]),
+    label = label,
+    nrounds=min.loss.idx)
[0]	train-mlogloss:0.551663
[1]	train-mlogloss:0.463063
[2]	train-mlogloss:0.405943
[3]	train-mlogloss:0.365365
[4]	train-mlogloss:0.337984
[5]	train-mlogloss:0.319499
[6]	train-mlogloss:0.306983
[7]	train-mlogloss:0.297759
[8]	train-mlogloss:0.287817
[9]	train-mlogloss:0.282251
[10]	train-mlogloss:0.279499
[11]	train-mlogloss:0.277898
[12]	train-mlogloss:0.276564
[13]	train-mlogloss:0.275641
[14]	train-mlogloss:0.276720
[15]	train-mlogloss:0.275977
[16]	train-mlogloss:0.275237
[17]	train-mlogloss:0.276514
```

6. Predict the results

```
> predictedValue <- predict(bst, as.matrix(santanderTest[,predictors]))
> head(predictedValue)
[1] 1 0 1 1 1 1
> santanderTestPredicted <- santanderTest
> santanderTestPredicted <- cbind(santanderTestPredicted,predictedValue)
> head(santanderTestPredicted)
     Sex Age RelAtBegMonth ResIndex ForIndex Channel ProvinceCode ActivityIndex Income Segment
[1,]   2  52             2        2        1      19           28             0   6731       3
[2,]   2  16             1        2        1      35           28             1   7629       3
[3,]   2  68             2        2        1      29           28             0   3918       3
[4,]   1  42             1        2        1      19           11             1   1478       3
[5,]   2  40             1        2        1      19           48             1      1       2
[6,]   2  41             1        2        2      35           50             0   3014       3
     predictedValue
[1,]              1
[2,]              0
[3,]              1
[4,]              1
[5,]              1
[6,]              1
```



Feature importance
8
5
1
6
0
Features
0.0
0.1
0.2
0.3
0.4
0.5
Gain
Cluster
1
2

Conclusion(s)

From market basket analysis / association rules, we learned that

- if a Santander customer has Direct Debit, e-account, and pensions, they will also likely have a Payroll Account.
- Clearly, these people are employed, and likely to use Santander in the future, so long as they are happy customers.

From xgboost, we learned that

- Efficient in predicting the customer product with the set of provided features.
- With the results of XGBoost, we found that Gross Income is a major feature that is used in classifying Cluster 1(No Current Account) and Channel, Age and Province Code are important in classifying Cluster 2(Potential Customers)
- Validation(sum(abs(pred-orig) upon train data provided an accuracy of 80% in prediction.