BLACKWELL ELECTRONICS

Brand preference prediction

Blackwell
Electronics

Maja Jurcan
maja.jurcan@gmail.com

Table of Contents

Goals

One of the objectives of the survey was to find out which of the two brands of computers Blackwell Electronics customers prefer. This information will be helpful when deciding with which manufacturer to pursue a deeper strategic relationship. Unfortunately, the answer to the brand preference question was not properly captured for all of the survey respondents.

The goal was to investigate if customer responses to some survey questions (e.g. income, age, etc.) enable us to predict the answer to the brand preference question. If it can be done with large enough level of confidence those predictions will provide the sales team with a complete view of what brand Blackwell customers prefer.

To do this we will run and optimize at least two different classification methods in R - *k*-nearest-neighbor and a decision tree - and compare which one works better for this data set.

Overview of the data

The dataset consists of 7 attributes:

- Salary – yearly salary in numeric format, not including bonuses
- Age – age in numeric format
- Education level – 5 levels of education, from less than high school to masters or doctoral
- Car – the make of customer's primary car (20 manufacturers to choose from)
- Zip code – 9 regions of the U.S.
- Credit – amount of available credit
- Brand – preference between Sony and Acer computers; 0 – Acer, 1 - Sony

The exploration of data showed that the most relevant attribute for brand preference is salary while all the others attributes have very low or non-existent connection to brand preference.

Methodology

After the initial exploration of data through looking at the type, minimums, maximums, averages and initial plots and histograms, the chi squared test was used to determine the attribute relevancy towards the brand attribute. As mentioned before, the test showed that salary is the only attribute in the survey with relevant connection to the brand preference.

```
> res <- gain.ratio(brand~., SurveyData)
> res
        attr_importance
salary     9.019048e-02
age        0.000000e+00
elevel     2.045922e-05
car        2.038119e-04
zipcode    1.689241e-04
credit     0.000000e+00
> res2 <- chi.squared(brand~., SurveyData)
> res2
        attr_importance
salary      0.498927279
age         0.000000000
elevel      0.008118016
car         0.034958140
zipcode     0.027182372
credit      0.000000000
```

The data had to be slightly transformed in order to make our classification models run smoothly. Education level, car, zip code and brand attributes were all transformed from numerical to factor data.

The decision tree was built first to further explore attribute relevancy to our brand preference. The data was split in train and test sets (75:25). Also, the 10 fold cross validation was used across all of the algorithms. After that, 3 classification algorithms were run (KNN, RF, SVM) in order to build our prediction models.

The relevant metrics (accuracy and kappa) were used to determine the level of confidence. After going through all the metrics and outputs, the best model was chosen.

The chosen model was built with the random forest algorithm ("rf" from caret package):

```
> RFFit
Random Forest

7501 samples
   6 predictor
   2 classes: '0', '1'

Pre-processing: centered (34), scaled (34)
Resampling: Cross-Validated (10 fold, repeated 6 times)
Summary of sample sizes: 6751, 6750, 6751, 6751, 6750, 6750, ...
Resampling results across tuning parameters:

  mtry  Accuracy   Kappa
   2    0.6216727  7.292306e-05
  18    0.9230755  8.366354e-01
  34    0.9181209  8.261324e-01

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was mtry = 18.
```

Confusion matrix (with accuracy and kappa):

```
> confusionMatrix(predictionRF, testSet$brand)
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0  860  107
         1   85 1447

               Accuracy : 0.9232
                 95% CI : (0.912, 0.9333)
    No Information Rate : 0.6218
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.8374
 Mcnemar's Test P-Value : 0.1296

            Sensitivity : 0.9101
            Specificity : 0.9311
         Pos Pred Value : 0.8893
         Neg Pred Value : 0.9445
             Prevalence : 0.3782
         Detection Rate : 0.3441
   Detection Prevalence : 0.3870
      Balanced Accuracy : 0.9206

       'Positive' Class : 0
```

Results

Applying the trained and tested model to the incomplete survey dataset showed that Blackwell Electronics customers prefer the Sony brand (60:40).

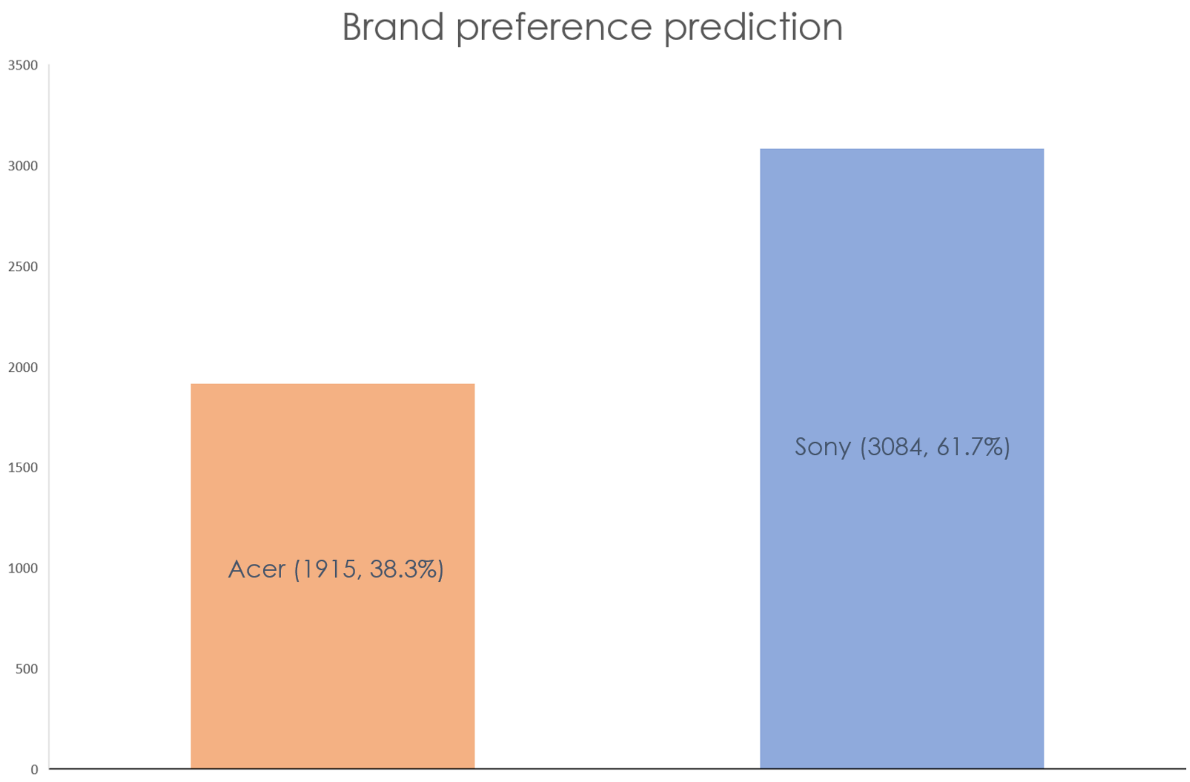

Figure: Brand preference prediction (incomplete survey)

If we look at the complete survey data (collected and predicted), it can be seen that Sony brand for laptops has bigger buying base among Blackwell electronics customers than the Acer brand. The percentages stay the same (60:40) even in this larger dataset.

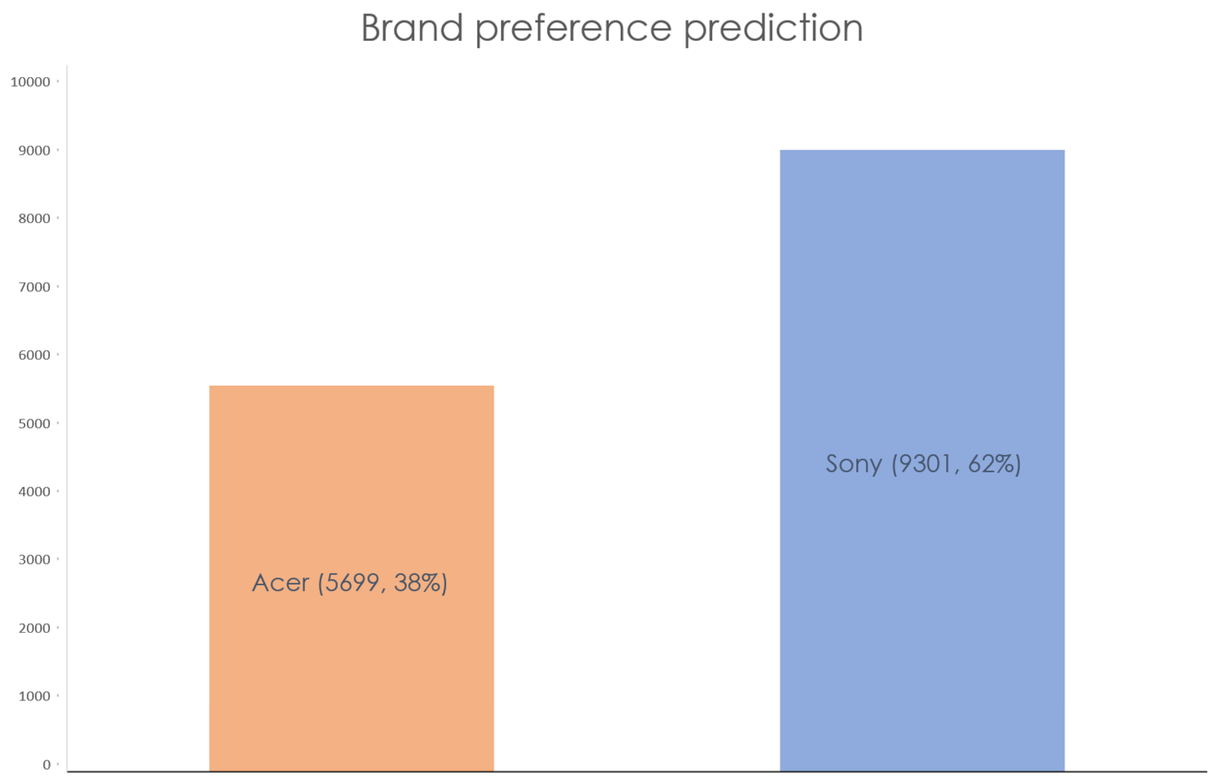

Figure: Brand preference prediction (all surveys)

Future actions

The main focus of this research was to establish our customer's laptop brand preference so we can build better relationship with that manufacturer.

Research showed that Blackwell customers prefer the Sony brand over the Acer brand when it comes to laptops. The percentage is 62% for Sony, and 38% for the Acer brand.

Future steps for the sales team would be to deepen the connection with the Sony manufacturer, without forgetting about Acer. Almost 40% of our customers prefer that brand and that cannot be ignored or forgot.

Also, the primary data exploration of the surveys showed obvious connection between customer's salary and preferred brand and little to no relevance to other attributes from the survey. Something to keep in mind for the next survey.

It should be noted that the only easily readable pattern comes from the salary ~ brand connection. We can conclude that customers who make more than 130.000 USD choose Sony as their brand. The same goes for customer whose salary is around 20.000 USD. They also choose Sony as their preferred brand. Sony computers attract two very different subgroups between Blackwell customers, and that should be something to look more into. Also, that would imply that a high end and a budget Sony computer will have their interested customers.

After these findings, it can be concluded that the next survey or any kind of notation of customers buying patterns or brand preferences, should focus on their salaries and incomes.

Appendix

Summary:

```
> summary(SurveyData)
     salary            age            elevel           car           zipcode          credit           brand       
 Min.   : 20000   Min.   :20.00   Min.   :0.000   Min.   : 1.00   Min.   :0.000   Min.   :     0   Min.   :0.0000  
 1st Qu.: 52109   1st Qu.:35.00   1st Qu.:1.000   1st Qu.: 6.00   1st Qu.:2.000   1st Qu.:121155   1st Qu.:0.0000  
 Median : 84969   Median :50.00   Median :2.000   Median :11.00   Median :4.000   Median :250607   Median :1.0000  
 Mean   : 84897   Mean   :49.81   Mean   :1.983   Mean   :10.53   Mean   :4.037   Mean   :249245   Mean   :0.6217  
 3rd Qu.:117168   3rd Qu.:65.00   3rd Qu.:3.000   3rd Qu.:16.00   3rd Qu.:6.000   3rd Qu.:374872   3rd Qu.:1.0000  
 Max.   :150000   Max.   :80.00   Max.   :4.000   Max.   :20.00   Max.   :8.000   Max.   :500000   Max.   :1.0000  
```

Structure:

```
> str(SurveyData)
'data.frame':	10000 obs. of  7 variables:
 $ salary : num  119807 106880 78021 63690 50874 ...
 $ age    : int  45 63 23 51 20 56 24 62 29 41 ...
 $ elevel : int  0 1 0 3 3 3 4 3 4 1 ...
 $ car    : int  14 11 15 6 14 14 8 3 17 5 ...
 $ zipcode: int  4 6 2 5 4 3 5 0 0 4 ...
 $ credit : num  442038 45007 48795 40889 352951 ...
 $ brand  : int  0 1 0 1 0 1 1 1 0 1 ...
```

KNN without tuning:

```
> confusionMatrix(predictionKNN, testSet$brand)
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0  579  321
         1  366 1233
                                          
               Accuracy : 0.7251          
                 95% CI : (0.7071, 0.7425)
    No Information Rate : 0.6218          
    P-Value [Acc > NIR] : < 2e-16         
                                          
                  Kappa : 0.41            
 Mcnemar's Test P-Value : 0.09321         
                                          
            Sensitivity : 0.6127          
            Specificity : 0.7934          
         Pos Pred Value : 0.6433          
         Neg Pred Value : 0.7711          
             Prevalence : 0.3782          
         Detection Rate : 0.2317          
   Detection Prevalence : 0.3601          
      Balanced Accuracy : 0.7031          
                                          
       'Positive' Class : 0               
```

Knn with tuning (metric = “accuracy”)

```
> confusionMatrix(predictionKNN, testSet$brand)
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0  548  370
         1  397 1184

               Accuracy : 0.6931
                 95% CI : (0.6746, 0.7111)
    No Information Rate : 0.6218
    P-Value [Acc > NIR] : 5.546e-14

                  Kappa : 0.3437
 Mcnemar's Test P-Value : 0.3478

            Sensitivity : 0.5799
            Specificity : 0.7619
         Pos Pred Value : 0.5969
         Neg Pred Value : 0.7489
             Prevalence : 0.3782
         Detection Rate : 0.2193
   Detection Prevalence : 0.3673
      Balanced Accuracy : 0.6709

       'Positive' Class : 0
```

Knn with tuning (metric = “kappa”)

```
> confusionMatrix(predictionKNN, testSet$brand)
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0  577  318
         1  368 1236

               Accuracy : 0.7255
                 95% CI : (0.7075, 0.7429)
    No Information Rate : 0.6218
    P-Value [Acc > NIR] : < 2e-16

                  Kappa : 0.4102
 Mcnemar's Test P-Value : 0.06137

            Sensitivity : 0.6106
            Specificity : 0.7954
         Pos Pred Value : 0.6447
         Neg Pred Value : 0.7706
             Prevalence : 0.3782
         Detection Rate : 0.2309
   Detection Prevalence : 0.3581
      Balanced Accuracy : 0.7030

       'Positive' Class : 0
```

Decision tree (split = gini)

```
> confusionMatrix(predictionDT, testSet$brand)
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0  840   90
         1  105 1464
                                          
               Accuracy : 0.922           
                 95% CI : (0.9108, 0.9322)
    No Information Rate : 0.6218          
    P-Value [Acc > NIR] : <2e-16          
                                          
                  Kappa : 0.8336          
 Mcnemar's Test P-Value : 0.3161          
                                          
            Sensitivity : 0.8889          
            Specificity : 0.9421          
         Pos Pred Value : 0.9032          
         Neg Pred Value : 0.9331          
             Prevalence : 0.3782          
         Detection Rate : 0.3361          
   Detection Prevalence : 0.3721          
      Balanced Accuracy : 0.9155          
                                          
       'Positive' Class : 0               
```

Decision tree (split = information)

```
> confusionMatrix(predictionDT, testSet$brand)
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0  852  115
         1   93 1439
                                          
               Accuracy : 0.9168          
                 95% CI : (0.9052, 0.9273)
    No Information Rate : 0.6218          
    P-Value [Acc > NIR] : <2e-16          
                                          
                  Kappa : 0.8238          
 Mcnemar's Test P-Value : 0.1454          
                                          
            Sensitivity : 0.9016          
            Specificity : 0.9260          
         Pos Pred Value : 0.8811          
         Neg Pred Value : 0.9393          
             Prevalence : 0.3782          
         Detection Rate : 0.3409          
   Detection Prevalence : 0.3870          
      Balanced Accuracy : 0.9138          
                                          
       'Positive' Class : 0               
```