horizontal line

**Predicting Customer Brand Preferences with R**

# OVERVIEW

We were tasked with predicting customer computer brand preferences using R and RStudio to analyze customer survey data. Customer survey data includes information such as: customer salary, age, zipcode, credit, education level, and type of car they drive. The choice in brand the customers had was either Acer or Sony. Given completed survey data for ~10,000 customers, we were asked to see whether we could predict brand preference utilizing the answers to the other survey questions. We were then asked to apply the trained model to incomplete survey data to make predictions.

# MODELS

Three different decision tree classification models were evaluated. They are as follows:

1. Stochastic Gradient Boosting
2. C50
3. Random Forest

# TRAINING RESULTS

**Stochastic Gradient Boosting (GBM)**

Results:

Stochastic Gradient Boosting

7424 samples

6 predictor

2 classes: '0', '1'

No pre-processing

Resampling: Cross-Validated (10 fold, repeated 1 times)

Summary of sample sizes: 6681, 6681, 6681, 6682, 6681, 6683, ...

Resampling results across tuning parameters:

|  |  |  |  |
| --- | --- | --- | --- |
| interaction.depth | n.trees | Accuracy | Kappa |
| 1 | 50 | 0.7304699 | 0.4317212 |
| 1 | 100 | 0.7291238 | 0.4276247 |
| 1 | 150 | 0.7272378 | 0.4221843 |
| 2 | 50 | 0.8129082 | 0.6071113 |
| 2 | 100 | 0.8844311 | 0.7583407 |
| 2 | 150 | 0.9036879 | 0.7969886 |
| 3 | 50 | 0.8783671 | 0.7479574 |
| 3 | 100 | 0.9012646 | 0.7928966 |
| 3 | 150 | 0.91783 | 0.8263581 |

Tuning parameter 'shrinkage' was held constant at a value of 0.1

Tuning parameter 'n.minobsinnode' was held constant at a value of 10

Accuracy was used to select the optimal model using the largest value.

The final values used for the model were n.trees = 150, interaction.depth = 3,

shrinkage = 0.1 and n.minobsinnode = 10.

**C5.0**

Results:

C5.0

7424 samples

6 predictor

2 classes: '0', '1'

No pre-processing

Resampling: Cross-Validated (10 fold, repeated 1 times)

Summary of sample sizes: 6681, 6681, 6681, 6682, 6681, 6683, ...

Resampling results across tuning parameters:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| model | winnow | trials | Accuracy | Kappa |
| rules | FALSE | 1 | 0.8309535 | 0.6590006 |
| rules | FALSE | 10 | 0.9150011 | 0.8186639 |
| rules | FALSE | 20 | 0.916483 | 0.8221278 |
| rules | TRUE | 1 | 0.8382455 | 0.6725782 |
| rules | TRUE | 10 | 0.9159448 | 0.8200646 |
| rules | TRUE | 20 | 0.917294 | 0.8238717 |
| tree | FALSE | 1 | 0.8306845 | 0.6581194 |
| tree | FALSE | 10 | 0.9146017 | 0.8183922 |
| tree | FALSE | 20 | 0.9168939 | 0.8236669 |
| tree | TRUE | 1 | 0.8387839 | 0.6737648 |
| tree | TRUE | 10 | 0.9154074 | 0.8199129 |
| tree | TRUE | 20 | 0.9154074 | 0.8201416 |

Accuracy was used to select the optimal model using the largest value.

The final values used for the model were trials = 20, model = rules and winnow = TRUE.

**Random Forest (RF)**

Results:

Random Forest

7424 samples

6 predictor

2 classes: '0', '1'

No pre-processing

Resampling: Cross-Validated (10 fold, repeated 1 times)

Summary of sample sizes: 6681, 6681, 6681, 6682, 6681, 6683, ...

Resampling results across tuning parameters:

|  |  |  |
| --- | --- | --- |
| mtry | Accuracy | Kappa |
| 1 | 0.6217673 | 0 |
| 2 | 0.6217673 | 0 |
| 3 | 0.72858 | 0.3452674 |
| 4 | 0.8565433 | 0.6876232 |
| 5 | 0.8946619 | 0.7762752 |

Accuracy was used to select the optimal model using the largest value.

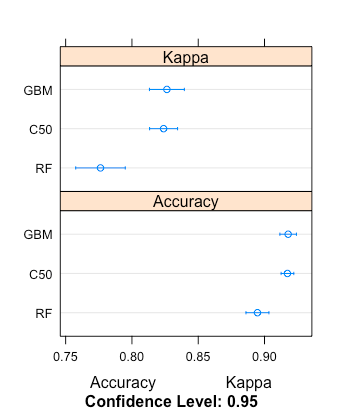
The final value used for the model was mtry = 5.

# PREDICTION RESULTS

## Model Selection

The accuracy and kappa scores for each model were compared. Stochastic gradient boosting achieved the best scores in these areas -- although only slightly higher than C5.0. Random forest had the worst performance of the three.

Additionally, each model was resampled and results are depicted in the chart below.



Again, stochastic gradient boosting performed slightly better than C5.0 (accuracy for GBM was 0.93 and for C5.0 was 0.924). Because of this, the stochastic gradient boosting model was chosen to make predictions.

## Prediction Results on Complete Survey Data

The stochastic gradient boosting model was first tested on the data that was set aside for testing from the complete survey results. Confusion matrix results are as follows:

**Confusion Matrix and Statistics**

Reference

Prediction 0 1

0 874 116

1 62 1422

Accuracy : 0.9281

95% CI : (0.9172, 0.9379)

No Information Rate : 0.6217

P-Value [Acc > NIR] : < 2.2e-16

Kappa : 0.8488

Mcnemar's Test P-Value : 7.112e-05

Sensitivity : 0.9338

Specificity : 0.9246

Pos Pred Value : 0.8828

Neg Pred Value : 0.9582

Prevalence : 0.3783

Detection Rate : 0.3533

Detection Prevalence : 0.4002

Balanced Accuracy : 0.9292

'Positive' Class : 0

Additionally, post resampling results are as follows:

**Post-Resampling**

Accuracy Kappa

0.9280517 0.8487549

The C5.0 model was also used to test predictions, and results for confusion matrix and post-resampling again scored slightly lower than stochastic gradient boosting.

## Prediction Results on Incomplete Survey Data

Finally, we were able to use our stochastic gradient boosting model to make predictions on the incomplete survey data. Results are as follows:

|  |  |
| --- | --- |
| Acer | Sony |
| 1946 | 3054 |

According to the variable importance of the model, salary and age were most influential in making these predictions--something that should be investigated further.

Additionally, the prediction results were added to the incomplete survey data set and imported to a .csv file, which is included in the zip folder I have attached. The file name is “newsurveypreds.csv”.