**CUSTOMER BRAND PREFERENCES REPORT**

# **RANDOM FOREST (AUTOMATIC GRID):**

## **CODING:**

>Library(caret)

>Set.seed(998)

>Intraining<- createDataPartition(CompleteResponses$brand, p=.75, list=FALSE)

>Training <- completeResponses[Intraining,]

>Testing <- completeResponses[-Intraining,]

>Fitcontrol<- traincontrol(method=”repeatdcv” , number = 10, repeats = 1)

>rfGrid<- expand.grid(mtry=c(1,2,3,4,5))

>system.time(rfFitm1 <- train(brand~., data = Training, method = "rf", trControl=fitControl, tuneGrid=rfGrid))

>rfFitm1

## **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: 6682, 6681, 6681, 6681, 6681, 6682, ...

Resampling results across tuning parameters:

mtry Accuracy Kappa

1 0.8523716 0.6741884

2 0.9240304 0.8390989

**3 0.9249740 0.8410133**

4 0.9234915 0.8377867

5 0.9222789 0.8351256

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

The final value used for the model was mtry = 3.

## **RESULTS WITH REDUCTION OF NUMBER OF PREDICTORS:**

>varImp(rfFitm1)

rf variable importance

Overall

Salary 100.000

age 65.100

credit 9.088

car 3.347

zipcode 1.454

elevel 0.000

>myvars<- c("salary", "age", "credit", "brand")

>newData<- newData[myvars]

>training2 <- newData[inTraining,]

>rfFit2 <- train(brand~., data = training2, method = "rf", trControl=fitControl, tuneGrid=rfGrid)

>rfFit2

Random Forest

7424 samples

3 predictor

2 classes: '0', '1'

No pre-processing

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

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

Resampling results across tuning parameters:

mtry Accuracy Kappa

**1 0.9229598 0.8368996**

2 0.9189165 0.8281459

3 0.9155474 0.8208990

4 0.9150076 0.8196457

5 0.9156827 0.8211168

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

The final value used for the model was mtry = 1.

🡺 The reduction of variables with less variable importance showed no difference with the results from the modeling including all variables.

## **PREDICTIONS:**

>rfResults<- predict(object = rfFitm1, newdata = testing)

>table(rfResults, testing$brand)

rfResults 0 1

0 2808 0

1 0 4616

# **C5.0**

## **CODING:**

>Cmodel<- train(brand ~ ., data = training, method = "C5.0", trcontrol=fitControl, tuneLength = 3)

>Cmodel

## **RESULTS:**

C5.0

7424 samples

6 predictor

2 classes: '0', '1'

No pre-processing

Resampling: Bootstrapped (25 reps)

Summary of sample sizes: 7424, 7424, 7424, 7424, 7424, 7424, ...

Resampling results across tuning parameters:

model winnow trials Accuracy Kappa

rules FALSE 1 0.8982511 0.7846188

rules FALSE 10 0.9148693 0.8183259

rules FALSE 20 0.9168865 0.8225826

rules TRUE 1 0.8982511 0.7846188

rules TRUE 10 0.9148693 0.8183259

rules TRUE 20 0.9168865 0.8225826

tree FALSE 1 0.8893621 0.7628711

tree FALSE 10 0.9147858 0.8187893

tree FALSE 20 0.9170540 0.8236255

tree TRUE 1 0.8893621 0.7628711

tree TRUE 10 0.9147858 0.8187893

**tree TRUE 20 0.9170540 0.8236255**

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

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

>varImp(Cmodel)

C5.0 variable importance

Overall

salary 100.00

age 85.17

elevel 10.57

zipcode 0.00

credit 0.00

car 0.00

## **RESULTS WITH REDUCTION OF NUMBER OF PREDICTORS:**

>myvarsC<- c("salary", "age", "elevel", "brand")

>newdataC<- CompleteResponses

>newdataC<- newdataC[myvarsC]

>trainingC<- newdataC[inTraining,]

>Cmodel2 <- train(brand ~ ., data = trainingC, method = "C5.0", trcontrol=fitControl, tuneLength = 3)

>Cmodel2

C5.0

7424 samples

3 predictor

2 classes: '0', '1'

No pre-processing

Resampling: Bootstrapped (25 reps)

Summary of sample sizes: 7424, 7424, 7424, 7424, 7424, 7424, ...

Resampling results across tuning parameters:

model winnow trials Accuracy Kappa

rules FALSE 1 0.9024847 0.7951979

rules FALSE 10 0.9154132 0.8202392

rules FALSE 20 0.9157442 0.8209384

rules TRUE 1 0.9024847 0.7951979

rules TRUE 10 0.9154132 0.8202392

rules TRUE 20 0.9157442 0.8209384

tree FALSE 1 0.8988456 0.7878730

tree FALSE 10 0.9151937 0.8199420

tree FALSE 20 0.9163392 0.8225347

tree TRUE 1 0.8988456 0.7878730

tree TRUE 10 0.9151937 0.8199420

**tree TRUE 20 0.9163392 0.8225347**

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

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

🡺 The reduction of variables with less variable importance showed no difference with the results from the modeling including all variables.

## **PREDICTIONS**

>Cresults<- predict(object=Cmodel, newdata=testing)

>table(Cresults, testing$brand)

Cresults 0 1

0 2622 305

1 186 4311

I have selected the **Random Forest** as it shows:

* Highest accuracy =0.9249740
* Response time=3-6 minutes
* Prediction assessment:

>postResample(rfResults, testing$brand)

Accuracy Kappa

1 1

Comparing to the C5.0 Model:

* Highest accuracy = 0.9170540
* Response time = 10-15 minutes
* Predictions Assessment:

>postResample(Cresults, testing$brand)

Accuracy Kappa

0.9338631 0.8605441

* 1. The predicted **answers** to the brand preference question for the instances of survey results that are missing that answer.🡺 SurveyIncomplete.csv

>SurveysResult<- predict(object = rfFitm1, newdata = SurveyIncomplete)

>SurveysResult

>table(SurveysResult, SurveyIncomplete$brand)

SurveysResult 0 1

0 1903 5

1 3034 58

**Actual Answers:**

>summary(rfResults)

0 1

2808 4616

**Predicted Answers:**

>summary(SurveysResult)

0 1

1908 3092

A **chart** that displays the customer preference for each brand based on the combination of the actual answers and the predicted answers to the brand preference survey question:

