**Results for all model run to predict brand preference**

1. **C5.0 results**

**With all the independent variables**

fitControl<-trainControl(method="repeatedcv",number = 10,repeats=1)  
c50Fit1<-train(brand~.,data=training,method="C5.0",trControl=fitControl,tuneLength=2)  
c50Fit1

## 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, 6682, 6683, ...   
## Resampling results across tuning parameters:  
##   
## model winnow trials Accuracy Kappa   
## rules FALSE 1 0.8593688 0.7152510  
## rules FALSE 10 0.9247055 0.8396015  
## rules TRUE 1 0.8597733 0.7161791  
## rules TRUE 10 0.9267260 0.8436584  
## tree FALSE 1 0.8593691 0.7152730  
## tree FALSE 10 0.9257826 0.8422156  
## tree TRUE 1 0.8593693 0.7153700  
## tree TRUE 10 0.9225504 0.8356068  
##   
## Accuracy was used to select the optimal model using the largest value.  
## The final values used for the model were trials = 10, model = rules  
## and winnow = TRUE.

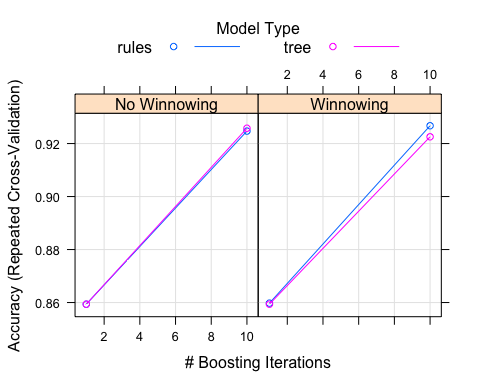
c50Fit1$bestTune

## trials model winnow  
## 8 10 rules TRUE

varImp(c50Fit1)

## C5.0 variable importance  
##   
## only 20 most important variables shown (out of 34)  
##   
## Overall  
## salary 100.00  
## age 95.85  
## car15 39.25  
## car16 33.24  
## zipcode5 3.93  
## car11 0.00  
## car20 0.00  
## car18 0.00  
## car13 0.00  
## car8 0.00  
## zipcode8 0.00  
## zipcode3 0.00  
## car6 0.00  
## elevel2 0.00  
## zipcode4 0.00  
## car7 0.00  
## credit 0.00  
## car2 0.00  
## car10 0.00  
## car17 0.00

plot(c50Fit1)



prediction1<-predict(c50Fit1,testing)

postResample(pred = prediction1,obs=testing$brand)

## Accuracy Kappa   
## 0.9086500 0.8053946

prediction\_incomplete1<-predict(c50Fit1,SurveyIncomplete)

**With salary**

library(C50)  
fitControl<-trainControl(method="repeatedcv",number = 10,repeats=1)  
c50Fit2<-train(brand~salary,data=training,method="C5.0",trControl=fitControl,tuneLength=2)  
c50Fit2

## C5.0   
##   
## 7424 samples  
## 1 predictor  
## 2 classes: '0', '1'   
##   
## No pre-processing  
## Resampling: Cross-Validated (10 fold, repeated 1 times)   
## Summary of sample sizes: 6682, 6681, 6681, 6681, 6682, 6682, ...   
## Resampling results across tuning parameters:  
##   
## model winnow trials Accuracy Kappa   
## rules FALSE 1 0.7370627 0.4480452  
## rules FALSE 10 0.7280405 0.4082226  
## rules TRUE 1 0.7370627 0.4480452  
## rules TRUE 10 0.7280405 0.4082226  
## tree FALSE 1 0.7370627 0.4480452  
## tree FALSE 10 0.7370627 0.4480452  
## tree TRUE 1 0.7370627 0.4480452  
## tree TRUE 10 0.7370627 0.4480452  
##   
## Accuracy was used to select the optimal model using the largest value.  
## The final values used for the model were trials = 1, model = rules  
## and winnow = TRUE.

c50Fit2$bestTune

## trials model winnow  
## 7 1 rules TRUE

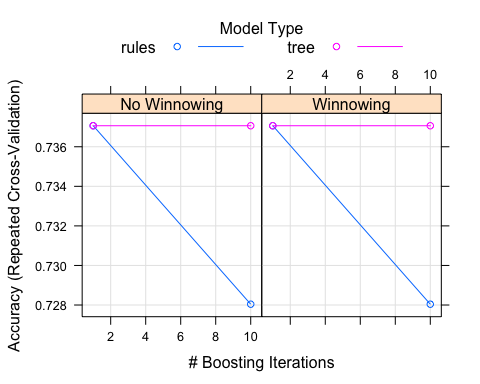
varImp(c50Fit2)

## C5.0 variable importance

## Warning in FUN(newX[, i], ...): no non-missing arguments to max; returning  
## -Inf

## Overall  
## salary NaN

plot(c50Fit2)



prediction2<-predict(c50Fit2,testing)

postResample(pred = prediction2,obs=testing$brand)

## Accuracy Kappa   
## 0.7134196 0.4089337

prediction\_incomplete2<-predict(c50Fit2,SurveyIncomplete)

**With salary and age**

library(C50)  
fitControl<-trainControl(method="repeatedcv",number = 10,repeats=1)  
c50Fit3<-train(brand~salary+age,data=training,method="C5.0",trControl=fitControl,tuneLength=2)  
c50Fit3

## C5.0   
##   
## 7424 samples  
## 2 predictor  
## 2 classes: '0', '1'   
##   
## No pre-processing  
## Resampling: Cross-Validated (10 fold, repeated 1 times)   
## Summary of sample sizes: 6682, 6681, 6682, 6682, 6682, 6681, ...   
## Resampling results across tuning parameters:  
##   
## model winnow trials Accuracy Kappa   
## rules FALSE 1 0.9035449 0.8003115  
## rules FALSE 10 0.9283405 0.8467770  
## rules TRUE 1 0.9035449 0.8003115  
## rules TRUE 10 0.9283405 0.8467770  
## tree FALSE 1 0.9035449 0.8003593  
## tree FALSE 10 0.9247039 0.8399285  
## tree TRUE 1 0.9035449 0.8003593  
## tree TRUE 10 0.9247039 0.8399285  
##   
## Accuracy was used to select the optimal model using the largest value.  
## The final values used for the model were trials = 10, model = rules  
## and winnow = TRUE.

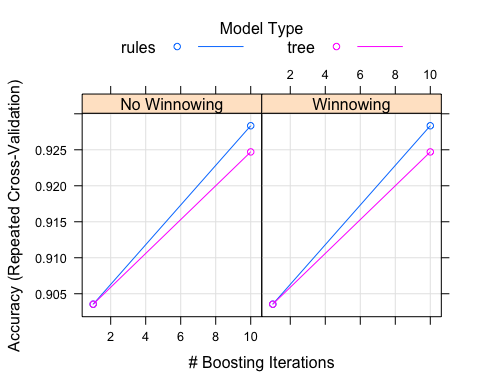
c50Fit3$bestTune

## trials model winnow  
## 8 10 rules TRUE

varImp(c50Fit3)

## C5.0 variable importance  
##   
## Overall  
## salary 100  
## age 0

plot(c50Fit3)



prediction3<-predict(c50Fit3,testing)

postResample(pred = prediction3,obs=testing$brand)

## Accuracy Kappa   
## 0.9110752 0.8086365

prediction\_incomplete3<-predict(c50Fit3,SurveyIncomplete)

**With salary, age and credit**

library(C50)  
fitControl<-trainControl(method="repeatedcv",number = 10,repeats=1)  
c50Fit4<-train(brand~salary+age+credit,data=training,method="C5.0",trControl=fitControl,tuneLength=2)  
c50Fit4

## C5.0   
##   
## 7424 samples  
## 3 predictor  
## 2 classes: '0', '1'   
##   
## No pre-processing  
## Resampling: Cross-Validated (10 fold, repeated 1 times)   
## Summary of sample sizes: 6682, 6683, 6681, 6681, 6681, 6681, ...   
## Resampling results across tuning parameters:  
##   
## model winnow trials Accuracy Kappa   
## rules FALSE 1 0.8623308 0.7224877  
## rules FALSE 10 0.9268616 0.8438374  
## rules TRUE 1 0.8616578 0.7211957  
## rules TRUE 10 0.9263229 0.8426043  
## tree FALSE 1 0.8624655 0.7228334  
## tree FALSE 10 0.9251114 0.8410916  
## tree TRUE 1 0.8616578 0.7212629  
## tree TRUE 10 0.9252469 0.8413908  
##   
## Accuracy was used to select the optimal model using the largest value.  
## The final values used for the model were trials = 10, model = rules  
## and winnow = FALSE.

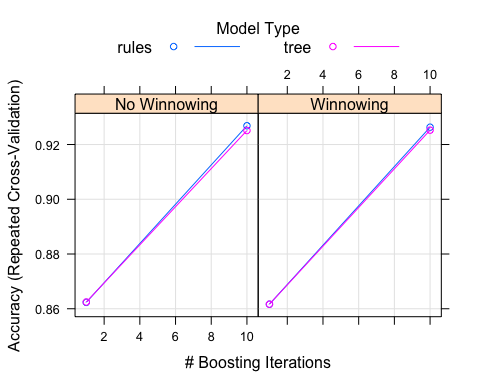
c50Fit4$bestTune

## trials model winnow  
## 6 10 rules FALSE

varImp(c50Fit4)

## C5.0 variable importance  
##   
## Overall  
## salary 100  
## age 100  
## credit 0

plot(c50Fit4)



prediction4<-predict(c50Fit4,testing)

postResample(pred = prediction4,obs=testing$brand)

## Accuracy Kappa   
## 0.9074373 0.8010185

prediction\_incomplete4<-predict(c50Fit4,SurveyIncomplete)

**With salary, age, credit and elevel**

library(C50)  
fitControl<-trainControl(method="repeatedcv",number = 10,repeats=1)  
c50Fit5<-train(brand~salary+age+credit+elevel,data=training,method="C5.0",trControl=fitControl,tuneLength=2)  
c50Fit5

## C5.0   
##   
## 7424 samples  
## 4 predictor  
## 2 classes: '0', '1'   
##   
## No pre-processing  
## Resampling: Cross-Validated (10 fold, repeated 1 times)   
## Summary of sample sizes: 6682, 6682, 6682, 6682, 6682, 6681, ...   
## Resampling results across tuning parameters:  
##   
## model winnow trials Accuracy Kappa   
## rules FALSE 1 0.8736388 0.7441517  
## rules FALSE 10 0.9271294 0.8441143  
## rules TRUE 1 0.8857683 0.7669123  
## rules TRUE 10 0.9278023 0.8455331  
## tree FALSE 1 0.8735044 0.7439184  
## tree FALSE 10 0.9261871 0.8432052  
## tree TRUE 1 0.8856339 0.7666759  
## tree TRUE 10 0.9243005 0.8393910  
##   
## Accuracy was used to select the optimal model using the largest value.  
## The final values used for the model were trials = 10, model = rules  
## and winnow = TRUE.

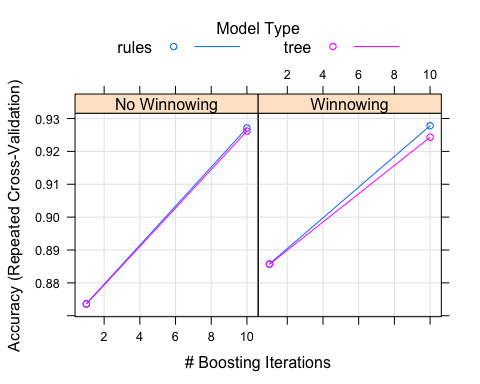
c50Fit5$bestTune

## trials model winnow  
## 8 10 rules TRUE

varImp(c50Fit5)

## C5.0 variable importance  
##   
## Overall  
## salary 100.00  
## age 96.11  
## credit 0.00  
## elevel3 0.00  
## elevel4 0.00  
## elevel2 0.00  
## elevel1 0.00

plot(c50Fit5)



prediction5<-predict(c50Fit5,testing)

postResample(pred = prediction5,obs=testing$brand)

## Accuracy Kappa   
## 0.9110752 0.8086365

prediction\_incomplete5<-predict(c50Fit5,SurveyIncomplete)

**With salary, age, credit, elevel and zipcode**

library(C50)  
fitControl<-trainControl(method="repeatedcv",number = 10,repeats=1)  
c50Fit6<-train(brand~salary+age+credit+elevel+zipcode,data=training,method="C5.0",trControl=fitControl,tuneLength=2)  
c50Fit6

## C5.0   
##   
## 7424 samples  
## 5 predictor  
## 2 classes: '0', '1'   
##   
## No pre-processing  
## Resampling: Cross-Validated (10 fold, repeated 1 times)   
## Summary of sample sizes: 6682, 6682, 6682, 6682, 6681, 6681, ...   
## Resampling results across tuning parameters:  
##   
## model winnow trials Accuracy Kappa   
## rules FALSE 1 0.8794492 0.7538852  
## rules FALSE 10 0.9273946 0.8447215  
## rules TRUE 1 0.8795841 0.7541714  
## rules TRUE 10 0.9247019 0.8387321  
## tree FALSE 1 0.8794492 0.7538298  
## tree FALSE 10 0.9268595 0.8448304  
## tree TRUE 1 0.8795839 0.7540702  
## tree TRUE 10 0.9280704 0.8470787  
##   
## Accuracy was used to select the optimal model using the largest value.  
## The final values used for the model were trials = 10, model = tree  
## and winnow = TRUE.

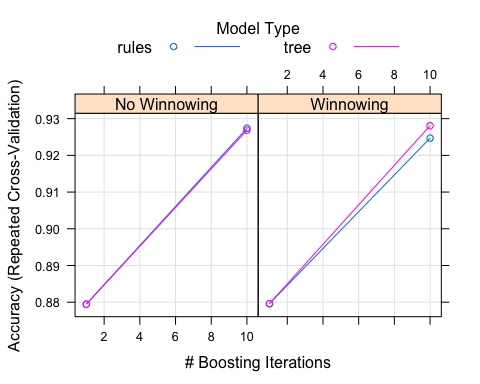
c50Fit6$bestTune

## trials model winnow  
## 4 10 tree TRUE

varImp(c50Fit6)

## C5.0 variable importance  
##   
## Overall  
## salary 100.00  
## age 84.15  
## zipcode7 80.66  
## elevel4 7.92  
## zipcode4 7.52  
## zipcode1 1.17  
## elevel2 0.00  
## zipcode3 0.00  
## zipcode2 0.00  
## elevel3 0.00  
## zipcode6 0.00  
## zipcode8 0.00  
## credit 0.00  
## elevel1 0.00  
## zipcode5 0.00

plot(c50Fit6)



prediction6<-predict(c50Fit6,testing)

postResample(pred = prediction6,obs=testing$brand)

## Accuracy Kappa   
## 0.9122878 0.8125552

prediction\_incomplete6<-predict(c50Fit6,SurveyIncomplete)

**With salary, age, credit and zipcode**

library(C50)  
fitControl<-trainControl(method="repeatedcv",number = 10,repeats=1)  
c50Fit7<-train(brand~salary+age+credit+zipcode,data=training,method="C5.0",trControl=fitControl,tuneLength=2)  
c50Fit7

## C5.0   
##   
## 7424 samples  
## 4 predictor  
## 2 classes: '0', '1'   
##   
## No pre-processing  
## Resampling: Cross-Validated (10 fold, repeated 1 times)   
## Summary of sample sizes: 6681, 6682, 6682, 6682, 6681, 6682, ...   
## Resampling results across tuning parameters:  
##   
## model winnow trials Accuracy Kappa   
## rules FALSE 1 0.8723088 0.7392100  
## rules FALSE 10 0.9255134 0.8413487  
## rules TRUE 1 0.8716357 0.7385641  
## rules TRUE 10 0.9238967 0.8381718  
## tree FALSE 1 0.8719045 0.7384760  
## tree FALSE 10 0.9275326 0.8459176  
## tree TRUE 1 0.8715009 0.7383331  
## tree TRUE 10 0.9263210 0.8438640  
##   
## Accuracy was used to select the optimal model using the largest value.  
## The final values used for the model were trials = 10, model = tree  
## and winnow = FALSE.

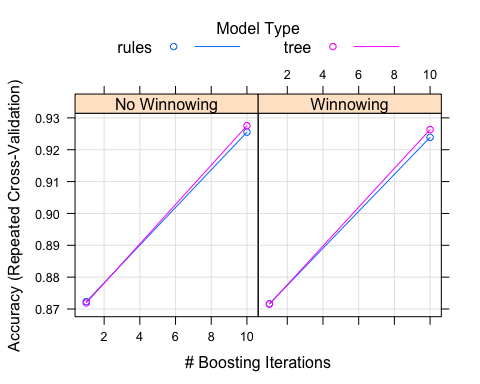
c50Fit7$bestTune

## trials model winnow  
## 2 10 tree FALSE

varImp(c50Fit7)

## C5.0 variable importance  
##   
## Overall  
## salary 100.00  
## age 84.15  
## zipcode8 28.72  
## zipcode6 8.50  
## credit 5.72  
## zipcode4 0.00  
## zipcode3 0.00  
## zipcode2 0.00  
## zipcode7 0.00  
## zipcode5 0.00  
## zipcode1 0.00

plot(c50Fit7)



prediction7<-predict(c50Fit7,testing)

postResample(pred = prediction7,obs=testing$brand)

## Accuracy Kappa   
## 0.9078416 0.8031775

prediction\_incomplete7<-predict(c50Fit7,SurveyIncomplete)

1. **Random forest results**

**With all independent variables**

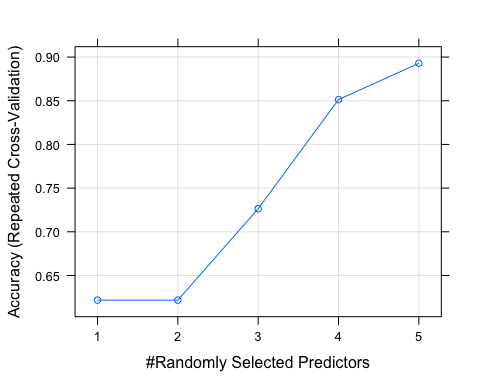
rfGrid<-expand.grid(mtry=c(1,2,3,4,5))  
rfFit1<-train(brand~.,data=training,method="rf",trControl=fitControl,tuneGrid=rfGrid)  
rfFit1

## 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, 6683, 6682, 6682, 6681, ...   
## Resampling results across tuning parameters:  
##   
## mtry Accuracy Kappa   
## 1 0.6217673 0.0000000  
## 2 0.6217673 0.0000000  
## 3 0.7264281 0.3385882  
## 4 0.8512934 0.6756583  
## 5 0.8929168 0.7722468  
##   
## Accuracy was used to select the optimal model using the largest value.  
## The final value used for the model was mtry = 5.

varImp(rfFit1)

## rf variable importance  
##   
## only 20 most important variables shown (out of 34)  
##   
## Overall  
## salary 100.0000  
## age 38.3719  
## credit 19.6062  
## elevel4 0.9084  
## elevel3 0.8763  
## elevel1 0.8559  
## elevel2 0.8180  
## zipcode3 0.4627  
## zipcode6 0.4474  
## zipcode1 0.4371  
## car15 0.3981  
## zipcode7 0.3885  
## zipcode4 0.3694  
## zipcode5 0.3195  
## zipcode2 0.2912  
## car7 0.2075  
## car12 0.2030  
## zipcode8 0.1917  
## car18 0.1870  
## car10 0.1860

plot(rfFit1)



prediction2<-predict(rfFit1,testing)

postResample(pred=prediction2,obs=testing$brand)

## Accuracy Kappa   
## 0.8823767 0.7494765

prediction\_incomplete2<-predict(rfFit1,SurveyIncomplete)

**3: KNN results**

**With all independent variables**

KNNfit1<-train(brand~.,data=training,method="knn",trControl=fitControl,preProcess = c("center","scale"), tuneLength=20)  
KNNfit1

## k-Nearest Neighbors   
##   
## 7424 samples  
## 6 predictor  
## 2 classes: '0', '1'   
##   
## Pre-processing: centered (34), scaled (34)   
## Resampling: Cross-Validated (10 fold, repeated 1 times)   
## Summary of sample sizes: 6681, 6681, 6682, 6681, 6681, 6681, ...   
## Resampling results across tuning parameters:  
##   
## k Accuracy Kappa   
## 5 0.5533296 0.02929107  
## 7 0.5727251 0.05712977  
## 9 0.5771715 0.05488148  
## 11 0.5874032 0.06088282  
## 13 0.5954900 0.06930221  
## 15 0.6042472 0.08331167  
## 17 0.6020934 0.07239369  
## 19 0.6064039 0.07336135  
## 21 0.6033058 0.06239969  
## 23 0.5984541 0.04608297  
## 25 0.5999353 0.04334803  
## 27 0.6003381 0.04132338  
## 29 0.5960273 0.02535789  
## 31 0.6018253 0.03352018  
## 33 0.6038471 0.03346834  
## 35 0.6066769 0.03596760  
## 37 0.6080213 0.03508316  
## 39 0.6126052 0.04111258  
## 41 0.6138176 0.03888102  
## 43 0.6151662 0.03897147  
##   
## Accuracy was used to select the optimal model using the largest value.  
## The final value used for the model was k = 43.

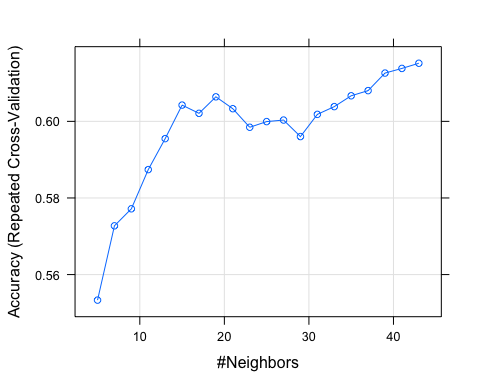
KNNfit1$bestTune

## k  
## 20 43

varImp(KNNfit1)

## ROC curve variable importance  
##   
## Importance  
## salary 100.00000  
## age 4.30856  
## elevel 3.11199  
## credit 2.06643  
## car 0.04176  
## zipcode 0.00000

plot(KNNfit1)



predictions3 <- predict(KNNfit1,testing)

postResample(pred=predictions3,obs=testing$brand)

## Accuracy Kappa   
## 0.6010509297 -0.0006392714

prediction\_incomplete3<-predict(KNNfit1,SurveyIncomplete)