task3_rfe

Code

#rfe

Hide

```
# Let's sample the data before using RFE
iphone_smallMatrix_Sample <- iphone_smallMatrix[sample(1:nrow(iphone_smallMatrix), 1000,
replace=FALSE),]

# Set up rfeControl with randomforest, repeated cross validation and no updates
ctrl <- rfeControl(functions = rfFuncs,
                   method = "repeatedcv",
                   repeats = 5,
                   verbose = FALSE)

# Use rfe and omit the response variable (attribute 11 iphonesentiment)
rfeResults <- rfe(iphone_smallMatrix_Sample[,1:10],
                  iphone_smallMatrix_Sample$iphonesentiment,
                  sizes=(1:10),
                  rfeControl=ctrl)

# Get results
rfeResults
```

```
Recursive feature selection

Outer resampling method: Cross-Validated (10 fold, repeated 5 times)

Resampling performance over subset size:
```

Variables <S3: AsIs>	RMSE <S3: AsIs>	Rsquared <S3: AsIs>	MAE <S3: AsIs>	RMSESD <S3: AsIs>	RsquaredSD <S3: AsIs>	MAESD <S3: AsIs>	Selected <S3: AsIs>
1	1.532	0.2995	1.154	0.1141	0.09049	0.08241	
2	1.472	0.3553	1.140	0.1154	0.09329	0.07607	
3	1.479	0.3560	1.147	0.1159	0.09835	0.08162	
4	1.479	0.3639	1.147	0.1173	0.10175	0.08402	
5	1.488	0.3644	1.165	0.1156	0.10276	0.08327	
6	1.449	0.3741	1.086	0.1278	0.10297	0.08547	
7	1.449	0.3751	1.087	0.1258	0.10141	0.08426	
8	1.448	0.3758	1.087	0.1251	0.10122	0.08464	
9	1.444	0.3762	1.062	0.1277	0.10091	0.08644	*

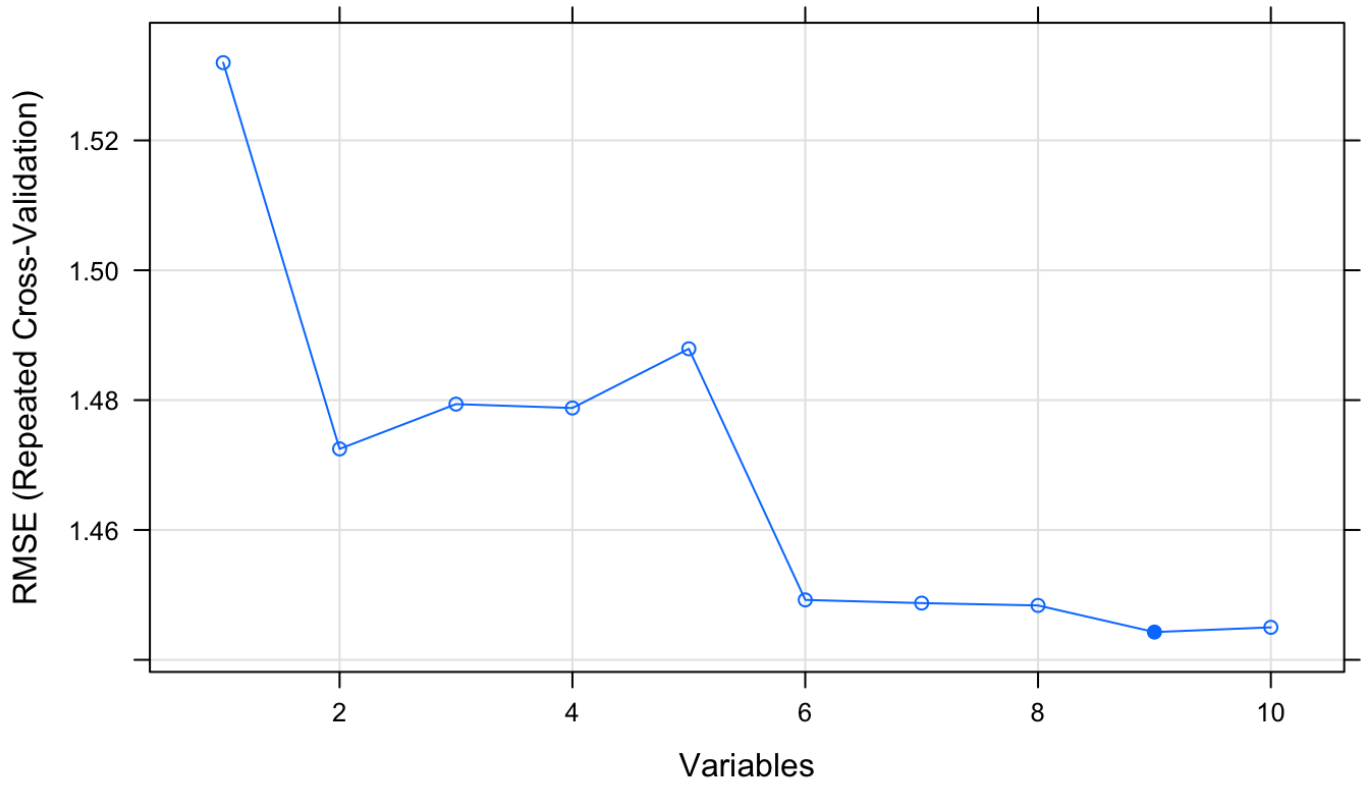
	Variables <S3: AsIs>	RMSE <S3: AsIs>	Rsquared <S3: AsIs>	MAE <S3: AsIs>	RMSESD <S3: AsIs>	RsquaredSD <S3: AsIs>	MAESD <S3: AsIs>	Selected <S3: AsIs>
10	10	1.445	0.3763	1.072	0.1268	0.10093	0.08518	

1-10 of 10 rows

```
The top 5 variables (out of 9):
   iphone, htcphone, samsunggalaxy, googleandroid, sonyxperia
```

Hide

```
# Plot results
plot(rfeResults, type=c("g", "o"))
```



Hide

```
# create new data set with rfe recommended features
iphone_smallMatrix_RFE <- iphone_smallMatrix[,predictors(rfeResults)]

# add the dependent variable to iphoneRFE
iphone_smallMatrix_RFE$iphonesentiment <- iphone_smallMatrix$iphonesentiment

# review outcome
str(iphone_smallMatrix_RFE)
```

```
'data.frame':   12973 obs. of  10 variables:
 $ iphone         : int  1 1 1 1 1 41 1 1 1 1 ...
 $ htcphone       : int  0 0 0 0 0 0 0 0 0 0 ...
 $ samsunggalaxy  : int  0 0 0 0 0 0 0 0 0 0 ...
 $ googleandroid  : int  0 0 0 0 0 0 0 0 0 0 ...
 $ sonyxperia     : int  0 0 0 0 0 0 0 0 0 0 ...
 $ samsungcampos  : int  0 0 0 0 0 0 0 0 0 0 ...
 $ ios            : int  0 0 0 0 0 6 0 0 0 0 ...
 $ iphonecampos   : int  0 0 0 0 0 1 1 0 0 0 ...
 $ nokialumina    : int  0 0 0 0 0 0 0 0 0 0 ...
 $ iphonesentiment: int  0 0 0 0 0 4 4 0 0 0 ...
```

Model Building

Hide

```
# create 10-fold cross validation fitcontrol
fitControl <- trainControl(method = "cv", number = 10)
```

Model of dataframe without highly correlated variables

Hide

```
# convert variable types, categorical
iphone_smallMatrix_RFE$iphonesentiment <- as.factor(iphone_smallMatrix_RFE$iphonesentime
nt)
```

Train and Test Set:

Hide

```
# Create Train and Test Set for iphoneDFBig
# create 75% sample of row indices
in_training <-createDataPartition(iphone_smallMatrix_RFE$iphonesentiment, p = .7, list =
FALSE)
# create 75% sample of data and save it to trainData
trainData_iphone_smallMatrix_RFE <- iphone_smallMatrix_RFE[in_training, ]
 # create 25% sample of data and save it to test_data
testData_iphone_smallMatrix_RFE <- iphone_smallMatrix_RFE[-in_training, ]
# verify split percentages
nrow(trainData_iphone_smallMatrix_RFE) / nrow(iphone_smallMatrix_RFE)
```

```
[1] 0.7001465
```

Hide

```
#c5
c5_iphone_smallMatrix_RFE <- train(iphonesentiment ~., data = trainData_iphone_smallMatr
ix_RFE, method = "C5.0",
                trControl = fitControl)
```

Hide

```
# randomforest
rf_iphone_smallMatrix_RFE <- train(iphonesentiment ~., data = trainData_iphone_smallMatr
ix_RFE, method = "rf",
                trControl = fitControl)
```

We won't try the following models because in the main dataset, the dataset with correlated variables removed, and nearzerovariance variables, removed, these models did not do very well.

Hide

```
# svm   (kernlab)
svm_iphone_smallMatrix_RFE <- train(iphonesentiment ~., data = trainData_iphone_smallMat
rix_RFE, method = "svmLinear",
                trControl = fitControl)
```

Hide

```
# kknn
kknn_iphone_smallMatrix_RFE <- train(iphonesentiment ~., data = trainData_iphone_smallMa
trix_RFE, method = "kknn",
                trControl = fitControl)
```

Hide

```
# gbm
#gbm_iphone_smallMatrix_RFE <- train(iphonesentiment ~., data = trainData_iphone_smallMa
trix_RFE, method = "gbm",
#                 trControl = fitControl)
```

Compare Accuracy on Prediction Results:

Hide

```
#c5
prediction_c5_iphone_smallMatrix_RFE <- predict(c5_iphone_smallMatrix_RFE, testData_ipho
ne_smallMatrix_RFE)
postResample(prediction_c5_iphone_smallMatrix_RFE, testData_iphone_smallMatrix_RFE$iphon
esentiment)
```

```
 Accuracy     Kappa
0.7311054 0.4554451
```

Hide

```
#randomforest
prediction_rf_iphone_smallMatrix_RFE  <- predict(rf_iphone_smallMatrix_RFE, testData_iph
one_smallMatrix_RFE)
postResample(prediction_rf_iphone_smallMatrix_RFE, testData_iphone_smallMatrix_RFE$iphon
esentiment)
```

```
 Accuracy     Kappa
0.7329049 0.4594732
```

Hide

```
#svm
prediction_svm_iphone_smallMatrix_RFE  <- predict(svm_iphone_smallMatrix_RFE, testData_i
phone_smallMatrix_RFE)
postResample(prediction_svm_iphone_smallMatrix_RFE, testData_iphone_smallMatrix_RFE$ipho
nesentiment)
# kknn
prediction_kknn_iphone_smallMatrix_RFE  <- predict(kknn_iphone_smallMatrix_RFE, testData
_iphone_smallMatrix_RFE)
postResample(prediction_kknn_iphone_smallMatrix_RFE, testData_iphone_smallMatrix_RFE$iph
onesentiment)
```

Hide

```
modelData_iphone_smallMatrix_RFE <- resamples(list(C50 = c5_iphone_smallMatrix_RFE, rand
omForest = rf_iphone_smallMatrix_RFE))

#svMLinear = svm_iphone_smallMatrix_RFE,kknn = kknn_iphone_smallMatrix_RFE))
```

Hide

```
summary(modelData_iphone_smallMatrix_RFE)
```

```

Call:
summary.resamples(object = modelData_iphone_smallMatrix_RFE)

Models: C50, randomForest
Number of resamples: 10

Accuracy
                  Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
C50          0.7133407 0.7148361 0.7222222 0.7221139 0.7263476 0.7381738    0
randomForest 0.7124040 0.7193956 0.7260711 0.7259796 0.7334067 0.7364939    0

Kappa
                  Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
C50          0.4089919 0.4148607 0.4348437 0.4342250 0.4463892 0.4783986    0
randomForest 0.4069394 0.4271471 0.4406466 0.4422311 0.4622529 0.4683431    0
```