task3_galaxy_rfe

Code

data

Hide

```
samsung <- read.csv("galaxy_smallmatrix_labeled_9d.csv")
```

#rfe

Hide

```
# Let's sample the data before using RFE
samsung_Sample <- samsung[sample(1:nrow(samsung), 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 galaxysentiment)
rfeResults <- rfe(samsung_Sample[,1:10],
                  samsung_Sample$galaxysentiment,
                  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	1.490	0.3375	1.135	0.1207	0.09974	0.07391	
2	1.434	0.3936	1.126	0.1031	0.10483	0.06379	
3	1.427	0.4034	1.111	0.1041	0.10240	0.07186	
4	1.415	0.4175	1.099	0.1043	0.10309	0.06637	
5	1.427	0.4155	1.120	0.1022	0.10427	0.06546	
6	1.390	0.4245	1.037	0.1196	0.10696	0.07170	*

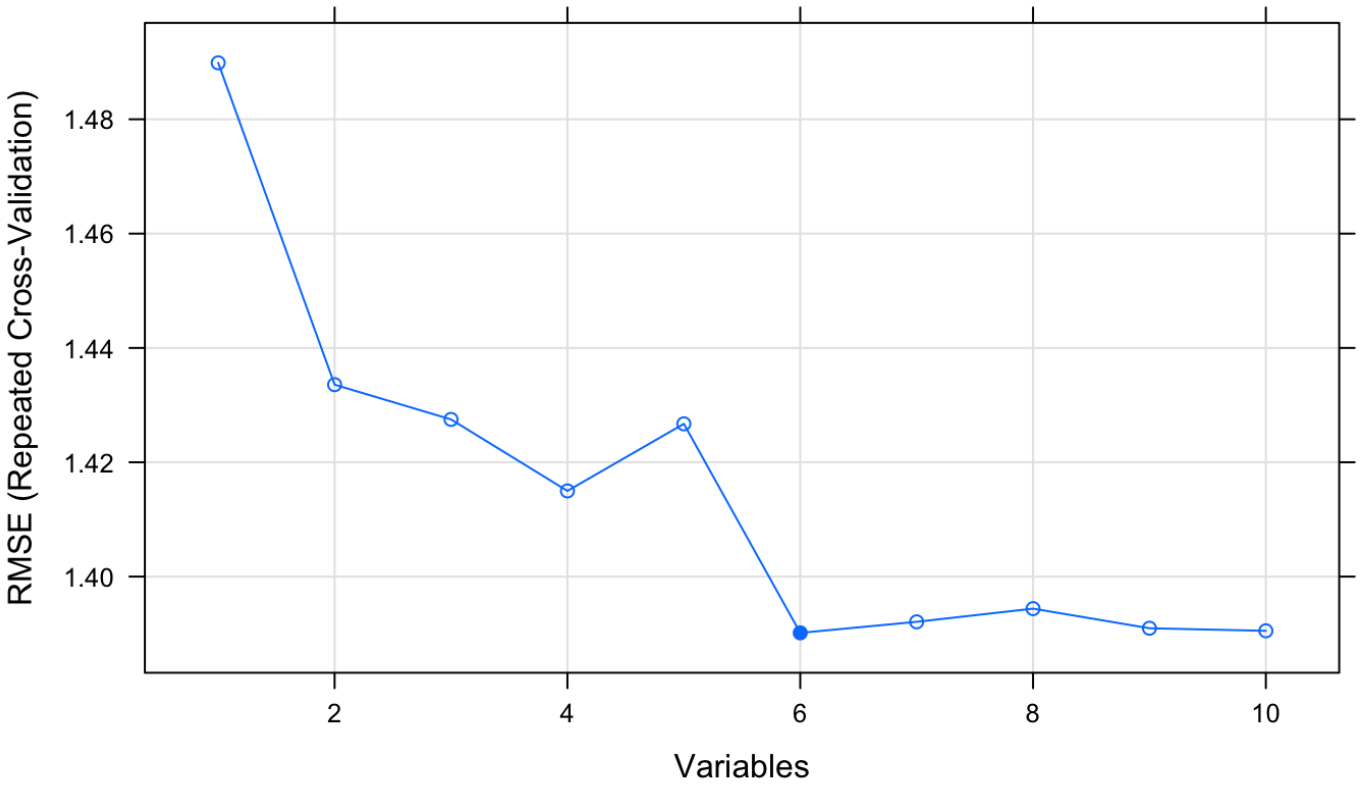
	Variables	RMSE	Rsquared	MAE	RMSESD	RsquaredSD	MAESD	Selected
	<S3: AsIs>	<S3: AsIs>	<S3: AsIs>	<S3: AsIs>	<S3: AsIs>	<S3: AsIs>	<S3: AsIs>	<S3: AsIs>
7	7	1.392	0.4232	1.041	0.1199	0.10753	0.07167	
8	8	1.394	0.4219	1.047	0.1192	0.10781	0.07221	
9	9	1.391	0.4214	1.022	0.1264	0.10910	0.07632	
10	10	1.391	0.4223	1.028	0.1248	0.10858	0.07611	

1-10 of 10 rows

```
The top 5 variables (out of 6):
   iphone, samsunggalaxy, htcphone, googleandroid, sonyxperia
```

Hide

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



create data with rfe features

Hide

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

# add the dependent variable to iphoneRFE
samsung_RFE$galaxysentiment <- samsung$galaxysentiment

# review outcome
str(samsung_RFE)
```

```
'data.frame':   12911 obs. of  7 variables:
 $ iphone         : int  1 1 1 0 1 2 1 1 4 1 ...
 $ samsunggalaxy  : int  0 0 1 0 0 0 0 0 0 0 ...
 $ htcphone       : int  0 0 0 1 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 1 0 0 0 0 0 0 0 ...
 $ galaxysentiment: int  5 3 3 0 1 0 3 5 5 5 ...
```

convert variable types

Hide

```
# convert variable types, categorical
samsung_RFE$galaxysentiment <- as.factor(samsung_RFE$galaxysentiment)
```

Train and Test Set

Hide

```


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

Train and Test Set:

Hide

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

```
[1] 0.7001781
```

Train Models

Hide

```
#c5
c5_samsung_RFE <- train(galaxysentiment ~., data = trainData_samsung_RFE, method = "C5.
0",
                trControl = fitControl)
```

Hide

```
# randomforest
rf_samsung_RFE <- train(galaxysentiment ~., data = trainData_samsung_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_samsung_RFE <- train(galaxysentiment ~., data = trainData_samsung_RFE, method = "sv
mLinear",
#                trControl = fitControl)
```

Hide

```
# kknn
kknn_samsung_RFE <- train(galaxysentiment ~., data = trainData_samsung_RFE, method = "kk
nn",
               trControl = fitControl)
```

Hide

```
# gbm
#gbm_samsung_RFE <- train(galaxysentiment ~., data = trainData_samsung_RFE, method = "gb
m",
#                trControl = fitControl)
```

ModelSummary

Compare Accuracy on Prediction Results:

Hide

```
#c5
prediction_c5_samsung_RFE <- predict(c5_samsung_RFE, testData_samsung_RFE)
postResample(prediction_c5_samsung_RFE, testData_samsung_RFE$galaxysentiment)
```

```
 Accuracy     Kappa
0.7207440 0.4107961
```

Hide

```
#randomforest
prediction_rf_samsung_RFE  <- predict(rf_samsung_RFE, testData_samsung_RFE)
postResample(prediction_rf_samsung_RFE, testData_samsung_RFE$galaxysentiment)
```

```
 Accuracy     Kappa
0.7235856 0.4178725
```

Hide

```
# kknn
prediction_kknn_samsung_RFE  <- predict(kknn_samsung_RFE, testData_samsung_RFE)
postResample(prediction_kknn_samsung_RFE, testData_samsung_RFE$galaxysentiment)
```

```
 Accuracy     Kappa
0.2544562 0.1268873
```

Hide

```
#svm
#prediction_svm_samsung_RFE  <- predict(svm_samsung_RFE, testData_samsung_RFE)
#postResample(prediction_svm_samsung_RFE, testData_samsung_RFE$galaxysentiment)
```

Hide

```
modelData_samsung_RFE <- resamples(list(C50 = c5_samsung_RFE, randomForest = rf_samsung_
RFE, kknn = kknn_samsung_RFE))

#svMLinear = svm_samsung_RFE,kknn = kknn_samsung_RFE))
```

Hide

```
summary(modelData_samsung_RFE)
```

```
Call:
summary.resamples(object = modelData_samsung_RFE)

Models: C50, randomForest, kknn
Number of resamples: 10

Accuracy
                  Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
C50          0.7090708 0.7171321 0.7226516 0.7238919 0.7307912 0.7386489    0
randomForest 0.7079646 0.7236516 0.7276246 0.7270981 0.7350755 0.7411504    0
kknn         0.2101770 0.2404536 0.4580574 0.4525807 0.6566999 0.7082873    0

Kappa
                  Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
C50          0.3705352 0.3943972 0.4104220 0.4136406 0.4344960 0.4566995    0
randomForest 0.3713226 0.4139719 0.4229353 0.4220274 0.4408536 0.4542620    0
kknn         0.1125298 0.1243480 0.2371886 0.2400347 0.3462085 0.3825112    0
```