justIphone_variables

Code ▾

Hide

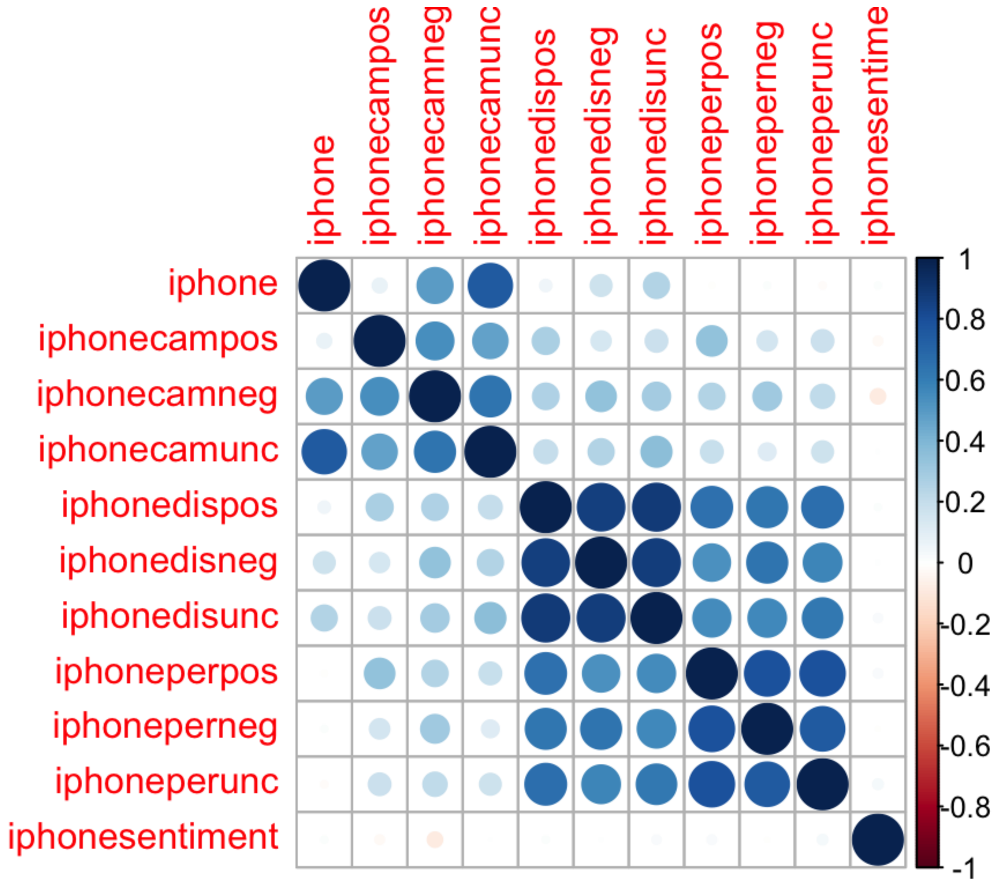
```
# keep only variables for iphones
toFilter <- grepl('iphone', colnames(iphone_smallMatrix))
 iphones <- iphone_smallMatrix[toFilter]
colnames(iphones)
```

```
 [1] "iphone"          "iphonecampos"    "iphonecamneg"    "iphonecamunc"    "iphonedisp
os"
 [6] "iphonedisneg"    "iphonedisunc"    "iphoneperpos"    "iphoneperneg"    "iphoneperu
nc"
[11] "iphonesentiment"
```

Correlation

Hide

```
corr_matrix <- cor(iphones)
corr_plot <- corrplot(as.matrix(corr_matrix))
```



Hide

```
corr_plot
```

```
                     iphone iphonecampos iphonecamneg iphonecamunc iphonedispos iphoned
isneg
iphone          1.000000000   0.07815733   0.49052359  0.750403174   0.05262462  0.1755
72621
iphonecampos    0.078157326   1.00000000   0.54133997  0.473266316   0.27258655  0.1486
50674
iphonecamneg    0.490523588   0.54133997   1.00000000  0.643460020   0.26198314  0.3468
78956
iphonecamunc    0.750403174   0.47326632   0.64346002  1.000000000   0.20900762  0.2532
53711
iphonedispos    0.052624621   0.27258655   0.26198314  0.209007616   1.00000000  0.8687
65387
iphonedisneg    0.175572621   0.14865067   0.34687896  0.253253711   0.86876539  1.0000
00000
iphonedisunc    0.250929821   0.18831003   0.29907429  0.361321734   0.88302623  0.8799
50578
iphoneperpos   -0.009507666   0.34833242   0.25756896  0.190248578   0.65935383  0.5308
88336
iphoneperneg    0.013863107   0.15191863   0.30887521  0.113175498   0.63776843  0.6409
95104
iphoneperunc   -0.016037424   0.18725962   0.21757939  0.174433158   0.66523752  0.5700
44418
iphonesentiment  0.014858654  -0.02973122  -0.08396314  0.001443485   0.01454682  0.0031
44905
               iphonedisunc iphoneperpos iphoneperneg iphoneperunc iphonesentiment
iphone           0.25092982 -0.009507666  0.013863107  -0.01603742     0.014858654
iphonecampos     0.18831003  0.348332416  0.151918629   0.18725962    -0.029731217
iphonecamneg     0.29907429  0.257568960  0.308875213   0.21757939    -0.083963139
iphonecamunc     0.36132173  0.190248578  0.113175498   0.17443316     0.001443485
iphonedispos     0.88302623  0.659353827  0.637768430   0.66523752     0.014546824
iphonedisneg     0.87995058  0.530888336  0.640995104   0.57004442     0.003144905
iphonedisunc     1.00000000  0.554364879  0.564479458   0.62392944     0.027172723
iphoneperpos     0.55436488  1.000000000  0.794832452   0.79182763     0.029637900
iphoneperneg     0.56447946  0.794832452  1.000000000   0.75948372    -0.004804058
iphoneperunc     0.62392944  0.791827630  0.759483720   1.00000000     0.037199859
iphonesentiment  0.02717272  0.029637900 -0.004804058   0.03719986     1.000000000
```

Hide

```
# run this for any features that are h
any_over_80 <- function(my_matrix) any(my_matrix > .8 & my_matrix < 1, na.rm = TRUE)
any_under_80 <- function(my_matrix) any(my_matrix < -.8 & my_matrix > -1, na.rm = TRUE)
```

Hide

```
# remove features with collinearity, correlation greater than .8, FOR small corr_matrix
corr_matrix %>%
  focus_if(any_over_80, mirror = TRUE)
```

rowname <chr>	iphonedispos <dbl>	iphonedisneg <dbl>	iphonedisunc <dbl>
iphonedispos	*NA*	0.8687654	0.8830262
iphonedisneg	0.8687654	*NA*	0.8799506
iphonedisunc	0.8830262	0.8799506	*NA*

3 rows

Let’s drop these variables: iphonedispos, iphonedisneg, iphonedisunc

Remove columns 5,6,7

Not all models are affected by collinearity

Hide

```
# keep columns only for iphone and samsunggalaxy
iphones_corr <- iphones[,-(5:7)]
```

NZR

Hide

```
#nearZeroVar() with saveMetrics = TRUE returns an object containing a table including: frequency ratio, percentage unique, zero variance and near zero variance

nzvMetrics <- nearZeroVar(iphones, saveMetrics = TRUE)
nzvMetrics
```

	freqRatio <dbl>	percentUnique <dbl>	zeroVar <lgl>	nzv <lgl>
iphone	5.041322	0.2081246	FALSE	FALSE
iphonecampos	10.524697	0.2312495	FALSE	FALSE
iphonecamneg	19.517529	0.1310414	FALSE	TRUE
iphonecamunc	16.764205	0.1618747	FALSE	FALSE
iphonedispos	6.792440	0.2466662	FALSE	FALSE
iphonedisneg	10.084428	0.1849996	FALSE	FALSE
iphonedisunc	11.471875	0.2081246	FALSE	FALSE
iphoneperpos	9.297834	0.1927079	FALSE	FALSE
iphoneperneg	11.054137	0.1695830	FALSE	FALSE
iphoneperunc	13.018349	0.1233331	FALSE	FALSE

1-10 of 11 rows Previous **1** 2 Next

Hide

```
# returns column 2, iphonecamunc, same as nvzMetrics
# nearZeroVar() with saveMetrics = FALSE returns an vector
nzv <- nearZeroVar(iphones, saveMetrics = FALSE)
nzv
```

```
[1] 3
```

Hide

```
# create a new data set and remove near zero variance features
iphones_nvz <- iphones[,-nzv]
str(iphones_nvz)
```

```
'data.frame':   12973 obs. of  10 variables:
 $ iphone         : int  1 1 1 1 1 41 1 1 1 1 ...
 $ iphonecampos   : int  0 0 0 0 0 1 1 0 0 0 ...
 $ iphonecamunc   : int  0 0 0 0 0 7 1 0 0 0 ...
 $ iphonedispos   : int  0 0 0 0 0 1 13 0 0 0 ...
 $ iphonedisneg   : int  0 0 0 0 0 3 10 0 0 0 ...
 $ iphonedisunc   : int  0 0 0 0 0 4 9 0 0 0 ...
 $ iphoneperpos   : int  0 1 0 1 1 0 5 3 0 0 ...
 $ iphoneperneg   : int  0 0 0 0 0 0 4 1 0 0 ...
 $ iphoneperunc   : int  0 0 0 1 0 0 5 0 0 0 ...
 $ iphonesentiment: Factor w/ 6 levels "0","1","2","3",..: 1 1 1 1 1 5 5 1 1 1 ...
```

visualize variable with nzv

Hide

```
plot_ly(iphone_smallMatrix, x= ~iphone_smallMatrix$iphone, type='histogram')
plot_ly(iphone_smallMatrix, x= ~iphone_smallMatrix$iphonecampos, type='histogram')
plot_ly(iphone_smallMatrix, x= ~iphone_smallMatrix$iphonecamneg, type='histogram')
plot_ly(iphone_smallMatrix, x= ~iphone_smallMatrix$iphonecamunc, type='histogram')
plot_ly(iphone_smallMatrix, x= ~iphone_smallMatrix$iphonecamneg, type='histogram')
plot_ly(iphone_smallMatrix, x= ~iphone_smallMatrix$iphonedispos, type='histogram')
plot_ly(iphone_smallMatrix, x= ~iphone_smallMatrix$iphonedisneg, type='histogram')
plot_ly(iphone_smallMatrix, x= ~iphone_smallMatrix$iphonedisunc, type='histogram')
plot_ly(iphone_smallMatrix, x= ~iphone_smallMatrix$iphoneperpos, type='histogram')
plot_ly(iphone_smallMatrix, x= ~iphone_smallMatrix$iphoneperneg, type='histogram')
plot_ly(iphone_smallMatrix, x= ~iphone_smallMatrix$iphoneperunc, type='histogram')
plot_ly(iphone_smallMatrix, x= ~iphone_smallMatrix$iphonesentiment, type='histogram')
```

#rfe

Hide

```
# Let's sample the data before using RFE
iphoneSample <- iphones[sample(1:nrow(iphones), 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)
rfeResultsSMALL <- rfe(iphoneSample[,1:10],
                  iphoneSample$iphonesentiment,
                  sizes=(1:10),
                  rfeControl=ctrl)

# Get results
rfeResultsSMALL
```

```


Recursive feature selection

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

Resampling performance over subset size:
```

	Variables	RMSE	Rsquared	MAE	RMSESD	RsquaredSD	MAESD	Selected
	<S3: AsIs>	<S3: AsIs>	<S3: AsIs>	<S3: AsIs>	<S3: AsIs>	<S3: AsIs>	<S3: AsIs>	<S3: AsIs>
1	1	1.567	0.2675	1.198	0.1407	0.10288	0.10058	
2	2	1.547	0.2876	1.203	0.1420	0.10489	0.09385	
3	3	1.555	0.2834	1.227	0.1359	0.10263	0.08864	
4	4	1.563	0.2811	1.239	0.1347	0.10285	0.08785	
5	5	1.572	0.2753	1.250	0.1333	0.10264	0.08771	
6	6	1.552	0.2815	1.182	0.1425	0.09936	0.09356	
7	7	1.550	0.2833	1.180	0.1439	0.09888	0.09569	
8	8	1.546	0.2872	1.179	0.1414	0.09685	0.09200	
9	9	1.541	0.2920	1.154	0.1407	0.09595	0.09551	
10	10	1.540	0.2929	1.157	0.1425	0.09848	0.09683	*

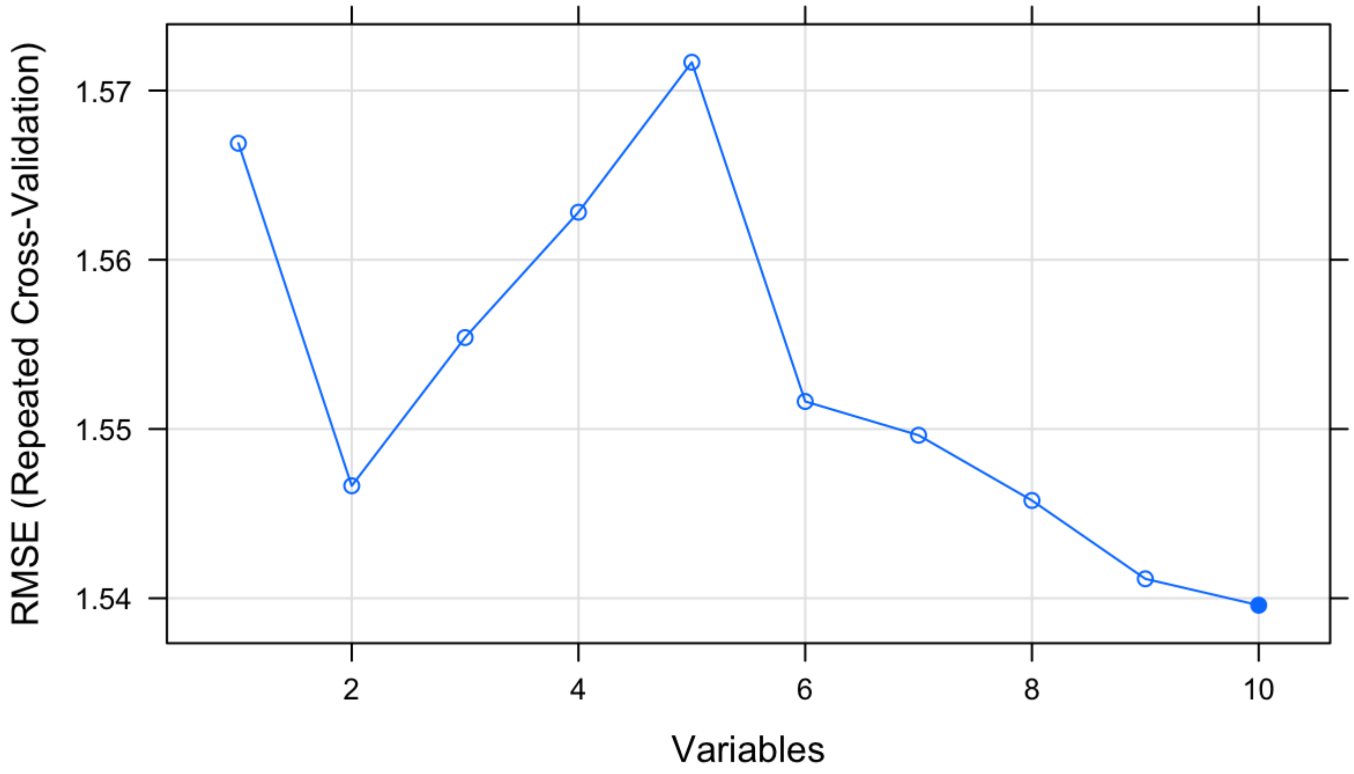
1-10 of 10 rows

```


The top 5 variables (out of 10):
   iphone, iphonecamneg, iphoneperunc, iphonedisneg, iphonedisunc
```

Hide

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



Hide

```
# create new data set with rfe recommended features
iphones_RFE <- iphones[,predictors(rfeResultsSMALL)]

# add the dependent variable to iphoneRFE
iphones_RFE$iphonesentiment <- iphones$iphonesentiment

# review outcome
str(iphones_RFE)
```

```
'data.frame':	12973 obs. of  11 variables:
 $ iphone         : int  1 1 1 1 1 41 1 1 1 1 ...
 $ iphonecamneg   : int  0 0 0 0 0 3 1 0 0 0 ...
 $ iphoneperunc   : int  0 0 0 1 0 0 5 0 0 0 ...
 $ iphonedisneg   : int  0 0 0 0 0 3 10 0 0 0 ...
 $ iphonedisunc   : int  0 0 0 0 0 4 9 0 0 0 ...
 $ iphonedispos   : int  0 0 0 0 0 1 13 0 0 0 ...
 $ iphoneperneg   : int  0 0 0 0 0 0 4 1 0 0 ...
 $ iphonecampos   : int  0 0 0 0 0 1 1 0 0 0 ...
 $ iphoneperpos   : int  0 1 0 1 1 0 5 3 0 0 ...
 $ iphonecamunc   : int  0 0 0 0 0 7 1 0 0 0 ...
 $ iphonesentiment: int  0 0 0 0 0 4 4 0 0 0 ...
```

Model for Regular Data: Iphones

Hide

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

Train and Test Set:

Hide

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

```
[1] 0.7001465
```

Hide

```
#c5
c5_iphones <- train(iphonesentiment ~., data = trainData_iphones, method = "C5.0",
                trControl = fitControl)
```

Hide

```
# randomforest
rf_iphones <- train(iphonesentiment ~., data = trainData_iphones, method = "rf",
                trControl = fitControl)
```

Hide

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

Hide

```
# kknn
kknn_iphones <- train(iphonesentiment ~., data = trainData_iphones, method = "kknn",
                trControl = fitControl)
```

Hide

```
# gbm
#gbm_iphones <- train(iphonesentiment ~., data = trainData_iphones, method = "gbm",
#                 trControl = fitControl)
```

Compare Accuracy on Prediction Results:

Hide

```
#c5
prediction_c5_iphones <- predict(c5_iphones, testData_iphones)
postResample(prediction_c5_iphones, testData_iphones$iphonesentiment)
```

```
 Accuracy     Kappa 
0.7167095 0.4224468 
```

Hide

```
#randomforest
prediction_rf_iphones  <- predict(rf_iphones, testData_iphones)
postResample(prediction_rf_iphones, testData_iphones$iphonesentiment)
```

```
 Accuracy     Kappa 
0.7226221 0.4350340 
```

Hide

```
#svm
prediction_svm_iphones  <- predict(svm_iphones, testData_iphones)
postResample(prediction_svm_iphones, testData_iphones$iphonesentiment)
```

```
  Accuracy      Kappa 
0.60385604 0.09009688 
```

Hide

```
# kknn
prediction_kknn_iphones  <- predict(kknn_iphones, testData_iphones)
postResample(prediction_kknn_iphones, testData_iphones$iphonesentiment)
```

```
 Accuracy     Kappa 
0.3017995 0.1179396 
```

Hide

```
modelData_iphones <- resamples(list(C50 = c5_iphones, randomForest = rf_iphones, svMLine
ar = svm_iphones,kknn = kknn_iphones))
```

Hide

```
summary(modelData_iphones)
```

```


Call:
summary.resamples(object = modelData_iphones)

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

Accuracy 
                  Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
C50          0.6993392 0.7215974 0.7306211 0.7246528 0.7332593 0.7337734    0
randomForest 0.7183718 0.7228969 0.7272729 0.7276227 0.7297732 0.7436744    0
svMLinear    0.5984598 0.6075358 0.6112330 0.6100399 0.6126547 0.6211454    0
kknn         0.2797357 0.2845692 0.2990626 0.2981422 0.3071429 0.3241455    0

Kappa 
                   Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
C50          0.37825534 0.4353286 0.4557795 0.4418250 0.4578631 0.4674848    0
randomForest 0.42670894 0.4380634 0.4467810 0.4484161 0.4505314 0.4863522    0
svMLinear    0.07693399 0.1001085 0.1071164 0.1072959 0.1135268 0.1434714    0
kknn         0.09471739 0.1099662 0.1199002 0.1196086 0.1278466 0.1513078    0
```

Model for Correlated Data:

Hide

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

Train and Test Set:

Hide

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

```
[1] 0.7001465
```

Hide

```
#c5
c5_iphones_corr <- train(iphonesentiment ~., data = trainData_iphones_corr, method = "C
5.0",
                trControl = fitControl)
```

Hide

```
# randomforest
rf_iphones_corr <- train(iphonesentiment ~., data = trainData_iphones_corr, method = "r
f",
                trControl = fitControl)
```

No:

Hide

```
# svm   (kernlab)
svm_iphones_corr <- train(iphonesentiment ~., data = trainData_iphones_corr, method = "s
vmLinear",
                trControl = fitControl)
# kknn
kknn_iphones_corr <- train(iphonesentiment ~., data = trainData_iphones_corr, method =
"kknn",
                trControl = fitControl)

# gbm
#gbm_iphones_corr <- train(iphonesentiment ~., data = trainData_iphones_corr, method =
 "gbm",
#                trControl = fitControl)
```

Compare Accuracy on Prediction Results:

Hide

```
#c5
prediction_c5_iphones_corr <- predict(c5_iphones_corr, testData_iphones_corr)
postResample(prediction_c5_iphones_corr, testData_iphones_corr$iphonesentiment)
```

```
 Accuracy     Kappa
0.6861183 0.3389545
```

Hide

```
#randomforest
prediction_rf_iphones_corr  <- predict(rf_iphones_corr, testData_iphones_corr)
postResample(prediction_rf_iphones_corr, testData_iphones_corr$iphonesentiment)
```

```
 Accuracy     Kappa
0.6884319 0.3455847
```

No:

Hide

```
#svm
prediction_svm_iphones_corr  <- predict(svm_iphones_corr, testData_iphones_corr)
postResample(prediction_svm_iphones_corr, testData_iphones_corr$iphonesentiment)
# kknn
prediction_kknn_iphones_corr  <- predict(kknn_iphones_corr, testData_iphones_corr)
postResample(prediction_kknn_iphones_corr, testData_iphones_corr$iphonesentiment)
```

Hide

```
modelData_iphones_corr <- resamples(list(C50 = c5_iphones_corr, randomForest = rf_iphone
s_corr))

# svMLinear = svm_iphones_corr,kknn = kknn_iphones_corr))
```

Hide

```
summary(modelData_iphones_corr)
```

```


Call:
summary.resamples(object = modelData_iphones_corr)

Models: C50, randomForest
Number of resamples: 10

Accuracy
                  Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
C50          0.6828194 0.6864297 0.6919692 0.6913998 0.6936418 0.7015419    0
randomForest 0.6824697 0.6835304 0.6870177 0.6930564 0.7044070 0.7106711    0

Kappa
                  Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
C50          0.3252032 0.3357033 0.3479342 0.3486942 0.3587904 0.3774182    0
randomForest 0.3275320 0.3382338 0.3438833 0.3576174 0.3828510 0.4027938    0
```

Model for NZR Data:

Hide

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

Train and Test Set:

Hide

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

```
[1] 0.7001465
```

Hide

```
#c5
c5_iphones_nvz <- train(iphonesentiment ~., data = trainData_iphones_nvz, method = "C5.0",
                trControl = fitControl)
```

Hide

```
# randomforest
rf_iphones_nvz <- train(iphonesentiment ~., data = trainData_iphones_nvz, method = "rf",
                trControl = fitControl)
```

No:

Hide

```
# svm   (kernlab)
svm_iphones_nvz <- train(iphonesentiment ~., data = trainData_iphones_nvz, method = "svmLinear",
                trControl = fitControl)
# kknn
kknn_iphones_nvz <- train(iphonesentiment ~., data = trainData_iphones_nvz, method = "kknn",
                trControl = fitControl)

# gbm
#gbm_iphones_nvz <- train(iphonesentiment ~., data = trainData_iphones_nvz, method = "gbm",
#                trControl = fitControl)
```

Compare Accuracy on Prediction Results:

Hide

```
#c5
prediction_c5_iphones_nvz <- predict(c5_iphones_nvz, testData_iphones_nvz)
postResample(prediction_c5_iphones_nvz, testData_iphones_nvz$iphonesentiment)
```

```
 Accuracy     Kappa 
0.7239075 0.4427599 
```

Hide

```
#randomforest
prediction_rf_iphones_nvz  <- predict(rf_iphones_nvz, testData_iphones_nvz)
postResample(prediction_rf_iphones_nvz, testData_iphones_nvz$iphonesentiment)
```

```
 Accuracy     Kappa 
0.7293059 0.4502492 
```

No:

Hide

```
#svm
prediction_svm_iphones_nvz  <- predict(svm_iphones_nvz, testData_iphones_nvz)
postResample(prediction_svm_iphones_nvz, testData_iphones_nvz$iphonesentiment)
# kknn
prediction_kknn_iphones_nvz  <- predict(kknn_iphones_nvz, testData_iphones_nvz)
postResample(prediction_kknn_iphones_nvz, testData_iphones_nvz$iphonesentiment)
```

Hide

```
modelData_iphones_nvz <- resamples(list(C50 = c5_iphones_nvz, randomForest = rf_iphones_
nvz))

# svMLinear = svm_iphones_nvz,kknn = kknn_iphones_nvz))
```

Hide

```
summary(modelData_iphones_nvz)
```

```


Call:
summary.resamples(object = modelData_iphones_nvz)

Models: C50, randomForest 
Number of resamples: 10 

Accuracy 
                  Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
C50          0.6993392 0.7166850 0.7238745 0.7226659 0.7318482 0.7386990    0
randomForest 0.6949339 0.7206163 0.7225277 0.7247615 0.7331127 0.7535754    0

Kappa 
                  Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
C50          0.3851717 0.4245681 0.4389983 0.4383170 0.4608489 0.4750914    0
randomForest 0.3653074 0.4289318 0.4392082 0.4421312 0.4653474 0.5125422    0
```

Models for RFE Data:

Hide

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

Train and Test Set:

Hide

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

```
[1] 0.7001465
```

Hide

```
#c5
c5_iphones_RFE <- train(iphonesentiment ~., data = trainData_iphones_RFE, method = "C5.
0",
                trControl = fitControl)
```

Hide

```
# randomforest
rf_iphones_RFE <- train(iphonesentiment ~., data = trainData_iphones_RFE, method = "rf",
                trControl = fitControl)
```

No:

Hide

```
# svm   (kernlab)
svm_iphones_RFE <- train(iphonesentiment ~., data = trainData_iphones_RFE, method = "svm
Linear",
                trControl = fitControl)
# kknn
kknn_iphones_RFE <- train(iphonesentiment ~., data = trainData_iphones_RFE, method = "kk
nn",
                trControl = fitControl)

# gbm
#gbm_iphones_RFE <- train(iphonesentiment ~., data = trainData_iphones_RFE, method = "gb
m",
#                trControl = fitControl)
```

Compare Accuracy on Prediction Results:

Hide

```
#c5
prediction_c5_iphones_RFE <- predict(c5_iphones_RFE, testData_iphones_RFE)
postResample(prediction_c5_iphones_RFE, testData_iphones_RFE$iphonesentiment)
```

```
 Accuracy     Kappa 
0.7269923 0.4465710 
```

Hide

```
#randomforest
prediction_rf_iphones_RFE  <- predict(rf_iphones_RFE, testData_iphones_RFE)
postResample(prediction_rf_iphones_RFE, testData_iphones_RFE$iphonesentiment)
```

```
 Accuracy     Kappa 
0.7295630 0.4506718 
```

No:

Hide

```
#svm
prediction_svm_iphones_RFE  <- predict(svm_iphones_RFE, testData_iphones_RFE)
postResample(prediction_svm_iphones_RFE, testData_iphones_RFE$iphonesentiment)
# kknn
prediction_kknn_iphones_RFE  <- predict(kknn_iphones_RFE, testData_iphones_RFE)
postResample(prediction_kknn_iphones_RFE, testData_iphones_RFE$iphonesentiment)
```

Hide

```
modelData_iphones_RFE <- resamples(list(C50 = c5_iphones_RFE, randomForest = rf_iphones_
RFE))

# svMLinear = svm_iphones_RFE,kknn = kknn_iphones_RFE))
```

Hide

```
summary(modelData_iphones_RFE)
```

```


Call:
summary.resamples(object = modelData_iphones_RFE)

Models: C50, randomForest 
Number of resamples: 10 

Accuracy 
                  Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
C50          0.6945976 0.7148753 0.7224670 0.7227723 0.7346563 0.7414741    0
randomForest 0.7051705 0.7178285 0.7222237 0.7250933 0.7320295 0.7477974    0

Kappa 
                  Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
C50          0.3669234 0.4215689 0.4399212 0.4385360 0.4653061 0.4829802    0
randomForest 0.3959714 0.4221486 0.4339812 0.4424809 0.4593732 0.5002932    0
```