task3_correlation

Code

Hide

```
# correlation matrix
corr_matrixBIG <- cor(iphone_smallMatrix)
corr_plotBIG <- corrplot(as.matrix(corr_matrixBIG))
corr_plotBIG
```

Hide

```
# returns correlation  greater than .9
corr_df_big <- correlate(iphone_smallMatrix, diagonal = NA) %>% stretch()
# Examine variables with correlation above .9
corr_df_big_filtered <- corr_df_big %>% filter(r > .9)
```

Hide

```
#columns to remove

corr_to_remove <- c("googleperneg","googleperpos","htcdispos","htcphone","ios","iosperne
g","iosperpos","iosperunc","iphone","nokiacamneg","nokiacampos","nokiacamunc","nokiadisn
eg","nokiadispos","nokiadisunc","nokiaperneg","nokiaperpos","nokiaperunc","samsungdisne
g","samsungdispos","samsungdisunc","samsungperneg","samsungperunc")

# We may consider trying this, with keeping the iphone variable
```

Hide

```
iphoneDFBigCOR <- iphone_smallMatrix[ , -which(names(iphone_smallMatrix) %in% corr_to_re
move)]
```

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
iphoneDFBigCOR$iphonesentiment <- as.factor(iphoneDFBigCOR$iphonesentiment)
```

Train and Test Set:

Hide

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

```
[1] 0.7001465
```

Hide

```
#c5
c5_iphoneDFBigCOR <- train(iphonesentiment ~., data = trainData_iphoneDFBigCOR, method =
"C5.0",
                trControl = fitControl)
```

Hide

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

Hide

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

Hide

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

Hide

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

Compare Accuracy on Prediction Results:

Hide

```
#c5
prediction_c5_iphoneDFBigCOR <- predict(c5_iphoneDFBigCOR, testData_iphoneDFBigCOR)
postResample(prediction_c5_iphoneDFBigCOR, testData_iphoneDFBigCOR$iphonesentiment)
```

```
 Accuracy     Kappa
0.7341902 0.4789958
```

Hide

```
#randomforest
prediction_rf_iphoneDFBigCOR  <- predict(rf_iphoneDFBigCOR, testData_iphoneDFBigCOR)
postResample(prediction_rf_iphoneDFBigCOR, testData_iphoneDFBigCOR$iphonesentiment)
```

```
 Accuracy     Kappa
0.7334190 0.4803428
```

Hide

```
#svm
prediction_svm_iphoneDFBigCOR  <- predict(svm_iphoneDFBigCOR, testData_iphoneDFBigCOR)
postResample(prediction_svm_iphoneDFBigCOR, testData_iphoneDFBigCOR$iphonesentiment)
```

```
 Accuracy     Kappa
0.6822622 0.3420238
```

Hide

```
# kknn
prediction_kknn_iphoneDFBigCOR  <- predict(kknn_iphoneDFBigCOR, testData_iphoneDFBigCOR)
postResample(prediction_kknn_iphoneDFBigCOR, testData_iphoneDFBigCOR$iphonesentiment)
```

```
 Accuracy     Kappa
0.2958869 0.1278011
```

Hide

```
modelData_iphoneDFBigCOR <- resamples(list(C50 = c5_iphoneDFBigCOR, randomForest = rf_ip
honeDFBigCOR, svMLinear = svm_iphoneDFBigCOR,kknn = kknn_iphoneDFBigCOR))
```

Hide

```
summary(modelData_iphoneDFBigCOR)
```

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

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

Accuracy
                  Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
C50          0.7114537 0.7132249 0.7164112 0.7180459 0.7238148 0.7260726    0
randomForest 0.7034179 0.7107863 0.7224559 0.7211219 0.7306733 0.7403740    0
svMLinear    0.6604190 0.6766711 0.6818905 0.6803879 0.6863987 0.6912088    0
kknn         0.2855568 0.2943282 0.3153554 0.3102416 0.3220484 0.3314978    0

Kappa
                  Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
C50          0.4221056 0.4345884 0.4389593 0.4433198 0.4538423 0.4645744    0
randomForest 0.4152328 0.4302072 0.4536354 0.4546140 0.4791705 0.4985824    0
svMLinear    0.2869824 0.3344127 0.3417492 0.3370345 0.3464884 0.3693383    0
kknn         0.1141310 0.1194301 0.1463461 0.1392616 0.1535684 0.1658100    0
```