Project-Classification

```r
train_x <- read.csv ('./data/data_2022-05-08/even/10/X_train.csv',
                   header = TRUE,
                   sep = ",",
                   strip.white = TRUE,
                   na.strings = "?",
                   stringsAsFactors = TRUE,
                   fill = TRUE)




train_y <- read.csv ('./data/data_2022-05-08/even/10/y_train.csv',
                   header = TRUE,
                   sep = ",",
                   strip.white = TRUE,
                   na.strings = "?",
                   stringsAsFactors = TRUE,
                   fill = TRUE)


test_x <- read.csv ('./data/data_2022-05-08/even/10/X_test.csv',
                   header = TRUE,
                   sep = ",",
                   strip.white = TRUE,
                   na.strings = "?",
                   stringsAsFactors = TRUE,
                   fill = TRUE)

test_y <- read.csv ('./data/data_2022-05-08/even/10/y_test.csv',
                   header = TRUE,
                   sep = ",",
                   strip.white = TRUE,
                   na.strings = "?",
                   stringsAsFactors = TRUE,
                   fill = TRUE)


train_x["interesting_removal"] = list(as.logical(unlist(train_y["interesting_removal_2"])))
test_x["interesting_removal"] = list(as.logical(unlist(test_y["interesting_removal_2"])))






train = train_x
```

```
test = test_x
```

```
model <- glm(interesting_removal ~  lr  + AntiSJW + AntiTheist + Black + Conspiracy + LGBT + LateNightTa
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Logistic Regression

trainPreds = as.integer(predict(model, train, type="response")>=0.5)
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading
```

```
testPreds = as.integer(predict(model, test, type="response")>=0.5)
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading
```

```
trainAccuracy = sum(trainPreds==train$interesting_removal)/dim(train)[1]
testAccuracy = sum(testPreds==test$interesting_removal)/dim(test)[1]






TP_train = sum(as.integer((trainPreds==1) & (train$interesting_removal==1)))
FP_train = sum(as.integer((trainPreds==0) & (train$interesting_removal==1)))
FN_train = sum(as.integer((trainPreds==1) & (train$interesting_removal==0)))

TP_test = sum(as.integer((testPreds==1) & (test$interesting_removal==1)))
FP_test = sum(as.integer((testPreds==0) & (test$interesting_removal==1)))
FN_test = sum(as.integer((testPreds==1) & (test$interesting_removal==0)))


prec_train = TP_train/(TP_train + FP_train)
prec_test = TP_test/(TP_test + FP_test)


recall_train = TP_train/(TP_train + FN_train)
recall_test = TP_test/(TP_test + FN_test)


sprintf("Train Accuracy: %.4f", trainAccuracy)
```

```
## [1] "Train Accuracy: 0.8640"
```

```
sprintf("Test Accuracy: %.4f", testAccuracy)
```

```
## [1] "Test Accuracy: 0.8103"
```

```
sprintf("Train Precision: %.4f", prec_train)
```

```
## [1] "Train Precision: 0.9561"
```

```
sprintf("Test Precision: %.4f", prec_test)
```

```
## [1] "Test Precision: 0.9655"
```

```r
sprintf("Train Recall: %.4f", recall_train)
```

```
## [1] "Train Recall: 0.8074"
```

```r
sprintf("Test Recall: %.4f", recall_test)
```

```
## [1] "Test Recall: 0.7368"
```

```r
#summary(model)

#sink("summary.txt")
#summary(model)
#sink()
```

```r
require(tree)
```

```
## Loading required package: tree
```

```r
model <- tree(interesting_removal ~  lr  + AntiSJW + AntiTheist + Black + Conspiracy + LGBT + LateNight




## Random Forest

trainPreds = as.integer(predict(model, train)>=0.5)
testPreds = as.integer(predict(model, test)>=0.5)

trainAccuracy = sum(trainPreds==train$interesting_removal)/dim(train)[1]
testAccuracy = sum(testPreds==test$interesting_removal)/dim(test)[1]





TP_train = sum(as.integer((trainPreds==1) & (train$interesting_removal==1)))
FP_train = sum(as.integer((trainPreds==0) & (train$interesting_removal==1)))
FN_train = sum(as.integer((trainPreds==1) & (train$interesting_removal==0)))

TP_test = sum(as.integer((testPreds==1) & (test$interesting_removal==1)))
FP_test = sum(as.integer((testPreds==0) & (test$interesting_removal==1)))
FN_test = sum(as.integer((testPreds==1) & (test$interesting_removal==0)))


prec_train = TP_train/(TP_train + FP_train)
prec_test = TP_test/(TP_test + FP_test)


recall_train = TP_train/(TP_train + FN_train)
recall_test = TP_test/(TP_test + FN_test)


sprintf("Train Accuracy: %.4f", trainAccuracy)
```

```
## [1] "Train Accuracy: 0.8596"
```

```
sprintf("Test Accuracy: %.4f", testAccuracy)
```

```
## [1] "Test Accuracy: 0.7759"
```

```
sprintf("Train Precision: %.4f", prec_train)
```

```
## [1] "Train Precision: 0.8596"
```

```
sprintf("Test Precision: %.4f", prec_test)
```

```
## [1] "Test Precision: 0.7931"
```

```
sprintf("Train Recall: %.4f", recall_train)
```

```
## [1] "Train Recall: 0.8596"
```

```
sprintf("Test Recall: %.4f", recall_test)
```

```
## [1] "Test Recall: 0.7667"
```

```
#sink("summary.txt")
#summary(model)
#sink()
```

~~~
```r
library(randomForest)
~~~

```
## randomForest 4.7-1

## Type rfNews() to see new features/changes/bug fixes.
```

```
model <- randomForest(interesting_removal ~  lr  + AntiSJW + AntiTheist + Black + Conspiracy + LGBT + La
```

```
## Warning in randomForest.default(m, y, ...): The response has five or fewer
## unique values. Are you sure you want to do regression?
```

```
## Random Forest

trainPreds = as.integer(predict(model, train, type="response")>=0.5)
testPreds = as.integer(predict(model, test, type="response")>=0.5)

trainAccuracy = sum(trainPreds==train$interesting_removal)/dim(train)[1]
testAccuracy = sum(testPreds==test$interesting_removal)/dim(test)[1]




TP_train = sum(as.integer((trainPreds==1) & (train$interesting_removal==1)))
FP_train = sum(as.integer((trainPreds==0) & (train$interesting_removal==1)))
FN_train = sum(as.integer((trainPreds==1) & (train$interesting_removal==0)))

TP_test = sum(as.integer((testPreds==1) & (test$interesting_removal==1)))
FP_test = sum(as.integer((testPreds==0) & (test$interesting_removal==1)))
FN_test = sum(as.integer((testPreds==1) & (test$interesting_removal==0)))
```

```r
prec_train = TP_train/(TP_train + FP_train)
prec_test = TP_test/(TP_test + FP_test)


recall_train = TP_train/(TP_train + FN_train)
recall_test = TP_test/(TP_test + FN_test)


sprintf("Train Accuracy: %.4f", trainAccuracy)
```

```
## [1] "Train Accuracy: 0.8728"
```

```r
sprintf("Test Accuracy: %.4f", testAccuracy)
```

```
## [1] "Test Accuracy: 0.8448"
```

```r
sprintf("Train Precision: %.4f", prec_train)
```

```
## [1] "Train Precision: 0.9561"
```

```r
sprintf("Test Precision: %.4f", prec_test)
```

```
## [1] "Test Precision: 0.9655"
```

```r
sprintf("Train Recall: %.4f", recall_train)
```

```
## [1] "Train Recall: 0.8195"
```

```r
sprintf("Test Recall: %.4f", recall_test)
```

```
## [1] "Test Recall: 0.7778"
```