# Machine Learning Pset 2

*Karl Jiang*

*January 26, 2017*

## Question 2 - Movie Reviews)

### Data engineering stuff. Data into train, cross validation, test

```
#install.packages("randomForest")
#library(randomForest)
library(Matrix)
MovieReview_train <- read.csv("MovieReview_train.csv")
MovieReview_test <- read.csv("MovieReview_test.csv")


n = nrow(MovieReview_train)
tr_ind = sample(seq_len(n), size = floor(n * 0.8) )

mr_tr = MovieReview_train[tr_ind, ]
y_tr = MovieReview_train$sentiment[tr_ind]

mr_cv = MovieReview_train[-tr_ind, ]
y_cv =  MovieReview_train$sentiment[-tr_ind]

mr_tr = mr_tr[, 1:(ncol(MovieReview_train) - 1) ]
mr_cv = mr_cv[, 1:(ncol(MovieReview_train) - 1)]

tr_sparse = Matrix(as.matrix(mr_tr), sparse = TRUE)
cv_sparse = Matrix(as.matrix(mr_cv), sparse = TRUE)
```

### Random Forest in case nothing works

```
rfc <- randomForest(as.factor(sentiment) ~ ., data = MovieReview_train, randomForest.default = )
dim(rfc$confusion)
summary(rfc)
sentiment <- predict(rfc, newdata = MovieReview_test)
names(sentiment) <- NULL
results <- data.frame(sentiment = sentiment)
write.csv(results, "hw2-2-karljiang.csv")
```

### XGBoost

```
install.packages("xgboost")
library(xgboost)
```

```r
xgb_tr <- xgb.DMatrix(data = tr_sparse, label = y_tr)
xgb_cv <- xgb.DMatrix(data = cv_sparse, label = y_cv)

bstSparse <- xgboost(data = xgb_tr, max.depth = 2, nthread = 2, nrounds = 1000, objective = "binary:logi
pred = predict(bstSparse, newdata = xgb_cv)
prediction <- as.numeric(pred > 0.5)
err <- mean(as.numeric(pred > 0.5) != y_cv)
print(paste("test-error=", err))

params = list(objective = "binary:logistic", eta = 0.1, max_depth = 2:10  )
xgb_cv = xgb.cv(params = params, data = xgb_tr, nrounds = 1, stratified = TRUE, nfold = 4, metric = 'au

summary(xgb_cv)
print(xgb_cv$evaluation_log)
xgb_cv$pred


bstSparse <- xgboost(data = train$data, label = train$label, max.depth = 2, eta = 1, nthread = 2, nroun

data(agaricus.train, package='xgboost')
data(agaricus.test, package='xgboost')
train <- agaricus.train
attributes()
test <- agaricus.test

bstSparse <- xgboost(data = train$data, label = train$label, max.depth = 2, eta = 1, nthread = 2, nroun


summary(bstDense)
help(xgboost)
```