test

Levi Lee

11/17/2020

Set Working Directories and Import Data

```
setwd("~/GitHub/Fall2020-Project4-group-4/doc")
```

```
df_high <- read.csv("../data/highDim_dataset.csv")
df_low <- read.csv("../data/lowDim_dataset.csv")
```

```
packages.used <- c("ggplot2", "WeightedROC", "rpart")

# check packages that need to be installed.
packages.needed <- setdiff(packages.used, intersect(installed.packages()[,1], packages.used))

# install additional packages
if(length(packages.needed) > 0){
   install.packages(packages.needed, dependencies = TRUE)
}


library(ggplot2)
library(WeightedROC)
```

```
## Warning: package 'WeightedROC' was built under R version 4.0.3
```

```
library(rpart)
```

```
## Warning: package 'rpart' was built under R version 4.0.3
```

Introduction

Background: Trees

Cross-Validation

Step 1: Set Controls

```r
K <- 5  # number of CV folds
sample.reweight <- TRUE # run sample reweighting in model training

run.cv.trees_high <- FALSE # run cross-validation on the training set for trees on high dim data
#run.train.trees_high <- TRUE # run evaluation on entire train set on high dim data
#run.test.trees_high <- TRUE # run evaluation on an independent test set on high dim data

run.cv.trees_low <- FALSE # run cross-validation on the training set for trees on low dim data
#run.train.trees_low <- TRUE # run evaluation on entire train set on low dim data
#run.test.trees_low <- TRUE # run evaluation on an independent test set on low dim data
```

```r
# hyperparameters for trees
hyper_grid_trees <- expand.grid(
  cp = c(0.001, 0.005, 0.01, 0.05, 0.1, 0.15),
  maxdepth = c(5, 10, 15, 20, 25, 30)
)
```

Step 2: Train a classification model with training features and responses

```r
feature_train_high = df_high[, -1:-2]
label_train_high = df_high[, 2]

feature_train_low = df_low[, -1:-2]
label_train_low = df_low[, 2]
```

```r
set.seed(5243)

if(run.cv.trees_high){
  res_cv_trees_high <- matrix(0, nrow = nrow(hyper_grid_trees), ncol = 4)
  for(i in 1:nrow(hyper_grid_trees)){
    cat("complexity = ", hyper_grid_trees$cp[i], ", max depth = ", hyper_grid_trees$maxdepth[i],"\n", s
    res_cv_trees_high[i,] <- cv.function(features = feature_train_high, labels = label_train_high,
                                         cp = hyper_grid_trees$cp[i],
                                         maxdepth = hyper_grid_trees$maxdepth[i],
                                         K, reweight = sample.reweight)
  save(res_cv_trees_high, file="../output/res_cv_trees_high.RData")
  }
}else{
  load("../output/res_cv_trees_high.RData")
}
```

High Dimensional Data

```r
set.seed(5243)
```

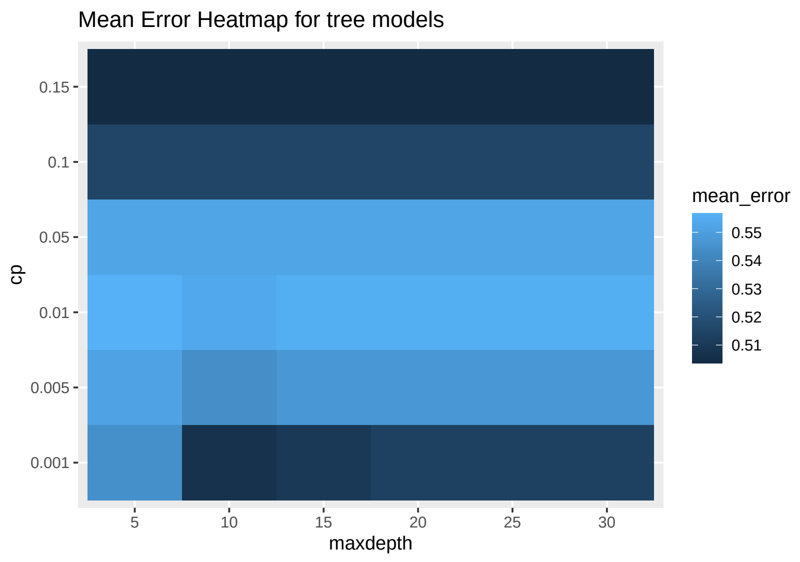
```r
if(run.cv.trees_low){
  res_cv_trees_low <- matrix(0, nrow = nrow(hyper_grid_trees), ncol = 4)
  for(i in 1:nrow(hyper_grid_trees)){
    cat("complexity = ", hyper_grid_trees$cp[i], ", max depth = ", hyper_grid_trees$maxdepth[i],"\n", se
    res_cv_trees_low[i,] <- cv.function(features = feature_train_low, labels = label_train_low,
                                        cp = hyper_grid_trees$cp[i],
                                        maxdepth = hyper_grid_trees$maxdepth[i],
                                        K, reweight = sample.reweight)
  save(res_cv_trees_low, file="../output/res_cv_trees_low.RData")
  }
}else{
  load("../output/res_cv_trees_low.RData")
}
```

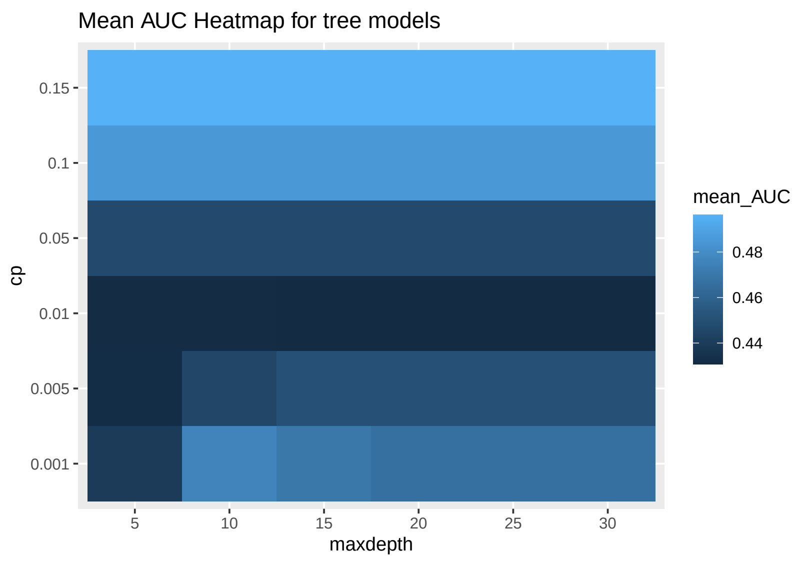
Low Dimensional Data

Step 3: Visualize CV Error and AUC

High Dimensional Data

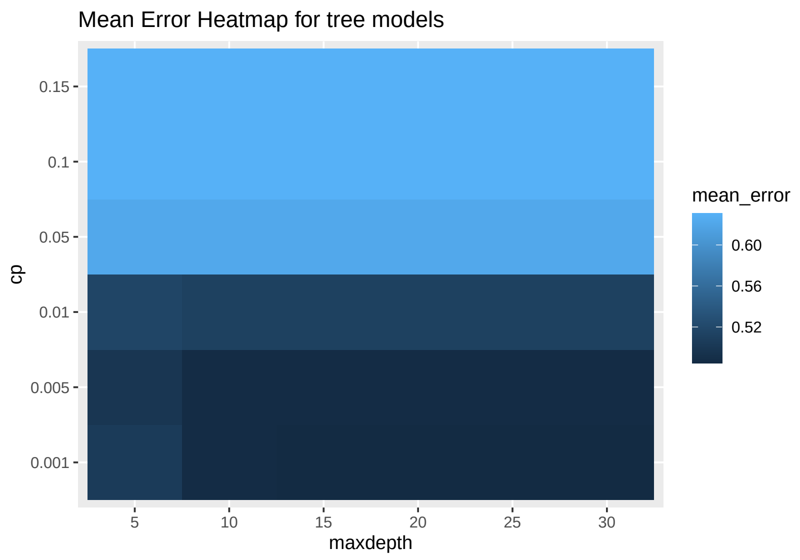
```
##      cp maxdepth mean_error    sd_error  mean_AUC      sd_AUC
## 6  0.15        5  0.5036364 0.008131156 0.4963636 0.008131156
## 12 0.15       10  0.5036364 0.008131156 0.4963636 0.008131156
## 18 0.15       15  0.5036364 0.008131156 0.4963636 0.008131156
## 24 0.15       20  0.5036364 0.008131156 0.4963636 0.008131156
## 30 0.15       25  0.5036364 0.008131156 0.4963636 0.008131156
```

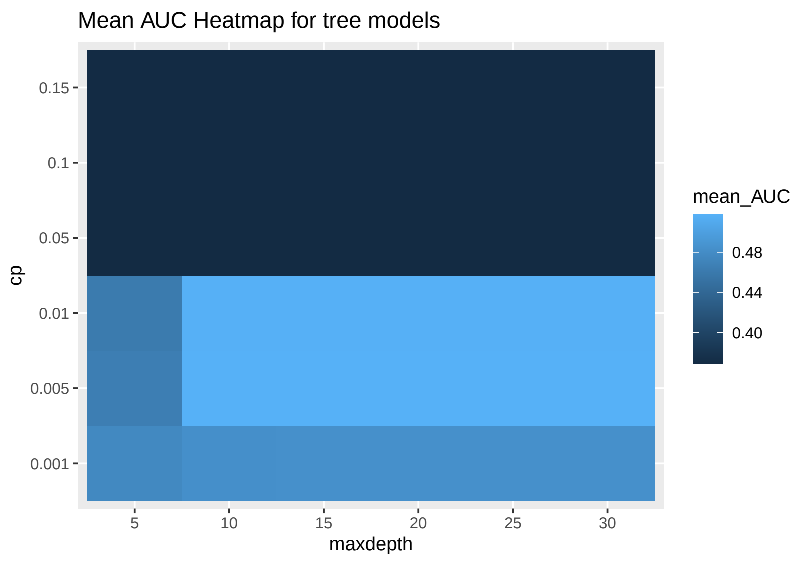




Low Dimensional Data

```
##       cp maxdepth mean_error  sd_error  mean_AUC     sd_AUC
## 8  0.005       10  0.4869266 0.0600334 0.5177229 0.09633321
## 14 0.005       15  0.4869266 0.0600334 0.5177229 0.09633321
## 20 0.005       20  0.4869266 0.0600334 0.5177229 0.09633321
## 26 0.005       25  0.4869266 0.0600334 0.5177229 0.09633321
## 32 0.005       30  0.4869266 0.0600334 0.5177229 0.09633321
```





Propensity Score Estimation

Stratification

Regression Adjustment

Stratification and Regression Adjustment

Results

Insert Comparison Here

Conclusion