testing

Joshua Crisologo: 1009860438

2024-03-31

```
# filtering data and lakes to most recent date of test
 df = data %>%
      group_by(Lake.Name) %>%
      filter(phos_date==max(phos_date)) %>%
      filter(trans_date==max(trans_date))
```

```
# t.test sample test
t.test(df$avg_phos_ug_l)
```

```
##
##  One Sample t-test
##
## data:  df$avg_phos_ug_l
## t = 34.777, df = 1169, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  10.21926 11.44126
## sample estimates:
## mean of x
##  10.83026
```

```
# bootstrapping sample, creating a 95% confidence interval for the
# average phosphorous level
boot_function = function() {
  boot_data = df[sample(nrow(df), replace = TRUE), ]

  boot_mean = mean(boot_data$avg_phos_ug_l, na.rm = TRUE)

  return(boot_mean)
}

quantile(replicate(100,boot_function()), c(0.025, 0.975))
```

```
##     2.5%    97.5%
## 10.30918 11.42726
```

```
# using formulas in regards to TSI to calculate the TSI of each lake
# 0 = oligotrophic
# 1 = mesotrophic
# 2 = eutrophic
df = df %>% mutate(TSI_Depth = (60 - 14.41*log(secchi_depth_m))) %>%
  mutate(TSI_Phos = (14.42*log(avg_phos_ug_l) + 4.15)) %>%
  mutate(TSI = ((TSI_Depth + TSI_Phos) / 2)) %>%
  mutate(classification = case_when(TSI <= 40 ~ "0",
```

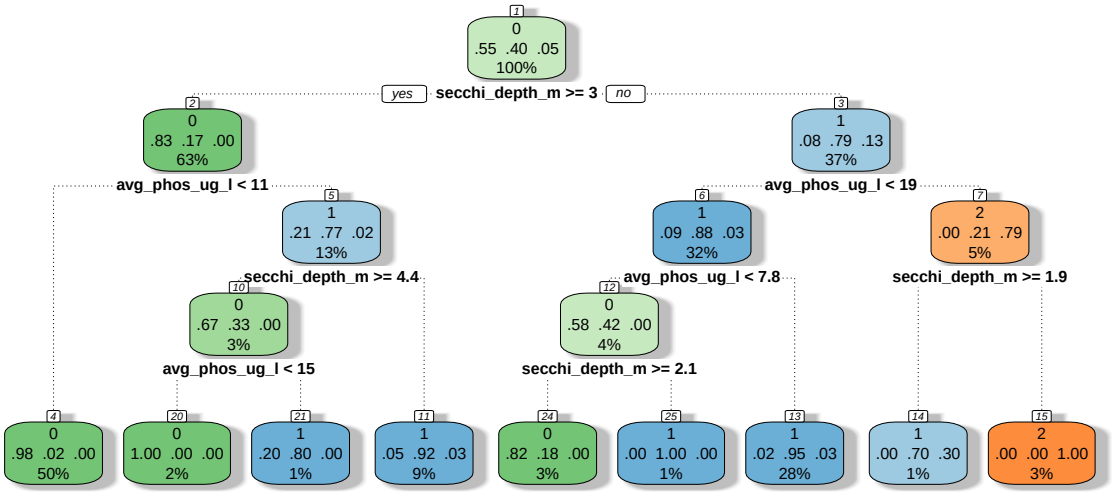
```
                                    TSI <= 50 ~ "1",
                                    TRUE ~ "2"))

# dropping NA values (8)
df_rm = df %>% drop_na()
```

```
# cross validation: splitting df into train and test of 60/40% for model testing
final = df_rm
final = final %>% mutate(group_ind = sample(c("train", "test"),
                                            size=1,
                                            prob = c(0.6, 0.4),
                                            replace = T))

final_train = final %>% filter(group_ind == "train")
final_test = final %>% filter(group_ind == "test")

# implementing decision tree analysis
# relating classification level (0,1,2) to avg phos lvl and secchi depth
library(rpart)
library(rattle)
```

```
## Loading required package: bitops
```

```
## Rattle: A free graphical interface for data science with R.
## Version 5.5.1 Copyright (c) 2006-2021 Togaware Pty Ltd.
## Type 'rattle()' to shake, rattle, and roll your data.
```

```
tree.m = rpart(classification ~ avg_phos_ug_l + secchi_depth_m, data = final_train,
               method = "class")
fancyRpartPlot(tree.m)
```



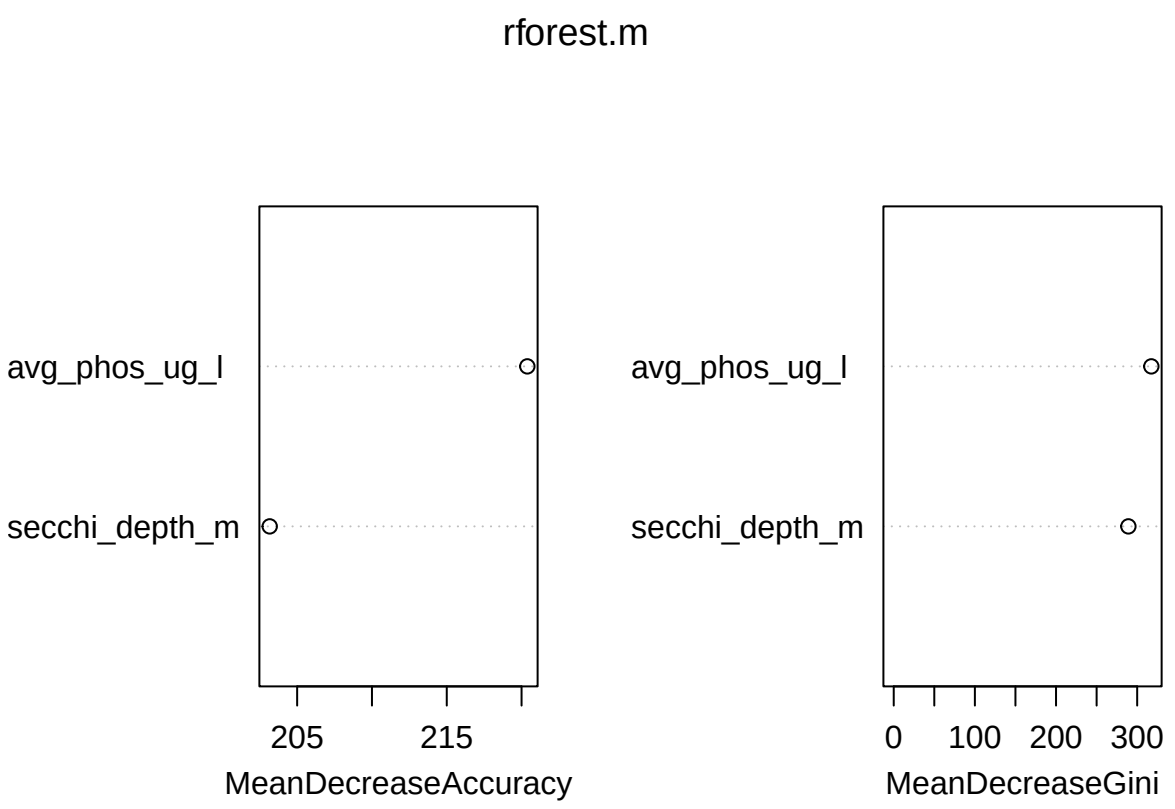
Rattle 2024−Mar−31 21:13:54 rstudio

```
# another test: using randomForest with the same parameters to predict
library(randomForest)
```

```
## randomForest 4.7-1.1
## Type rfNews() to see new features/changes/bug fixes.
##
## Attaching package: 'randomForest'
## The following object is masked from 'package:rattle':
##
##     importance
## The following object is masked from 'package:dplyr':
##
##     combine
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
rforest.m = randomForest(as.factor(classification) ~ avg_phos_ug_l + secchi_depth_m, data=df_rm,
                         ntree=500, importance=TRUE)
```

```
# plotting variable importance
varImpPlot(rforest.m)
```




```
# prediction using decision tree
final_test = final_test %>% ungroup(.) %>% mutate(tree_predictions =
                                        predict(tree.m, newdata = final_test, type = "class"))

# prediction using random forest

final_test = final_test %>% mutate(rforest_predict =
```

```
                                    predict(rforest.m, newdata = final_test))

glimpse(final_test)
```

```
## Rows: 453
## Columns: 19
## $ lat              <int> 432112, 432844, 434143, 435755, 440050, 440830, 44101~
## $ long             <int> 803211, 803806, 802646, 804843, 774233, 804402, 81030~
## $ STN              <int> 7691, 7597, 7110, 205, 7103, 7522, 7248, 1138, 7171, ~
## $ Site.ID          <int> 1, 1, 3, 1, 6, 1, 1, 1, 1, 2, 1, 1, 9, 1, 1, 6, 3, 2,~
## $ Township         <chr> "NEW DUNDEE", "WILMOT", "PUSLINCH", "MINTO", "HILLIER~
## $ Lake.Name        <chr> "ALDER LAKE", "SUNFISH LAKE", "PUSLINCH LAKE", "PIKE ~
## $ Site.Description <chr> "Deep spot", "Mid Lake, Deep Spot", "McCormick Pt", "~
## $ avg_phos_ug_l    <dbl> 23.5, 11.5, 17.5, 24.3, 12.7, 6.2, 21.0, 10.2, 19.6, ~
## $ phos_is_outlier  <chr> "Yes", "No", "No", "No", "No", "No", "No", "No", "No"~
## $ phos_date        <date> 2021-11-20, 2022-11-17, 2018-06-03, 2019-06-24, 2022~
## $ secchi_depth_m   <dbl> 0.8, 3.6, 1.5, 4.5, 2.1, 5.4, 3.0, 3.0, 1.3, 3.0, 6.0~
## $ trans_date       <date> 2021-11-02, 2018-11-11, 2018-07-29, 2017-10-28, 2022~
## $ TSI_Depth        <dbl> 63.21550, 41.54174, 54.15725, 38.32624, 49.30868, 35.~
## $ TSI_Phos         <dbl> 49.67395, 39.36864, 45.42294, 50.15667, 40.79990, 30.~
## $ TSI              <dbl> 56.44472, 40.45519, 49.79009, 44.24146, 45.05429, 33.~
## $ classification   <chr> "2", "1", "1", "1", "1", "0", "1", "1", "2", "0", "0"~
## $ group_ind        <chr> "test", "test", "test", "test", "test", "test", "test~
## $ tree_predictions <fct> 2, 1, 1, 1, 1, 0, 1, 0, 2, 0, 0, 0, 1, 2, 2, 1, 0, 0,~
## $ rforest_predict  <fct> 2, 1, 1, 1, 1, 0, 1, 1, 2, 0, 0, 0, 1, 2, 2, 2, 0, 0,~
```

```
# create the confusion matrix using decision tree model and test accuracy
conmat = table(final_test$classification, final_test$tree_predictions)
sum(diag(conmat))/sum(conmat)
```

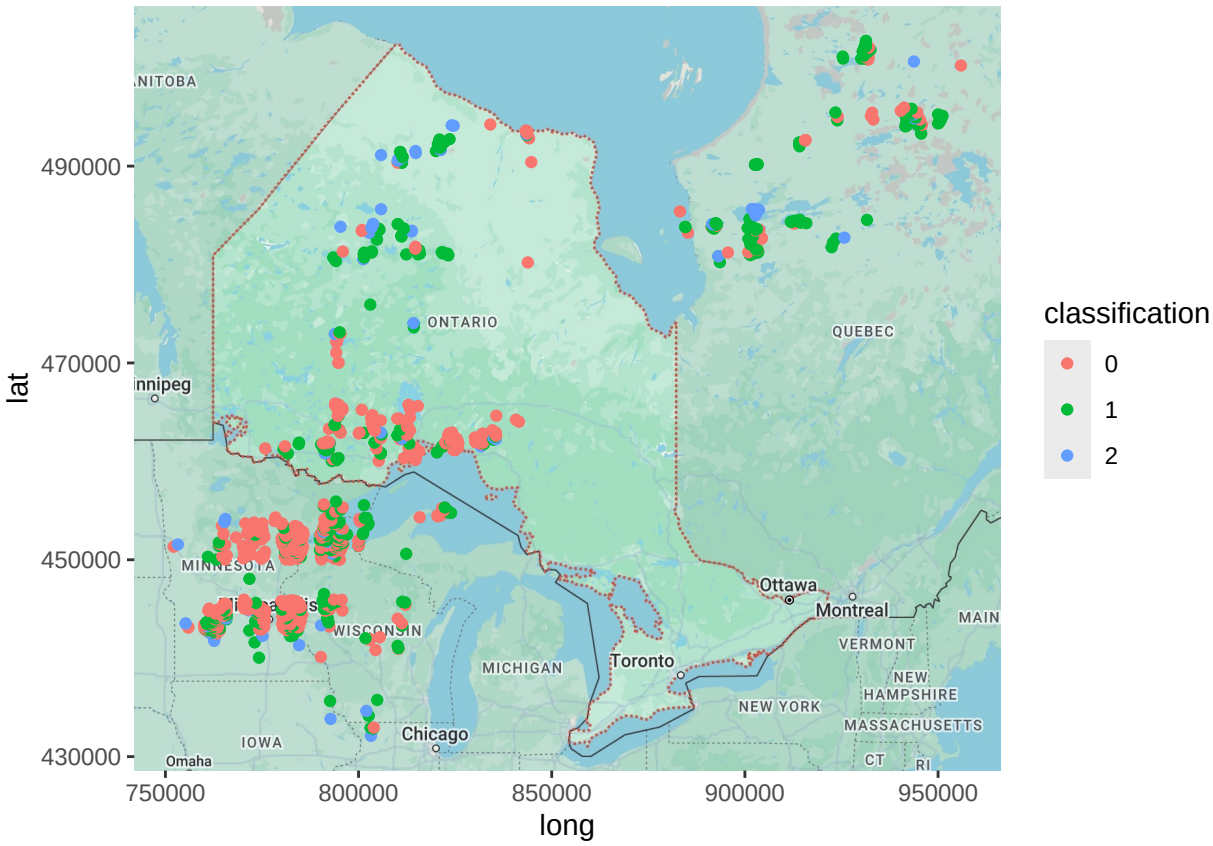
```
## [1] 0.9227373
```

```
# create confusion matrix using random forest model and test accuracy
conmat = table(final_test$classification, final_test$rforest_predict)
sum(diag(conmat))/sum(conmat)
```

```
## [1] 1
```

```
#lat long map according to classification (W.I.P.)
ggplot(df_rm, aes(y=lat, x=long, col=classification)) + geom_point()
```


```
# attempt at relating coords to map of Ontario (W.I.P.)
library(ggpubr)
library(jpeg)
img=readJPEG("ontario.jpg")

ggplot(df_rm, aes(y=lat, x=long, col=classification)) + background_image(img) +
  geom_point()
```


lat
490000
470000
450000
430000
750000
800000
850000
900000
950000
long
ONTARIO
QUEBEC
MINNESOTA
WISCONSIN
MICHIGAN
IOWA
Chicago
Toronto
Ottawa
Montreal
VERMONT
NEW YORK
MASSACHUSETTS
Omaha
classification
0
1
2