Table of Contents

[Load and Prepare Data 1](#_Toc196226509)

[Random Forest with Imputed Data 2](#_Toc196226510)

[Tune Model Hyperparameters 4](#_Toc196226511)

[Random Forest on Complete Cases Only 6](#_Toc196226512)

[Conclusion 10](#_Toc196226513)

## Load and Prepare Data

alldata <- read.csv("../data/csv/alldata-2004-2016v2.csv") %>%  
 filter(!is.na(Total.Biovolume)) %>%  
 select(-YearMonth, -Month, -Cryto.Biovolume, -Cyano.Biovolume, -Diatoms.Biovolume, -Greens.Biovolume, -RB5.ChlA) %>%  
 mutate(Year = as.numeric(Year), Season = as.factor(Season))  
summary(alldata)

## Year Season Total.Biovolume waterlevel.masl  
## Min. :2004 Autumn:36 Min. : 104270 Min. :106.7   
## 1st Qu.:2007 Spring:36 1st Qu.: 2015874 1st Qu.:107.0   
## Median :2010 Summer:37 Median : 4290946 Median :107.2   
## Mean :2010 Winter:31 Mean : 7579646 Mean :107.2   
## 3rd Qu.:2013 NA's : 2 3rd Qu.:10160024 3rd Qu.:107.4   
## Max. :2016 Max. :87549484 Max. :108.0   
## NA's :2 NA's :40   
## Daphnia RB5.NO3 RB5.SRP RB5.SRSi   
## Min. : 0.0000 Min. :0.0100 Min. : 1.020 Min. : 0.0100   
## 1st Qu.: 0.4708 1st Qu.:0.1150 1st Qu.: 4.322 1st Qu.: 0.5825   
## Median : 1.6750 Median :0.6700 Median : 6.520 Median : 1.4025   
## Mean : 6.2361 Mean :0.9035 Mean : 9.927 Mean : 2.2136   
## 3rd Qu.: 6.5250 3rd Qu.:1.6062 3rd Qu.:10.348 3rd Qu.: 3.1012   
## Max. :56.8000 Max. :3.5050 Max. :72.700 Max. :10.1900   
## NA's :102 NA's :6 NA's :40   
## RB5.TP RB5.SD RB5.Temp RB5.pH   
## Min. : 15.45 Min. :0.620 Min. : 1.600 Min. :6.640   
## 1st Qu.: 31.62 1st Qu.:1.150 1st Qu.: 5.292 1st Qu.:7.845   
## Median : 39.84 Median :1.437 Median :10.575 Median :8.360   
## Mean : 46.22 Mean :1.621 Mean :10.574 Mean :8.264   
## 3rd Qu.: 55.80 3rd Qu.:1.850 3rd Qu.:15.600 3rd Qu.:8.793   
## Max. :126.30 Max. :4.245 Max. :21.065 Max. :9.400   
## NA's :6 NA's :35 NA's :2 NA's :43   
## RB5.DO   
## Min. : 77.30   
## 1st Qu.: 93.01   
## Median : 96.52   
## Mean : 96.97   
## 3rd Qu.:100.12   
## Max. :120.95   
## NA's :56

## Random Forest with Imputed Data

### Impute Missing Values

data\_impute <- rfImpute(Total.Biovolume ~ ., data = alldata, ntree = 500)

## | Out-of-bag |  
## Tree | MSE %Var(y) |  
## 500 | 8.908e+13 90.04 |  
## | Out-of-bag |  
## Tree | MSE %Var(y) |  
## 500 | 8.853e+13 89.48 |  
## | Out-of-bag |  
## Tree | MSE %Var(y) |  
## 500 | 9.256e+13 93.56 |  
## | Out-of-bag |  
## Tree | MSE %Var(y) |  
## 500 | 9.231e+13 93.30 |  
## | Out-of-bag |  
## Tree | MSE %Var(y) |  
## 500 | 9.126e+13 92.24 |

### Train-Test Split

trainIndex1 <- createDataPartition(data\_impute$Total.Biovolume, p = 0.7, list = FALSE)  
trainData1 <- data\_impute[trainIndex1, ]  
testData1 <- data\_impute[-trainIndex1, ]

### Fit Random Forest

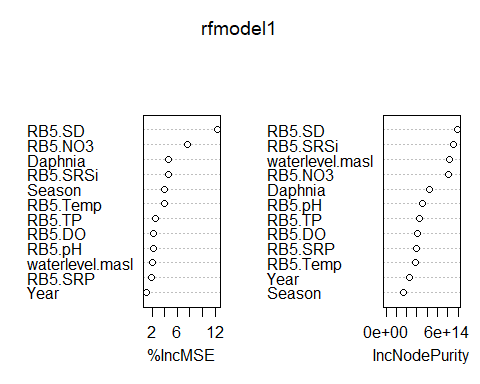
rfmodel1 <- randomForest(Total.Biovolume ~ ., data = trainData1, ntree = 500, mtry = sqrt(ncol(trainData1)-1), importance = TRUE)  
print(rfmodel1)

##   
## Call:  
## randomForest(formula = Total.Biovolume ~ ., data = trainData1, ntree = 500, mtry = sqrt(ncol(trainData1) - 1), importance = TRUE)   
## Type of random forest: regression  
## Number of trees: 500  
## No. of variables tried at each split: 3  
##   
## Mean of squared residuals: 4.594635e+13  
## % Var explained: 14.54

rfmodel1$importance

## %IncMSE IncNodePurity  
## Year 4.197329e+11 2.289577e+14  
## Season 1.471044e+12 1.707812e+14  
## waterlevel.masl 9.991508e+11 6.201864e+14  
## Daphnia 2.409907e+12 4.231888e+14  
## RB5.NO3 4.790721e+12 6.029073e+14  
## RB5.SRP 8.322700e+11 2.951942e+14  
## RB5.SRSi 2.935821e+12 6.515032e+14  
## RB5.TP 1.045831e+12 3.243275e+14  
## RB5.SD 9.020735e+12 6.990395e+14  
## RB5.Temp 1.713124e+12 2.887353e+14  
## RB5.pH 1.060274e+12 3.521908e+14  
## RB5.DO 1.052345e+12 3.056314e+14

varImpPlot(rfmodel1)



### Evaluate Model

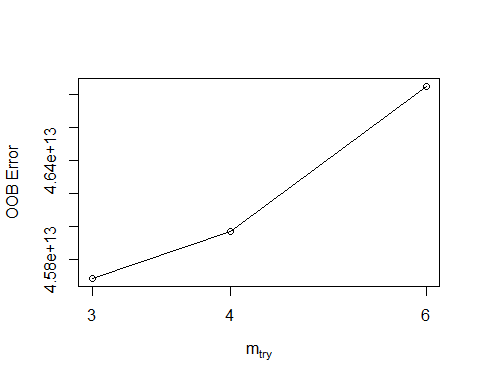
predictions <- predict(rfmodel1, testData1)  
rmse <- sqrt(mean((predictions - testData1$Total.Biovolume)^2))  
rsq <- cor(predictions, testData1$Total.Biovolume)^2  
cat("R²:", rsq, "\nRMSE:", rmse)

## R²: 0.04814227   
## RMSE: 14318130

## Tune Model Hyperparameters

tuned\_model <- tuneRF(trainData1[, -which(names(trainData1) == "Total.Biovolume")], trainData1$Total.Biovolume, stepFactor = 1.5, improve = 0.01, ntreeTry = 500, trace = TRUE)

## mtry = 4 OOB error = 4.596766e+13   
## Searching left ...  
## mtry = 3 OOB error = 4.568297e+13   
## 0.006193347 0.01   
## Searching right ...  
## mtry = 6 OOB error = 4.684678e+13   
## -0.0191247 0.01



### Final Tuned Model

rfmodel1opt <- randomForest(Total.Biovolume ~ ., data = trainData1, ntree = 500, mtry = 6, importance = TRUE)  
print(rfmodel1opt)

##   
## Call:  
## randomForest(formula = Total.Biovolume ~ ., data = trainData1, ntree = 500, mtry = 6, importance = TRUE)   
## Type of random forest: regression  
## Number of trees: 500  
## No. of variables tried at each split: 6  
##   
## Mean of squared residuals: 4.654127e+13  
## % Var explained: 13.43

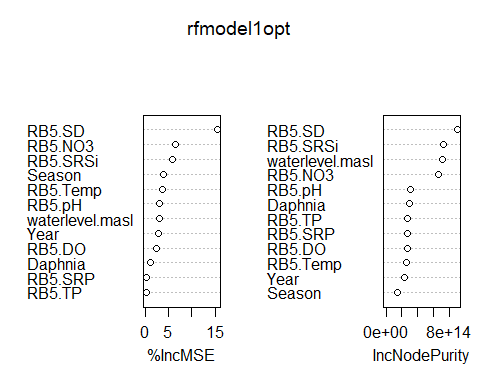
rfmodel1opt$importance

## %IncMSE IncNodePurity  
## Year 1.059677e+12 2.314507e+14  
## Season 1.489809e+12 1.435579e+14  
## waterlevel.masl 1.481133e+12 7.167117e+14  
## Daphnia 5.039160e+11 2.988729e+14  
## RB5.NO3 4.391759e+12 6.564095e+14  
## RB5.SRP 1.197435e+11 2.685283e+14  
## RB5.SRSi 3.446938e+12 7.289899e+14  
## RB5.TP 5.630557e+10 2.737771e+14  
## RB5.SD 1.534417e+13 9.065972e+14  
## RB5.Temp 1.632840e+12 2.557006e+14  
## RB5.pH 1.221846e+12 3.033555e+14  
## RB5.DO 1.156705e+12 2.667971e+14

predictions1opt <- predict(rfmodel1opt, testData1)  
rmse1opt <- sqrt(mean((predictions1opt - testData1$Total.Biovolume)^2))  
rsq1opt <- cor(predictions1opt, testData1$Total.Biovolume)^2  
cat("R²:", rsq1opt, "\nRMSE:", rmse1opt)

## R²: 0.04315796   
## RMSE: 14361269

varImpPlot(rfmodel1opt)



## Random Forest on Complete Cases Only

set.seed(40)  
noNAdata <- alldata %>% drop\_na()

### Train-Test Split

trainIndex2 <- createDataPartition(noNAdata$Total.Biovolume, p = 0.7, list = FALSE)  
trainData2 <- noNAdata[trainIndex2, ]  
testData2 <- noNAdata[-trainIndex2, ]

### Fit Model

rfmodel2 <- randomForest(Total.Biovolume ~ ., data = trainData2, ntree = 500, mtry = sqrt(ncol(trainData2)-1), importance = TRUE)  
print(rfmodel2)

##   
## Call:  
## randomForest(formula = Total.Biovolume ~ ., data = trainData2, ntree = 500, mtry = sqrt(ncol(trainData2) - 1), importance = TRUE)   
## Type of random forest: regression  
## Number of trees: 500  
## No. of variables tried at each split: 3  
##   
## Mean of squared residuals: 7.809481e+13  
## % Var explained: 31.54

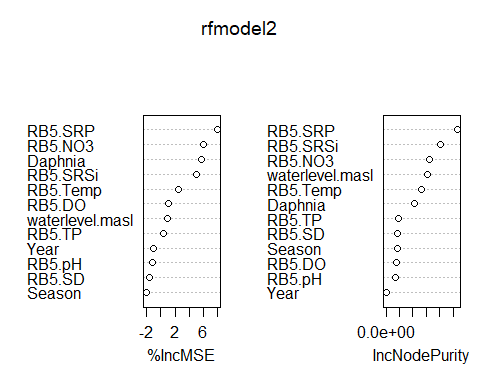
rfmodel2$importance

## %IncMSE IncNodePurity  
## Year -2.621286e+10 1.668237e+12  
## Season -2.325194e+12 4.117518e+13  
## waterlevel.masl 1.466149e+12 1.524454e+14  
## Daphnia 1.277275e+13 1.042920e+14  
## RB5.NO3 1.463132e+13 1.611270e+14  
## RB5.SRP 2.504472e+13 2.669766e+14  
## RB5.SRSi 1.406638e+13 2.040490e+14  
## RB5.TP 4.451574e+11 4.491608e+13  
## RB5.SD -1.951062e+12 4.412091e+13  
## RB5.Temp 4.492040e+12 1.309090e+14  
## RB5.pH -7.652727e+11 3.509055e+13  
## RB5.DO 1.349827e+12 3.827267e+13

predictions2 <- predict(rfmodel2, testData2)  
rmse2 <- sqrt(mean((predictions2 - testData2$Total.Biovolume)^2))  
rsq2 <- cor(predictions2, testData2$Total.Biovolume)^2  
cat("R²:", rsq2, "\nRMSE:", rmse2)

## R²: 0.4345314   
## RMSE: 37353204

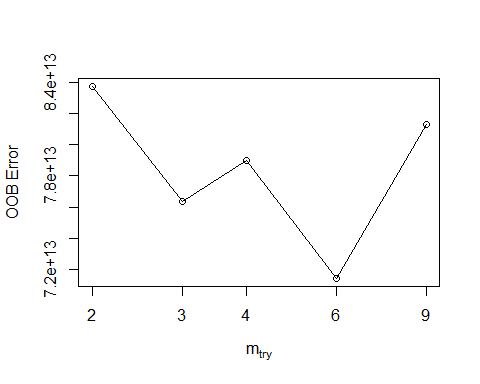
varImpPlot(rfmodel2)



### Tune Final Model

tuned\_model2 <- tuneRF(trainData2[, -which(names(trainData2) == "Total.Biovolume")], trainData2$Total.Biovolume, stepFactor = 1.5, improve = 0.01, ntreeTry = 500, trace = TRUE)

## mtry = 4 OOB error = 7.89978e+13   
## Searching left ...  
## mtry = 3 OOB error = 7.632914e+13   
## 0.03378144 0.01   
## mtry = 2 OOB error = 8.372919e+13   
## -0.09694922 0.01   
## Searching right ...  
## mtry = 6 OOB error = 7.14342e+13   
## 0.06412938 0.01   
## mtry = 9 OOB error = 8.129322e+13   
## -0.1380154 0.01



rfmodel2opt <- randomForest(Total.Biovolume ~ ., data = trainData2, ntree = 500, mtry = 4, importance = TRUE)  
print(rfmodel2opt)

##   
## Call:  
## randomForest(formula = Total.Biovolume ~ ., data = trainData2, ntree = 500, mtry = 4, importance = TRUE)   
## Type of random forest: regression  
## Number of trees: 500  
## No. of variables tried at each split: 4  
##   
## Mean of squared residuals: 7.18227e+13  
## % Var explained: 37.04

rfmodel2opt$importance

## %IncMSE IncNodePurity  
## Year -8.465859e+10 2.618577e+12  
## Season -3.653437e+12 6.577670e+13  
## waterlevel.masl 9.305259e+12 1.619684e+14  
## Daphnia 1.078503e+13 8.345826e+13  
## RB5.NO3 1.891470e+13 2.167845e+14  
## RB5.SRP 1.972447e+13 2.645145e+14  
## RB5.SRSi 1.365172e+13 2.014512e+14  
## RB5.TP -1.467488e+12 3.168691e+13  
## RB5.SD -9.548901e+11 2.833717e+13  
## RB5.Temp 5.520234e+12 9.475459e+13  
## RB5.pH -6.353382e+11 1.565802e+13  
## RB5.DO 9.002710e+11 3.550268e+13

predictions2opt <- predict(rfmodel2opt, testData2)  
rmse2opt <- sqrt(mean((predictions2opt - testData2$Total.Biovolume)^2))  
rsq2opt <- cor(predictions2opt, testData2$Total.Biovolume)^2  
cat("R²:", rsq2opt, "\nRMSE:", rmse2opt)

## R²: 0.4212883   
## RMSE: 37758025

### Variable Importance Plot (Renamed)

imp\_df <- as.data.frame(importance(rfmodel2opt, type = 1))  
imp\_df$vars <- rownames(imp\_df)  
imp\_df$vars <- recode(imp\_df$vars,  
 "RB5.SRP" = "SRP",  
 "RB5.NO3" = "Nitrate",  
 "RB5.SRSi" = "SRSi",  
 "RB5.Temp" = "Temperature",  
 "waterlevel.masl" = "Water Level",  
 "RB5.DO" = "DO",  
 "RB5.pH" = "pH",  
 "RB5.TP" = "TP",  
 "RB5.SD" = "Secchi Depth")  
imp\_df <- imp\_df[order(imp\_df$`%IncMSE`), ]  
dotchart(imp\_df$`%IncMSE`, labels = imp\_df$vars, xlim = c(-2, max(imp\_df$`%IncMSE`, na.rm = TRUE) \* 1.1),  
 pch = 1, xlab = "% Increase in MSE", ylab = "Predictors")

A graph with text overlay

AI-generated content may be incorrect.