peat\_depth\_model\_selection.R

user

Mon Aug 21 23:10:10 2017

#Peat Depths: model selection script  
#Aug 2017  
  
#packages ----  
library(raster)

## Loading required package: sp

library(rgdal)

## Warning: package 'rgdal' was built under R version 3.3.3

## rgdal: version: 1.2-8, (SVN revision 663)  
## Geospatial Data Abstraction Library extensions to R successfully loaded  
## Loaded GDAL runtime: GDAL 2.0.1, released 2015/09/15  
## Path to GDAL shared files: E:/Portable\_Stats/R/R-3.3.2/library/rgdal/gdal  
## Loaded PROJ.4 runtime: Rel. 4.9.2, 08 September 2015, [PJ\_VERSION: 492]  
## Path to PROJ.4 shared files: E:/Portable\_Stats/R/R-3.3.2/library/rgdal/proj  
## Linking to sp version: 1.2-5

library(randomForest)

## Warning: package 'randomForest' was built under R version 3.3.3

## randomForest 4.6-12

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

library(caTools)

## Warning: package 'caTools' was built under R version 3.3.3

library(ggplot2)

##   
## Attaching package: 'ggplot2'

## The following object is masked from 'package:randomForest':  
##   
## margin

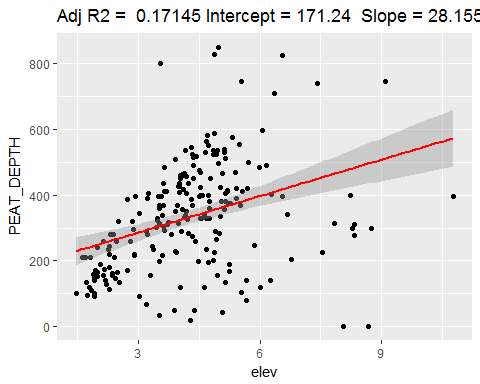
#helper functions----  
  
#plot output of regression model using ggplot  
ggplotRegression <- function (fit) {  
   
 require(ggplot2)  
   
 ggplot(fit$model, aes\_string(x = names(fit$model)[2], y = names(fit$model)[1])) +   
 geom\_point() +  
 stat\_smooth(method = "lm", col = "red") +  
 labs(title = paste("Adj R2 = ",signif(summary(fit)$adj.r.squared, 5),  
 "Intercept =",signif(fit$coef[[1]],5 ),  
 " Slope =",signif(fit$coef[[2]], 5),  
 " P =",signif(summary(fit)$coef[2,4], 5)))  
}  
  
  
  
  
  
#import data ----  
input.data <- readRDS("E://National\_Peat\_Depth\_map/data/input.data.rds")   
  
  
# Split into training and test set ----  
  
set.seed(123)  
split = sample.split(input.data$PEAT\_DEPTH, SplitRatio = 2/3)  
training\_set = subset(input.data, split == TRUE)  
test\_set = subset(input.data, split == FALSE)  
  
# Feature Scaling ----   
#not sure needed  
  
# training\_set = scale(training\_set)  
# test\_set = scale(test\_set)  
  
  
  
#run model ----  
 #linear model  
M.lm <- lm(PEAT\_DEPTH ~ elev + aspect + slope + aap, data=training\_set)  
 summary(M.lm)

##   
## Call:  
## lm(formula = PEAT\_DEPTH ~ elev + aspect + slope + aap, data = training\_set)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -538.86 -97.18 0.48 89.49 501.37   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.712e+02 3.200e+01 5.351 2.26e-07 \*\*\*  
## elev 2.816e+01 5.999e+01 0.469 0.639292   
## aspect 5.151e-03 1.010e-01 0.051 0.959373   
## slope -3.892e+03 9.886e+02 -3.937 0.000112 \*\*\*  
## aap 1.486e+01 5.988e+01 0.248 0.804265   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 149.4 on 212 degrees of freedom  
## Multiple R-squared: 0.1868, Adjusted R-squared: 0.1715   
## F-statistic: 12.17 on 4 and 212 DF, p-value: 6.299e-09

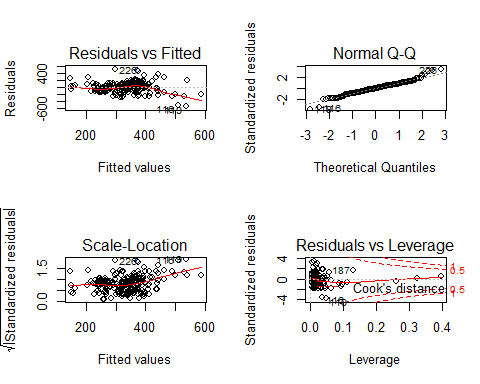
M.lm

##   
## Call:  
## lm(formula = PEAT\_DEPTH ~ elev + aspect + slope + aap, data = training\_set)  
##   
## Coefficients:  
## (Intercept) elev aspect slope aap   
## 1.712e+02 2.816e+01 5.151e-03 -3.892e+03 1.486e+01

ggplotRegression(M.lm)



par(mfrow = c(2,2))  
 plot(M.lm)



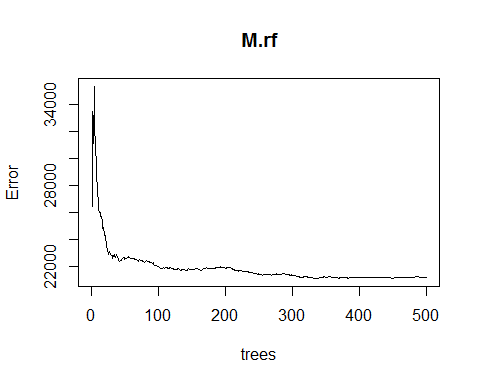
par(mfrow = c(1,1))  
  
   
 #random forest  
M.rf <- randomForest(PEAT\_DEPTH ~ elev + aspect + slope + aap, data=training\_set)   
 summary(M.rf)

## Length Class Mode   
## call 3 -none- call   
## type 1 -none- character  
## predicted 217 -none- numeric   
## mse 500 -none- numeric   
## rsq 500 -none- numeric   
## oob.times 217 -none- numeric   
## importance 4 -none- numeric   
## importanceSD 0 -none- NULL   
## localImportance 0 -none- NULL   
## proximity 0 -none- NULL   
## ntree 1 -none- numeric   
## mtry 1 -none- numeric   
## forest 11 -none- list   
## coefs 0 -none- NULL   
## y 217 -none- numeric   
## test 0 -none- NULL   
## inbag 0 -none- NULL   
## terms 3 terms call

M.rf

##   
## Call:  
## randomForest(formula = PEAT\_DEPTH ~ elev + aspect + slope + aap, data = training\_set)   
## Type of random forest: regression  
## Number of trees: 500  
## No. of variables tried at each split: 1  
##   
## Mean of squared residuals: 21203.47  
## % Var explained: 20.91

plot(M.rf)



n.variables <- ncol(training\_set)-1   
oob.err=double(n.variables)  
test.err=double(n.variables)  
   
#mtry is no of Variables randomly chosen at each split  
 for(mtry in 1:n.variables)   
 {  
 rf=randomForest(PEAT\_DEPTH ~ . , data = training\_set, mtry=mtry,ntree=400)   
 oob.err[mtry] = rf$mse[400] #Error of all Trees fitted  
   
 pred<-predict(rf, test\_set) #Predictions on Test Set for each Tree  
 test.err[mtry]= with(test\_set, mean( (PEAT\_DEPTH - pred)^2)) #Mean Squared Test Error  
   
 cat(mtry," ") #printing the output to the console  
   
 }

## 1 2 3 4

#test error  
  
test.err

## [1] 20414.51 21512.51 21701.44 22256.28

#Out of Bag Error Estimation  
oob.err

## [1] 20940.45 21996.13 21911.43 22011.49

#Plotting both Test Error and Out of Bag Error  
  
matplot(  
 1:mtry ,  
 cbind(oob.err, test.err),  
 pch = 19 ,  
 col = c("red", "blue"),  
 type = "b",  
 ylab = "Mean Squared Error",  
 xlab = "Number of Predictors Considered at each Split"  
)  
  
legend(  
 "bottomright",  
 legend = c("Out of Bag Error", "Test Error"),  
 pch = 19,  
 col = c("red", "blue")  
)

