Final Project

Hussam Taj

4/11/2020

## -- Attaching packages --------

## v ggplot2 3.2.1 v purrr 0.3.3  
## v tibble 2.1.3 v dplyr 0.8.3  
## v tidyr 1.0.0 v stringr 1.4.0  
## v readr 1.3.1 v forcats 0.4.0

## -- Conflicts -----------------  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

## Warning: package 'zoo' was built under R version 3.6.3

##   
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':  
##   
## as.Date, as.Date.numeric

##   
## Attaching package: 'lubridate'

## The following object is masked from 'package:base':  
##   
## date

## Registered S3 method overwritten by 'GGally':  
## method from   
## +.gg ggplot2

##   
## Attaching package: 'GGally'

## The following object is masked from 'package:dplyr':  
##   
## nasa

## Warning: package 'gam' was built under R version 3.6.3

## Loading required package: splines

## Loading required package: foreach

##   
## Attaching package: 'foreach'

## The following objects are masked from 'package:purrr':  
##   
## accumulate, when

## Loaded gam 1.16.1

## randomForest 4.6-14

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

##   
## Attaching package: 'randomForest'

## The following object is masked from 'package:dplyr':  
##   
## combine

## The following object is masked from 'package:ggplot2':  
##   
## margin

Reading in the data

YC <- read\_csv("data/Yeild Curve.csv", col\_names = c("Date", "InterestRate"))

## Parsed with column specification:  
## cols(  
## Date = col\_character(),  
## InterestRate = col\_character()  
## )

UnE <- read\_csv("data/Unemployment.csv")

## Parsed with column specification:  
## cols(  
## Year = col\_double(),  
## Jan = col\_double(),  
## Feb = col\_double(),  
## Mar = col\_double(),  
## Apr = col\_double(),  
## May = col\_double(),  
## Jun = col\_double(),  
## Jul = col\_double(),  
## Aug = col\_double(),  
## Sep = col\_double(),  
## Oct = col\_double(),  
## Nov = col\_double(),  
## Dec = col\_double()  
## )

GDP <- read\_csv("data/GDPPOT.csv", col\_names = c("Date", "GDP"))

## Parsed with column specification:  
## cols(  
## Date = col\_character(),  
## GDP = col\_character()  
## )

Formatting the Yield Curve Dataset

YC <- YC[-1,]  
YC$Date <- format(strptime(YC$Date, format = "%m/%d/%Y"), "%Y-%m-%d")  
YC$Date <- as.Date(YC$Date)  
YC <- as.data.frame(YC)

Formatting GDP Output Data

GDP <- GDP[-1,]  
GDP$Date <- format(strptime(GDP$Date, format = "%m/%d/%Y"), "%Y-%m-%d")  
GDP$Date <- as.Date(GDP$Date)  
monthly <- seq(GDP$Date[1], tail(GDP$Date,1), by="month")  
GDP <- data.frame(Date = monthly, GDP = spline(GDP, method="fmm", xout = monthly)$y)

Formatiing Unemployment Dataset

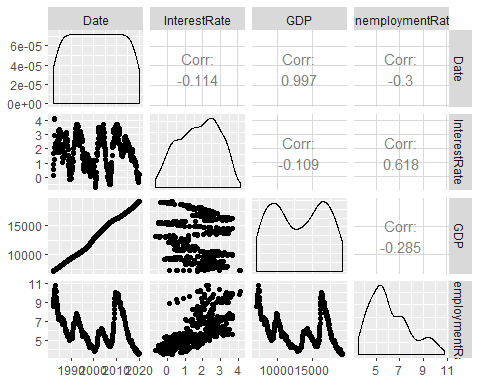
UnE <- UnE %>%   
 pivot\_longer(c("Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug", "Sep", "Oct", "Nov", "Dec"), names\_to = "Months", values\_to = "UnemploymentRate")  
UnE <- as.data.frame(UnE)  
UnE <- UnE[-c(458:468), ]  
UnE$Months <- match(UnE$Months,month.abb)  
i1 <- grepl("^[0-9]$", UnE$Months)  
UnE$Months[i1] <- paste0("0", UnE$Months[i1])  
UnE <- unite(UnE, "Date", c(Year, Months), sep = "-", remove = TRUE, na.rm = FALSE)  
UnE$Date <- YC$Date

Megrging the datasets

Data <- merge(YC, GDP, by = "Date")  
Data <- merge(Data, UnE, by = "Date")

Matrix Plot

Data$InterestRate <- as.numeric(Data$InterestRate)  
ggpairs(Data)



Cross-validation

set.seed(39)  
Data <- Data %>% mutate(IRIV = ifelse(InterestRate >= 0, 0,1))  
Data$IRIV <- as.factor(Data$IRIV)  
Data <- mutate(Data, id = row\_number())  
train <- sample\_frac(Data, .75)  
test <- anti\_join(Data, train, by = "id")  
train <- train[c(-10)]  
test <- test[c(-10)]  
Data <- Data[ , !(names(Data) %in% c("id"))]  
train <- train[ , !(names(train) %in% c("id"))]  
test <- test[ , !(names(test) %in% c("id"))]

Fitting Generalized additive model

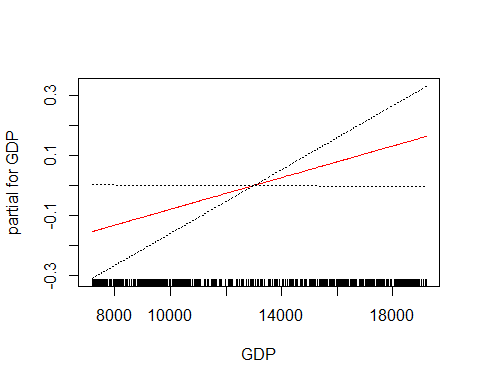
gam <- gam(InterestRate ~ GDP + UnemploymentRate, data = train)

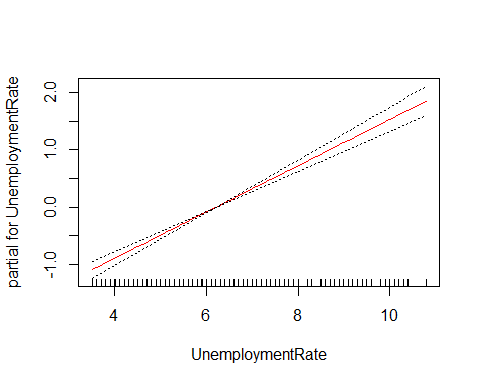
## Warning in model.matrix.default(mt, mf, contrasts): non-list contrasts argument  
## ignored

summary(gam)

##   
## Call: gam(formula = InterestRate ~ GDP + UnemploymentRate, data = train)  
## Deviance Residuals:  
## Min 1Q Median 3Q Max   
## -2.55386 -0.68649 0.03797 0.67818 2.11110   
##   
## (Dispersion Parameter for gaussian family taken to be 0.7721)  
##   
## Null Deviance: 428.0004 on 342 degrees of freedom  
## Residual Deviance: 262.5038 on 340 degrees of freedom  
## AIC: 889.6514   
##   
## Number of Local Scoring Iterations: 2   
##   
## Anova for Parametric Effects  
## Df Sum Sq Mean Sq F value Pr(>F)   
## GDP 1 4.708 4.708 6.0978 0.01403 \*   
## UnemploymentRate 1 160.789 160.789 208.2566 < 2e-16 \*\*\*  
## Residuals 340 262.504 0.772   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

plot.Gam(gam, se = TRUE, col = "red")





pred <- predict(gam, newdata = test)  
mean((pred - test$InterestRate)^2)

## [1] 0.7380704

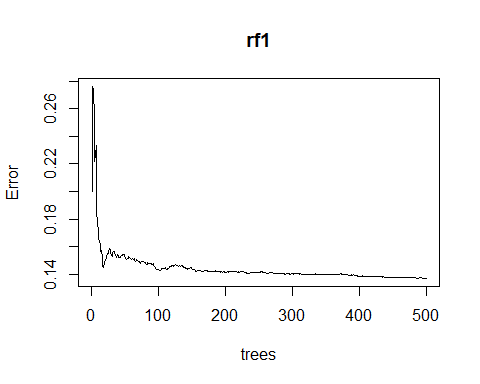
Indicates that this model leads to test predictions that are within around 0.8591102 of the true median.

Fitting Random Forest regression tree method

rf1 <- randomForest(InterestRate ~ GDP + UnemploymentRate,   
 data = train,  
 importance = TRUE)  
pred\_rf1 <- predict(rf1, newdata = test)  
summary(rf1)

## Length Class Mode   
## call 4 -none- call   
## type 1 -none- character  
## predicted 343 -none- numeric   
## mse 500 -none- numeric   
## rsq 500 -none- numeric   
## oob.times 343 -none- numeric   
## importance 4 -none- numeric   
## importanceSD 2 -none- numeric   
## 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 343 -none- numeric   
## test 0 -none- NULL   
## inbag 0 -none- NULL   
## terms 3 terms call

plot(rf1)



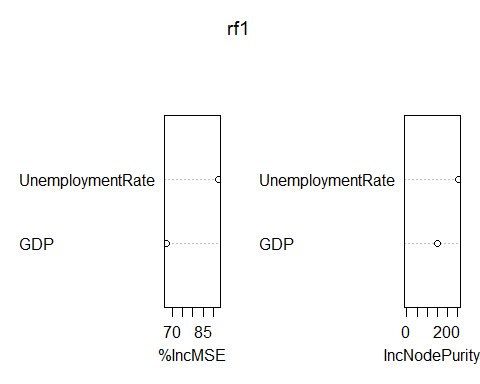
mean((pred\_rf1 - test$InterestRate)^2)

## [1] 0.1212357

importance(rf1)

## %IncMSE IncNodePurity  
## GDP 67.51874 150.4242  
## UnemploymentRate 92.47268 254.0372

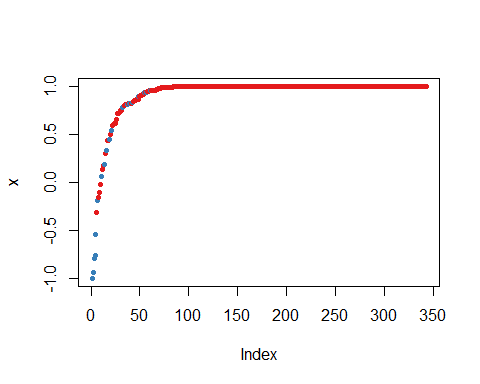
varImpPlot(rf1)

 Indicates that this model leads to test predictions that are within around 0.3474779 of the true median.

Fitting Random Forest Classification Tree

rf <- randomForest(IRIV ~ GDP + UnemploymentRate,   
 data = train,  
 importance = TRUE)  
pred\_rf <- predict(rf, newdata = test)  
tb <- table(pred\_rf, test$IRIV)  
mcr <- mean(pred\_rf != test$IRIV)  
plot(margin(rf, test$IRIV))

## Warning in RColorBrewer::brewer.pal(nlevs, "Set1"): minimal value for n is 3, returning requested palette with 3 different levels



accuracy <- (sum(diag(tb)))/sum(tb)

The model has a high overall accuracy that is 0.9736842.