report_2

Kevork Sulahian

August 7, 2019

```
library(readxl)
library(MASS)
library(caret)
library(rpart)
library(rpart.plot)
library(randomForest)
library(glmnet)
library(xgboost)
```

Cleaning the Data

```
df <- read_xls('economy.xls', sheet='2011-2019 NACE 2')
df <- df[c(4,6,11,30,34),]

df <- df[,-c(1,3)]

my_months <- c('JAN','FEB','MARCH','APRIL','MAY','JUNE','JULY','AUG','SEP', 'OCT','NOV','DEC')
my_years <- c(2011:2019)
df_months = c()
j = 0
for (i in 2:length(df)) {

  if((i-2)%%12==0) {
    j = j + 1
    df_months = c(df_months, paste0(my_years[j], '_', my_months))
  }
}


df_months = head(df_months,-6)

df <- as.data.frame(t(df))
cols <- df[1,]
cols = as.character(unlist(cols))
df <- df[-c(1),]
colnames(df) <- as.character(cols)
rownames(df) <- df_months
df[] <- sapply(df[],function(x) as.numeric(as.character(x)))

df$`Total industry` = df$`Total industry` * 1000000
df$`mining and quarrying` = df$`mining and quarrying` * 1000000
df$manufacturing = df$manufacturing * 1000000
df$`Electricity, gas, steam and air conditioning supply` = df$`Electricity, gas, steam and air condition
df$`Water supply, sewerage, waste management and remediation activites` = df$`Water supply, sewerage, wa
```

```
head(df)
```

```
##            Total industry mining and quarrying manufacturing
## 2011_JAN       63316800000          13164400000   32688300000
## 2011_FEB       68950700000          12337100000   38793800000
## 2011_MARCH     76904600000          13334900000   48051400000
## 2011_APRIL     75019400000          15377500000   46263600000
## 2011_MAY       82815300000          14881400000   53648900000
## 2011_JUNE      87199000000          14519800000   58033900000
##            Electricity, gas, steam and air conditioning supply
## 2011_JAN                                            16059900000
## 2011_FEB                                            16327300000
## 2011_MARCH                                          14049200000
## 2011_APRIL                                          11845900000
## 2011_MAY                                            12772400000
## 2011_JUNE                                           13069300000
##            Water supply, sewerage, waste management and remediation activites
## 2011_JAN                                                          1404200000
## 2011_FEB                                                          1492500000
## 2011_MARCH                                                        1469100000
## 2011_APRIL                                                        1532400000
## 2011_MAY                                                          1512600000
## 2011_JUNE                                                         1575900000
```

Spliting the data into train.*85 and test*.15

```
index <- sample(1:nrow(df),round(0.85*nrow(df)))

train <- df[index,]
test <- df[-index,]

n <- names(train)
```

```
df$`Total industry`
```

```
##   [1]  63316800000  68950700000  76904600000  75019400000  82815300000
##   [6]  87199000000  78206100000  82340200000  91564700000  89221300000
##  [11]  93969200000 102601000000  74121200000  80296600000  85180900000
##  [16]  83637700000  95768500000  94252400000  95346800000  93504300000
##  [21]  94322000000  98881600000 107133800000 118030000000  95473900000
##  [26]  93167800000  99722300000  85554000000  90153900000 103338200000
##  [31]  95634300000 101106900000 110464000000 118994600000 116280000000
##  [36] 130688800000  87916800000  87808200000  98631100000  95404300000
##  [41] 101988000000 107586700000 114318300000 105561000000 119757000000
##  [46] 121128900000 120359200000 127696200000  88660200000  99366900000
##  [51] 106852900000 104876300000 105518700000 114280400000 110330100000
##  [56] 113534000000 118789900000 117244200000 117806100000 129815800000
##  [61]  94430400000  99675100000 111202400000 115680100000 112485600000
##  [66] 126229400000 117745900000 121268100000 129288300000 124218000000
##  [71] 133420400000 143400300000 110043500000 124444200000 139515900000
##  [76] 132504200000 145671500000 140072100000 144456900000 147917100000
```

```
##  [81] 161314700000 179763200000 186876600000 223798900000 130511600000
##  [86] 131634500000 153538600000 133147900000 143461700000 152379300000
##  [91] 155670300000 154683600000 163500600000 194080500000 189553000000
##  [96] 233100000000 124934500000 136840600000 160751900000 151122400000
## [101] 159462400000 167388400000
```

```
f <- as.formula(`Total industry`~ `mining and quarrying` + manufacturing + `Electricity, gas, steam and
```

Prediction Models

Linear Regression

```
attach(df)
fit <- lm(`Total industry`~., train)

pred1<-predict(fit, newdata = test)

RMSE1<-RMSE(test$`Total industry`, pred1)

MAE1 <- MAE(test$`Total industry`, pred1)
```

```
## [1] "RMSE for Linear Regression: 3870.66038529419"
```

```
## [1] "MAE for Linear Regression: 2986.47204996745"
```

K- Nearest Neighbors

```
ctrl <- trainControl(method = "cv", number = 5)

knn_c <- train(f, data = train, method = "knn",
               trControl = ctrl, preProcess = c("center", "scale"), tuneLength = 5)
```

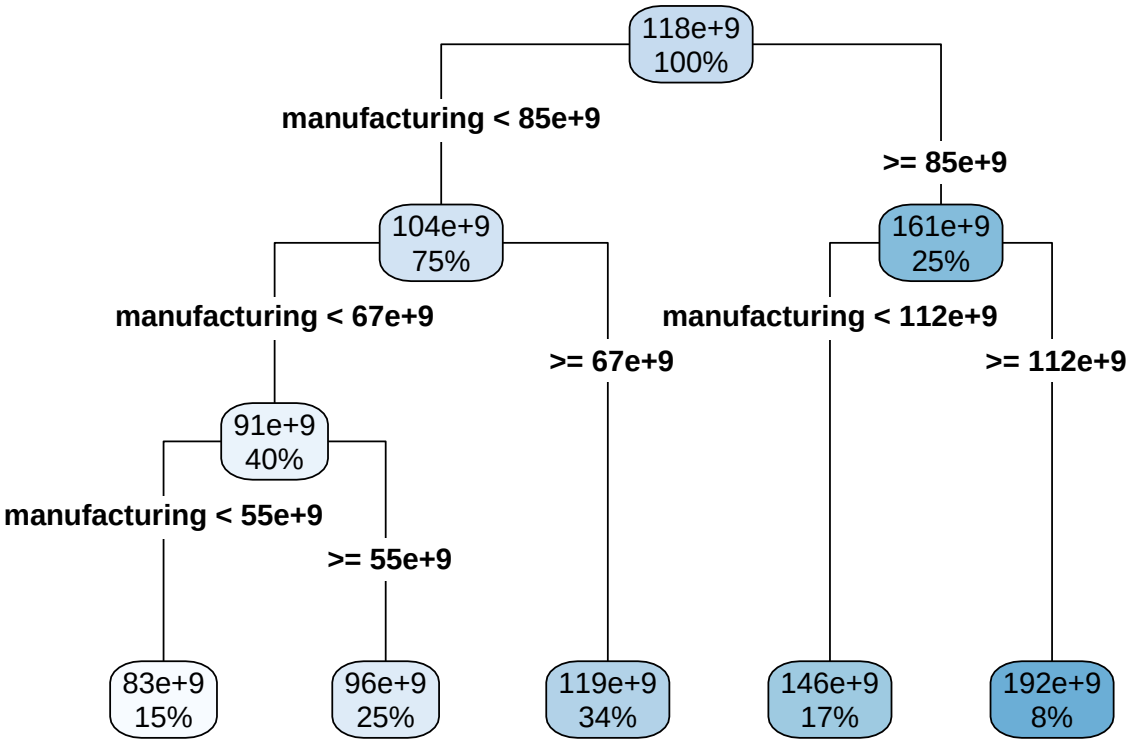
```
model_predict_test = predict(knn_c, newdata = test)
# which.min(c(sqrt(mean(abs(model_predict_test - test$main)^2)),sqrt(mean((test$main- predict(fit, test

RMSE_KNN <- RMSE(test$`Total industry`, model_predict_test)

MAE_KNN <- MAE(test$`Total industry`, model_predict_test)
```

```
## [1] "RMSE for KNN: 9477854538.78109"
```

```
## [1] "MAE for KNN: 6982962666.66667"
```

Tree

```
my_model <- rpart(f,subset = index, data= df)
```




```
predictions<-predict(my_model,newdata=test)

RMSE_Tree <- RMSE(predictions, test$`Total industry`)

MAE_Tree <- MAE(predictions, test$`Total industry`)
```

```
## [1] "RMSE for Tree: 8971005007.14326"
```

```
## [1] "MAE for Tree: 7445594176.26818"
```

forrest

```
# set.seed(1)
# bag.black <- randomForest(f,data=df, subset=index,importance =TRUE)
#
# prediction_forest = predict(bag.black, newdata=test[1,])
```

```
# getTree(bag.black,1,labelVar=TRUE)
```

```
# MAE_forest <- MAE(prediction_forest, test$main)
# RMSE_forest <- RMSE(prediction_forest, test$main)
```

Ridge Regression

```
x = model.matrix(f, data = df)[,c(-1)]
y = df$`Total industry`

grid = 10^seq(10,-2, length = 100)

ridge.mod = glmnet(x[index,], y[index], alpha = 0, lambda = grid,
                   thresh = 1e-12)
ridge.pred =predict(ridge.mod, s = 4, newx = x[-index,])

RMSE_ridge <- RMSE(ridge.pred, y[-index])

MAE_ridge <- MAE(ridge.pred, y[-index])
```

```
## [1] "RMSE for Ridge Regression: 3870.8311910365"
```

```
## [1] "MAE for Ridge Regression: 2985.62620747884"
```

Lasso Regression

```
set.seed(1)

lasso.mod=glmnet(x[index,],y[index],alpha=1,lambda=grid)
cv.out=cv.glmnet(x[index,],y[index],alpha=1)

bestlam=cv.out$lambda.min

lasso.pred=predict(lasso.mod,s=bestlam,newx=x[-index,])
RMSE_lasso <- RMSE(lasso.pred, y[-index])

MAE_lasso <- MAE(lasso.pred, y[-index])
```

```
## [1] "RMSE for Lasso Regression: 993034396.912752"
```

```
## [1] "MAE for Lasso Regression: 796001104.886188"
```

Extreme Gradient Boosting

```
set.seed(1)
dtrain2 <- xgb.DMatrix(data = x[index,], label = y[index])
dtest2 <- xgb.DMatrix(data = x[-index,], label = y[-index])
watchlist <- list(train= dtrain2, test= dtest2)
```

```
set.seed(1)
bst2 <- xgb.train(data= dtrain2, max.depth=10, eta=0.09, nrounds=120,watchlist=watchlist,
                  base_score = 0.1)

# xgb_test <- predict(bst2, data.matrix(test[,-c(1)]))

# RMSE_xgboost <- RMSE(test$main, xgb_test)
#
# MAE_xgboost <- MAE(test$main,xgb_test)
```

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

```
## [1] "the Algorithm with the least error is: 3870.66038529419"
```

```
options(scipen = 999)
## Ridge Regression
mean_error(difference(ridge.pred,test$`Total industry`))
```

```
## [1] 0.00000003038319
```

```
# Linear Regression
mean_error(difference(pred1,test$`Total industry`))
```

```
## [1] 0.00000003038359
```