СЛУЧАЙНЫЙ ЛЕС\_\_ДЕРЕВЬЯ РЕГРЕССИИ (03.12.21)

stat\_1<-read.csv2("Initial\_Data\_3.csv")

View(stat\_1)

library(randomForest)

library(dplyr)

# Формируются обучающая (train) и тестовая выборки (test)

train <- stat\_1 %>% sample\_frac(.70)

test <- stat\_1 %>% anti\_join(train, by = 'N')

# В полученных массивах исключаем столбец N (чтобы он не участвоал в группировках)

train <- train %>% select(-N)

View (train)

test <- test %>% select(-N)

View(test)

## вывод наименований переменных

names(train)

names(test)

## Построение случайного леса деревьев регрессии для пепеменной (inv\_c\_19)

rf <- randomForest(inv\_c\_19 ~ .,data=train, importance=TRUE, proximity=TRUE)

plot(rf)

### Построение случайного леса деревьев регрессии для пепеменной (inv\_g\_19)

rf\_1 <- randomForest(inv\_g\_19 ~ .,data=train, importance=TRUE, proximity=TRUE)

plot(rf\_1)

## Результат оценивания предсказанных значений. Вывод на экран исходных и предсказанных значений зависимой переменной (inv\_c\_19)

Result <- data.frame(inv\_c\_19=test$inv\_c\_19, Predict=predict(rf, test))

View(Result)

### Результат оценивания предсказанных значений. Вывод на экран исходных

### и предсказанных значений зависимой переменной (inv\_g\_19)

Result\_1 <- data.frame(inv\_g\_19=test$inv\_g\_19, Predict=predict(rf\_1, test))

View(Result\_1)

## Сохранить результат в Excel по переменной inv\_c\_19

write.csv2(Result, 'Predict\_inv\_c\_19.csv')

### Сохранить результат в Excel по переменной inv\_g\_19

write.csv2(Result\_1, 'Predict\_inv\_g\_19.csv')

# Визуализация оценок importance по переменным-предикторам

## по переменной inv\_c\_19

importance(rf)

## округлить до 3-х знаков после запятой

round(importance(rf), 3)

# Визуализация оценок importance по переменным-предикторам

## по переменной inv\_g\_19

importance(rf\_1)

## округлить до 3-х знаков после запятой

round(importance(rf\_1), 3)

# Построение графика оценок значимости переменных-предикторов:

## мера точности предсказания (%IncMSE) и мера загрязнения расщепления (IncNodePurity)

varImpPlot(rf)

varImpPlot(rf\_1)

# Для визуализации деревьев решений требуется

## создать список деревьев, упорядоченных по числу узлов

tree\_number\_nodes <- data.frame(tree\_id=1:length(rf$forest$ndbigtree),

number\_nodes=rf$forest$ndbigtree) %>% arrange(number\_nodes)

head(tree\_number\_nodes)

tail(tree\_number\_nodes)

## Подключается специальная функция для визуализации деревьев по ссылке:

# https://shiring.github.io/machine\_learning/2017/03/16/rf\_plot\_ggraph

source('tree\_func.R')

tree\_func(final\_model = rf, tree\_number\_nodes[1, 'tree\_id'])

tree\_func(final\_model = rf, tree\_number\_nodes[50, 'tree\_id'])

tree\_func(final\_model = rf, tree\_number\_nodes[100, 'tree\_id'])

tree\_func(final\_model = rf, tree\_number\_nodes[200, 'tree\_id'])

tree\_func(final\_model = rf, tree\_number\_nodes[500, 'tree\_id'])

tree\_func(final\_model = rf\_1, tree\_number\_nodes[1, 'tree\_id'])

tree\_func(final\_model = rf\_1, tree\_number\_nodes[50, 'tree\_id'])

tree\_func(final\_model = rf\_1, tree\_number\_nodes[100, 'tree\_id'])

tree\_func(final\_model = rf\_1, tree\_number\_nodes[200, 'tree\_id'])

tree\_func(final\_model = rf\_1, tree\_number\_nodes[500, 'tree\_id'])