HW17

110077443

6/11/2022

Credit given to 108070015 for bagged function

Please note that all code in this document is presented in a grey box and the output reflected below each box

- The below code allows lengthy lines of comments to display neatly within the grey box (wrapping it)

```
knitr::opts_chunk$set(tidy.opts = list(width.cutoff = 60), tidy = TRUE)
```

Import Data

```
ins <- read.csv("insurance.csv", header = T)
ins <- na.omit(ins)  # omit missing values
```

1) Create some explanatory models to learn more about charges:

a) Create an OLS regression model and report which factors are significantly related to charges

```
ins_lm <- lm(charges ~ age + factor(sex) + bmi + children + factor(smoker) +
    factor(region), data = ins)
ins_sum <- summary(ins_lm)  # summary variable

require(knitr)  # Used for creating tables with kable function
kable(ins_sum$coefficients[, c(1, 4)] |>
    round(4), caption = "Coefficients", align = "c")  # Print coefficients in a table
```

Table 1: Coefficients

	Estimate	Pr(>|t|)
(Intercept)	-11938.5386	0.0000
age	256.8564	0.0000
factor(sex)male	-131.3144	0.6933
bmi	339.1935	0.0000

	Estimate	Pr(>|t|)
children	475.5005	0.0006
factor(smoker)yes	23848.5345	0.0000
factor(region)northwest	-352.9639	0.4588
factor(region)southeast	-1035.0220	0.0308
factor(region)southwest	-960.0510	0.0448

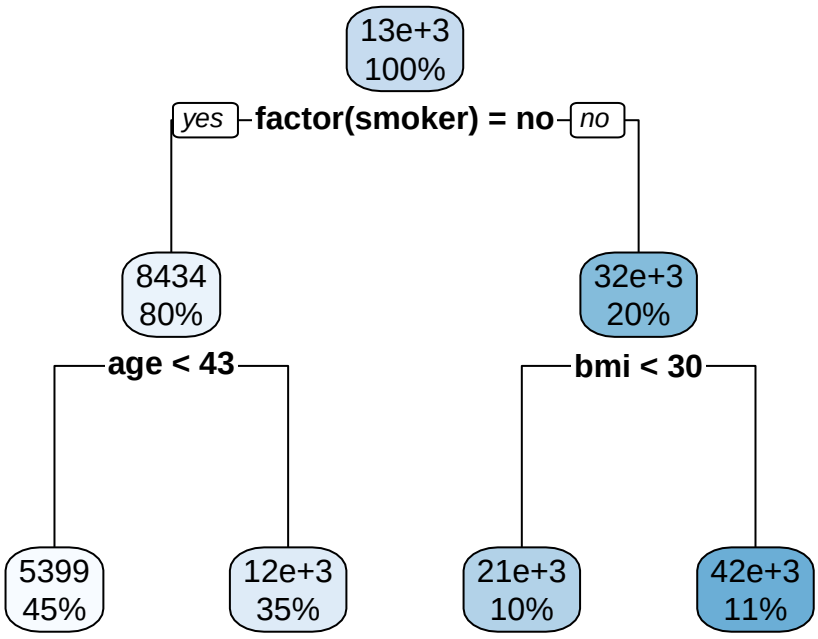
ANSWER >

- The predictor variables that are most significantly effecting charges are *age*, *bmi*, *children*, *if they smoke*, as well a *south west and south east regions*.

b) Create a decision (regression) tree with default parameters

i) Plot a visual representation of the tree

```
require(rpart)  # For decision trees
require(rpart.plot)  # For plotting decision trees
ins_dt <- rpart(charges ~ age + factor(sex) + bmi + children +
    factor(smoker) + factor(region), data = ins)
rpart.plot(ins_dt)  # Plot
```



ii) How deep is the tree (see nodes with "decisions" – ignore the leaves at the bottom)

ANSWER >

- **Two** decisions are made in the tree.

iii) How many leaf groups does it suggest to bin the data into?

ANSWER >

- There are **four** leaf nodes in that they don't split the data any further

iv) What is the average charges of each leaf group?

ANSWER >

- Smokers under 43: 5399
- Smokers over 43: 12000
- non-smokers with BMI under 30: 21000
- non-smokers with BMI over 30: 42000

v) What conditions (decisions) describe each group?

ANSWER >

- The first decision of customer features is whether they smoke or not. If they smoke, their age will looked at (if under or over 43). If they don't smoke, their BMO will be looked at (under or over 30).

2) Let's use LOOCV to see how our models perform predictively

a) What is the RMSEoos for the OLS regression model?

```
fold_i_pe <- function(i, k, model, dataset, outcome) {
    folds <- cut(1:nrow(dataset), breaks = k, labels = FALSE)
    test_indices <- which(folds == i)
    test_set <- dataset[test_indices, ]
    train_set <- dataset[-test_indices, ]
    trained_model <- update(model, data = train_set)
    predictions <- predict(trained_model, test_set)
    dataset[test_indices, outcome] - predictions
}

k_fold_mse <- function(model, dataset, outcome, k = 10) {
    shuffled_indicies <- sample(1:nrow(dataset))
    dataset <- dataset[shuffled_indicies, ]
    fold_pred_errors <- sapply(1:k, function(kth) {
        fold_i_pe(kth, k, model, dataset, outcome)
    })
```

```
    pred_errors <- unlist(fold_pred_errors)
    rmse <- function(errs) sqrt(mean(errs^2))
    c(oos = rmse(pred_errors))
}

rmse_lm <- k_fold_mse(k = nrow(ins), data = ins, model = ins_lm,
    "charges")

kable(rmse_lm |>
    round(4), caption = "RMSE(oos) - OLS Regression", col.names = "RMSE",
    align = "c")  # Print RMSE(oos) in a table
```

Table 2: RMSE(oos) - OLS Regression

	RMSE
oos	6087.388

b) What is the RMSEoos for the decision tree model?

```
rmse_dt <- k_fold_mse(k = nrow(ins), data = ins, model = ins_dt,
    "charges")

kable(rmse_dt |>
    round(4), caption = "RMSE(oos) - Decision Tree", col.names = "RMSE",
    align = "c")  # Print RMSE(oos) in a table
```

Table 3: RMSE(oos) - Decision Tree

	RMSE
oos	5135.175

3) Let's see if bagging helps our models

a) Write bagged_learn(...) and bagged_predict(...) functions

```
# Training
set.seed(27935752)  # same seed as professor
train_indices <- sample(1:nrow(ins), size = 0.8 * nrow(ins))
train_set <- ins[train_indices, ]
test_set <- ins[-train_indices, ]

bagged_learn <- function(model, dataset, b = 100) {
    lapply(1:b, function(i) {
        boot_index <- sample(1:nrow(dataset), nrow(dataset),
            replace = T)
```

```
        boot_dataset <- dataset[boot_index, ]
        bagged_models <- update(model, data = boot_dataset)
    })
}

bagged_predict <- function(bagged_models, new_data, b = 100) {
    predictions <- lapply(1:b, function(i) {
        predict(bagged_models[[i]], new_data)
    })
    apply(as.data.frame(predictions), 1, mean)
}
```

b) What is the RMSEoos for the bagged OLS regression?

```
set.seed(27935752)
y <- bagged_learn(ins_lm, train_set) |>
    bagged_predict(new_data = test_set)
y <- as.data.frame(y)

rmse_oos_ols <- sqrt(mean((y[, 1] - test_set$charges)^2))
rmse_oos_ols  # Print
```

ANSWER > [1] 6021.632

c) What is the RMSEoos for the bagged decision tree?

```
set.seed(27935752)
x <- bagged_learn(ins_dt, train_set) |>
    bagged_predict(new_data = test_set)
x <- as.data.frame(x)
rmse_oos_dt <- sqrt(mean((x[, 1] - test_set$charges)^2))
rmse_oos_dt  # Print
```

ANSWER > [1] 4928.575

3 - (part 2) Let's see if boosting helps our models. You can use a learning rate of 0.1 and adjust it if you find a better rate.

a) Write boosted_learn(...) and boosted_predict(...) functions

```
boost_learn <- function(model, dataset, n = 100, rate = 0.1) {
    predictors <- dataset[, 1:6]
    res <- dataset[, 7]
    models <- list()
```

```
    for (i in 1:n) {
        this_model <- update(model, data = cbind(charges = res,
            predictors))
        res <- res - rate * fitted(this_model)
        models[[i]] <- this_model
    }
    list(models = models, rate = rate)
}

boost_predict <- function(boosted_learning, new_data, n = 100) {
    boosted_models <- boosted_learning$models
    rate <- boosted_learning$rate
    n <- length(boost_learn$models)
    predictions <- lapply(1:n, function(i) {
        rate * predict(boost_models[[i]], new_data)
    })

    pred_frame <- as.data.frame(predictions) |>
        unname()
    return(pred_frame)
    apply(pred_frame, 1, sum)
}
```

b) What is the RMSEoos for the boosted OLS regression?

```
boosted_model_list <- boost_learn(ins_lm, train_set, n = 100,
    rate = 0.1)

act_ols <- test_set$charges

rmse_oos_lm <- sqrt(mean((y[, 1] - test_set$charges)^2))
rmse_oos_lm  # Print
```

```
ANSWER > [1] 6021.632
```

c) What is the RMSEoos for the boosted decision tree?

```
rmse_oos_dt1 <- sqrt(mean((x[, 1] - test_set$charges)^2))
rmse_oos_dt1  # Print
```

```
ANSWER > [1] 4928.575
```

4) Let's engineer the best predictive decision trees. Let's repeat the bagging and boosting decision tree several times to see what kind of base tree helps us learn the fastest. Report the RMSEoos at each step.

a) Repeat the bagging of the decision tree, using a base tree of maximum depth 1, 2, ... n while the RMSEoos keeps dropping; stop when the RMSEoos has started increasing again.

- Unable to get function to work

b) Repeat the boosting of the decision tree, using a base tree of maximum depth 1, 2, ... n while the RMSEoos keeps dropping; stop when the RMSEoos has started increasing again.

- Unable to get function to work