Assignment 2

Anthony Fundzak

4/10/2022

Question A1: The key idea behind bagging is randomly selecting data points (each data point can further be replaced in the data set and used multiple times), train the weak models independently inn order to reduce the overall variance of the model. Bagging deals with reducing variance but it cannot deal with high bias

Question A2: Bagging models are computationally more efficient because the modules are trained independently while in boosting the models are trained sequentially.

Question A3: I do think creating an ensemble model can boost the performance of the model. By building a ensemble model that can learn from the method of stacking or staked learning. This allows the new models to learn off the previous models work in order to reach a higher accuracy than its previous model. Stacked generalization will also allow the model to become more precise and can have an overall boost the performance effect on the ensemble model.

Question A4: Information gain is the expected reduction in entropy of the target variable in the data sample set after splitting. In this instance, based on the size attribute, the information gain would be: entropy(parent)-entropy(average of children) which comes to .085 information gain based off size.

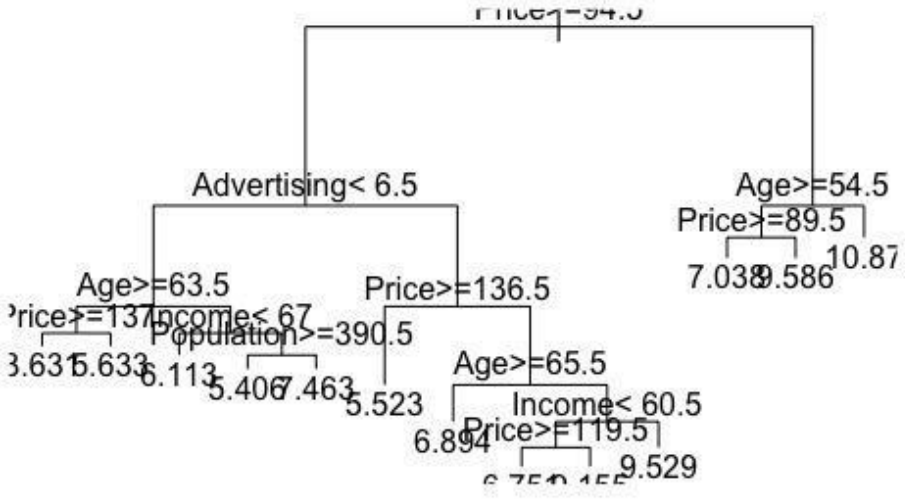
Question A5: It is important for the m parameter to be optimally split at each random forest model because we want to achieve the most possible information gained and diversity in data selected from the nodes so the first decision tree ran to the last will have maximum diversity/entropy within the nodes. If we choose a model that has a to large or to small m parameter, the model could be impure by choosing almost all attributes within a node, thus we will not be enough information gain and randomness/diversity among the data. The tuneRF function can show you the optimal number of m parameters needed.

Part B

> library(rmarkdown) > library(ISLR) > library(dplyr) > library(caret) > attach(Carseats)

Question B1

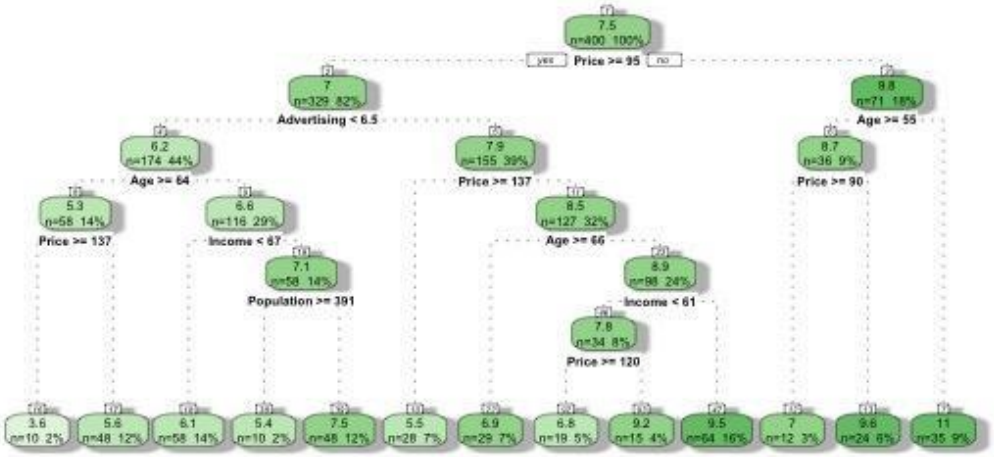
>library(rpart)
> library(rpart.plot)
>carseats_Filtered <- Carseats %>% select("Sales", "Price", "Advertising","Population","Age","Income","Education") model_1=rpart (Sales~.,data=mydata, method='anova')
>plot(model_1)
>text(model_1)
Price is used at the top of the tree for splitting.



Question B2

```
>library(rpart.plot)
>library(rattle)
```

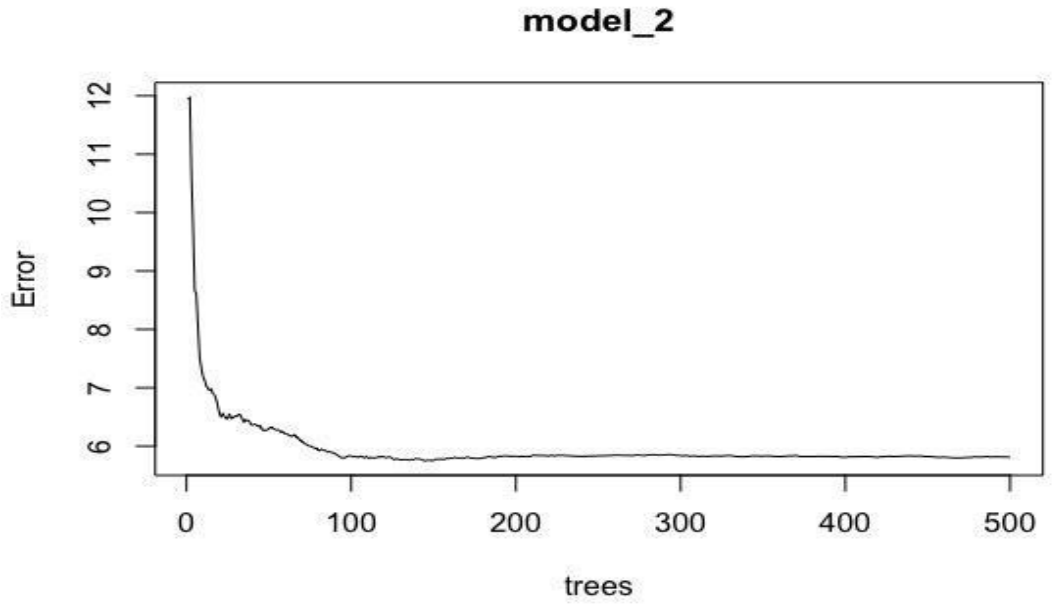
The estimated sales for this record will be 9.6, located in node 13.



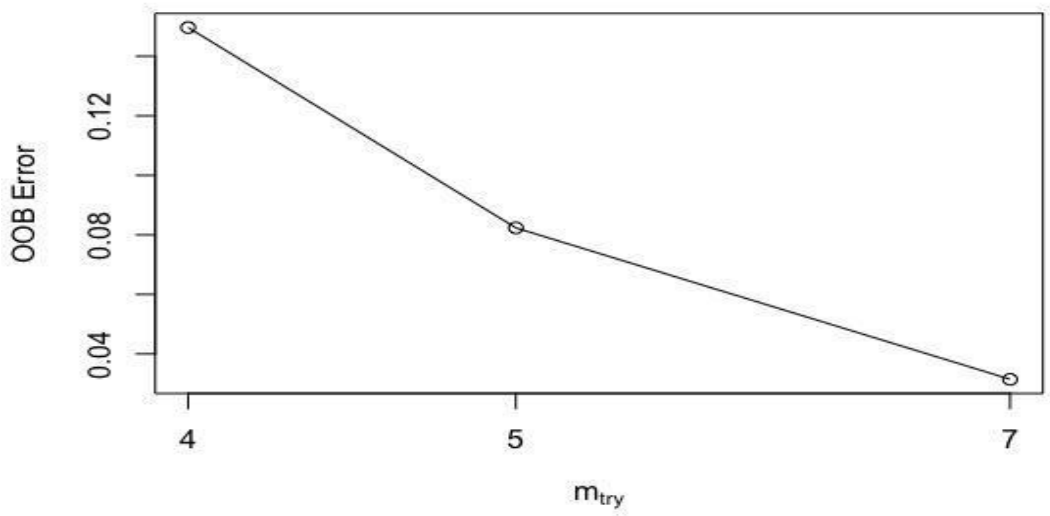
Rattle 2022-Apr-10 16:49:31 anthonyfundzak

Question B3

```
>library(ranger)
>library(randomForest)
> library(rsample)
>library(h2o)
> library(broom)
>carseats_split <- initial_split(mydata, prop = .7)
>carseats_train <- training(carseats_split)
>carseats_test <- testing(carseats_split)
> set.seed(123)
>model_2 <- randomForest(formula = Sales~., data=carseats_train)
>plot(model_2)
```




```
>model_3 <- tuneRF(x=carseats_train[features], y=carseats_train$Sales,
ntreeTry=500, mtryStart=5, stepFactor=1.5, improve=.01, trace=FALSE)
0.817746 0.01
0.6182017 0.01
## mtry with the value of 7 gives the best performance.
```



Question B4

> hyper_grid <- expand.grid(mtry=seq(2,3,5), node_size=seq(3,9, by=2), sample_size=c(.55, .632, .70, .80), OOB_RMSE=0)

>for (i in 1:nrow(hyper_grid)) {model_5 <- ranger(formula=Sales~., data=carseats_train, num.trees =500,

mtry=hyper_grid$mtry[i]$, min.node.size = hyper_grid$node_size[i]$,

sample.fraction=hyper_grid$sample_size[i]$, $seed = 123$)

+ $hyper_grid$OOB_RMSE[i] <- sqrt(model_5$prediction.error)

+ }

>hyper_grid %>% dplyr::arrange(OOB_RMSE) %>%

+ head(10)

	mtry	node_size	sample_size	OOB_RMSE
1	2	3	0.550	2.391077
2	2	5	0.800	2.391290
3	2	5	0.700	2.392326
4	2	7	0.550	2.397911
5	2	7	0.700	2.398612
6	2	5	0.550	2.399040
7	2	9	0.632	2.399164
8	2	9	0.550	2.401196
9	2	3	0.700	2.401409
10	2	5	0.632	2.401616

The model with the best performance is mtry 2, node_size 3, sample size, 55% with OOB_RMSE of 2.39