STATS 790
Assignment 4

Yilong Zhai

05 April, 2023

Q1

The dataset we selected for Q1 is Penguins from Kaggle(Gorman 2021), which contain the data from 344 penguins, including their species(Adelie, Gentoo, and Chinstrap), their island(Biscoe, Dream, and Torgersen), their sex, their bill length, bill depeth, fipper length, and body mass, the units for these quantitative variables are mm, mm, mm,and gram respectively. The response variable in this question is the sex, a categorical random variable, there are two categorical variables. island, and species, four quantitative variables, bill length, bill depth, body mass, and flipper length There are a few missing values, and the rows which contained missing value would be removed. To investigate the relationship between the sex of penguin and Their bill length, bill depth, flipper length and body mass, we will do some classification under the environment of R(R Core Team 2020) and RMarkdown(Xie, Dervieux, and Riederer 2020). This dataset contains categorical and continuous variable, which is great for prediction and easy to compare the performance for different parameter.

The model we select is random forest, which is a traditional tree model. This model could gain high accuracy and is robutness. Moreover, random forest would not relay on the data type.

we would use the factor island to split the training set and testing set.

In our model, there does not exist a loss function, it uses a combination of techniques like bagging, random feature selection, and majority voting to reduce overfitting and improve the accuracy of the predictions.

the parameter we choose for random forest is mtry=1, ntree=850. Usually mtry would perform better when it equals to square root if predictor variables, in our case we find that when mtry=2, we achieve the best performance, which means the highest accuracy with OOB error=8.74%. ntree means the number of decision trees to include in the model, usually larger ntree performance better, but it will need longer running time, so we select 850 in our model.

in the last plot, we could find that body mass is the most important predictor in this model. From the perspective of importance, the order of variables is body mass, bill depth, bill length, flipper length, species, and island.

Also, our error rate in this model is 0.08510638.

```
library(tidymodels)
```

```
## -- Attaching packages -------------------------------------- tidymodels 1.0.0 --
## v broom        1.0.3     v recipes      1.0.5
## v dials        1.1.0     v rsample      1.1.1
## v dplyr        1.1.0     v tibble       3.2.1
## v ggplot2      3.4.1     v tidyr        1.3.0
## v infer        1.0.4     v tune         1.0.1
## v modeldata    1.1.0     v workflows    1.1.3
## v parsnip      1.0.4     v workflowsets 1.0.0
## v purrr        1.0.1     v yardstick    1.1.0

## -- Conflicts ----------------------------------------- tidymodels_conflicts() --
## x purrr::discard() masks scales::discard()
## x dplyr::filter()  masks stats::filter()
## x dplyr::lag()     masks stats::lag()
## x recipes::step()  masks stats::step()
## * Dig deeper into tidy modeling with R at https://www.tmwr.org
```

```
library(dplyr)
library(randomForest)
```

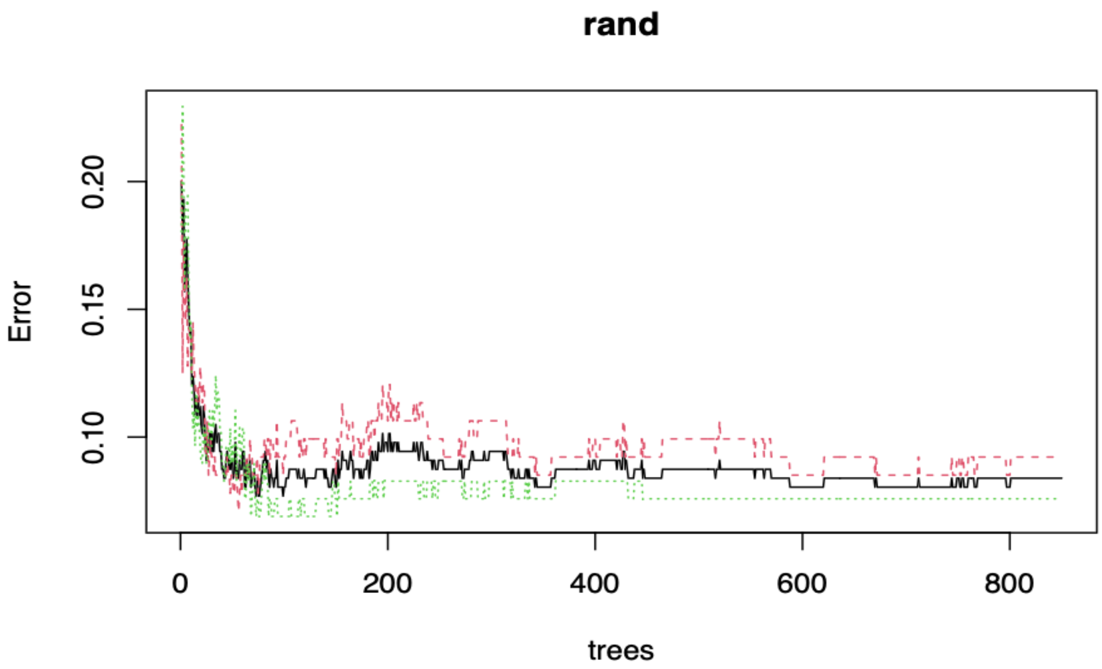
```
## randomForest 4.7-1.1

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

##
## Attaching package: 'randomForest'

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

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

from here, we import the packages we needed.

```
penguin=read.csv("penguins.csv")
penguin=na.omit(penguin)
penguin=penguin[,-1]
penguin=penguin[,-8]
penguin$sex=as.factor(penguin$sex)
penguin$species=as.factor(penguin$species)
penguin$island=as.factor(penguin$island)
```

From here, we load the dataset and scale the categorical variables.

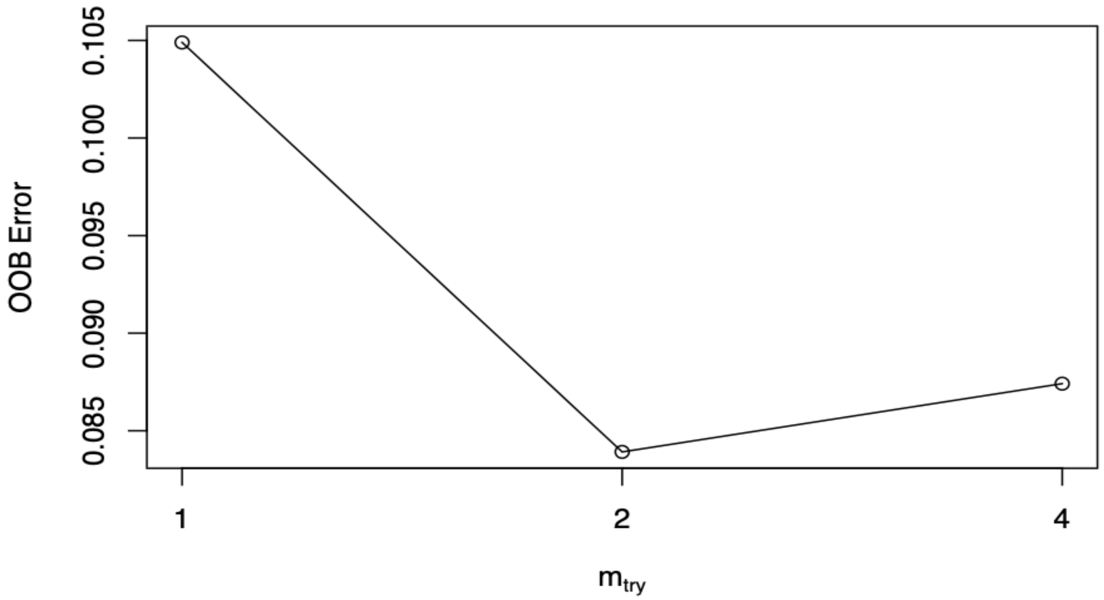
```
set.seed(1)
penguin_train1=penguin %>%
  mutate(train=(island!="Torgersen"))%>%
  filter(train)
penguin_test1=penguin %>%
  mutate(train=(island!="Torgersen"))%>%
  filter(!train)
rand=randomForest(sex~.,data=penguin_train1,mtry=2,imoprtance=TRUE,ntree=850)
plot(rand)
```



from here, we split the dataset into training set and testing set and also fit the random forest model.

```
set.seed(1)
mtry=tuneRF(penguin_train1[,-7],penguin_train1[,7],ntreeTry = 850,stepFactor = 0.5,
            improve=0.05,trace=TRUE,plot=TRUE)
```

```
## mtry = 2  OOB error = 8.39%
## Searching left ...
## mtry = 4     OOB error = 8.74%
## -0.04166667 0.05
## Searching right ...
## mtry = 1     OOB error = 10.49%
## -0.25 0.05
```



from here, we find the best value of mtry

```
table(predict(rand,penguin_test1[,-7]),penguin_test1$sex)
```

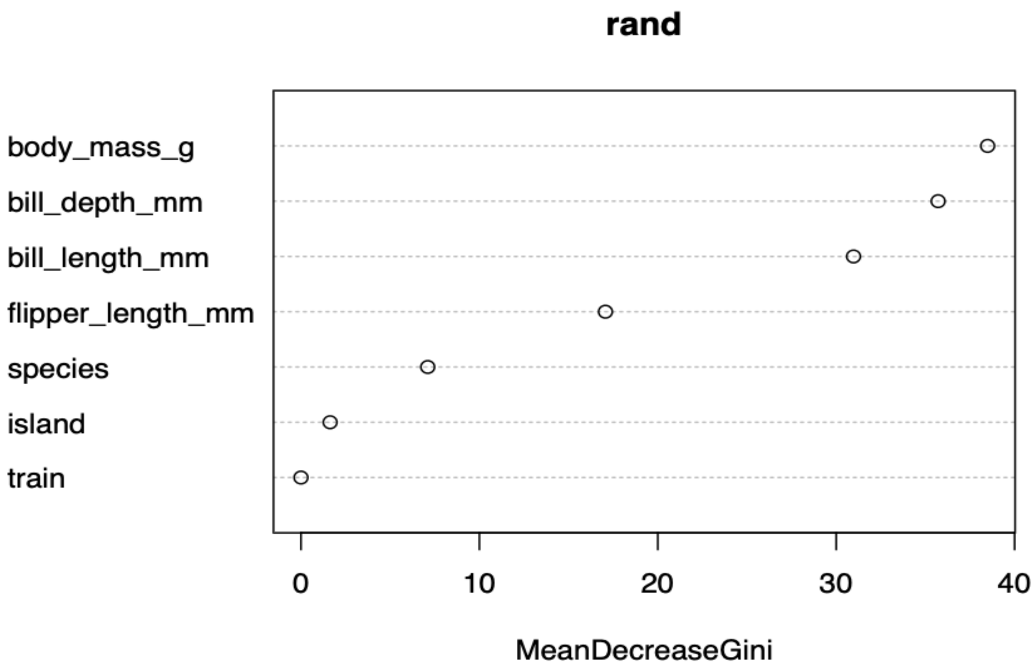
```
##
##         female male
##   female     22    2
##   male        2   21
```

```
tablerf=table(predict(rand,penguin_test1[,-7]),penguin_test1$sex)
1-sum(diag(tablerf))/sum(tablerf)
```

```
## [1] 0.08510638
```

from here, we showed the accuracy of our prediction.

```
varImpPlot(rand)
```



from here, we showed the importance of variables.

Q2

a)

For MSE:

loss function for MSE is $L = |y_i - f_{jm}(x_i)|_2^2$

to find the optimal value of weights γ_{jm}, we want $r_{im} = -\frac{dL}{df_{jm}(x_i)} = 2(y_i - f_{jm}(x_i))$

$f(\gamma) = \sum_{i=1}^N L(y_i, f_{m-1}(x_i) + \gamma)$ (Hastie, Tibshirani, and Friedman 2008)

$\frac{df}{d\gamma} = \sum_{i=1}^N 2(\gamma - y_i + f_{m-1}(x_i)) = 0$

so $\gamma_{jm} = \frac{1}{N}\sum_{i=1}^N (y_i - f_{m-1}(x_i))$

For binomial deviance

loss function for binomial deviance is $L(y_i, f_{m-1}(x_i)) = -log(1 + e^{-2y_i f_{m-1}(x_i)})$

$f(\gamma) = \sum_{i=1}^N L(y_i, f_{m-1}(x_i) + \gamma)$

$\frac{df}{d\gamma} = \sum_{i \in R_{jm}} \frac{2y_i e^{-2y_i(f_{m-1}(x_i)+\gamma)}}{1+e^{-2y_i(f_{m-1}(x_i)+\gamma)}} = \sum_{i \in R_{jm}} \frac{2y_i}{1+e^{2y_i(f_{m-1}(x_i)+\gamma)}} = 0$

since for binomial deviance, y could only be 1 or -1, so we know, $P(y_i = 1|x) = \frac{e^{2(f_{m-1}(x_i)+\gamma)}}{1+e^{2(f_{m-1}(x_i)+\gamma)}}, P(y_i = -1|x) = \frac{1}{1+e^{2(f_{m-1}(x_i)+\gamma)}}$

we have $\sum_{i \in R_{jm}} \frac{2y_i}{1+e^{2y_i(f_{m-1}(x_i)+\gamma_{jm})}} = 0$

b)

For newton boosting, for a fixed value of q(x), we have the second order approximation $w_j = -\frac{\sum_{i \in I_j} g_i}{\sum_{i \in I_j} h_i + \lambda}$ where w_j is the optimal weights and $g_i = \frac{dL(y_i, f_{m-1}(x_i))}{df_{jm}(x_i)}, hi = \frac{d^2L(y_i, f_{m-1}(x_i))}{df_{jm}(x_i)^2}$ (Chen and Guestrin 2016)

for MSE:

$g_i = -2(y_i - f_{m-1}(x_i)), h_i = 2$

so optimal weights $w_j = -\frac{\sum_{i \in I_j} -2(y_i - f_{m-1}(x_i))}{\sum_{i \in I_j} 2 + \lambda}$

for binomial deviance:

$g_i = \frac{2y_i}{1+e^{2y_i f_{m-1}(x_i)}}, h_i = \frac{-4y_i^2 e^{2y_i f_{m-1}(x_i)}}{(1+e^{2y_i f_{m-1}(x_i)})^2}$

we replace w_j by the natation we used here γ_{jm}

optimal weights $\gamma_{jm} = -\frac{\sum_{i\in I_j} \frac{2y_i}{1+e^{2y_i f_{m-1}(x_i)}}}{\sum_{i\in I_j} \frac{-4y_i^2 e^{2y_i f_{m-1}(x_i)}}{(1+e^{2y_i f_{m-1}(x_i)})^2}+\lambda}$

Reference

Chen, Tianqi, and Carlos Guestrin. 2016. “XGBoost: A Scalable Tree Boosting System.” *In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–94.

Gorman, Kristen. 2021. “Penguins.” https://www.kaggle.com/datasets/larsen0966/penguins?resource=download.

Hastie, Trevor, Robert Tibshirani, and Jerome Friedman. 2008. *The Elements of Statistical Learning*. Springer.

R Core Team. 2020. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.

Xie, Yihui, Christophe Dervieux, and Emily Riederer. 2020. *R Markdown Cookbook*. CRC Press.