Mushroom Capstone Project

Chanya Limdamnern

22/5/2564

Contents

Introduction

This capstone project is the second part of PH125.9x: Data Science: Capstone course. For this project, mushroom class(edible or poisonous) prediction model will be developed using mushrooms dataset downloaded from Kaggle and kept in my GitHub.

The goal of this project is to develop model for mushroom class prediction which maximize poisonous mushroom prediction accuracy.

Overview

The first step of coding is to install and load required package which is not shown the code here. Next step is to import mushroom dataset from my GitHub into R.

```
### import mushroom dataset from my github
url<- "https://raw.githubusercontent.com/ChanyaLim/Individual_Project/master/mushrooms.csv"
## assign column names
col_name <- c("class","cap_shape","cap_surface","cap_color",
              "bruises","odor","gill_attachment","gill_spacing",
              "gill_size","gill_color","stalk_shape","stalk_root",
              "stalk_surface_above_ring","stalk_surface_below_ring",
              "stalk_color_above_ring","stalk_color_below_ring",
              "veil_type","veil_color","ring_number","ring_type","spore_print_color",
              "population","habitat")
## import mushrooms.csv file
mushroom <- fread(url,col.names=col_name)
## change character data to factor
mushroom <- mushroom %>% mutate_if(is.character, as.factor)
```

Data structure below shows that mushroom dataset have 23 variables and 8124 observations. You can see that all variables are factor(categorical data). Meaning of levels of each variable is in the Appendix A.

```
str(mushroom,width=80,strict.width="cut")
```

```
## Classes 'data.table' and 'data.frame':   8124 obs. of  23 variables:
##  $ class                   : Factor w/ 2 levels "e","p": 2 1 1 2 1 1 1 1 2 1 ...
##  $ cap_shape               : Factor w/ 6 levels "b","c","f","k",..: 6 6 1 6 6 ..
##  $ cap_surface             : Factor w/ 4 levels "f","g","s","y": 3 3 3 4 3 4 3..
##  $ cap_color               : Factor w/ 10 levels "b","c","e","g",..: 5 10 9 9 ..
##  $ bruises                 : Factor w/ 2 levels "f","t": 2 2 2 2 1 2 2 2 2 2 ...
##  $ odor                    : Factor w/ 9 levels "a","c","f","l",..: 7 1 4 7 6 ..
##  $ gill_attachment         : Factor w/ 2 levels "a","f": 2 2 2 2 2 2 2 2 2 2 ...
##  $ gill_spacing            : Factor w/ 2 levels "c","w": 1 1 1 1 2 1 1 1 1 1 ...
##  $ gill_size               : Factor w/ 2 levels "b","n": 2 1 1 2 1 1 1 1 2 1 ...
##  $ gill_color              : Factor w/ 12 levels "b","e","g","h",..: 5 5 6 6 5..
##  $ stalk_shape             : Factor w/ 2 levels "e","t": 1 1 1 1 2 1 1 1 1 1 ...
##  $ stalk_root              : Factor w/ 5 levels "?","b","c","e",..: 4 3 3 4 4 ..
##  $ stalk_surface_above_ring: Factor w/ 4 levels "f","k","s","y": 3 3 3 3 3 3 3..
##  $ stalk_surface_below_ring: Factor w/ 4 levels "f","k","s","y": 3 3 3 3 3 3 3..
##  $ stalk_color_above_ring  : Factor w/ 9 levels "b","c","e","g",..: 8 8 8 8 8 ..
##  $ stalk_color_below_ring  : Factor w/ 9 levels "b","c","e","g",..: 8 8 8 8 8 ..
##  $ veil_type               : Factor w/ 1 level "p": 1 1 1 1 1 1 1 1 1 1 ...
##  $ veil_color              : Factor w/ 4 levels "n","o","w","y": 3 3 3 3 3 3 3..
```

```
##  $ ring_number             : Factor w/ 3 levels "n","o","t": 2 2 2 2 2 2 2 2 2..
##  $ ring_type               : Factor w/ 5 levels "e","f","l","n",..: 5 5 5 5 1 ..
##  $ spore_print_color       : Factor w/ 9 levels "b","h","k","n",..: 3 4 4 3 4 ..
##  $ population              : Factor w/ 6 levels "a","c","n","s",..: 4 3 3 4 1 ..
##  $ habitat                 : Factor w/ 7 levels "d","g","l","m",..: 6 2 4 6 2 ..
##  - attr(*, ".internal.selfref")=<externalptr>
```

Executive summary

Now mushroom dataset is imported into R already. Next step is data cleaning including NA, duplicate, unknown data and non-significant data such as single level factor variable validation.

After data cleaning finished, data exploration and visualization will be introduced. This section is to select significant variable with mushroom class for model development.

Once we get the variables for our model development, next step is to split mushroom dataset to be train set for model development and test set for model validation by using confusion matrix. The model which maximize poisonous mushroom prediction accuracy will be the final model for mushroom class prediction.

Methods and analysis

Data cleaning

NA

The result from code below shows FALSE means there is no NA in our mushroom dataset.

```
any(is.na(mushroom))
```

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

Duplicated data

The result from code below shows 0 means there is no duplicated observation in our mushroom dataset.

```
sum(duplicated(mushroom))
```

```
## [1] 0
```

Unknown and non significant data

Below dataframe shows summary of level number and levels for each variable. From this summary, you will see that there is "?" level in stalk root variable which means unknown data and there is only 1 level in veil type which is non-significant for model development so I decided to remove these two variables from the dataset.

```
level <- sapply(mushroom,levels)
Number_level <- sapply(mushroom,uniqueN)
dataoverview <- as.data.frame(Number_level) %>% mutate(Level=level)
dataoverview
```

```
##                          Number_level                          Level
## class                               2                            e, p
## cap_shape                           6                 b, c, f, k, s, x
## cap_surface                         4                       f, g, s, y
## cap_color                          10       b, c, e, g, n, p, r, u, w, y
## bruises                             2                            f, t
## odor                                9          a, c, f, l, m, n, p, s, y
## gill_attachment                     2                            a, f
## gill_spacing                        2                            c, w
## gill_size                           2                            b, n
## gill_color                         12 b, e, g, h, k, n, o, p, r, u, w, y
## stalk_shape                         2                            e, t
## stalk_root                          5                    ?, b, c, e, r
## stalk_surface_above_ring            4                       f, k, s, y
## stalk_surface_below_ring            4                       f, k, s, y
## stalk_color_above_ring              9          b, c, e, g, n, o, p, w, y
## stalk_color_below_ring              9          b, c, e, g, n, o, p, w, y
## veil_type                           1                               p
## veil_color                          4                       n, o, w, y
## ring_number                         3                          n, o, t
## ring_type                           5                    e, f, l, n, p
## spore_print_color                   9          b, h, k, n, o, r, u, w, y
## population                          6                 a, c, n, s, v, y
## habitat                             7               d, g, l, m, p, u, w
```

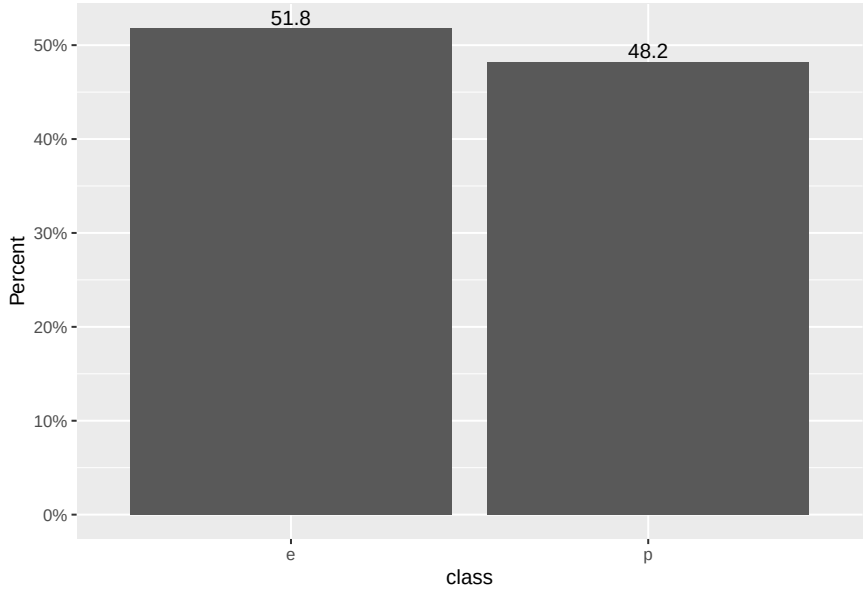
```
mushroom <- mushroom %>% select(-"veil_type")
mushroom <- mushroom %>% select(-"stalk_root")
```

Now data cleaning step have been finished. Our mushroom dataset have no NA data, no duplicated observation and already removed variables which is non-significant and have unknown data.

Data exploration and visualization

Class

This variable is the target prediction including 2 levels; e is edible and p is poisonous. Graph below shows proportion of each level in class variable. The proportion of e : p is 51.8 : 48.2 so this data is balance.



Next step is to explore each variable and select variable which will be used in model development step. For variable selection, proportion of each variable and proportion compare to class will be plotted.

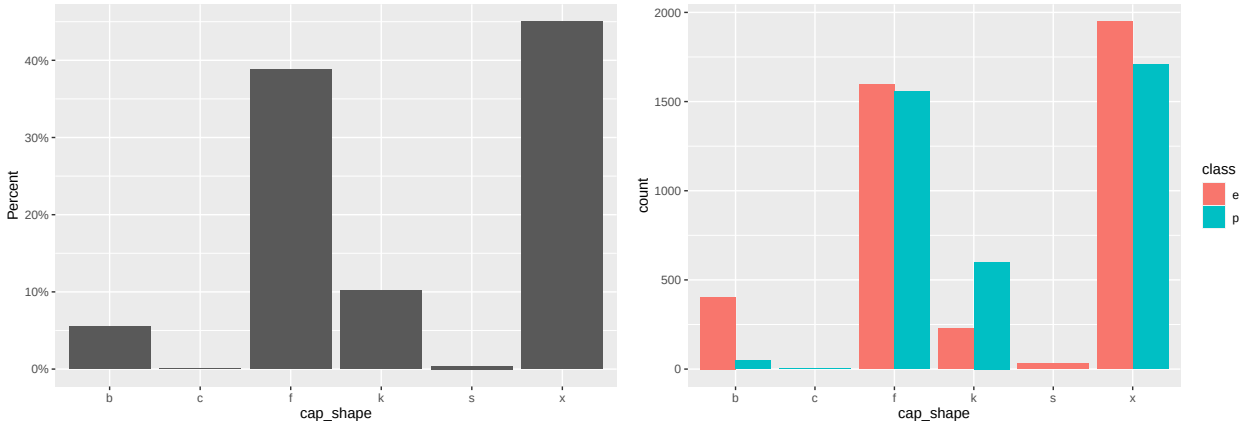
First group of selected variables are variable which each levels proportion is more than 1% and proportion compare to class variable is significant. The reason to select only more than 1% each level is to minimize overfitting problem in the prediction model.

Second group of selected variables will consider others variable further from first group and add variable which proportion compare to class variable of big portion level is significant. Due to this group consider less than 1% each level, it may lead to overfitting problem so during model development, we need to check variable important and see if less portion level is top important. Then, adjust the model to minimize overfitting.

Cap shape

The first graph below shows level c and s are less than 1% and big portion are level f and x.

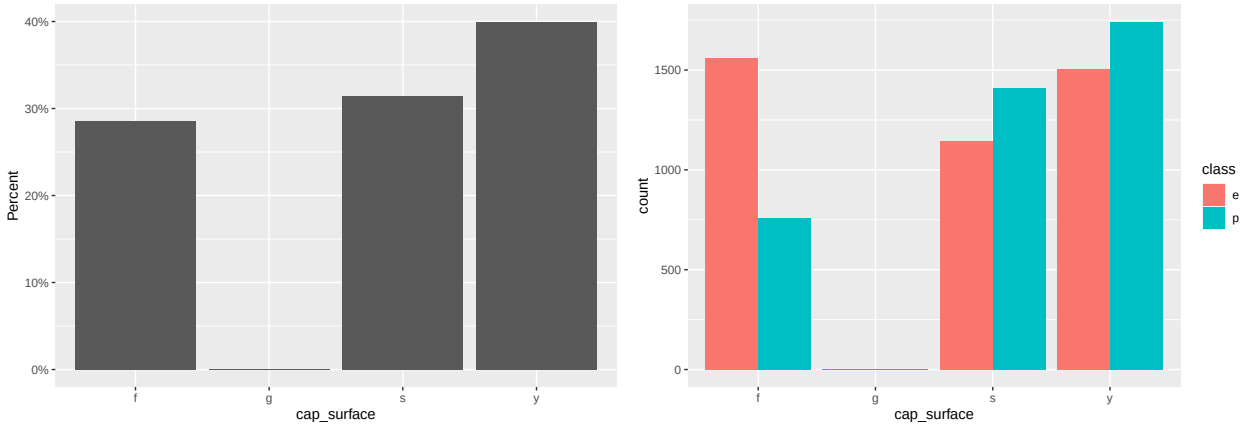
The second graph shows big portion levels are not significant to class because proportion between poisonous and edible of level f and x are not much different so this variable is not used in model development step.



Cap surface

The first graph below shows level g is less than 1% and big portion are f,s and y level.

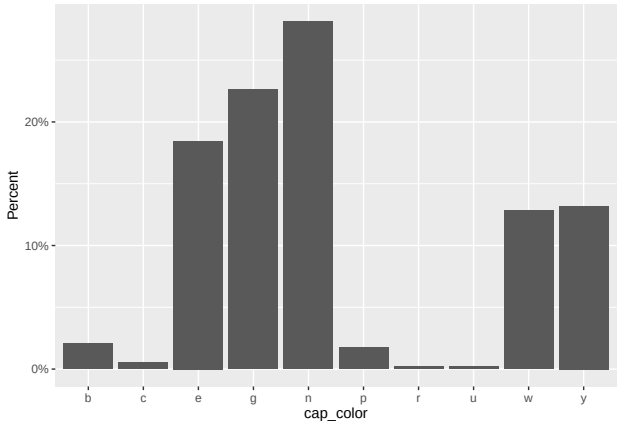
The second graph shows f level is significant to class because large majority of f is edible so this variable will be in second group.

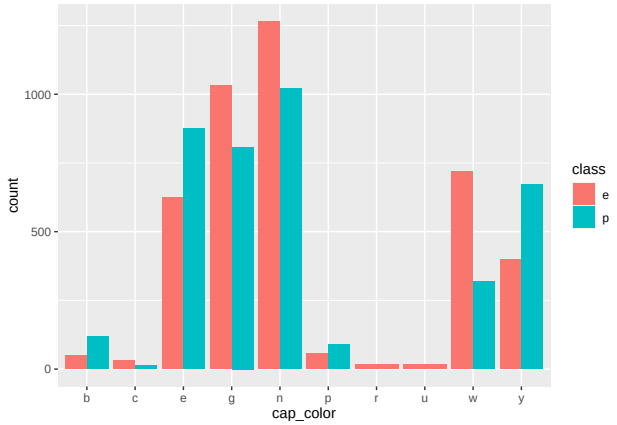


Cap color

The first graph below shows level c, r and u are less than 1% and big portion are e, g, n, w and y level.

The second graph shows e, g, n, w and y level are significant to class because large majority of w is edible, majority of e, y are poisonous and majority of g, n are edible so this variable will be in second group.

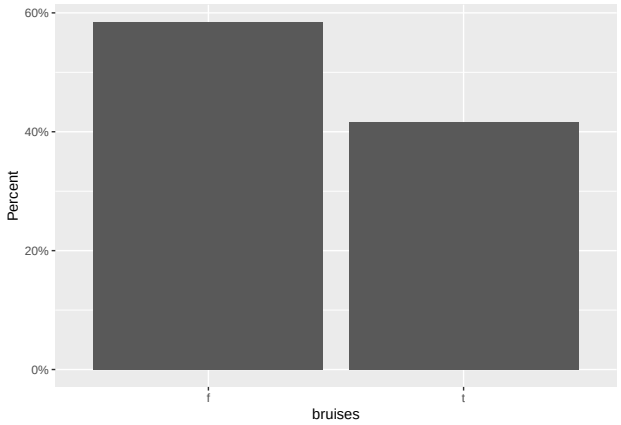


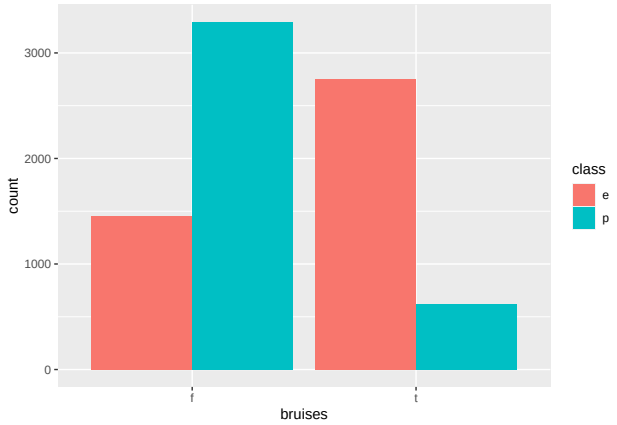


Bruises

The first graph below shows no level is less than 1% and both f and t level are big portion.

The second graph shows f and t level are significant to class because large majority of f is poisonous and large majority of t is edible so this variable will be in first group.

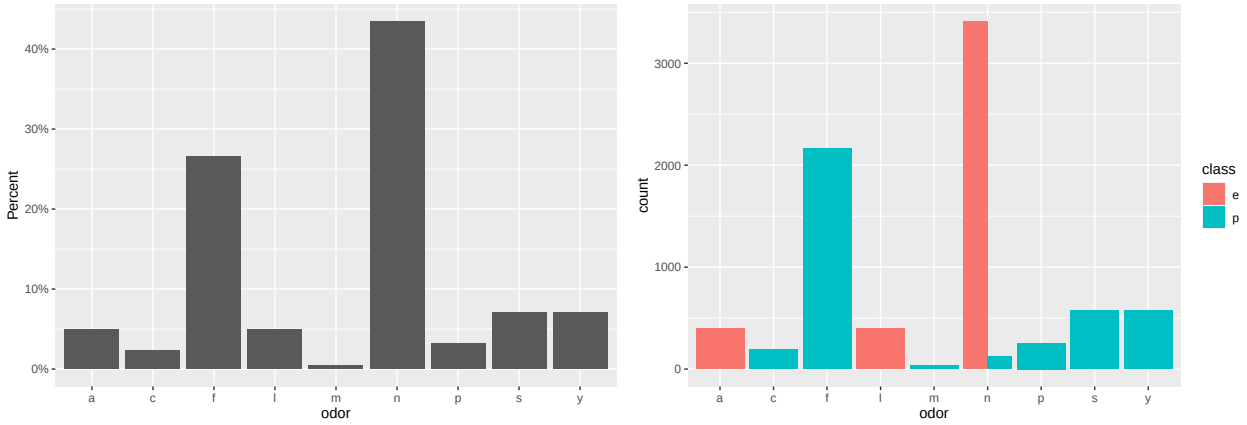




Odor

The first graph below shows level m is less than 1% and big portion are f and n level.

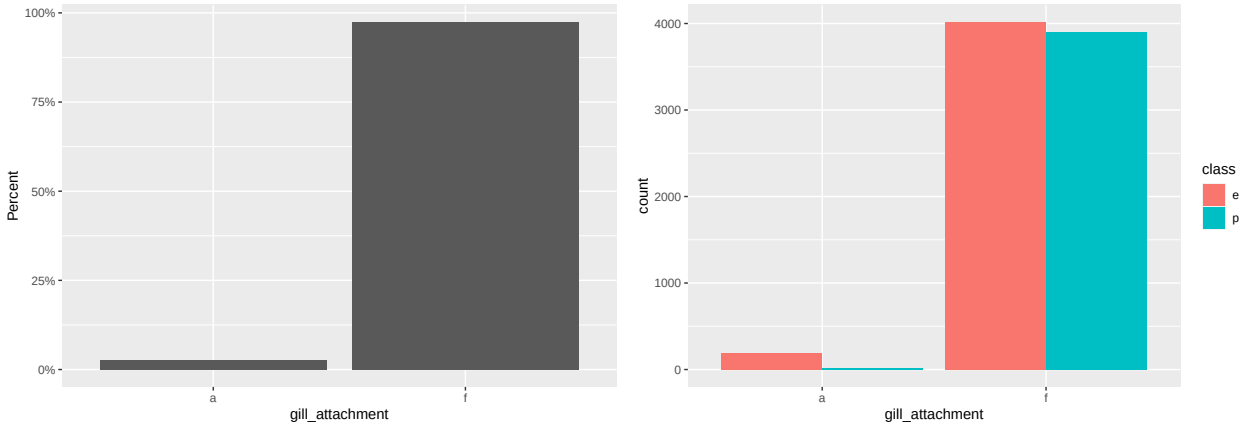
The second graph shows both f and n level are significant to class because all f is poisonous and large majority of n is edible so this variable will be in second group.



Gill attachment

The first graph below shows level a is less than 1% and big portion is level f.

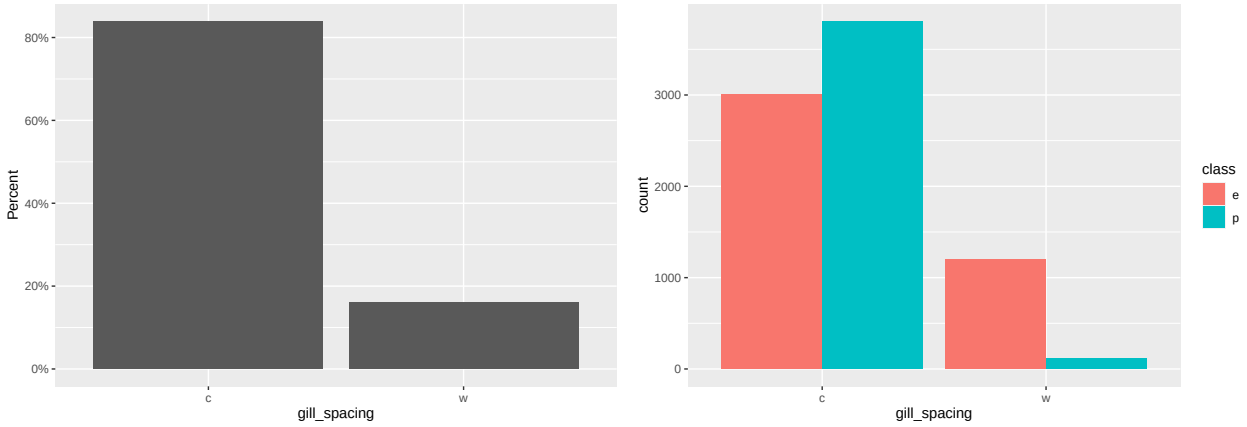
The second graph shows level f is not significant to class so this variable is not used in model development step.



Gill spacing

The first graph below shows no level is less than 1% and c level is big portion.

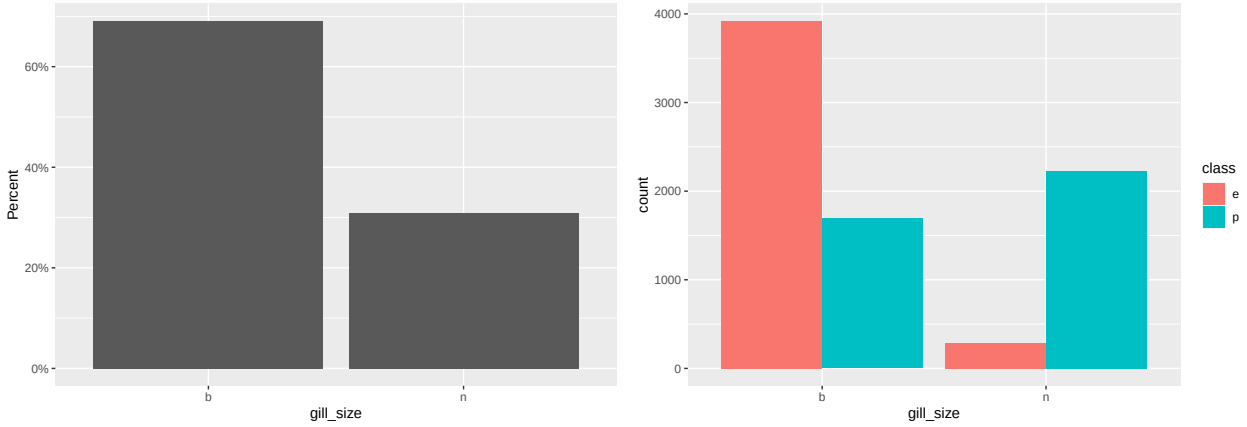
The second graph shows c level is not quite significant to class even though majority of c is poisonous but proportion between poisonous and edible is not much different. And you can see w level which is small portion but large majority of w is edible. From this exploration, this variable will also be in first group.



Gill size

The first graph below shows no level is less than 1% and both b and n level are big portion.

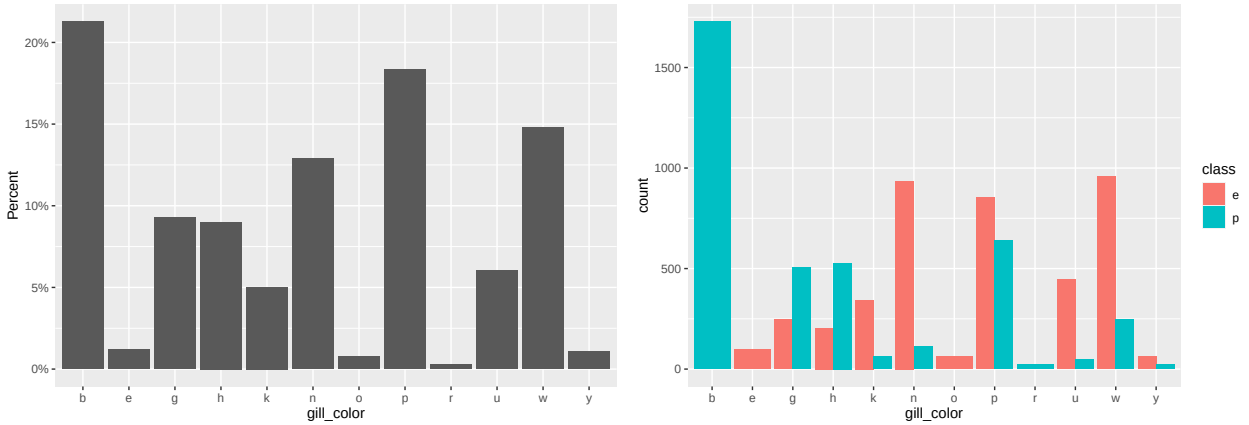
The second graph shows b and n level are significant to class because large majority of b is edible and large majority of n is poisonous so this variable will be in first group.



Gill color

The first graph below shows level r is less than 1% and big portion are b, g, h, n, p and w level.

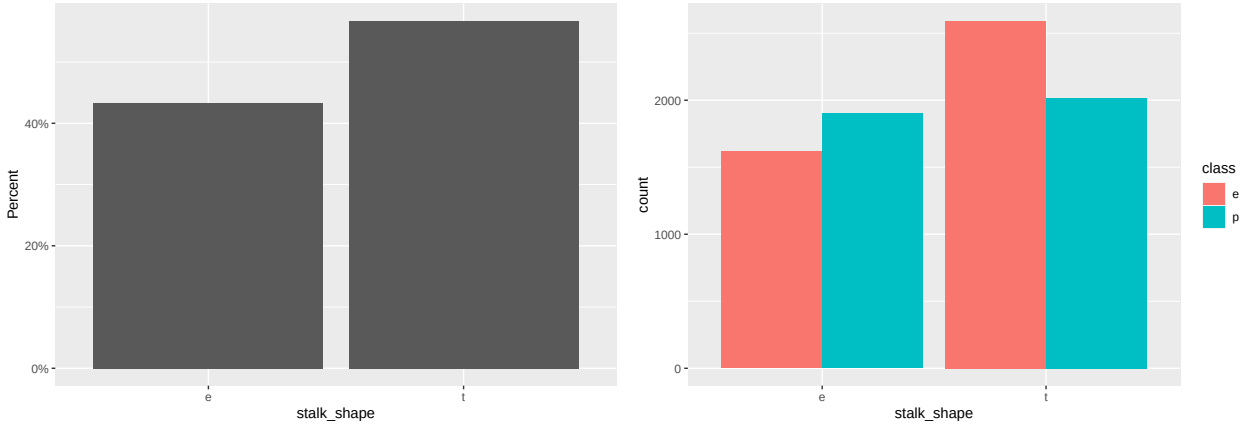
The second graph shows b, h, n and w level are significant to class because all b is poisonous, large majority of h is poisonous and large majority of n and w are edible so this variable will be in second group.



Stalk shape

The first graph below shows no level is less than 1% and both e and t level are big portion.

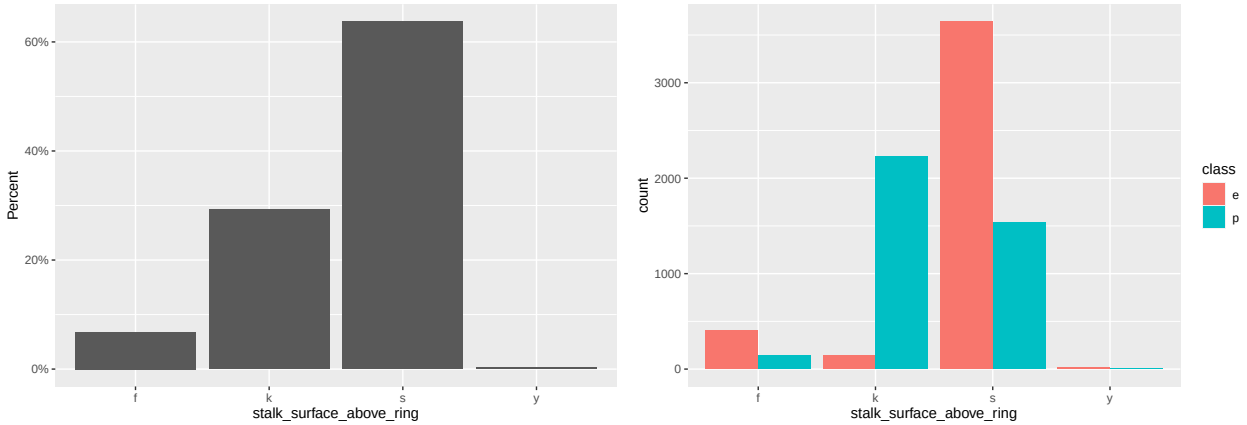
The second graph shows b and n level are not quite significant to class even though majority of e is poisonous and majority of t is edible but proportion between poisonous and edible is not much different. From this exploration, this variable will also be in first group and we can see in variable importance of the model if this variable is significant.



Stalk surface above ring

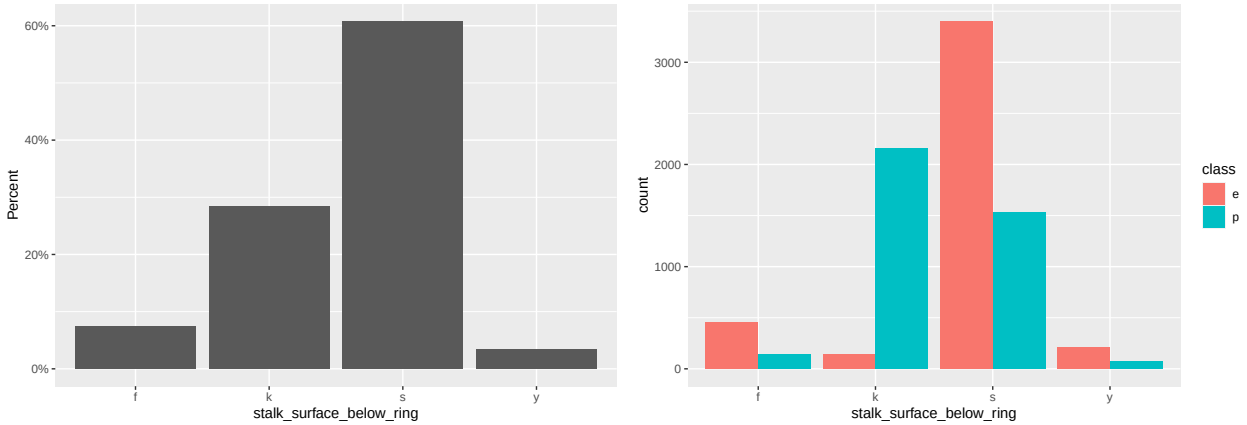
The first graph below shows level y is less than 1% and big portion are k and s level.

The second graph shows k and s level are significant to class because large majority of k is poisonous and large majority of s is edible so this variable will be in second group.



Stalk surface below ring

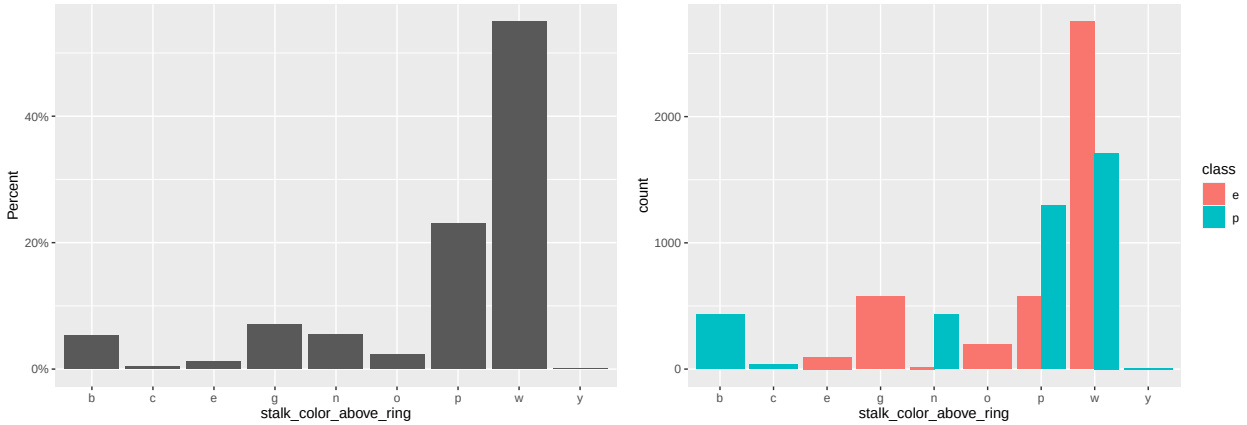
This variable characteristic is similar to stalk surface above ring so this variable will be in second group.



Stalk color above ring

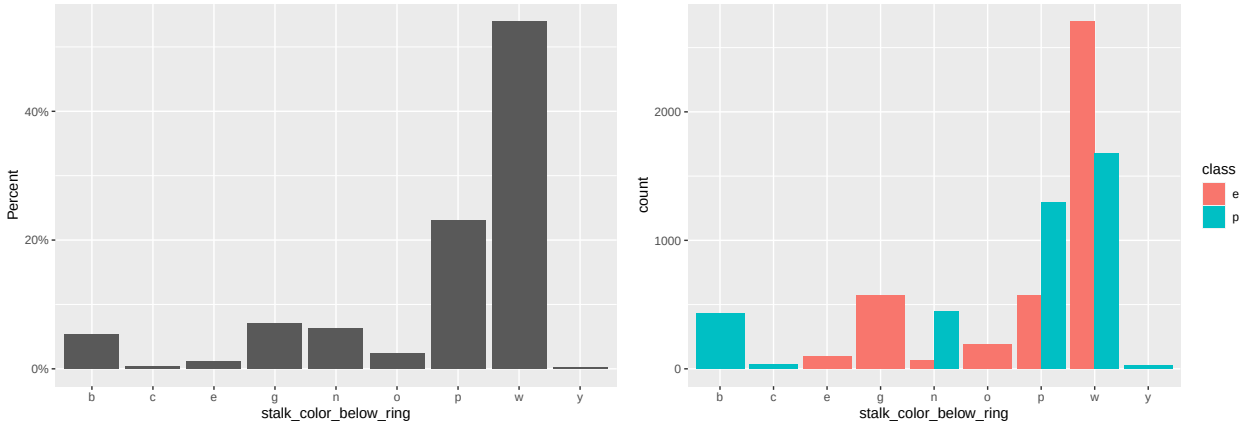
The first graph below shows level c and y are less than 1% and big portion are p and w level.

The second graph shows p and w level are significant to class because large majority of p is poisonous and majority of w is edible so this variable will be in second group.



Stalk color below ring

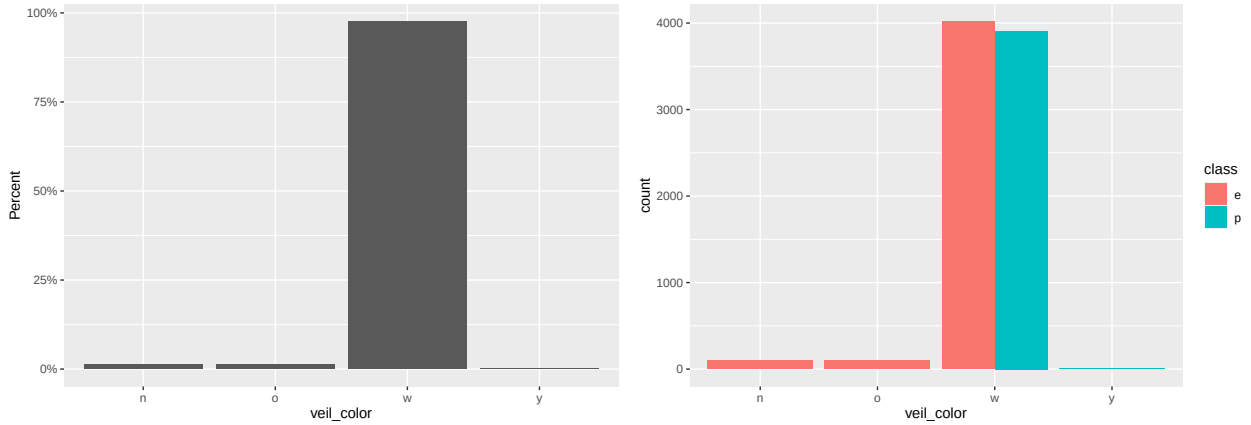
This variable characteristic is similar to stalk color above ring so this variable will be in second group.



Veil color

The first graph below shows level n, o and y are less than 1% and big portion is level w.

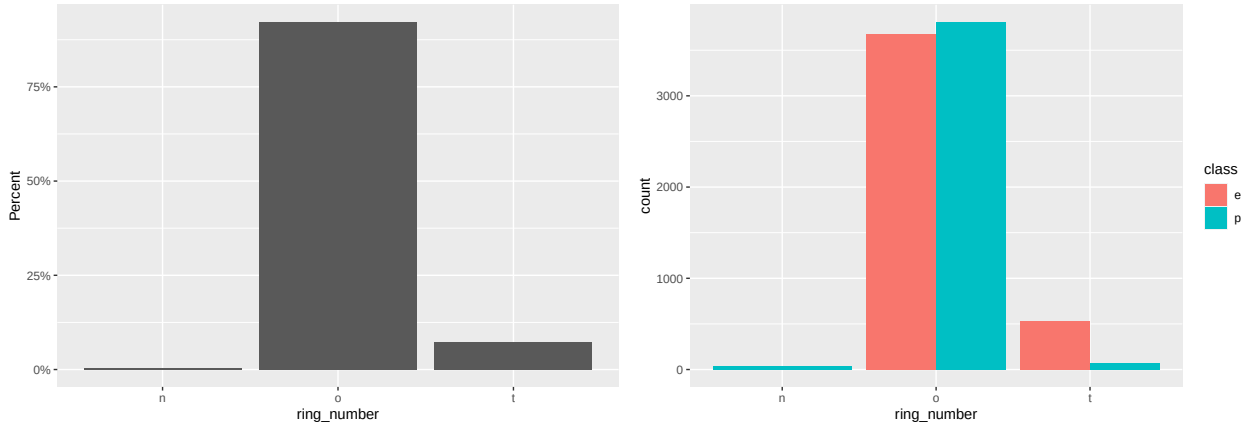
The second graph shows big portion level is not significant to class so this variable is not used in model development step.



Ring number

The first graph below shows level n is less than 1% and big portion is level o.

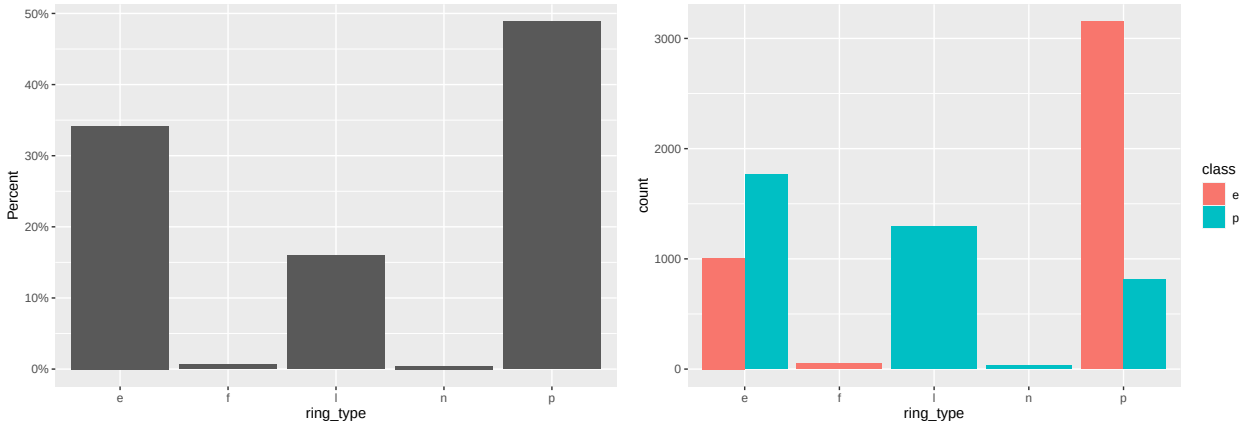
The second graph shows big portion level is not significant to class so this variable is not used in model development step.



Ring type

The first graph below shows level f and n are less than 1% and big portion are e, l and p level.

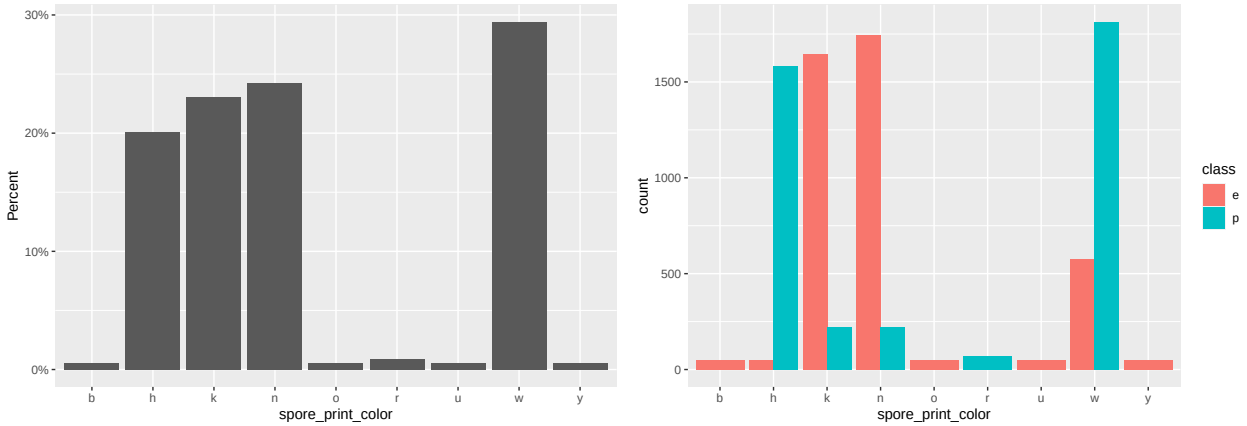
The second graph shows e, l and p level are significant to class because majority of e is poisonous, all of l is poisonous and large majority of p is edible so this variable will be in second group.



Spore print color

The first graph below shows level b,o,u and y are less than 1% and big portion are h, k, n and w level.

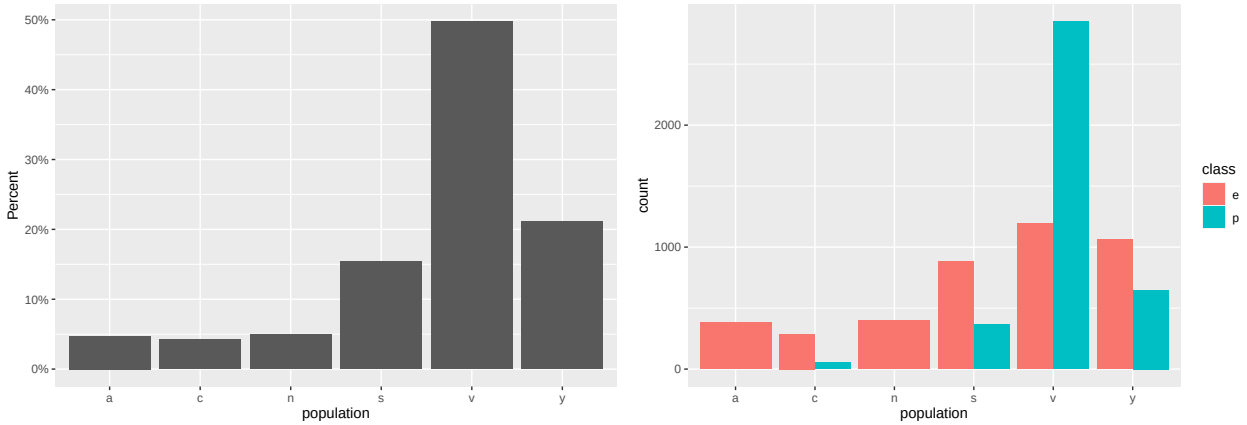
The second graph shows h, k, n and w level are significant to class because large majority of h and w are poisonous and large majority of k and n are edible so this variable will be in second group.



Population

The first graph below shows no level is less than 1% and s, v and y level are big portion.

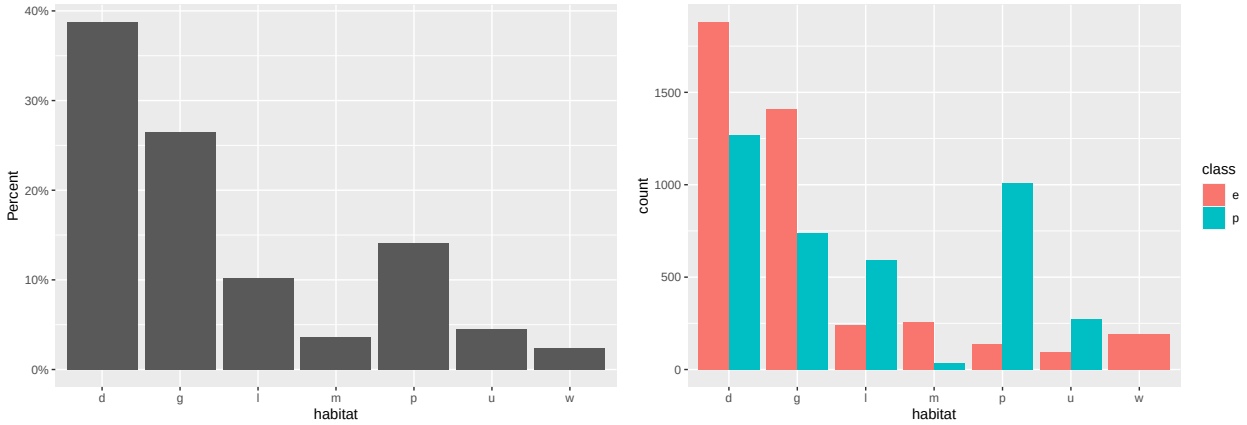
The second graph shows s, v and y level are significant to class because large majority of s is edible, large majority of v is poisonous and majority of y is edible so this variable will be in first group.



Habitat

The first graph below shows no level is less than 1% and d, g and p level are big portion.

The second graph shows d, g and p level are significant to class because majority of d and g are edible and large majority of p is poisonous so this variable will be in first group.



From all exploration above,

First group variables are bruises, gill spacing, gill size, stalk shape, population and habitat.

Second group variables are bruises, gill spacing, gill size, stalk shape, population, habitat, cap surface, cap color, odor, gill color, stalk surface above ring, stalk surface below ring, stalk color above ring, stalk color below ring, ring type and spore print color.

Model development

Split Train and Test set

Mushroom dataset is split to train set and test set with portion 0.8 and 0.2 respectively. Train set is used for prediction model development and test set is used for performance validation.

```
set.seed(1, sample.kind = "Rounding")
test_index <- createDataPartition(mushroom$class,times = 1,p=0.2,list=FALSE)
train_set <- mushroom[-test_index,]
test_set <- mushroom[test_index,]
```

Model

GLM Let's start with a simple model, generalized linear model(GLM) with first group variables.

```
set.seed(1, sample.kind = "Rounding")
train_set_glm1 <- train_set %>% select(class,bruises,gill_spacing,gill_size,stalk_shape,
                                       population,habitat)
glm1 <- train(class ~ ., method = "glm", data = train_set_glm1)
predict_glm1 <- predict(glm1,test_set)
confusionMatrix(predict_glm1,test_set$class)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   e   p
##          e 808  66
##          p  34 718
##
##                Accuracy : 0.9385
##                  95% CI : (0.9257, 0.9497)
##     No Information Rate : 0.5178
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.8767
##
##  Mcnemar's Test P-Value : 0.001935
##
##             Sensitivity : 0.9596
##             Specificity : 0.9158
##          Pos Pred Value : 0.9245
##          Neg Pred Value : 0.9548
##              Prevalence : 0.5178
##          Detection Rate : 0.4969
##    Detection Prevalence : 0.5375
##       Balanced Accuracy : 0.9377
##
##        'Positive' Class : e
##
```

Method	Accuracy	Sensitivity	Specificity
GLM(6 features)	0.9384994	0.95962	0.9158163

Table above shows performance of this GLM model from confusion matrix. This model performance is quite good with 93.8% accuracy, 96% sensitivity and 91.6% specificity. Next, random forest method will be applied. Then, we will see if the performance is better compare to GLM.

Random forest Random forest method with first group variables will be applied and Then, compare performance with GLM method.

```
set.seed(1, sample.kind = "Rounding")
train_set_rf1 <- train_set %>% select(class,bruises,gill_spacing,gill_size,stalk_shape,
                                      population,habitat)
rf1 <- train(class~.,
             method="rf",
             data = train_set_rf1,
             tuneGrid = data.frame(mtry = 2))
predict_rf1 <- predict(rf1,test_set)
confusionMatrix(predict_rf1,test_set$class)
```

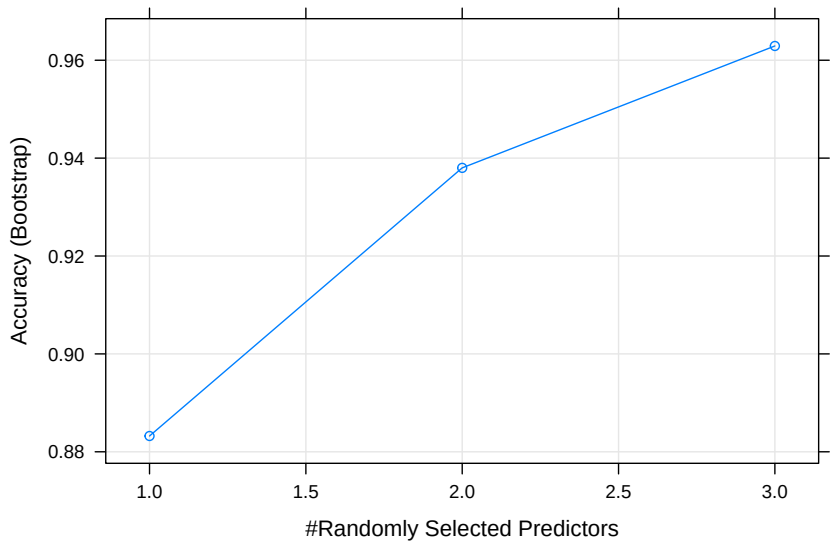
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   e   p
##          e 788  42
##          p  54 742
##
##                Accuracy : 0.941
##                  95% CI : (0.9284, 0.9519)
##     No Information Rate : 0.5178
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.8818
##
##  Mcnemar's Test P-Value : 0.2616
##
##             Sensitivity : 0.9359
##             Specificity : 0.9464
##          Pos Pred Value : 0.9494
##          Neg Pred Value : 0.9322
##              Prevalence : 0.5178
##          Detection Rate : 0.4846
##    Detection Prevalence : 0.5105
##       Balanced Accuracy : 0.9411
##
##        'Positive' Class : e
##
```

Method	Accuracy	Sensitivity	Specificity
GLM(6 features)	0.9384994	0.959620	0.9158163
Random Forest(6 features)	0.9409594	0.935867	0.9464286

The result of random forest method with first group variables is 94.1% accuracy, 93.6% sensitivity and 94.6% specificity. Compare to previous GLM method, random forest accuracy and specificity are better but sensitivity is lower. From our goal of this project to maximize poisonous mushroom prediction accuracy means maximize specificity. This random forest model is better than GLM method.

Next, I tuned mtry which is tuning parameter of random forest method from 1 to 3 to get the best performance of this method with first group variables.

```
set.seed(1, sample.kind = "Rounding")
rf1_cv <- train(class~.,
                method="rf",
                data = train_set_rf1,
                tuneGrid = data.frame(mtry = seq(1,3,1)))
plot(rf1_cv)
```



This plot shows result of mtry tuning from 1 to 3. The best performance comes from mtry = 3 with about 96% accuracy. Full result of the model is in Appendix B. Next, validate the model performance with test set.

```
predict_rf1_cv <- predict(rf1_cv,test_set)
confusionMatrix(predict_rf1_cv,test_set$class)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   e   p
##          e 804  27
```

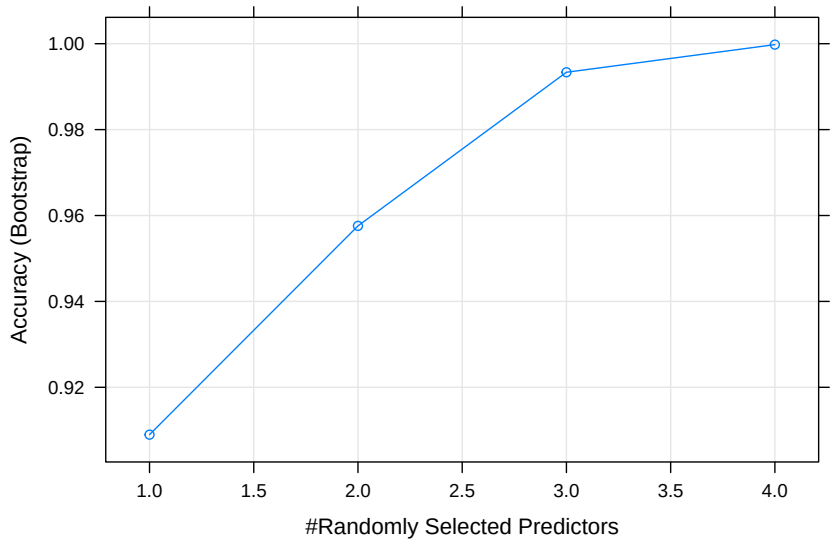
```
##          p  38 757
##
##                Accuracy : 0.96
##                  95% CI : (0.9493, 0.969)
##     No Information Rate : 0.5178
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.92
##
##  Mcnemar's Test P-Value : 0.2148
##
##             Sensitivity : 0.9549
##             Specificity : 0.9656
##          Pos Pred Value : 0.9675
##          Neg Pred Value : 0.9522
##              Prevalence : 0.5178
##          Detection Rate : 0.4945
##    Detection Prevalence : 0.5111
##       Balanced Accuracy : 0.9602
##
##        'Positive' Class : e
##
```

Method	Accuracy	Sensitivity	Specificity
GLM(6 features)	0.9384994	0.9596200	0.9158163
Random Forest(6 features)	0.9409594	0.9358670	0.9464286
Random Forest with mtry tuning(6 features)	0.9600246	0.9548694	0.9655612

The result of random forest method and mtry tuning with first group variables is 96% accuracy, 95.5% sensitivity and 96.6% specificity. Compare to previous two methods, random forest with mtry tuning accuracy and specificity are better but sensitivity is a little bit lower than GLM method. From our goal of this project to maximize specificity. This random forest model with mtry tuning is the best for now.

From the exploration data analysis section, there is second group variables. Let's apply these variables with random forest method with mtry tuning.

```
set.seed(1, sample.kind = "Rounding")
train_set_rf2 <- train_set %>% select(class,bruises,gill_spacing,gill_size,stalk_shape,
                                      population,habitat, cap_surface,cap_color,
                                      odor,gill_color,stalk_surface_above_ring,
                                      stalk_surface_below_ring, stalk_color_above_ring,
                                      stalk_color_below_ring,ring_type,spore_print_color)
rf2_cv <- train(class~.,
             method="rf",
             data = train_set_rf2,
             tuneGrid = data.frame(mtry = seq(1,4,1)))
plot(rf2_cv)
```



This plot shows result of mtry tuning from 1 to 4. The best performance comes from mtry = 4 with about 100% accuracy. Full result of the model is in Appendix B.

This model gave us the best performance but this model may lead to overfitting because we including variable which have less portion level(<1%). Top 20 variable importance is used to validate and adjust the model by removing the variable which is not in this top 20 and variable level is small portion(<5%).

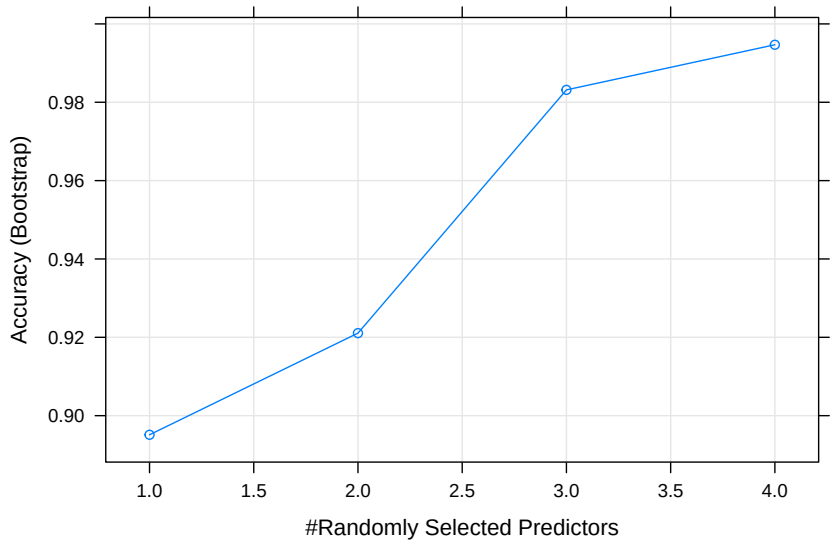
```
varImp(rf2_cv)
```

```
## rf variable importance
##
##   only 20 most important variables shown (out of 80)
##
##                            Overall
## odorn                       100.00
## gill_sizen                   55.53
## odorf                        51.90
## stalk_surface_above_ringk    42.20
## ring_typep                   40.42
## spore_print_colorh           34.88
## stalk_surface_below_ringk    33.74
## bruisest                     33.01
## stalk_surface_above_rings    27.27
## populationv                  26.50
## gill_spacingw                24.52
## spore_print_colorw           20.08
## ring_typel                   17.52
## spore_print_colorn           14.24
## stalk_surface_below_rings    14.10
## odorp                        14.00
## spore_print_colork           13.83
## stalk_shapet                 13.22
## habitatg                     10.14
## stalk_color_below_ringw       8.43
```

From variable importance, variables which are in these top 20 are bruises, gill spacing, gill size, stalk shape, population, habitat, odor, stalk surface above ring, stalk surface below ring, stalk color below ring, ring type and spore print color but odor variable will be removed because odorp portion is less than 5%.

Let's try to fit model again with this adjusted variables.

```
set.seed(1, sample.kind = "Rounding")
train_set_rf3 <- train_set %>% select(class,bruises,gill_spacing,gill_size,stalk_shape,
                                      population,habitat,stalk_surface_above_ring,
                                      stalk_surface_below_ring,stalk_color_below_ring,
                                      ring_type,spore_print_color)
rf3_cv <- train(class~.,
                method="rf",
                data = train_set_rf3,
                tuneGrid = data.frame(mtry = seq(1,4,1)))
plot(rf3_cv)
```



This plot show result of mtry tuning from 1 to 4. The best performance comes from mtry = 4 with about 99.4% accuracy. Full result of the model is in Appendix B.

```
varImp(rf3_cv)$importance %>% arrange(desc(Overall)) %>% head(10)
```

```
##                              Overall
## gill_sizen                 100.00000
## spore_print_colorh          58.37807
## stalk_surface_above_ringk   53.24895
## stalk_surface_below_ringk   52.65014
## ring_typep                  52.27092
## populationv                 40.16129
## bruisest                    37.39794
## gill_spacingw               32.77003
## stalk_surface_above_rings   30.69299
## spore_print_colorw          28.90286
```

Let's check variable importance again. This time I checked only top 10 because less variables applied to the model. From top 10 variable importance, there is no small portion level(<5%) so this model is acceptable. Then, apply this model to test set and see the performance.

```
predict_rf3_cv<- predict(rf3_cv,test_set)
confusionMatrix(predict_rf3_cv,test_set$class)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   e   p
##          e 829   0
##          p  13 784
##
##                Accuracy : 0.992
##                  95% CI : (0.9864, 0.9957)
##     No Information Rate : 0.5178
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.984
##
##  Mcnemar's Test P-Value : 0.0008741
##
##             Sensitivity : 0.9846
##             Specificity : 1.0000
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 0.9837
##              Prevalence : 0.5178
##          Detection Rate : 0.5098
##    Detection Prevalence : 0.5098
##       Balanced Accuracy : 0.9923
##
##        'Positive' Class : e
##
```

Method	Accuracy	Sensitivity	Specificity
GLM(6 features)	0.9384994	0.9596200	0.9158163
Random Forest(6 features)	0.9409594	0.9358670	0.9464286
Random Forest with mtry tuning(6 features)	0.9600246	0.9548694	0.9655612
Random Forest with mtry tuning(11 features)	0.9920049	0.9845606	1.0000000

The result of random forest method and mtry tuning with adjusted second group variables is 99.2% accuracy, 98.5% sensitivity and 100% specificity. From the result, this is the best model. Next, XgBoost method will be applied and we will see it performance compare to previous methods.

XgBoost Let's start applied XgBoost model with first group variables and compare the result with GLM and random forest.

```
set.seed(1, sample.kind = "Rounding")
train_set_xg1 <- train_set %>%
  select(class,bruises,gill_spacing,gill_size,stalk_shape,population,habitat)
xg1 <- train(class~.,
             method="xgbTree",
             data = train_set_xg1)
```

The final values used for the model were nrounds = 50, max_depth = 3, eta = 0.4, gamma = 0, colsample_bytree = 0.6, min_child_weight = 1 and subsample = 0.75. Full result of the model is in Appendix B. Next, validate the model performance with test set.

```
predict_xg1 <- predict(xg1,test_set)
confusionMatrix(predict_xg1,test_set$class)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   e   p
##          e 834   0
##          p   8 784
##
##                Accuracy : 0.9951
##                  95% CI : (0.9903, 0.9979)
##     No Information Rate : 0.5178
##     P-Value [Acc > NIR] : < 2e-16
##
##                   Kappa : 0.9902
##
##  Mcnemar's Test P-Value : 0.01333
##
##             Sensitivity : 0.9905
##             Specificity : 1.0000
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 0.9899
##              Prevalence : 0.5178
##          Detection Rate : 0.5129
##    Detection Prevalence : 0.5129
##       Balanced Accuracy : 0.9952
##
##        'Positive' Class : e
##
```

Method	Accuracy	Sensitivity	Specificity
GLM(6 features)	0.9384994	0.9596200	0.9158163
Random Forest(6 features)	0.9409594	0.9358670	0.9464286
Random Forest with mtry tuning(6 features)	0.9600246	0.9548694	0.9655612
Random Forest with mtry tuning(11 features)	0.9920049	0.9845606	1.0000000
XgBoost(6 features)	0.9950800	0.9904988	1.0000000

The result of XgBoost method with first group variables is 99.5% accuracy, 99.0% sensitivity and 100 % specificity. Compare to previous GLM method and random forest, XgBoost performance is better for all accuracy, sensitivity and specificity.

Next, second group variable will be applied.

```
set.seed(1, sample.kind = "Rounding")
train_set_xg2 <- train_set %>% select(class,bruises,gill_spacing,gill_size,stalk_shape,
                                      population,habitat,cap_surface,cap_color,odor,
                                      gill_color,stalk_surface_above_ring,stalk_surface_below_ring,
                                      stalk_color_above_ring,stalk_color_below_ring,
                                      ring_type,spore_print_color)
xg2 <- train(class~.,
             method="xgbTree",
             data = train_set_xg2)
```

This model gave us 100% accuracy with the final values used for the model were nrounds = 50, max_depth = 3, eta = 0.3, gamma = 0, colsample_bytree = 0.8, min_child_weight = 1 and subsample = 1 which you can see full result of the model in Appendix B, but this model may lead to overfitting because we including variable which have less portion level(<1%). Top 20 variable importance is used to validate and adjust the model by removing the variable which is not in this top 20 and variable level is small portion(<5%).

```
varImp(xg2)
```

```
## xgbTree variable importance
##
##   only 20 most important variables shown (out of 80)
##
##                             Overall
## odorn                      100.0000
## odorf                       32.4828
## bruisest                    24.0499
## gill_sizen                  20.3541
## spore_print_colorr           9.3421
## odorp                        8.7373
## odorl                        8.3729
## spore_print_colorw           5.0892
## habitatm                     3.9360
## stalk_shapet                 2.7895
## stalk_surface_below_ringy    2.7643
## gill_spacingw                1.7381
## ring_typep                   1.4113
## stalk_surface_above_ringk    0.8312
## cap_colory                   0.7315
## habitatu                     0.5296
## spore_print_colorh           0.3814
## odorc                        0.3759
## spore_print_coloru           0.3350
## populationc                  0.3329
```

From variable importance, variables which are in these top 20 are bruises, gill spacing, gill size, population, habitat,stalk shape, cap color, odor, stalk surface above ring, stalk surface below ring, spore print color and ring type but odor, spore print color, population, habitat and stalk surface below ring will be removed because some level portion of these variables in top 20 is less than 5% such as odorp and odorc.

Let's try to fit model again with this adjusted variables.

```
set.seed(1, sample.kind = "Rounding")
train_set_xg3 <- train_set %>% select(class,bruises,gill_spacing,gill_size,stalk_shape,
                                      cap_color,stalk_surface_above_ring,ring_type)
xg3 <- train(class~.,
             method="xgbTree",
             data = train_set_xg3)
```

The final values used for the model were nrounds = 100, max_depth = 3, eta = 0.4, gamma = 0, colsample_bytree = 0.6, min_child_weight = 1 and subsample = 0.75. Full result of the model is in Appendix B. Next, validate the model performance with test set.

```
predict_xg3 <- predict(xg3,test_set)
confusionMatrix(predict_xg3,test_set$class)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   e   p
##          e 841  14
##          p   1 770
##
##                Accuracy : 0.9908
##                  95% CI : (0.9848, 0.9948)
##     No Information Rate : 0.5178
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9815
##
##  Mcnemar's Test P-Value : 0.001946
##
##             Sensitivity : 0.9988
##             Specificity : 0.9821
##          Pos Pred Value : 0.9836
##          Neg Pred Value : 0.9987
##              Prevalence : 0.5178
##          Detection Rate : 0.5172
##    Detection Prevalence : 0.5258
##       Balanced Accuracy : 0.9905
##
##        'Positive' Class : e
##
```

Method	Accuracy	Sensitivity	Specificity
GLM(6 features)	0.9384994	0.9596200	0.9158163
Random Forest(6 features)	0.9409594	0.9358670	0.9464286
Random Forest with mtry tuning(6 features)	0.9600246	0.9548694	0.9655612
Random Forest with mtry tuning(11 features)	0.9920049	0.9845606	1.0000000
XgBoost(6 features)	0.9950800	0.9904988	1.0000000
XgBoost(7 features)	0.9907749	0.9988124	0.9821429

The result of XgBoost method with adjusted second group variables is 99.1% accuracy, 99.9% sensitivity and 98.2% specificity. You can see this model provide the highest sensitivity but our goal is to maximize specificity so this model is not the best.

Result

Method	Accuracy	Sensitivity	Specificity
GLM(6 features)	0.9384994	0.9596200	0.9158163
Random Forest(6 features)	0.9409594	0.9358670	0.9464286
Random Forest with mtry tuning(6 features)	0.9600246	0.9548694	0.9655612
Random Forest with mtry tuning(11 features)	0.9920049	0.9845606	1.0000000
XgBoost(6 features)	0.9950800	0.9904988	1.0000000
XgBoost(7 features)	0.9907749	0.9988124	0.9821429

From the goal of this project, to develop model for mushroom class prediction which maximize poisonous mushroom prediction accuracy means maximize specificity, Random Forest with mtry tuning(11 features) and XgBoost(6 features) gave the best result with 100% specificity but XgBoost(6 features) have higher accuracy and sensitivity so the best model is XgBoost(6 features).

Conclusion

From the result, random forest model performance improve when number of features are increased. When adding features which some levels are significant to mushroom class even though others level have less significant or less proportion but by building many trees and applied voting concept results performance improvement.

XgBoost is opposite compared to random forest. While increasing features, its performance doesn't improve. By boosting concept when you add feature which some levels are significant to mushroom class but others level have less significant, this less significant affect the model performance. In the other hand, using only features which significant to mushroom class and have lower noise; in this case from less significant levels results better model performance. So the final model to predict mushroom class is XgBoost with 6 features(First group variables); bruises, gill spacing, gill size, stalk shape, population and habitat.

In this mushroom dataset, there are many variables which have significant levels to mushroom class but others level are less portion such as ring type so to minimize overfitting problem, these variables are not used in the model. In the future, if this dataset is updated, it is challenging to revisit and improve the model to have better performance.

Appendix A

Definition of each variable in mushroom dataset

Class : e=edible, p=poisonous

Cap shape : b=bell, c=conical, x=convex, f=flat, k=knobbed, s=sunken

Cap surface : f=fibrous, g=grooves, y=scaly, s=smooth

Cap color : n=brown, b=buff, c=cinnamon, g=gray, r=green, p=pink, u=purple, e=red, w=white, y=yellow

Bruises : t=bruises, f=no

Odor : a=almond, l=anise, c=creosote, y=fishy, f=foul, m=musty, n=none, p=pungent, s=spicy

Gill attachment : a=attached, f=free

Gill spacing : c=close, w=crowded

Gill size : b=broad, n=narrow

Gill color : k=blank, n=brown, b=buff, h=chocolate, g=gray, r=green, o=orange, p=pink, u=purple, e=red, w=white, y=yellow

Stalk shape : e=enlarging, t=tapering

Stalk root : b=bulbous, c=club, u=cup, e=equal, z=rhizomorphs, r=rooted, ?=missing

Stalk surface above ring : f=fibrous, y=scaly, k=silky, s=smooth

Stalk surface below ring : f=fibrous, y=scaly, k=silky, s=smooth

Stalk color above ring : n=brown, b=buff, c=cinnamon, g=gray, o=orange, p=pink, e=red, w=white, y=yellow

Stalk color below ring : n=brown, b=buff, c=cinnamon, g=gray, o=orange, p=pink, e=red, w=white, y=yellow

Veil type : p=partial, u=universal

Veil color : n=brown, o=orange, w=white, y=yellow

Ring number : n=none, o=one, t=two

Ring type : c=cobwebby, e=evanescent, f=flaring, l=large, n=none, p=pendent, s=sheathing, z=zone

Spore print color : k=black, n=brown, b=buff, h=chocolate, r=green, o=orange, u=purple, w=white, y=yellow

Population : a=abundant, c=clustered, n=numerous, s=scattered, v=several, y=solitary

Habitat : g=grasses, l=leaves, m=meadows, paths=p, u=urban, w=waste, d=woods

Appendix B

Fit model result

GLM(6 features)

```
glm1
```

```
## Generalized Linear Model
##
## 6498 samples
##    6 predictor
##    2 classes: 'e', 'p'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 6498, 6498, 6498, 6498, 6498, 6498, ...
## Resampling results:
##
##   Accuracy   Kappa
##   0.9400649  0.8797092
```

Random Forest(6 features)

```
rf1
```

```
## Random Forest
##
## 6498 samples
##    6 predictor
##    2 classes: 'e', 'p'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 6498, 6498, 6498, 6498, 6498, 6498, ...
## Resampling results:
##
##   Accuracy   Kappa
##   0.9393405  0.8785122
##
## Tuning parameter 'mtry' was held constant at a value of 2
```

Random Forest with mtry tuning(6 features)

```
rf1_cv
```

```
## Random Forest
##
```

```
## 6498 samples
##    6 predictor
##    2 classes: 'e', 'p'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 6498, 6498, 6498, 6498, 6498, 6498, ...
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa
##   1     0.8832255  0.7653353
##   2     0.9380087  0.8758432
##   3     0.9629144  0.9257811
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 3.
```

Random Forest with mtry tuning(Second group variables)

```
rf2_cv
```

```
## Random Forest
##
## 6498 samples
##   16 predictor
##    2 classes: 'e', 'p'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 6498, 6498, 6498, 6498, 6498, 6498, ...
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa
##   1     0.9089750  0.8163102
##   2     0.9575816  0.9147177
##   3     0.9933537  0.9866751
##   4     0.9997823  0.9995640
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 4.
```

Random Forest with mtry tuning(11 features)

```
rf3_cv
```

```
## Random Forest
##
## 6498 samples
##   11 predictor
##    2 classes: 'e', 'p'
```

```
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 6498, 6498, 6498, 6498, 6498, 6498, ...
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa
##   1     0.8951192  0.7883584
##   2     0.9210716  0.8410676
##   3     0.9831590  0.9662382
##   4     0.9946765  0.9893374
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 4.
```

XgBoost(6 features)

```
xg1
```

```
## eXtreme Gradient Boosting
##
## 6498 samples
##    6 predictor
##    2 classes: 'e', 'p'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 6498, 6498, 6498, 6498, 6498, 6498, ...
## Resampling results across tuning parameters:
##
##   eta  max_depth  colsample_bytree  subsample  nrounds  Accuracy   Kappa
##   0.3  1          0.6               0.50        50      0.9154092  0.8301907
##   0.3  1          0.6               0.50       100      0.9226455  0.8446738
##   0.3  1          0.6               0.50       150      0.9308859  0.8611963
##   0.3  1          0.6               0.75        50      0.9121527  0.8236355
##   0.3  1          0.6               0.75       100      0.9226079  0.8445964
##   0.3  1          0.6               0.75       150      0.9268593  0.8531035
##   0.3  1          0.6               1.00        50      0.9104485  0.8201909
##   0.3  1          0.6               1.00       100      0.9190994  0.8375537
##   0.3  1          0.6               1.00       150      0.9255675  0.8505021
##   0.3  1          0.8               0.50        50      0.9125960  0.8245373
##   0.3  1          0.8               0.50       100      0.9250363  0.8494834
##   0.3  1          0.8               0.50       150      0.9319702  0.8633798
##   0.3  1          0.8               0.75        50      0.9125824  0.8244878
##   0.3  1          0.8               0.75       100      0.9238242  0.8470146
##   0.3  1          0.8               0.75       150      0.9295790  0.8585665
##   0.3  1          0.8               1.00        50      0.9104753  0.8202451
##   0.3  1          0.8               1.00       100      0.9190284  0.8374025
##   0.3  1          0.8               1.00       150      0.9259377  0.8512339
##   0.3  2          0.6               0.50        50      0.9806345  0.9612023
##   0.3  2          0.6               0.50       100      0.9927409  0.9854627
##   0.3  2          0.6               0.50       150      0.9942884  0.9885610
```

```
##   0.3  2          0.6              0.75         50       0.9806655  0.9612626
##   0.3  2          0.6              0.75        100       0.9921869  0.9843545
##   0.3  2          0.6              0.75        150       0.9943843  0.9887528
##   0.3  2          0.6              1.00         50       0.9822877  0.9645101
##   0.3  2          0.6              1.00        100       0.9922524  0.9844853
##   0.3  2          0.6              1.00        150       0.9944677  0.9889199
##   0.3  2          0.8              0.50         50       0.9826330  0.9652061
##   0.3  2          0.8              0.50        100       0.9934902  0.9869626
##   0.3  2          0.8              0.50        150       0.9945509  0.9890864
##   0.3  2          0.8              0.75         50       0.9830946  0.9661283
##   0.3  2          0.8              0.75        100       0.9933987  0.9867805
##   0.3  2          0.8              0.75        150       0.9945678  0.9891202
##   0.3  2          0.8              1.00         50       0.9825484  0.9650312
##   0.3  2          0.8              1.00        100       0.9930848  0.9861507
##   0.3  2          0.8              1.00        150       0.9944677  0.9889199
##   0.3  3          0.6              0.50         50       0.9938543  0.9876918
##   0.3  3          0.6              0.50        100       0.9946344  0.9892536
##   0.3  3          0.6              0.50        150       0.9946344  0.9892536
##   0.3  3          0.6              0.75         50       0.9940047  0.9879929
##   0.3  3          0.6              0.75        100       0.9946344  0.9892536
##   0.3  3          0.6              0.75        150       0.9946344  0.9892536
##   0.3  3          0.6              1.00         50       0.9940363  0.9880559
##   0.3  3          0.6              1.00        100       0.9946344  0.9892536
##   0.3  3          0.6              1.00        150       0.9946344  0.9892536
##   0.3  3          0.8              0.50         50       0.9944356  0.9888552
##   0.3  3          0.8              0.50        100       0.9946344  0.9892536
##   0.3  3          0.8              0.50        150       0.9946344  0.9892536
##   0.3  3          0.8              0.75         50       0.9945342  0.9890528
##   0.3  3          0.8              0.75        100       0.9946344  0.9892536
##   0.3  3          0.8              0.75        150       0.9946344  0.9892536
##   0.3  3          0.8              1.00         50       0.9946005  0.9891857
##   0.3  3          0.8              1.00        100       0.9946344  0.9892536
##   0.3  3          0.8              1.00        150       0.9946344  0.9892536
##   0.4  1          0.6              0.50         50       0.9162415  0.8318611
##   0.4  1          0.6              0.50        100       0.9291229  0.8576693
##   0.4  1          0.6              0.50        150       0.9355000  0.8704953
##   0.4  1          0.6              0.75         50       0.9151287  0.8296386
##   0.4  1          0.6              0.75        100       0.9272576  0.8538984
##   0.4  1          0.6              0.75        150       0.9345192  0.8685150
##   0.4  1          0.6              1.00         50       0.9136990  0.8266985
##   0.4  1          0.6              1.00        100       0.9245317  0.8484281
##   0.4  1          0.6              1.00        150       0.9293348  0.8580595
##   0.4  1          0.8              0.50         50       0.9191350  0.8376704
##   0.4  1          0.8              0.50        100       0.9296622  0.8587439
##   0.4  1          0.8              0.50        150       0.9374550  0.8744323
##   0.4  1          0.8              0.75         50       0.9152473  0.8298415
##   0.4  1          0.8              0.75        100       0.9266179  0.8525978
##   0.4  1          0.8              0.75        150       0.9348758  0.8692371
##   0.4  1          0.8              1.00         50       0.9139715  0.8272709
##   0.4  1          0.8              1.00        100       0.9247653  0.8488803
##   0.4  1          0.8              1.00        150       0.9292347  0.8578555
##   0.4  2          0.6              0.50         50       0.9887639  0.9774997
##   0.4  2          0.6              0.50        100       0.9935531  0.9870881
##   0.4  2          0.6              0.50        150       0.9945678  0.9891202
```

```
##   0.4  2          0.6              0.75        50      0.9900644  0.9801007
##   0.4  2          0.6              0.75       100      0.9942537  0.9884917
##   0.4  2          0.6              0.75       150      0.9946344  0.9892536
##   0.4  2          0.6              1.00        50      0.9892134  0.9783957
##   0.4  2          0.6              1.00       100      0.9940851  0.9881544
##   0.4  2          0.6              1.00       150      0.9946344  0.9892536
##   0.4  2          0.8              0.50        50      0.9892320  0.9784345
##   0.4  2          0.8              0.50       100      0.9945192  0.9890228
##   0.4  2          0.8              0.50       150      0.9946344  0.9892536
##   0.4  2          0.8              0.75        50      0.9898032  0.9795802
##   0.4  2          0.8              0.75       100      0.9945845  0.9891537
##   0.4  2          0.8              0.75       150      0.9946344  0.9892536
##   0.4  2          0.8              1.00        50      0.9903557  0.9806850
##   0.4  2          0.8              1.00       100      0.9943852  0.9887547
##   0.4  2          0.8              1.00       150      0.9946344  0.9892536
##   0.4  3          0.6              0.50        50      0.9944849  0.9889541
##   0.4  3          0.6              0.50       100      0.9945678  0.9891202
##   0.4  3          0.6              0.50       150      0.9946344  0.9892536
##   0.4  3          0.6              0.75        50      0.9946344  0.9892536
##   0.4  3          0.6              0.75       100      0.9946344  0.9892536
##   0.4  3          0.6              0.75       150      0.9946344  0.9892536
##   0.4  3          0.6              1.00        50      0.9945678  0.9891202
##   0.4  3          0.6              1.00       100      0.9946344  0.9892536
##   0.4  3          0.6              1.00       150      0.9946344  0.9892536
##   0.4  3          0.8              0.50        50      0.9946005  0.9891857
##   0.4  3          0.8              0.50       100      0.9946344  0.9892536
##   0.4  3          0.8              0.50       150      0.9946344  0.9892536
##   0.4  3          0.8              0.75        50      0.9946344  0.9892536
##   0.4  3          0.8              0.75       100      0.9946344  0.9892536
##   0.4  3          0.8              0.75       150      0.9946344  0.9892536
##   0.4  3          0.8              1.00        50      0.9946344  0.9892536
##   0.4  3          0.8              1.00       100      0.9946344  0.9892536
##   0.4  3          0.8              1.00       150      0.9946344  0.9892536
##
## Tuning parameter 'gamma' was held constant at a value of 0
## Tuning
##  parameter 'min_child_weight' was held constant at a value of 1
## Accuracy was used to select the optimal model using the largest value.
## The final values used for the model were nrounds = 50, max_depth = 3, eta
##  = 0.4, gamma = 0, colsample_bytree = 0.6, min_child_weight = 1 and subsample
##  = 0.75.
```

XgBoost(Second group variables)

```
xg2
```

```
## eXtreme Gradient Boosting
##
## 6498 samples
##   16 predictor
##    2 classes: 'e', 'p'
##
```

```
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 6498, 6498, 6498, 6498, 6498, 6498, ...
## Resampling results across tuning parameters:
##
##   eta  max_depth  colsample_bytree  subsample  nrounds  Accuracy   Kappa
##   0.3  1          0.6               0.50        50      0.9948860  0.9897533
##   0.3  1          0.6               0.50       100      0.9986030  0.9972002
##   0.3  1          0.6               0.50       150      0.9992172  0.9984311
##   0.3  1          0.6               0.75        50      0.9957437  0.9914707
##   0.3  1          0.6               0.75       100      0.9987210  0.9974365
##   0.3  1          0.6               0.75       150      0.9992840  0.9985651
##   0.3  1          0.6               1.00        50      0.9951223  0.9902275
##   0.3  1          0.6               1.00       100      0.9985027  0.9969991
##   0.3  1          0.6               1.00       150      0.9992172  0.9984311
##   0.3  1          0.8               0.50        50      0.9953934  0.9907691
##   0.3  1          0.8               0.50       100      0.9986210  0.9972359
##   0.3  1          0.8               0.50       150      0.9991839  0.9983643
##   0.3  1          0.8               0.75        50      0.9953798  0.9907432
##   0.3  1          0.8               0.75       100      0.9985196  0.9970328
##   0.3  1          0.8               0.75       150      0.9992840  0.9985651
##   0.3  1          0.8               1.00        50      0.9950915  0.9901653
##   0.3  1          0.8               1.00       100      0.9985193  0.9970325
##   0.3  1          0.8               1.00       150      0.9992840  0.9985651
##   0.3  2          0.6               0.50        50      0.9992186  0.9984335
##   0.3  2          0.6               0.50       100      0.9998998  0.9997992
##   0.3  2          0.6               0.50       150      0.9999500  0.9998999
##   0.3  2          0.6               0.75        50      0.9993341  0.9986655
##   0.3  2          0.6               0.75       100      1.0000000  1.0000000
##   0.3  2          0.6               0.75       150      1.0000000  1.0000000
##   0.3  2          0.6               1.00        50      0.9993674  0.9987321
##   0.3  2          0.6               1.00       100      1.0000000  1.0000000
##   0.3  2          0.6               1.00       150      1.0000000  1.0000000
##   0.3  2          0.8               0.50        50      0.9994009  0.9987992
##   0.3  2          0.8               0.50       100      0.9999334  0.9998666
##   0.3  2          0.8               0.50       150      0.9999500  0.9998999
##   0.3  2          0.8               0.75        50      0.9994511  0.9988999
##   0.3  2          0.8               0.75       100      1.0000000  1.0000000
##   0.3  2          0.8               0.75       150      1.0000000  1.0000000
##   0.3  2          0.8               1.00        50      0.9993509  0.9986992
##   0.3  2          0.8               1.00       100      1.0000000  1.0000000
##   0.3  2          0.8               1.00       150      1.0000000  1.0000000
##   0.3  3          0.6               0.50        50      0.9998662  0.9997319
##   0.3  3          0.6               0.50       100      0.9999500  0.9998999
##   0.3  3          0.6               0.50       150      0.9999500  0.9998999
##   0.3  3          0.6               0.75        50      0.9999334  0.9998666
##   0.3  3          0.6               0.75       100      1.0000000  1.0000000
##   0.3  3          0.6               0.75       150      1.0000000  1.0000000
##   0.3  3          0.6               1.00        50      0.9998998  0.9997992
##   0.3  3          0.6               1.00       100      1.0000000  1.0000000
##   0.3  3          0.6               1.00       150      1.0000000  1.0000000
##   0.3  3          0.8               0.50        50      0.9998995  0.9997987
##   0.3  3          0.8               0.50       100      0.9999500  0.9998999
##   0.3  3          0.8               0.50       150      0.9999500  0.9998999
```

```
##   0.3  3          0.8              0.75         50       0.9999164  0.9998326
##   0.3  3          0.8              0.75        100       0.9999500  0.9998999
##   0.3  3          0.8              0.75        150       0.9999500  0.9998999
##   0.3  3          0.8              1.00         50       1.0000000  1.0000000
##   0.3  3          0.8              1.00        100       1.0000000  1.0000000
##   0.3  3          0.8              1.00        150       1.0000000  1.0000000
##   0.4  1          0.6              0.50         50       0.9979538  0.9958983
##   0.4  1          0.6              0.50        100       0.9991014  0.9981988
##   0.4  1          0.6              0.50        150       0.9992840  0.9985651
##   0.4  1          0.6              0.75         50       0.9977212  0.9954326
##   0.4  1          0.6              0.75        100       0.9992507  0.9984983
##   0.4  1          0.6              0.75        150       0.9993343  0.9986658
##   0.4  1          0.6              1.00         50       0.9978232  0.9956366
##   0.4  1          0.6              1.00        100       0.9991347  0.9982656
##   0.4  1          0.6              1.00        150       0.9993174  0.9986321
##   0.4  1          0.8              0.50         50       0.9979887  0.9959685
##   0.4  1          0.8              0.50        100       0.9990193  0.9980342
##   0.4  1          0.8              0.50        150       0.9993174  0.9986321
##   0.4  1          0.8              0.75         50       0.9978021  0.9955942
##   0.4  1          0.8              0.75        100       0.9991685  0.9983337
##   0.4  1          0.8              0.75        150       0.9993174  0.9986321
##   0.4  1          0.8              1.00         50       0.9979714  0.9959336
##   0.4  1          0.8              1.00        100       0.9991839  0.9983643
##   0.4  1          0.8              1.00        150       0.9993174  0.9986321
##   0.4  2          0.6              0.50         50       0.9995830  0.9991644
##   0.4  2          0.6              0.50        100       0.9999500  0.9998999
##   0.4  2          0.6              0.50        150       0.9999500  0.9998999
##   0.4  2          0.6              0.75         50       0.9997674  0.9995341
##   0.4  2          0.6              0.75        100       1.0000000  1.0000000
##   0.4  2          0.6              0.75        150       1.0000000  1.0000000
##   0.4  2          0.6              1.00         50       0.9997338  0.9994667
##   0.4  2          0.6              1.00        100       1.0000000  1.0000000
##   0.4  2          0.6              1.00        150       1.0000000  1.0000000
##   0.4  2          0.8              0.50         50       0.9996500  0.9992989
##   0.4  2          0.8              0.50        100       1.0000000  1.0000000
##   0.4  2          0.8              0.50        150       1.0000000  1.0000000
##   0.4  2          0.8              0.75         50       0.9997661  0.9995313
##   0.4  2          0.8              0.75        100       1.0000000  1.0000000
##   0.4  2          0.8              0.75        150       1.0000000  1.0000000
##   0.4  2          0.8              1.00         50       0.9999334  0.9998666
##   0.4  2          0.8              1.00        100       1.0000000  1.0000000
##   0.4  2          0.8              1.00        150       1.0000000  1.0000000
##   0.4  3          0.6              0.50         50       0.9999500  0.9998999
##   0.4  3          0.6              0.50        100       0.9999500  0.9998999
##   0.4  3          0.6              0.50        150       0.9999500  0.9998999
##   0.4  3          0.6              0.75         50       1.0000000  1.0000000
##   0.4  3          0.6              0.75        100       1.0000000  1.0000000
##   0.4  3          0.6              0.75        150       1.0000000  1.0000000
##   0.4  3          0.6              1.00         50       1.0000000  1.0000000
##   0.4  3          0.6              1.00        100       1.0000000  1.0000000
##   0.4  3          0.6              1.00        150       1.0000000  1.0000000
##   0.4  3          0.8              0.50         50       0.9999334  0.9998666
##   0.4  3          0.8              0.50        100       0.9999500  0.9998999
##   0.4  3          0.8              0.50        150       0.9999500  0.9998999
```

```
##   0.4  3          0.8              0.75        50      0.9999500  0.9998999
##   0.4  3          0.8              0.75       100      1.0000000  1.0000000
##   0.4  3          0.8              0.75       150      0.9999500  0.9998999
##   0.4  3          0.8              1.00        50      1.0000000  1.0000000
##   0.4  3          0.8              1.00       100      1.0000000  1.0000000
##   0.4  3          0.8              1.00       150      1.0000000  1.0000000
##
## Tuning parameter 'gamma' was held constant at a value of 0
## Tuning
##  parameter 'min_child_weight' was held constant at a value of 1
## Accuracy was used to select the optimal model using the largest value.
## The final values used for the model were nrounds = 50, max_depth = 3, eta
##  = 0.3, gamma = 0, colsample_bytree = 0.8, min_child_weight = 1 and subsample
##  = 1.
```

XgBoost(7 features)

```
xg3
```

```
## eXtreme Gradient Boosting
##
## 6498 samples
##    7 predictor
##    2 classes: 'e', 'p'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 6498, 6498, 6498, 6498, 6498, 6498, ...
## Resampling results across tuning parameters:
##
##   eta  max_depth  colsample_bytree  subsample  nrounds  Accuracy   Kappa
##   0.3  1          0.6               0.50        50      0.9345629  0.8684867
##   0.3  1          0.6               0.50       100      0.9469503  0.8935069
##   0.3  1          0.6               0.50       150      0.9443046  0.8882437
##   0.3  1          0.6               0.75        50      0.9333791  0.8660889
##   0.3  1          0.6               0.75       100      0.9488131  0.8972404
##   0.3  1          0.6               0.75       150      0.9438028  0.8872309
##   0.3  1          0.6               1.00        50      0.9326810  0.8646965
##   0.3  1          0.6               1.00       100      0.9459147  0.8914014
##   0.3  1          0.6               1.00       150      0.9436282  0.8868614
##   0.3  1          0.8               0.50        50      0.9362788  0.8719654
##   0.3  1          0.8               0.50       100      0.9468928  0.8933897
##   0.3  1          0.8               0.50       150      0.9440125  0.8876609
##   0.3  1          0.8               0.75        50      0.9350643  0.8694972
##   0.3  1          0.8               0.75       100      0.9470903  0.8937828
##   0.3  1          0.8               0.75       150      0.9448542  0.8893342
##   0.3  1          0.8               1.00        50      0.9353095  0.8699865
##   0.3  1          0.8               1.00       100      0.9477313  0.8950595
##   0.3  1          0.8               1.00       150      0.9443715  0.8883587
##   0.3  2          0.6               0.50        50      0.9586956  0.9170969
##   0.3  2          0.6               0.50       100      0.9774643  0.9547918
##   0.3  2          0.6               0.50       150      0.9802593  0.9603994
```

```
##   0.3  2          0.6              0.75         50       0.9582617  0.9162282
##   0.3  2          0.6              0.75        100       0.9765307  0.9529242
##   0.3  2          0.6              0.75        150       0.9805590  0.9609997
##   0.3  2          0.6              1.00         50       0.9576056  0.9149104
##   0.3  2          0.6              1.00        100       0.9746893  0.9492404
##   0.3  2          0.6              1.00        150       0.9802438  0.9603714
##   0.3  2          0.8              0.50         50       0.9611471  0.9220350
##   0.3  2          0.8              0.50        100       0.9790375  0.9579422
##   0.3  2          0.8              0.50        150       0.9806267  0.9611362
##   0.3  2          0.8              0.75         50       0.9601751  0.9200721
##   0.3  2          0.8              0.75        100       0.9786004  0.9570725
##   0.3  2          0.8              0.75        150       0.9803413  0.9605597
##   0.3  2          0.8              1.00         50       0.9611273  0.9219825
##   0.3  2          0.8              1.00        100       0.9764433  0.9527513
##   0.3  2          0.8              1.00        150       0.9805102  0.9608990
##   0.3  3          0.6              0.50         50       0.9792800  0.9584306
##   0.3  3          0.6              0.50        100       0.9826923  0.9652793
##   0.3  3          0.6              0.50        150       0.9843056  0.9685192
##   0.3  3          0.6              0.75         50       0.9802453  0.9603699
##   0.3  3          0.6              0.75        100       0.9833393  0.9665788
##   0.3  3          0.6              0.75        150       0.9847395  0.9693896
##   0.3  3          0.6              1.00         50       0.9803932  0.9606666
##   0.3  3          0.6              1.00        100       0.9836039  0.9671106
##   0.3  3          0.6              1.00        150       0.9845881  0.9690872
##   0.3  3          0.8              0.50         50       0.9805093  0.9609012
##   0.3  3          0.8              0.50        100       0.9842383  0.9683844
##   0.3  3          0.8              0.50        150       0.9844386  0.9687886
##   0.3  3          0.8              0.75         50       0.9804765  0.9608324
##   0.3  3          0.8              0.75        100       0.9844533  0.9688170
##   0.3  3          0.8              0.75        150       0.9845881  0.9690872
##   0.3  3          0.8              1.00         50       0.9807074  0.9612978
##   0.3  3          0.8              1.00        100       0.9840569  0.9680229
##   0.3  3          0.8              1.00        150       0.9845881  0.9690872
##   0.4  1          0.6              0.50         50       0.9412015  0.8818935
##   0.4  1          0.6              0.50        100       0.9448274  0.8892995
##   0.4  1          0.6              0.50        150       0.9439320  0.8875153
##   0.4  1          0.6              0.75         50       0.9399790  0.8794224
##   0.4  1          0.6              0.75        100       0.9438798  0.8873782
##   0.4  1          0.6              0.75        150       0.9441300  0.8878986
##   0.4  1          0.6              1.00         50       0.9395907  0.8786302
##   0.4  1          0.6              1.00        100       0.9429247  0.8854376
##   0.4  1          0.6              1.00        150       0.9444666  0.8885620
##   0.4  1          0.8              0.50         50       0.9424267  0.8843601
##   0.4  1          0.8              0.50        100       0.9442037  0.8880382
##   0.4  1          0.8              0.50        150       0.9453302  0.8903193
##   0.4  1          0.8              0.75         50       0.9431825  0.8858774
##   0.4  1          0.8              0.75        100       0.9444198  0.8884663
##   0.4  1          0.8              0.75        150       0.9448610  0.8893699
##   0.4  1          0.8              1.00         50       0.9423900  0.8842856
##   0.4  1          0.8              1.00        100       0.9433055  0.8861983
##   0.4  1          0.8              1.00        150       0.9442841  0.8881978
##   0.4  2          0.6              0.50         50       0.9631024  0.9259700
##   0.4  2          0.6              0.50        100       0.9797632  0.9594052
##   0.4  2          0.6              0.50        150       0.9812600  0.9624064
```

```
##   0.4  2          0.6              0.75        50      0.9638066  0.9273814
##   0.4  2          0.6              0.75       100      0.9804231  0.9607271
##   0.4  2          0.6              0.75       150      0.9810546  0.9619912
##   0.4  2          0.6              1.00        50      0.9634960  0.9267464
##   0.4  2          0.6              1.00       100      0.9802927  0.9604665
##   0.4  2          0.6              1.00       150      0.9808580  0.9615975
##   0.4  2          0.8              0.50        50      0.9688478  0.9374995
##   0.4  2          0.8              0.50       100      0.9802754  0.9604315
##   0.4  2          0.8              0.50       150      0.9811734  0.9622352
##   0.4  2          0.8              0.75        50      0.9680953  0.9359805
##   0.4  2          0.8              0.75       100      0.9804560  0.9607902
##   0.4  2          0.8              0.75       150      0.9815375  0.9629613
##   0.4  2          0.8              1.00        50      0.9665717  0.9329241
##   0.4  2          0.8              1.00       100      0.9802606  0.9603971
##   0.4  2          0.8              1.00       150      0.9810063  0.9618965
##   0.4  3          0.6              0.50        50      0.9809759  0.9618332
##   0.4  3          0.6              0.50       100      0.9844380  0.9687867
##   0.4  3          0.6              0.50       150      0.9845881  0.9690872
##   0.4  3          0.6              0.75        50      0.9809739  0.9618288
##   0.4  3          0.6              0.75       100      0.9847395  0.9693896
##   0.4  3          0.6              0.75       150      0.9847395  0.9693896
##   0.4  3          0.6              1.00        50      0.9810388  0.9619626
##   0.4  3          0.6              1.00       100      0.9845881  0.9690872
##   0.4  3          0.6              1.00       150      0.9845881  0.9690872
##   0.4  3          0.8              0.50        50      0.9810572  0.9619981
##   0.4  3          0.8              0.50       100      0.9846229  0.9691571
##   0.4  3          0.8              0.50       150      0.9845900  0.9690910
##   0.4  3          0.8              0.75        50      0.9813728  0.9626326
##   0.4  3          0.8              0.75       100      0.9844549  0.9688195
##   0.4  3          0.8              0.75       150      0.9844386  0.9687886
##   0.4  3          0.8              1.00        50      0.9812065  0.9622987
##   0.4  3          0.8              1.00       100      0.9845881  0.9690872
##   0.4  3          0.8              1.00       150      0.9845881  0.9690872
##
## Tuning parameter 'gamma' was held constant at a value of 0
## Tuning
##  parameter 'min_child_weight' was held constant at a value of 1
## Accuracy was used to select the optimal model using the largest value.
## The final values used for the model were nrounds = 100, max_depth = 3, eta
##  = 0.4, gamma = 0, colsample_bytree = 0.6, min_child_weight = 1 and subsample
##  = 0.75.
```