

*Toronto*

**Module 2: Technique Practice**

Soni Manana (002952645)

a *College of Professional Studies****,*** *Master of Professional Studies in Analytics.*

***Subject****: ALY6040*

Under the guidance of

**Prof. Kamran Hootan**

**Introduction**

We have provided a dataset of mushrooms to develop a classifier that can discriminate poisonous mushrooms. Determine the distinctions between poisonous and edible mushrooms, then mark them as such using the mushroom's attributes such gill, caps, etc. The same dataset contains labels for each of the 23 distinct variables that describe the other features of mushrooms.

• In addition, we will implement the requirements that are determined to be the major contributors to a mushroom's toxicity.

• Separate the data into test and training sets.

• Create a decision tree that is more accurate at anticipating better results, and then apply the required penalty to generate better results.

**Data Cleaning and Processing**

Cleaning is crucial process for data but we need to observe our data first in order to get actual scenario. Now, looking at Fig1, all of the data is categorical which is obvious for classification problem. Str() function of R provides type of variables, number of rows and its initial values. We need to factorised the values of each column in order to develop a base classifier.

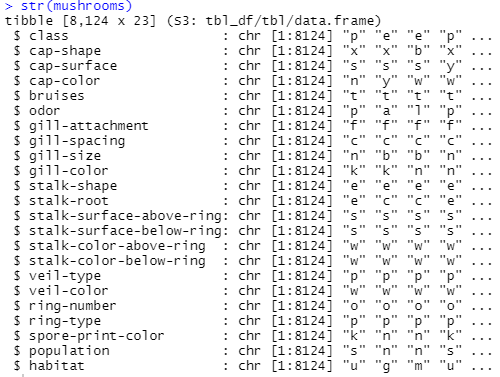
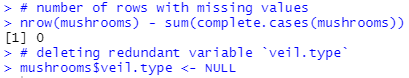
 

Fig.1 Data Description Fig.2 Cleaning Data

There are 0 null values in samples which allowing us to proceed to the following step and remove redundant data from the dataset's veil.type property. As a result, the dataset has been cleaned and is now ready for actual analysis.

**Initial Analysis**

Odour is crucial factor in determination of poisonous mushrooms. Our focus should be odour first.There are some samples having poisonous mushrooms such as having p are poisonous.

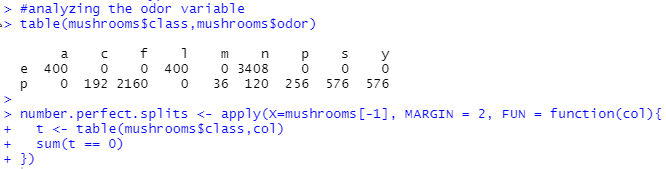
****

Fig.3 Odour Observation

As shown in the Fig.4, the features represent different features and splits are classes or attributes of mushrooms such as cap-shape, veil colour and etc.

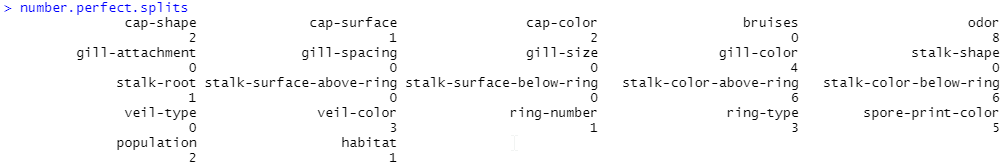


Fig. 5 Splits of mushroom

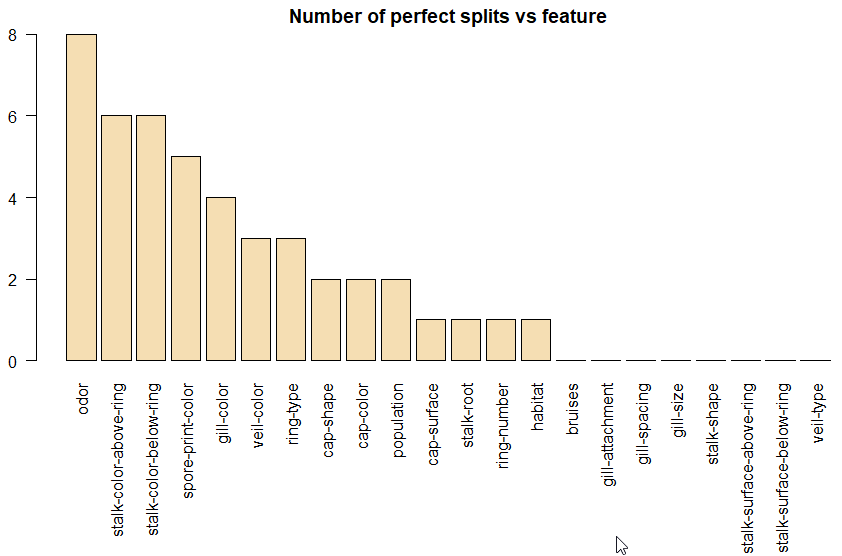


Fig.4 Number of Perfect Splits vs Features.

**Basic Classification Model**

In order to test the dataset with regard to other factors like odour, gill size, and gill shape, we need to separate the first column as it is our target variable for the primary focus, that column is class.

We must first divide the data into a training and a test dataframe in order to train our model.

To help the model evolve more effectively, we allow the threshold of 80% of the data for training. 20% of the data for testing.

While training a model it is necessary to provide proper weight distribution to the model in order to train and get the correct edible mushroom which should be main purpose of our model to classify.

To choose the right edible mushroom, which is what our model is really designed to predict, it is important to punish the model while it is being trained.

We will use Decision tree for classification which is considered as base model for other advance models like Random Forest, Gradient Boosting and Xtream Gradient Boosting (XgBoost). Showing in Fig.5 is our Decision tree which has depth of 4.

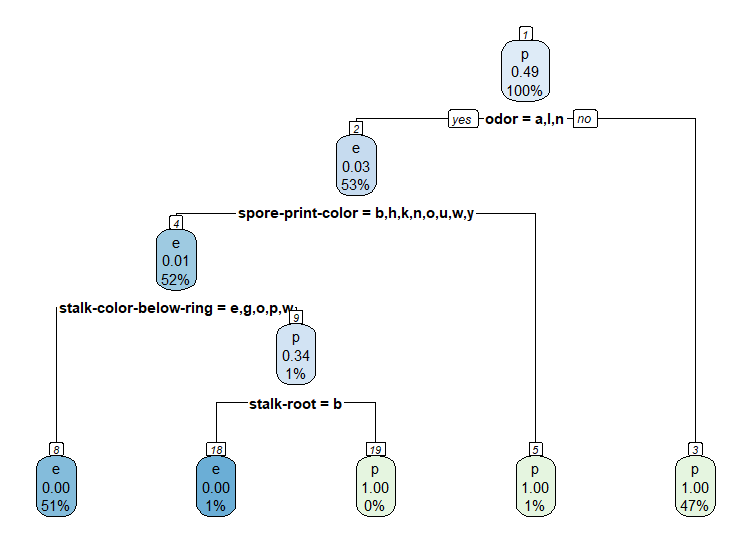


Fig. 5 Decision Tree

There are hardly any false classified according to confusion matrix. We got 100% accuracy for this data I believed that dataset is learning specific that’s why we have got this accuracy otherwise it can be Overfitting.

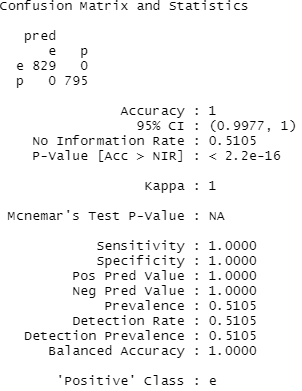
****

Fig.6 Performance of Decision Tree

**Conclusion**

This assignment is specific to learn the base model for classification which is decision tree. In addition, we learned that how to remove redundancy, create a cross tabulation, factorising, splitting between train and test split and prepare this model.

**Appendix**

#Installing libraries

install.packages('rpart')

install.packages('caret')

install.packages('rpart.plot')

install.packages('rattle')

install.packages('readxl')

#Loading libraries

library(rpart,quietly = TRUE)

library(caret,quietly = TRUE)

library(rpart.plot,quietly = TRUE)

library(rattle)

library(readxl)

#Reading the data set as a dataframe

mushrooms <- read\_excel("../../Mxnxn/Downloads/mushrooms.xlsx")

# structure of the data

str(mushrooms)

summary(mushrooms)

# number of rows with missing values

nrow(mushrooms) - sum(complete.cases(mushrooms))

# deleting redundant variable `veil.type`

mushrooms$veil.type <- NULL

#analyzing the odor variable

table(mushrooms$class,mushrooms$odor)

number.perfect.splits <- apply(X=mushrooms[-1], MARGIN = 2, FUN = function(col){

t <- table(mushrooms$class,col)

sum(t == 0)

})

number.perfect.splits

# Descending order of perfect splits

order <- order(number.perfect.splits,decreasing = TRUE)

number.perfect.splits <- number.perfect.splits[order]

# Plot graph

par(mar=c(10,2,2,2))

barplot(number.perfect.splits,

main="Number of perfect splits vs feature",

xlab="",ylab="Feature",las=2,col="wheat")

#data splicing

set.seed(12345)

train <- sample(1:nrow(mushrooms),size = ceiling(0.80\*nrow(mushrooms)),replace = FALSE)

# training set

mushrooms\_train <- mushrooms[train,]

# test set

mushrooms\_test <- mushrooms[-train,]

# penalty matrix

penalty.matrix <- matrix(c(0,1,10,0), byrow=TRUE, nrow=2)

# building the classification tree with rpart

tree <- rpart(class~.,

data=mushrooms\_train,

parms = list(loss = penalty.matrix),

method = "class")

# Visualize the decision tree with rpart.plot

rpart.plot(tree, nn=TRUE)

# choosing the best complexity parameter "cp" to prune the tree

cp.optim <- tree$cptable[which.min(tree$cptable[,"xerror"]),"CP"]

# tree prunning using the best complexity parameter. For more in

tree <- prune(tree, cp=cp.optim)

#Testing the model

pred <- predict(object=tree,mushrooms\_test[-1],type="class")

#Calculating accuracy

t <- table(mushrooms\_test$class,pred)

confusionMatrix(t)