**Project 3 Analysis**

**A. listing of all R functions that you have written**

**Library:**

library(rpart) # Popular decision tree algorithm

library(rattle) # Fancy tree plot

library(rpart.plot) # Enhanced tree plots

library(ROCR) # ROC plot and AUC score

library(c50)  # C5.0 decision trees and rule-based models for pattern recognition

library(randomForest) # random forest algorithm

**R function:**

Rpart

Prune

Prp

fancyRpartPlot

confusionmatrix

performance

randomForest

importance

varImpPlot

predict

C5.0

plot

**Dataset:**

We set seed to generate the same dataset every time for train (70%) and test (30%) data regard the randomness of train and test split.

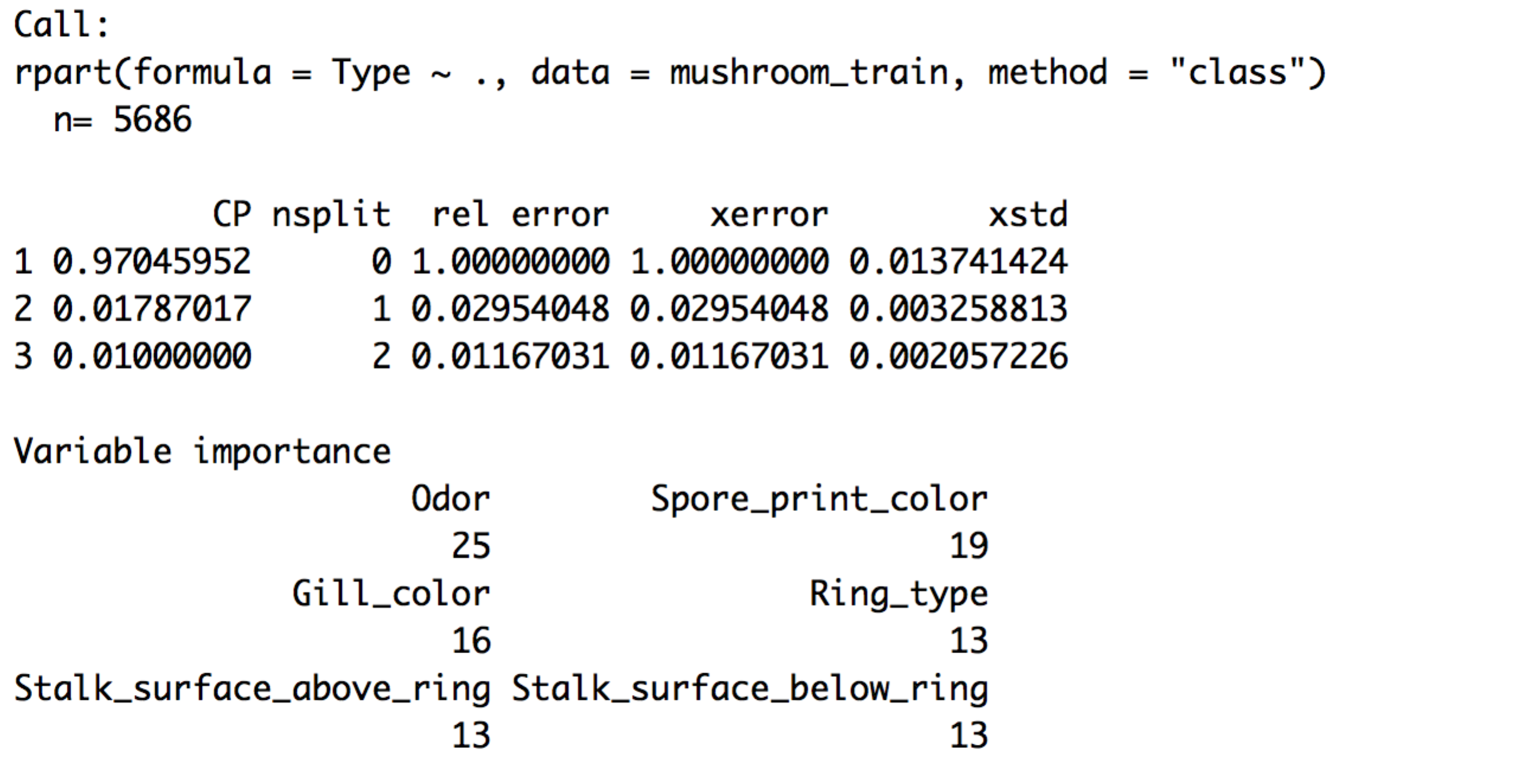
Result is showing below (Edible as 1. Poisson as 0):

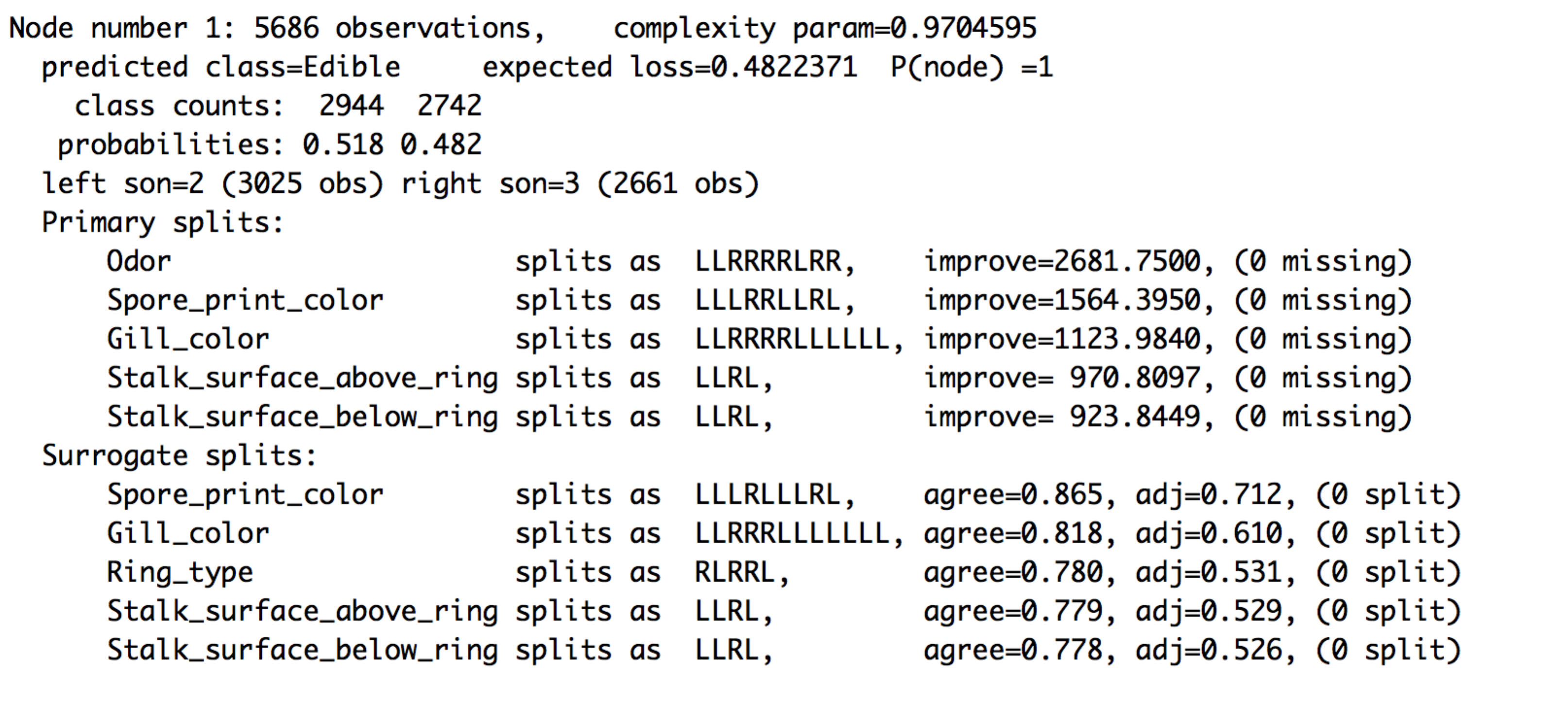
|  |  |  |  |
| --- | --- | --- | --- |
|  | edible | poisson | sum |
| Train (70%) | 2944 | 2742 | 5686 |
| Test (30%) | 1264 | 1173 | 2437 |
|  | 4208 | 3915 | 8123 |

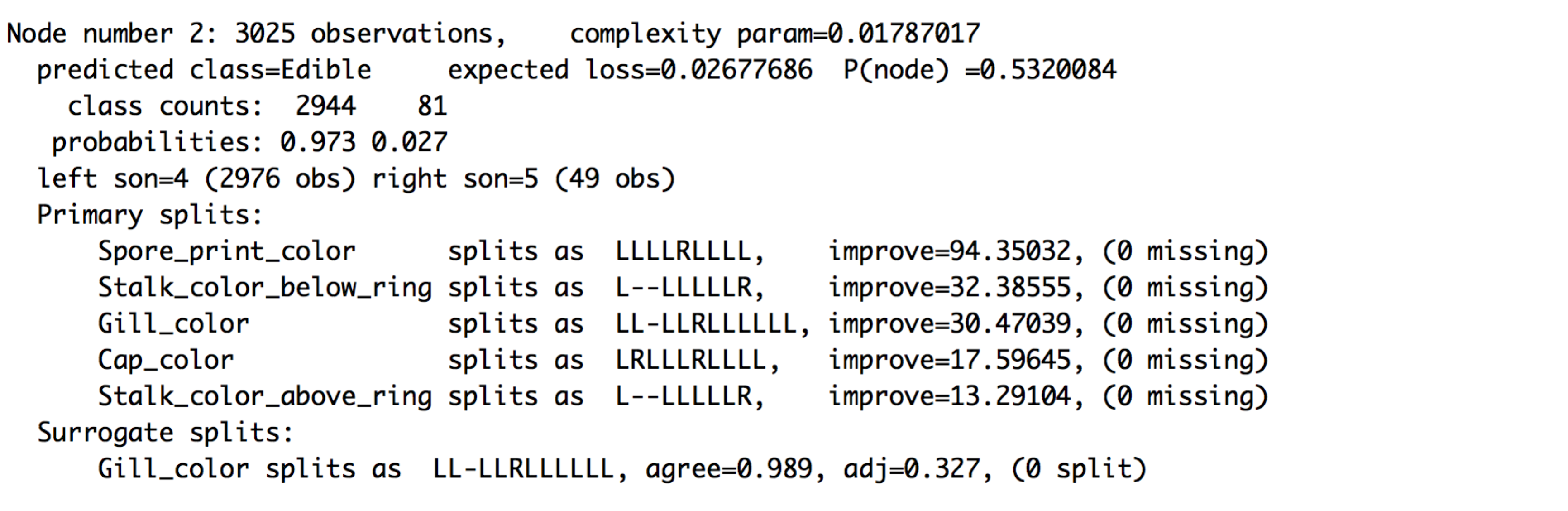
**B. Presentation & Discussion of Results**

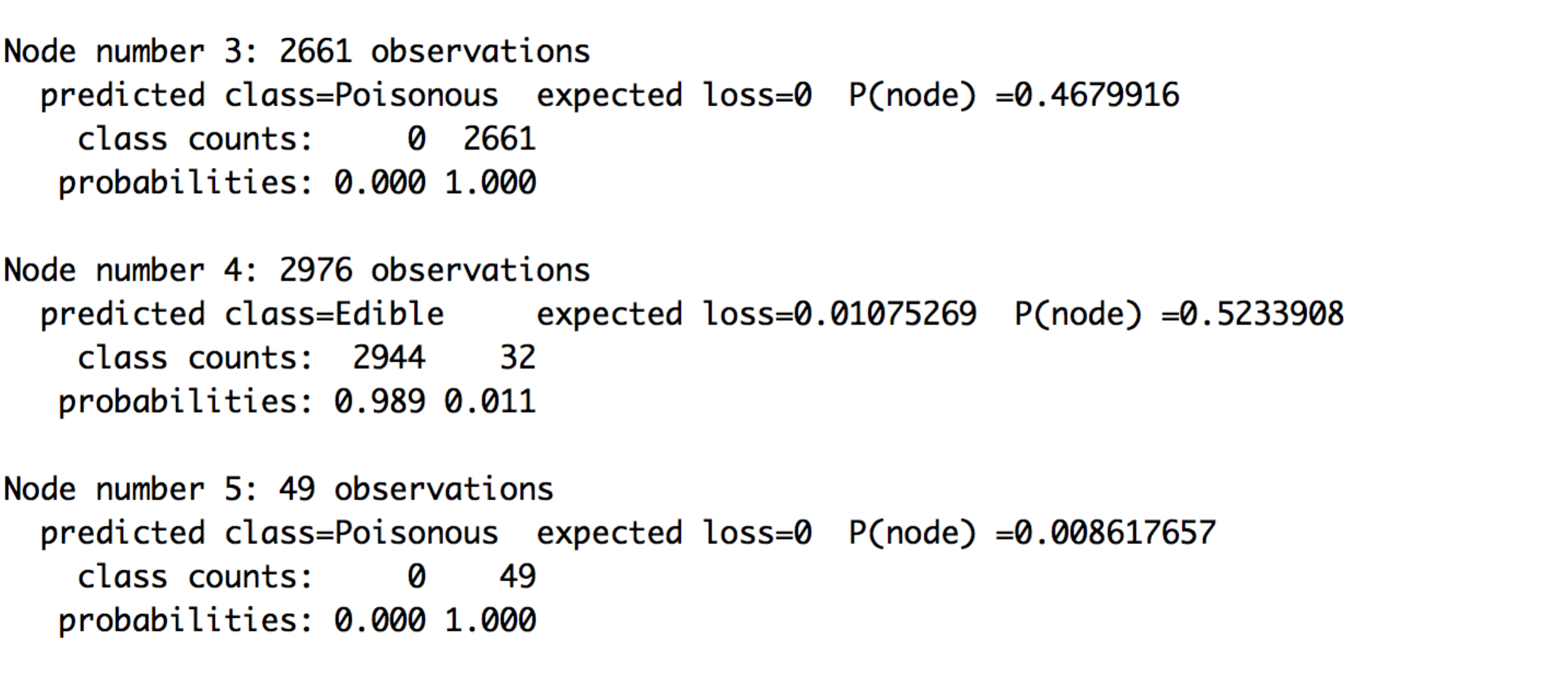
**Rpart:**

We train the decision tree model using training dataset (70%).









The R output for rpart include:

1. Cross validation error
2. Variable importance
3. For each node, the possible split and result

The above shows the possible split in each node. For example, in the first node, the root node, possible split variables are:

Odor; spore print color; gill color; Stalk surface above ring; Stalk surface below ring

The decision tree uses the root attribute Odor to be the best attribute to split the edibility of a mushroom. It is determined by “improve”, which is the improvement in deviance given by this split.

Odor gives the largest improvement. Thus it chooses odor as the first split variable. When odor is not almond or anise or none, the mushroom is poison type.

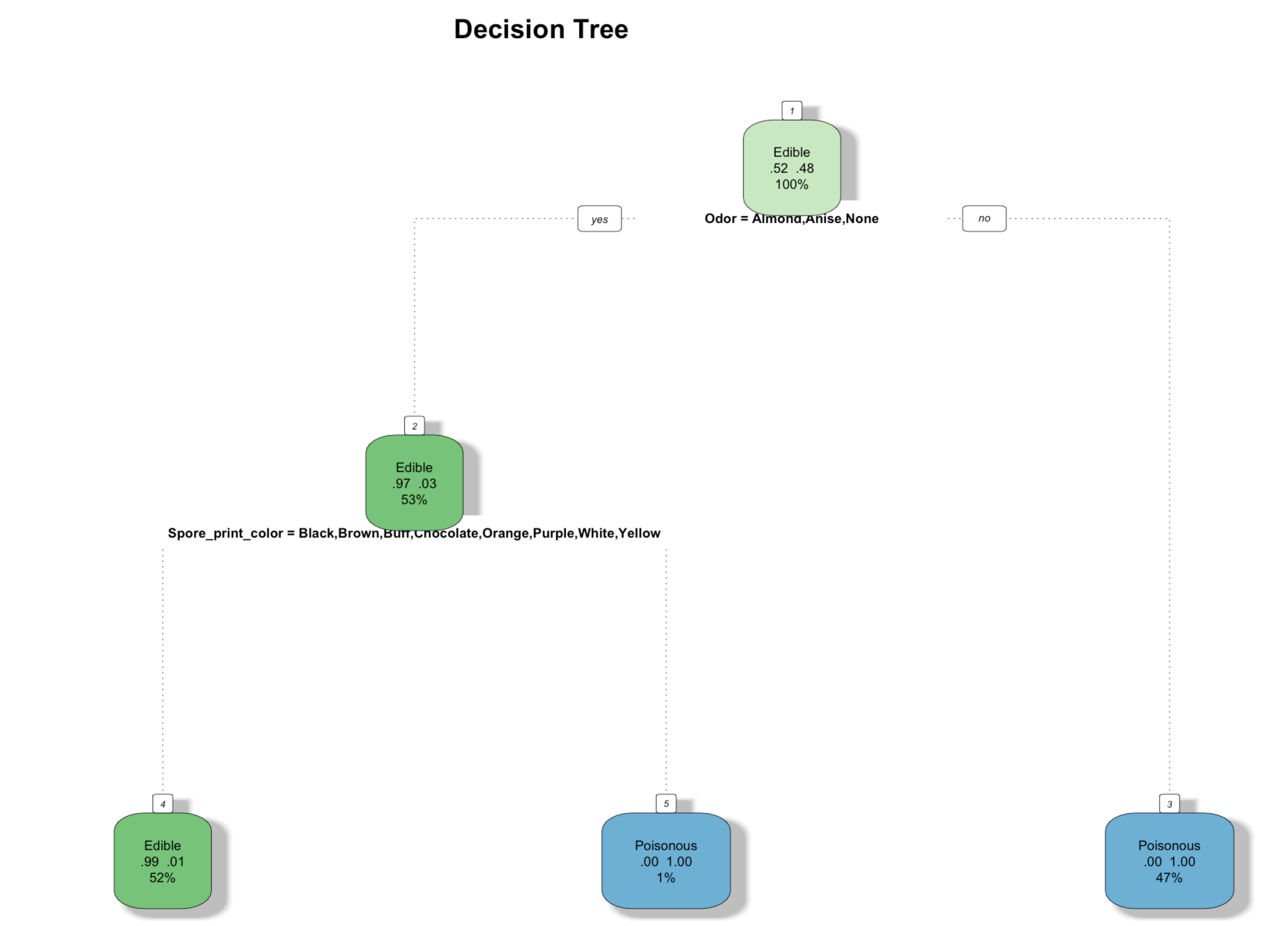
The second split variable is Spore print color. When mushroom odor is almond, anise or none. Also the spore print color is black or brown or burr or chocolate or orange or purple or white or yellow, the mushroom is edible. Others are poison. Only two attributes can gives a good result in training dataset: 16/2437 are misclassified.

In the decision tree model, it could continue split until every observation are correctly split. But we can see it stops in the second split. This is because “rpart” using cross-validation error to prune the tree.

**Prune the tree**

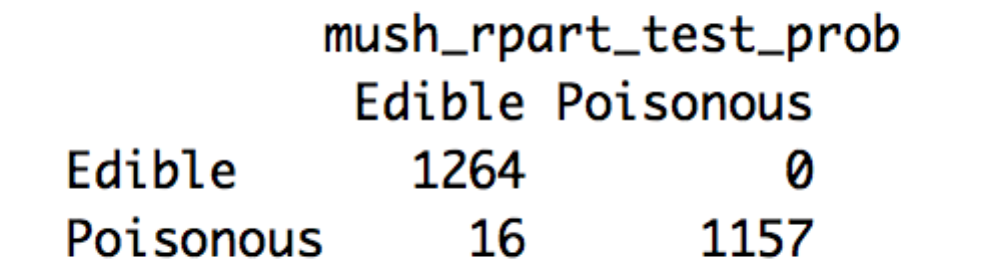
We use printcp( ) to examine the cross-validated error results by select the complexity parameter associated with the smallest cross-validated error.

After prune the tree, it has the same result as un-pruned tree. Thus we continue using the same model training model without prune.



**Evaluation:**

In confusion matrix:



16 edible mushroom are classified as poison mushroom. Error rate is 0.0065.

Overall, 2421 out of 2437 mushrooms were correctly predicted using the decision tree (raprt) model which is approximately 99.3%.

**C50:**

The C50 decision tree algorithm produces a decision tree using the categorical attributes of the mushroom dataset.

The algorithm is:

For each attribute A

Find the normalized information gain ratio from splitting on A

Let best attribute be the attribute with the highest normalized information gain

Create a decision node that splits on best attribute

Recur on the sub-splists obtained by splitting on best attribute, and add those nodes as children of node.

Using training set of 70% of the mushroom data, the C50 algorithm is run and produces the following tree using the R summary function:

Decision Tree Result from C50 Summary:

Decision tree:

Odor in {Creosote,Fishy,Foul,Musty,Pungent,Spicy}: Poisonous (2661)

Odor in {Almond,Anise,None}:

:...Spore\_print\_color = Green: Poisonous (49)

Spore\_print\_color in {Black,Brown,Buff,Chocolate,Orange,Purple,White,

: Yellow}:

:...Cap\_Surface = Grooves: Poisonous (2)

Cap\_Surface in {Fibrous,Scaly,Smooth}:

:...Stalk\_color\_below\_ring = N/A: Edible (67)

Stalk\_color\_below\_ring in {Buff,Cinnamon,Gray,N/A,Orange,Pink,

: White}: Edible (2836/1)

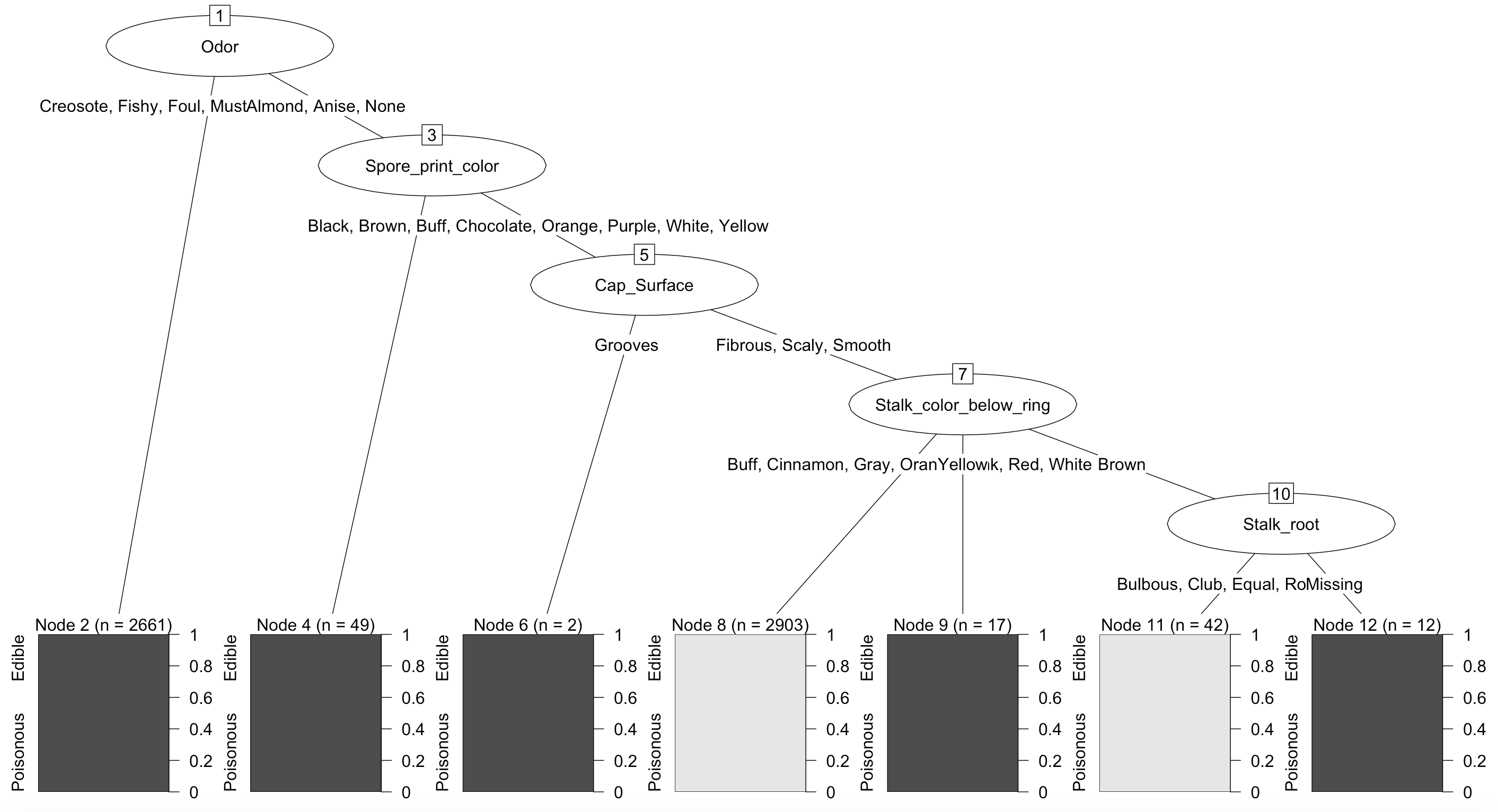
Stalk\_color\_below\_ring = Yellow: Poisonous (17)

Stalk\_color\_below\_ring = Brown:

:...Stalk\_root in {Bulbous,Club,Equal,Rooted}: Edible (42)

Stalk\_root = Missing: Poisonous (12)

The plot function is able to display a plot of the C50 model:



The C50 decision tree uses the root attribute Odor, which is determined to be the best attribute to split the data upon in order to determine edibility of a mushroom. Four other attributes are used (Spore\_print\_color, Cap\_Surface, Stalk, color\_below\_ring, and Stalk\_root) to split the data until the terminal leaf nodes are reached. Thus, these attributes have been determined by the algorithm to have the strongest relationship towards predicting the edibility of a mushroom. In order to determine the success of the algorithm, we computed a confusion matrix to see how the training model compares to a test data set. In this case, the test data set will be the remaining 30% of the mushroom data. Our confusion matrix shows that 1264 out of 1264 edible mushrooms were correctly predicted to be edible. Out of 1173 poisonous mushroom, 1170 were predicted correctly. Overall, 2434 out of 2437 mushrooms were correctly predicted using the C50 decision tree model which is approximately 99.8%. This technique compares favorably to the rpart decision tree model but not as well as the random forest algorithm.

## Random Forest

1. What’s Random Forest ?

Random forests are an ensemble learning method for classification (and regression) that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes output by individual trees.

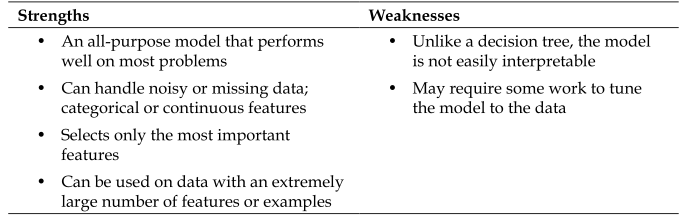
This method was championed by Leo Breiman and Adele Cutler, and combines the base principles of bagging with random feature selection to add additional diversity to the decision tree models.

Group learning is called ensemble learning

* Weak learner = a classification algorithm that performs better than chance
* Many weak learners = An ensemble learner
* We can apply ensemble learning to classification by creating multiple classification trees – called a random forest

Random forests combine versatility and power into a single machine learning approach. Because the ensemble uses only a small, random portion of the full feature set, random forests can handle extremely large datasets, where the so-called "curse of dimensionality" might cause other models to fail. At the same time, its error rates for most learning tasks are on par with nearly any other method.

The following table lists the general strengths and weaknesses of random forest models. It's worth noting that relative to other ensemble-based methods, random forests are quite competitive and offer key advantages relative to the competition. For instance, random forests tend to be easier to use and less prone to over-fitting.



## 2. How to build a random forest?

•Step 1: randomly select a subset of attributes

•Step 2: build a classification tree

•Step 3: repeat steps 1 and 2

**RandomForest Result:**

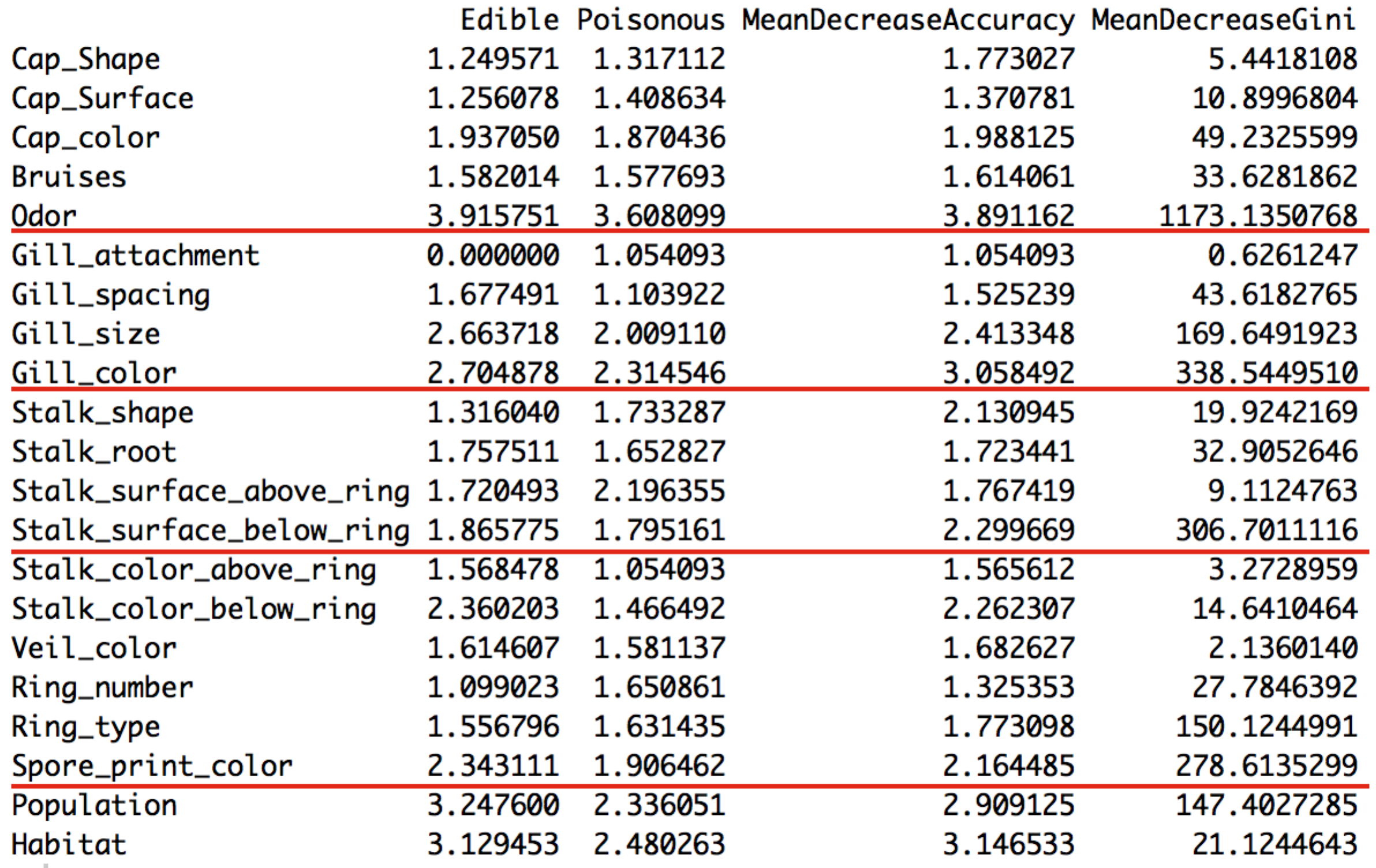
We set up the parameter in random forest:

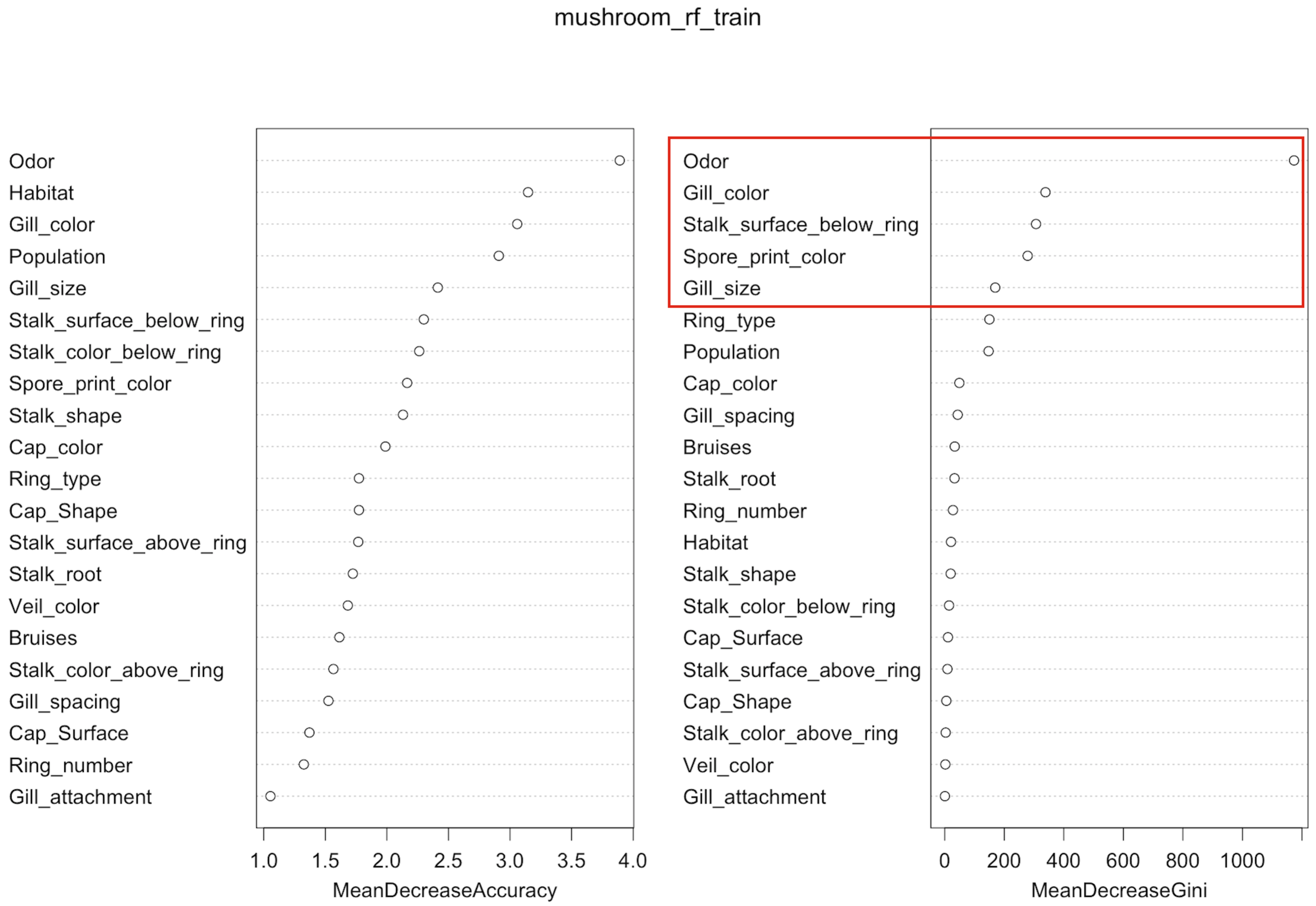
ntree (Number of trees to grow):10

Mtry (Number of variables randomly sampled as candidates at each split): 5

When building a random forest of classification trees, rule of thumb is variables being used in Random Forest to provide better result.

Using the importance() function, we can view the importance of each importance() variable. In order to get a better view of the important variables, we are using varImpPlot() function to plot the results.



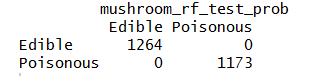


In conclusion, we have got the results from random forest algorithm are the top five most important variables for classifying the attributes for mushroom data:

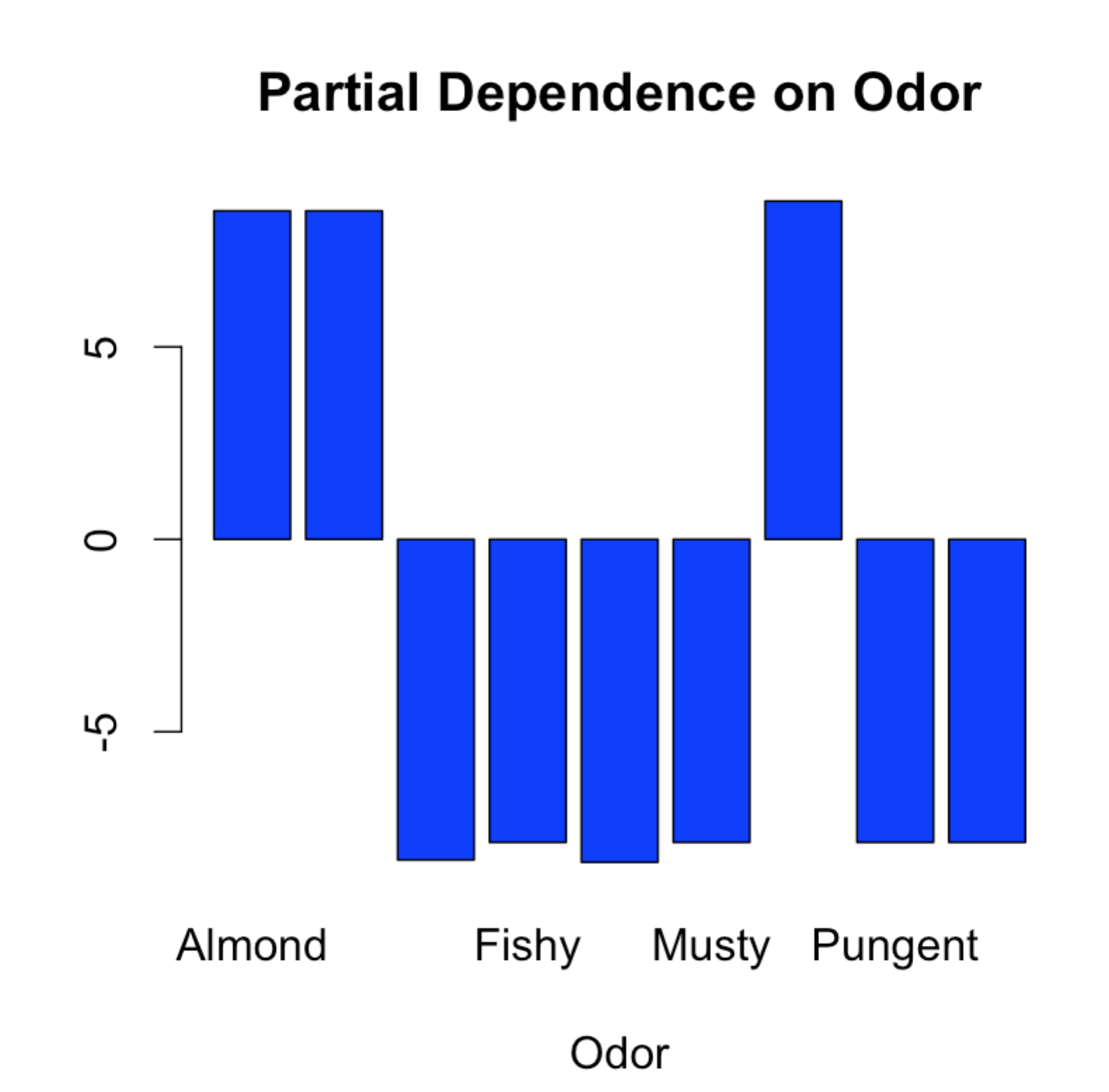
1. Odor
2. Gill color
3. Stalk surface below ring
4. Spore print color
5. Gill size

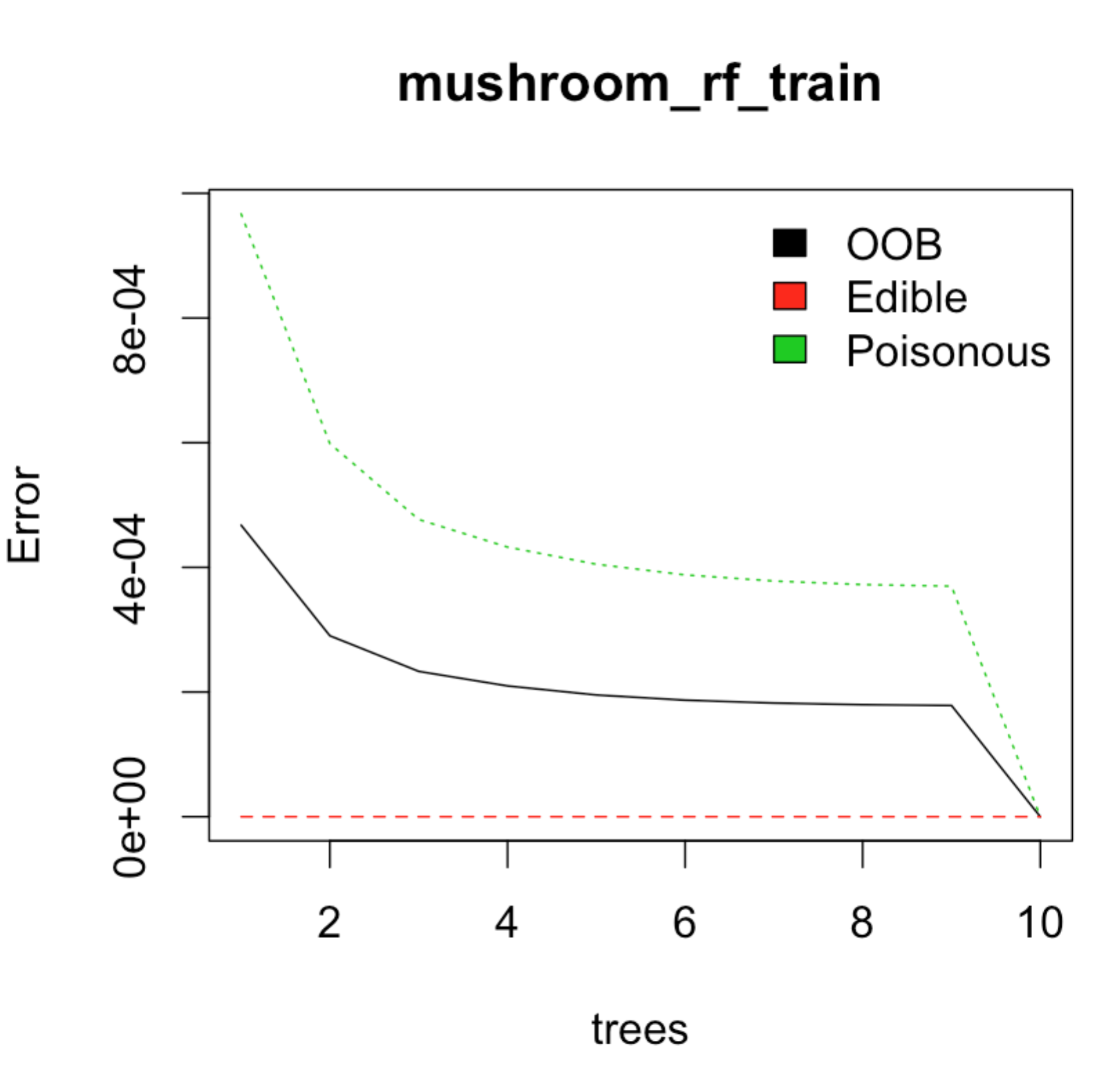
**Evaluation:**

Get the prediction by applying the model to test data, the results as following:



Marginal effect of a variable, odor, on the class probability (classification):

  
  
  
This gives the same result as rpart. When odor is almond or anise or none, the partial dependence is predicting mushroom as poison 100%.



Plot above is the error rates or MSE of Random Forest. Because Random Forest using a subset of attribute in each tree. It could have misclassification for “weak” attributes. We can see the poison mushroom was misclassified in a few trees. But as more different trees are built, misclassification rate is decreasing till 0.

## Summary: Compare those three algorithms from the points of strengths in algorithms themselves and our dataset.

|  |  |  |  |
| --- | --- | --- | --- |
|  | rpart | C5.0 | Random Forest |
| Correct rate | 99.3% | 99.8% | 100% |

Because Random Forest correctly predicted 100% of the edibility of each mushroom we feel that this algorithm is the best machine learning technique for this data set.

Random forests provide an improvement over trees by way of a small tweak that de-correlates the trees.

From the tree plot in rpat and C5.0, we can see, the first two split are the same: odor and stalk print color. But after the second split, rpart stop split because the cross validation error approach to minimum. But C5.0 continues split until information gained is not improved. Information gain is equal to the total entropy for an attribute if for each of the attribute values a unique classification can be made for the result attribute. In this case the relative entropies subtracted from the total entropy are 0. They use different criteria in stop splitting in C5.0 and rpart.

For Random Forest, it gives the best result towards this dataset. One reason is, Random Forest is ensemble method. Using multiple learning algorithms to obtain better predictive performance than could be obtained from any of the constituent learning algorithms. But overall, these three methods shows a good performance in correct rate.

**C.  What We Learned about Data Science**

In the project, I find that tree method is a really powerful method in classification. Even tree method are assumed easy to implement and explain, but cannot provide accurate results, people have create a lot of derivative method upon the tree method, like Random Forest, Boosting, Bagging to improve the result. Also, I learned a new algorithm, C5.0. I think it is the basic method for decision tree. From the result, we can see it is still quite powerful as other methods.

Form this project, I learned that we could not only pursue a good result, also a good explanation to the result and visualize the result. Number is convincing if you provide a great story for that.