Contents

[Rpart 2](#_Toc437000456)

[a. Prp () 2](#_Toc437000457)

[Decision tree description: 3](#_Toc437000458)

[b. prp(fit, extra =1) and prp(fit, extra = 100) 4](#_Toc437000459)

[Rules Derived: 5](#_Toc437000460)

[C50 7](#_Toc437000461)

[Package Information: 7](#_Toc437000462)

[Results: 8](#_Toc437000463)

[Decision tree description: 10](#_Toc437000464)

[Rules Derived: 10](#_Toc437000465)

[RandomForest 12](#_Toc437000466)

[a. varImpPlot(mushroomNum.rf) 12](#_Toc437000467)

[b. print(mushroomNum.rf) 12](#_Toc437000468)

[c. importance(mushroomNum.rf) 13](#_Toc437000469)

[d. getTree() 14](#_Toc437000470)

[e. Plot of the tree: 15](#_Toc437000471)

[f. Rules Derived 15](#_Toc437000472)

[Technique which gives the best results 17](#_Toc437000473)

[Learning from the Project about data science 17](#_Toc437000474)

# Rpart

The prp function plots rpart trees. It automatically scales and adjusts the displayed tree for best fit. It combines and extends the plot.rpart and text.rpart functions in the rpart package. Prp () plots an rpart model. The arguments of this function are a superset of those of rpart.plot.

Prp Function arguments used in this project:

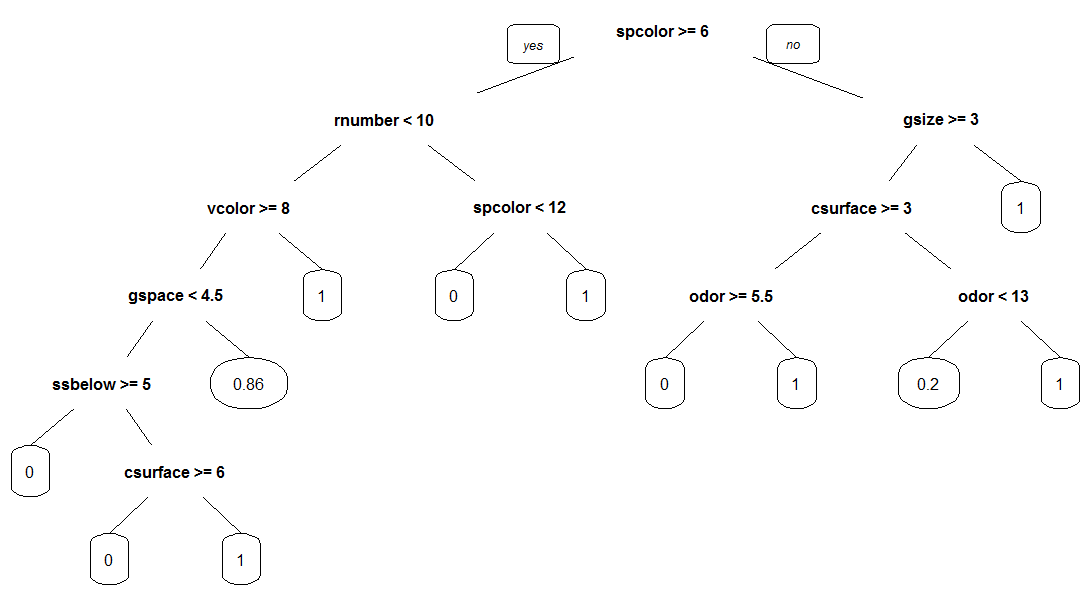
1. Type- To determine the basic plotting style.

* type = 0 (default)
* type = 1 label all nodes
* type = 2 split labels below node labels
* type = 3 left and right split labels
* type = 4 like type =3 but with interior labels

1. Extra- To add more details to the node labels.

* extra = 0 (default)
* extra = 1 number of objects
* extra = 100 percentage of objects
* extra = 101 number and percentage of objects

## Prp ()



=The above plot gives a decision tree on with the following attributes:

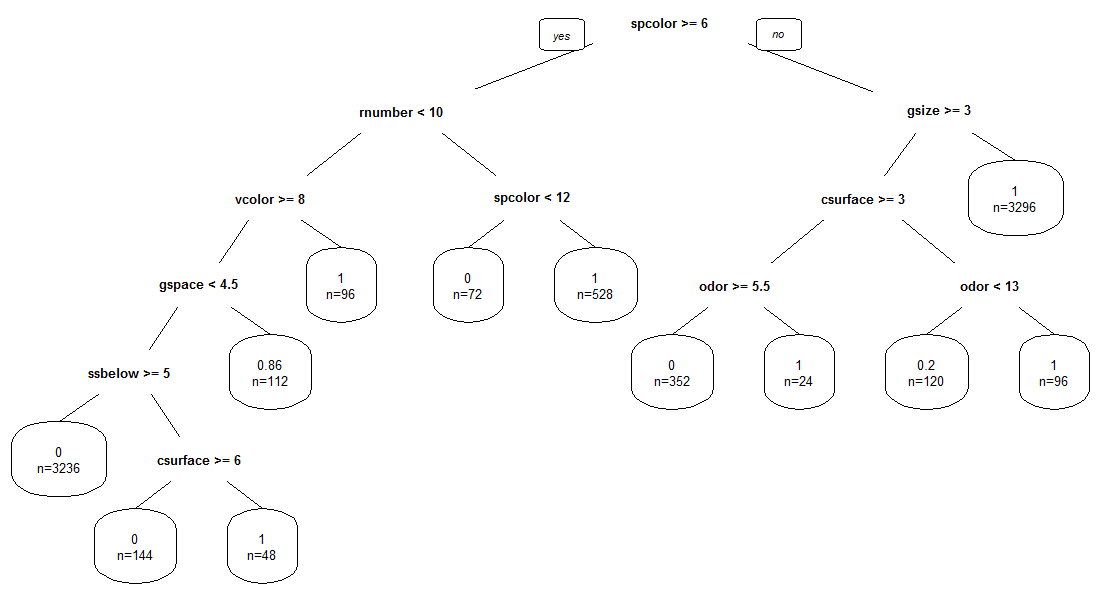
1. spore-print-color
2. ring-number
3. gill-size
4. veil-color
5. cap-surface
6. gill-spacing
7. odor
8. stalk-surface-below-ring

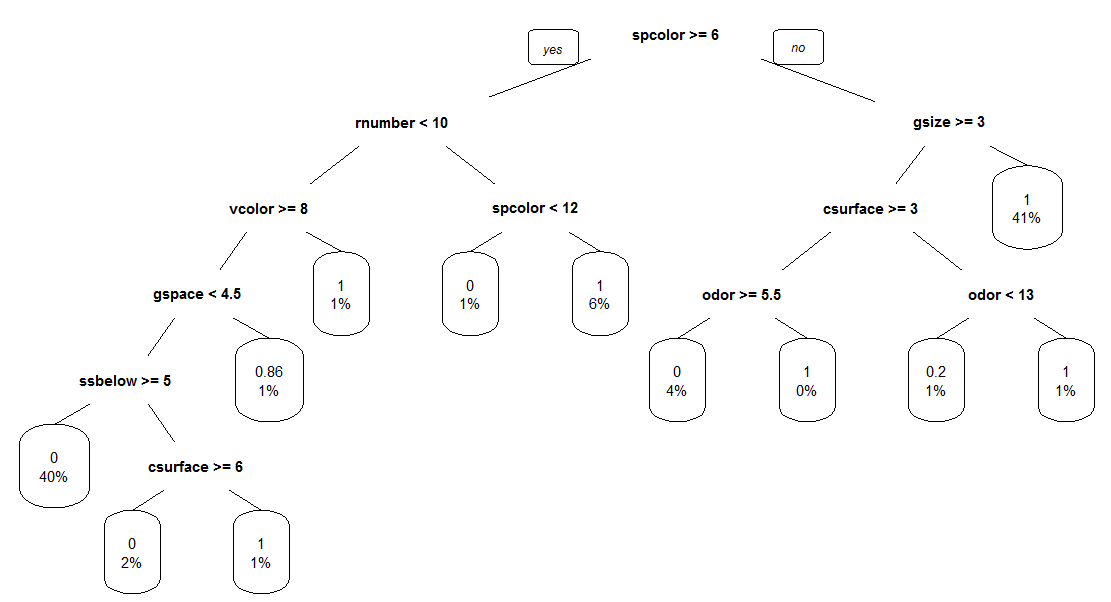
Poisonous -> 0 Edible -> 1

## Decision tree description:

The plot starts with spore-print-color. First we check if spore-print-color is greater than or equal to 6, if it is we check if ring-number is less than 10 or not, otherwise we check if gill-size is greater or equal to 3 or not. If ring-number is less than 10, we check if veil-color is greater than or equal to 8 or not, otherwise we check if spore-print-color is less than 12 or not. If gill-size is greater than or equal to 3 we check if cap-surface is greater than or equal to 3 or not, otherwise the mushrooms are edible. If veil-color is greater than or equal to 8 then we check if gill-spacing is less than 4.5 or not, otherwise mushrooms are edible. If spore-print-color is less than 12 then the mushrooms are poisonous, else if it is greater than 12 then mushrooms are edible. If cap-surface is greater than or equal to 3 we check if odor is greater than or equal to 5.5 or not, otherwise we check if odor is less than 13 or not. If gill-spacing is less than 4.5 we check if stalk-surface-below-ring is greater than or equal to 5 or not, otherwise 86 % mushrooms are edible rest are poisonous. If odor is greater than or equal to 5.5 then 20% mushrooms are edible rest are poisonous, else they are edible. If the odor is less than 13 then the mushrooms are poisonous else are edible. If stalk-surface-below-ring is greater than or equal to 5 then mushrooms are poisonous else we check if cap-surface is greater than or equal to 6. If cap-surface is greater than or equal to 6 the mushrooms are poisonous else they are edible.

## prp(fit, extra =1) and prp(fit, extra = 100)

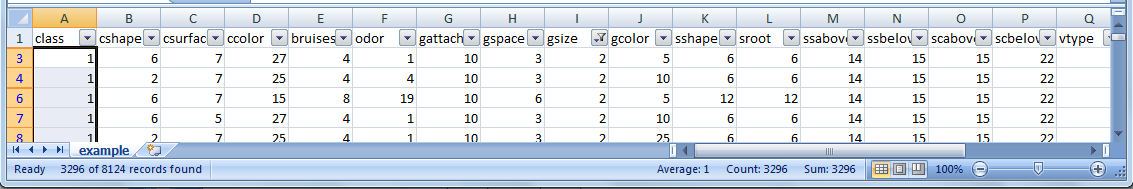




The total number of records according to the tree are = 3236 + 144 + 48 + 112 + 96 + 72 + 528 + 352 + 24 + 120 + 96 + 3296 = 8124

The number is exactly same as the number of records.

We verified the tree by putting different filters on the data using an excel sheet and the tree is accurate. Example: by adding spcolor < 6 and g-size < 3, we get 3296 records, all of which belong to edible class.



## Rules Derived:

We can draw the following deductions based on the tree and the data available (these are true for the data at hand and may change with more data):

1. If spore-print-color is black=k/brown=n/buff=b and gill-size is broad=b then the mushrooms are edible.
2. If spore-print-color is black=k/brown=n/buff=b, gill-size is narrow=n, cap-surface is fibrous=f/grooves=g and odor is almond=a/anise=l, then the mushrooms are poisonous.
3. If spore-print-color is black=k/brown=n/buff=b, gill-size is narrow=n, cap-surface is fibrous=f/grooves=g and odor is creosote=c/ fishy=y/ foul=f/ musty=m/ none=n/ pungent=p/ spicy=s, then the mushrooms are edible.
4. If spore-print-color is black=k/brown=n/buff=b, gill-size is narrow=n, cap-surface is scaly=y/ smooth=s and odor is almond=a/ anise=l/ creosote=c/ fishy=y then 20% mushrooms are edible rest are poisonous.
5. If spore-print-color is black=k/brown=n/buff=b, gill-size is narrow=n, cap-surface is scaly=y/ smooth=s and odor is foul=f/ musty=m/ none=n/ pungent=p/ spicy=s then mushrooms are edible.
6. If spore-print-color is chocolate=h/ green=r/ orange=o/ purple=u/ white=w/ yellow=y, ring-number is none=n/ one=o, veil-color is white=w/ yellow=y, gill-spacing close=c and stalk-surface-below-ring scaly=y/ silky=k/ smooth=s then mushroom is poisonous.
7. If spore-print-color is chocolate=h/ green=r/ orange=o/ purple=u/ white=w/ yellow=y, ring-number is none=n/ one=o, veil-color is white=w/ yellow=y, gill-spacing close=c, stalk-surface-below-ring fibrous=f and cap-surface is smooth=s then mushroom is poisonous.
8. If spore-print-color is chocolate=h/ green=r/ orange=o/ purple=u/ white=w/ yellow=y, ring-number is none=n/ one=o, veil-color is white=w/ yellow=y, gill-spacing close=c, stalk-surface-below-ring fibrous=f and cap-surface is fibrous=f/ grooves=g/ scaly=y then mushroom is edible.
9. If spore-print-color is chocolate=h/ green=r/ orange=o/ purple=u/ white=w/ yellow=y, ring-number is none=n/ one=o, veil-color is white=w/ yellow=y and gill-spacing crowded=w/ distant=d then 86% mushroom is edible and rest is poisonous.
10. If spore-print-color is chocolate=h/ green=r/ orange=o/ purple=u/ white=w/ yellow=y, ring-number is none=n/ one=o and veil-color is brown=n/ orange=o then mushroom is edible.
11. If spore-print-color is chocolate=h/ green=r/ orange=o and ring-number is two=t then mushroom is poisonous.
12. If spore-print-color is purple=u/ white=w/ yellow=y and ring-number is two=t then mushroom is edible.

# C50

## Package Information:

C5.0 Decision Trees and Rule-Based Models: This package provides fit classification tree models or rule-based models using Quinlan’s C5.0 algorithm.

**Quinlan’s C5.0 algorithm:**

Using the concept of [information entropy](https://en.wikipedia.org/wiki/Entropy_%28information_theory%29) C4.5 builds decision trees from a set of training data in the same way as [ID3](https://en.wikipedia.org/wiki/ID3_algorithm). The training data is a set of already classified samples. At each node of the tree, C4.5 chooses the attribute of the data that most effectively splits its set of samples into subsets enriched in one class or the other. The splitting criterion is the normalized [information gain](https://en.wikipedia.org/wiki/Information_gain) (difference in [entropy](https://en.wikipedia.org/wiki/Entropy_%28information_theory%29)). The attribute with the highest normalized information gain is chosen to make the decision. The C4.5 algorithm then recurs on the smaller sub lists. Source: Wikipedia

C5.0 offers many advantages over C4.5 like better speed, efficient memory usage, smaller decision trees, support for boosting, weighting and winnowing.

**Method Usage:**

C5.0(x, ...)

#Default S3 method:

C5.0(x, y, trials = 1, rules= FALSE, weights = NULL, control = C5.0Control(), costs = NULL, ...)

Arguments:

* **X**: a data frame or matrix of predictors.
* **Y**: a factor vector with 2 or more levels
* **Trials**: an integer specifying the number of boosting iterations. A value of one indicates that a single model is used.
* **Rules**: A logical- should the tree be decomposed into a rule-based model?
* **Weights**: an optional numeric vector of case weights. Note that the data used for the
* **Case**: weights will not be used as a splitting variable in the
* **Control**: a list of control parameters
* **Costs**: a matrix of costs associated with the possible errors. The matrix should have C columns and rows where C is the number of class levels.
* **Formula**: a formula, with a response and at least one predictor.
* **Data**: an optional data frame in which to interpret the variables named in the formula.
* **Subset**: optional expression saying that only a subset of the rows of the data should be used in the fit.
* **na.action**: a function which indicates what should happen when the data contain NAs. The default is to include missing values since the model can accommodate them.

## Results:

|  |
| --- |
| Call: |
| C5.0.default(x = mushroomNum.sample, y = as.factor(mushroomNum.df$class)) |
| C5.0 [Release 2.07 GPL Edition] Sat Nov 21 20:51:26 2015 |
| ------------------------------- |
| Class specified by attribute `outcome' |
| Read 8124 cases (23 attributes) from undefined.data |
| Decision tree: |
| spcolor <= 5: |
| :...odor > 19: 0 (256) |
| : odor <= 19: |
| : :...gsize <= 2: 1 (3296) |
| : gsize > 2: |
| : :...sroot > 6: 1 (96) |
| : sroot <= 6: |
| : :...bruises <= 4: 1 (48) |
| : bruises > 4: 0 (192) |
| spcolor > 5: |
| :...rnumber > 9: |
| :...spcolor <= 11: 0 (72) |
| : spcolor > 11: 1 (528) |
| rnumber <= 9: |
| :...vcolor <= 6: 1 (96) |
| vcolor > 6: |
| :...gspace > 3: |
| :...popnum <= 6: 0 (16) |
| : popnum > 6: 1 (96) |
| gspace <= 3: |
| :...ssbelow > 3: 0 (3236) |
| ssbelow <= 3: |
| :...csurface <= 5: 1 (48) |
| csurface > 5: 0 (144) |
|  |
| Evaluation on training data (8124 cases): |
| Decision Tree |
| ---------------- |
| Size Errors |
|  |
| 13 0( 0.0%) << |
| (a) (b) <-classified as |
| ---- ---- |
| 3916 (a): class 0 |
| 4208 (b): class 1 |
| Attribute usage: |
| 100.00% spcolor |
| 52.14% rnumber |
| 47.86% odor |
| 44.76% vcolor |
| 44.71% gsize |
| 43.57% gspace |
| 42.20% ssbelow |
| 4.14% sroot |
| 2.95% bruises |
| 2.36% csurface |
| 1.38% popnum |
| Time: 0.1 secs |

The C50 decision tree has the following attributes along with their usage in the tree:

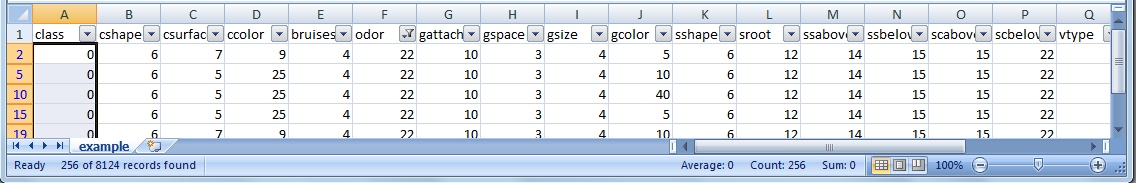
1. spore-print-color (100%)
2. ring-number (52.14%)
3. gill-size (44.71%)
4. veil-color (44.76%)
5. cap-surface (2.36%)
6. gill-spacing (43.57%)
7. odor (47.86%)
8. stalk-surface-below-ring (42.20%)
9. stalk-root (4.14%)
10. bruises (2.95%)
11. population (1.38%)

This tree has three extra attributes (stalk-root, bruises, population) when compared to rpart decision tree.

Poisonous -> 0 Edible -> 1

**The total Number of records according to the tree** are = 256 + 3296 + 96 + 48 + 192 + 72 + 528 + 96 + 16 + 96 + 3236 + 48 + 144 = 8124 same as the records at hand.

We verified the tree by putting different filters on the data using an excel sheet and the tree is accurate. **Example**: By adding filter spcolor<=5 and odor >19 we get 256 records all of which belong to poisonous class.



## Decision tree description:

The plot starts with spore-print-color. First we check if spore-print-color is greater than 5, if it is we check if ring-number is less than or equal to 9 or not, otherwise we check if odor is less than or equal to 19 or not. If ring-number is less than or equal to 9, we check if veil-color is greater than 6 or not, otherwise we check if spore-print-color is greater than 11 or not. If odor is less than 20 then, we check if gill-size is greater than 2 or not, otherwise mushrooms are poisonous. If spore-print-color is less than or equal to 11 then the mushrooms are poisonous, else if it is greater than 11 then mushrooms are edible. If gill-size is greater than 2 we check if stalk-root is less than or equal to 6 or not, otherwise the mushrooms are edible. If veil-color is greater than 6 then we check if gill-spacing is less than or equal to 3 or not, otherwise mushrooms are edible. If stalk-root is less than or equal to 6 then we check if bruises are greater than 4 or not, otherwise mushrooms are edible. If gill-spacing is less than or equal to 3 we check if stalk-surface-below-ring is less than or equal to 3 or not, otherwise we check if population is greater than 6 or not. If bruises are greater than 4 then mushrooms are poisonous else they are edible. If population is greater than 6 then the mushrooms are edible, else they are poisonous. If stalk-surface-below-ring is greater than 3 then mushrooms are poisonous else we check if cap-surface is greater than 5. If cap-surface is greater than 5 then the mushrooms are poisonous else they are edible.

## Rules Derived:

From the above deductions of the decision tree we can arrive at below rules (these are true for the data at hand and may change with more data):

1. If spore-print-color is black=k/brown=n/buff=b and odor is pungent=p/spicy=s then the mushrooms are poisonous.
2. If spore-print-color is black=k/brown=n/buff=b, odor is almond=a/ anise=l/ creosote=c/ fishy=y/ foul=f/ musty=m/ none=n and gill-size is broad=b then the mushrooms are edible.
3. If spore-print-color is black=k/brown=n/buff=b, odor is almond=a/ anise=l/ creosote=c/ fishy=y/ foul=f/ musty=m/ none=n, gill-size is narrow=n and stalk-root is cup=u/ equal=e/ rhizomorphs=z/ rooted=r/ missing then the mushrooms are edible.
4. If spore-print-color is black=k/brown=n/buff=b, odor is almond=a/ anise=l/ creosote=c/ fishy=y/ foul=f/ musty=m/ none=n, gill-size is narrow=n, stalk-root is cup=u/ equal=e/ rhizomorphs=z/ rooted=r/ missing and bruises are bruises=t then the mushrooms are edible.
5. If spore-print-color is black=k/brown=n/buff=b, odor is almond=a/ anise=l/ creosote=c/ fishy=y/ foul=f/ musty=m/ none=n, gill-size is narrow=n, stalk-root is cup=u/ equal=e/ rhizomorphs=z/ rooted=r/ missing and bruises are no=f then the mushrooms are poisonous.
6. If spore-print-color is chocolate=h/ green=r/ orange=o/ purple=u/ white=w/ yellow=y, ring-number is none=n/ one=o, veil-color is white=w/ yellow=y, gill-spacing close=c and stalk-surface-below-ring scaly=y/ silky=k/ smooth=s then mushroom is poisonous.
7. If spore-print-color is chocolate=h/ green=r/ orange=o/ purple=u/ white=w/ yellow=y, ring-number is none=n/ one=o, veil-color is white=w/ yellow=y, gill-spacing close=c, stalk-surface-below-ring fibrous=f and cap-surface is smooth=s then mushroom is poisonous.
8. If spore-print-color is chocolate=h/ green=r/ orange=o/ purple=u/ white=w/ yellow=y, ring-number is none=n/ one=o, veil-color is white=w/ yellow=y, gill-spacing close=c, stalk-surface-below-ring fibrous=f and cap-surface is fibrous=f/ grooves=g/ scaly=y then mushroom is edible.
9. If spore-print-color is chocolate=h/ green=r/ orange=o/ purple=u/ white=w/ yellow=y, ring-number is none=n/ one=o, veil-color is white=w/ yellow=y and gill-spacing crowded=w/ distant=d and population is abundant=a/ clustered=c/ numerous=n then the mushroom is poisonous.
10. If spore-print-color is chocolate=h/ green=r/ orange=o/ purple=u/ white=w/ yellow=y, ring-number is none=n/ one=o, veil-color is white=w/ yellow=y and gill-spacing crowded=w/ distant=d and population is scattered=s/ several=v/ solitary=y then the mushroom is edible.
11. If spore-print-color is chocolate=h/ green=r/ orange=o/ purple=u/ white=w/ yellow=y, ring-number is none=n/ one=o and veil-color is brown=n/ orange=o then mushroom is edible.
12. If spore-print-color is chocolate=h/ green=r/ orange=o and ring-number is two=t then mushroom is poisonous.
13. If spore-print-color is purple=u/ white=w/ yellow=y and ring-number is two=t then mushroom is edible.

From the deductions we can see that the rules are almost similar to the ones retrieved from the rpart decision tree.

# RandomForest

Random forests package helps in **creating random decision forests** that can be used for **classification, regression and other tasks**, that operate by **constructing a multitude of decision trees** at **training time and outputting the class** that is the mode of the classes (classification) or mean prediction (regression) of the individual trees.

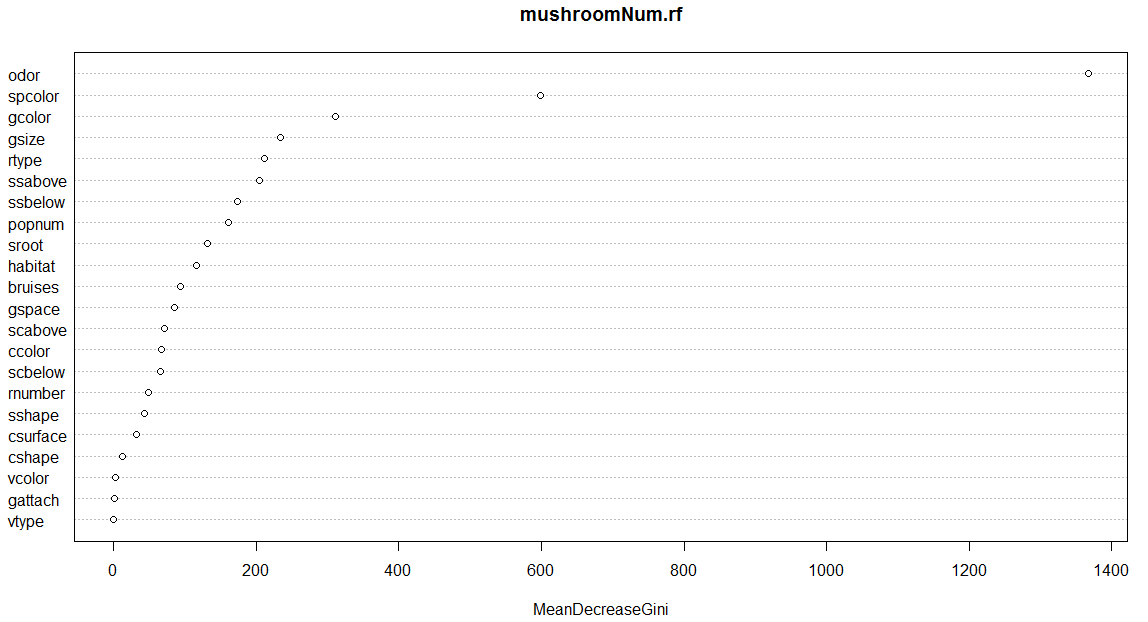
The algorithm for inducing Breiman's random forest was developed by Leo Breiman and Adele Cutler.

The method combines Breiman's "bagging" idea and the random selection of features, introduced independently by Ho, Amit and Geman in order to construct a collection of decision trees with controlled variance.

The selection of a random subset of features is an example of the random subspace method, which, in Ho's formulation, is a way to implement the "stochastic discrimination" approach to classification proposed by Eugene Kleinberg.

Source: Wikipedea

## varImpPlot(mushroomNum.rf)



The above plot shows the importance of the attributes of the data.

## print(mushroomNum.rf)

|  |
| --- |
| Call: |
| randomForest(formula = as.factor(mushroomNum.df$class) ~ ., data = mushroomNum.df[-1], do.trace = 10) |
| Type of random forest: classification |
| Number of trees: 500 |
| No. of variables tried at each split: 4 |
|  |
| OOB estimate of error rate: 0% |
| Confusion matrix: |
| 0 1 class.error |
| 0 3916 0 0 |
| 1 0 4208 0 |
|  |

The **four variables** are tried by the algorithm at each split. There are **500 trees in total.** The algorithm divides the data **into two categories** i.e**. edible and poisonous** and the result tallies with the actual data.

## importance(mushroomNum.rf)

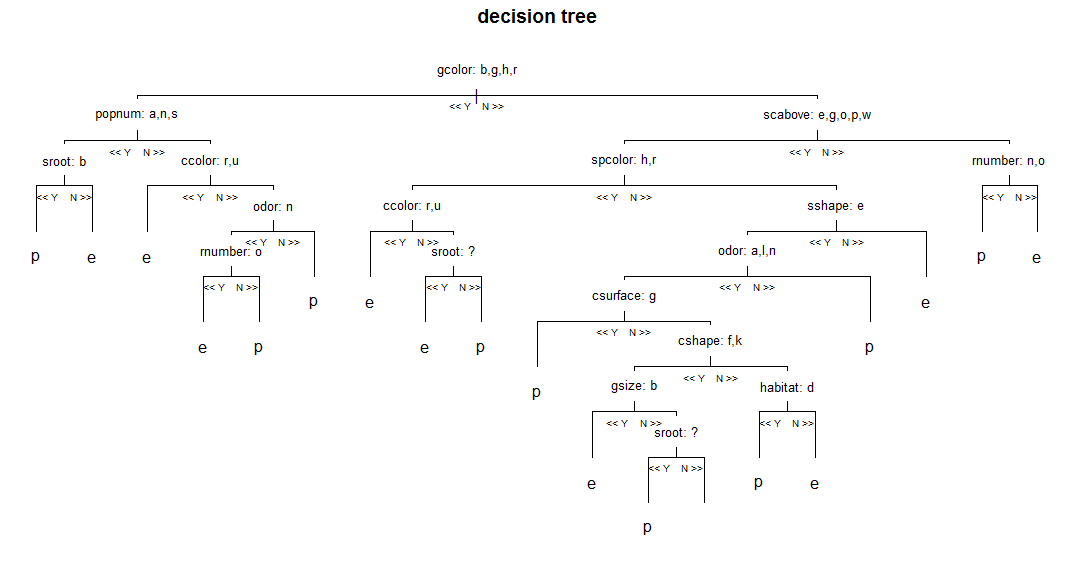
|  |  |
| --- | --- |
| MeanDecreaseGini | |
| .Odor | 1367.311 |
| spcolor | 599.2801 |
| Gcolor | 311.1388 |
| Gsize | 233.7357 |
| Rtype | 212.0833 |
| ssabove | 204.1884 |
| ssbelow | 174.0333 |
| Popnum | 160.6452 |
| Sroot | 132.1761 |
| habitat | 117.0057 |
| bruises | 94.33557 |
| Gspace | 85.41558 |
| scabove | 71.4773 |
| Ccolor | 67.21998 |
| scbelow | 65.90578 |
| rnumber | 49.51868 |
| Sshape | 42.96631 |
| csurface | 31.78152 |
| Cshape | 11.97569 |
| Vcolor | 2.642068 |
| gattach | 1.175474 |
| Vtype | 0 |

The above table gives the importance of each attribute present in the plot. The most important attribute according to the plot is odor and the least is vtype.

## getTree()

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Num | left-daughter | right-daughter | split-var | split-point | Status | prediction |
| 1 | 2 | 3 | 9 | 269 | 1 | 0 |
| 2 | 4 | 5 | 21 | 13 | 1 | 0 |
| 3 | 6 | 7 | 14 | 236 | 1 | 0 |
| 4 | 8 | 9 | 11 | 2 | 1 | 0 |
| 5 | 10 | 11 | 3 | 192 | 1 | 0 |
| 6 | 12 | 13 | 20 | 34 | 1 | 0 |
| 7 | 14 | 15 | 18 | 3 | 1 | 0 |
| 8 | 0 | 0 | 0 | 0 | -1 | 2 |
| 9 | 0 | 0 | 0 | 0 | -1 | 1 |
| 10 | 0 | 0 | 0 | 0 | -1 | 1 |
| 11 | 16 | 17 | 5 | 32 | 1 | 0 |
| 12 | 18 | 19 | 3 | 192 | 1 | 0 |
| 13 | 20 | 21 | 10 | 1 | 1 | 0 |
| 14 | 0 | 0 | 0 | 0 | -1 | 2 |
| 15 | 0 | 0 | 0 | 0 | -1 | 1 |
| 16 | 22 | 23 | 18 | 2 | 1 | 0 |
| 17 | 0 | 0 | 0 | 0 | -1 | 2 |
| 18 | 0 | 0 | 0 | 0 | -1 | 1 |
| 19 | 24 | 25 | 11 | 1 | 1 | 0 |
| 20 | 26 | 27 | 5 | 41 | 1 | 0 |
| 21 | 0 | 0 | 0 | 0 | -1 | 1 |
| 22 | 0 | 0 | 0 | 0 | -1 | 1 |
| 23 | 0 | 0 | 0 | 0 | -1 | 2 |
| 24 | 0 | 0 | 0 | 0 | -1 | 1 |
| 25 | 0 | 0 | 0 | 0 | -1 | 2 |
| 26 | 28 | 29 | 2 | 2 | 1 | 0 |
| 27 | 0 | 0 | 0 | 0 | -1 | 2 |
| 28 | 0 | 0 | 0 | 0 | -1 | 2 |
| 29 | 30 | 31 | 1 | 12 | 1 | 0 |
| 30 | 32 | 33 | 8 | 1 | 1 | 0 |
| 31 | 34 | 35 | 22 | 1 | 1 | 0 |
| 32 | 0 | 0 | 0 | 0 | -1 | 1 |
| 33 | 36 | 37 | 11 | 1 | 1 | 0 |
| 34 | 0 | 0 | 0 | 0 | -1 | 2 |
| 35 | 0 | 0 | 0 | 0 | -1 | 1 |
| 36 | 0 | 0 | 0 | 0 | -1 | 2 |
| 37 | 38 | 39 | 3 | 26 | 1 | 0 |
| 38 | 0 | 0 | 0 | 0 | -1 | 1 |
| 39 | 0 | 0 | 0 | 0 | -1 | 2 |

## Plot of the tree:



## Rules Derived

The above tree is the first tree of all the trees generated using random Forest. We can derive the following rules from the above tree:

1. If gill-color is b/ g/ h/ r, population is a/ n/ s and stalk-root is b then the mushrooms are poisonous.
2. If gill-color is b/ g/ h/ r, population is a/ n/ s and stalk-root is not b then the mushrooms are edible.
3. If gill-color is b/ g/ h/ r, population is c/ v/ y and cap-color is r/ u, then the mushrooms are edible.
4. If gill-color is b/ g/ h/ r, population is c/ v/ y, cap-color is not r/ u, odor is n and ring-number is o then the mushrooms are edible.
5. If gill-color is b/ g/ h/ r, population is c/ v/ y, cap-color is not r/ u, odor is n and ring-number is not o then the mushrooms are poisonous.
6. If gill-color is b/ g/ h/ r, population is c/ v/ y, cap-color is not r/ u and odor is not n then the mushrooms are poisonous.
7. If gill-color is k/ n/ o/ p/ u/ e/ w/ y, stalk-color-above-ring is e/ g/ o/ p/ w, spore-print-color is h/ r and cap-color is r/ u, then the mushrooms are edible.
8. If gill-color is k/ n/ o/ p/ u/ e/ w/ y, stalk-color-above-ring is e/ g/ o/ p/ w, spore-print-color is h/ r, cap-color is not r/ u and stalk root is missing then the mushrooms are edible.
9. If gill-color is k/ n/ o/ p/ u/ e/ w/ y, stalk-color-above-ring is e/ g/ o/ p/ w, spore-print-color is h/ r, cap-color is not r/ u and stalk root is not missing then the mushrooms are poisonous.
10. If gill-color is k/ n/ o/ p/ u/ e/ w/ y, stalk-color-above-ring is e/ g/ o/ p/ w, spore-print-color is not h/ r, stalk-shape is e, odor is a/ l/ n and cap-surface is g then mushrooms are poisonous.
11. If gill-color is k/ n/ o/ p/ u/ e/ w/ y, stalk-color-above-ring is e/ g/ o/ p/ w, spore-print-color is not h/ r, stalk-shape is e, odor is a/ l/ n, cap-surface is not g, cap-shape is f/ k and gill-size is b then mushrooms are edible.
12. If gill-color is k/ n/ o/ p/ u/ e/ w/ y, stalk-color-above-ring is e/ g/ o/ p/ w, spore-print-color is not h/ r, stalk-shape is e, odor is a/ l/ n, cap-surface is not g, cap-shape is f/ k, gill-size is not b and stalk-root is missing then mushrooms are poisonous.
13. If gill-color is k/ n/ o/ p/ u/ e/ w/ y, stalk-color-above-ring is e/ g/ o/ p/ w, spore-print-color is not h/ r, stalk-shape is e, odor is a/ l/ n, cap-surface is not g, cap-shape is not f/ k and habitat is d then mushrooms are poisonous.
14. If gill-color is k/ n/ o/ p/ u/ e/ w/ y, stalk-color-above-ring is e/ g/ o/ p/ w, spore-print-color is not h/ r, stalk-shape is e, odor is a/ l/ n, cap-surface is not g, cap-shape is not f/ k and habitat is not d then mushrooms are edible.
15. If gill-color is k/ n/ o/ p/ u/ e/ w/ y, stalk-color-above-ring is e/ g/ o/ p/ w, spore-print-color is not h/ r, stalk-shape is e and odor is not a/ l/ n then mushrooms are poisonous.
16. If gill-color is k/ n/ o/ p/ u/ e/ w/ y, stalk-color-above-ring is e/ g/ o/ p/ w, spore-print-color is not h/ r, stalk-shape is not e then mushrooms are edible.
17. If gill-color is k/ n/ o/ p/ u/ e/ w/ y, stalk-color-above-ring is not e/ g/ o/ p/ w and ring-number is n/ o then mushrooms are poisonous.
18. If gill-color is k/ n/ o/ p/ u/ e/ w/ y, stalk-color-above-ring is not e/ g/ o/ p/ w and ring-number is not n/ o then mushrooms are edible.

# Technique which gives the best results

**Rpart** gives the best decision tree plots in terms of accuracy, ease and clarity. Rpart provides functions like prp with a wide range of attributes, which can be used to plot very clear decision trees with many variations. These trees also provide important attributes in the data which are used to plot the decision tree. We can also see the exact number of records in each partition; this is very useful in verifying the tree and inspecting the records. C50 also gave a decision tree but it is not very elaborate and lacks flexibility when it comes to exploration of the decision trees and data. Random tree also provides many functions but it is a little complex to understand. Plotting the decision tree requires you to write a very complex code.

# Learning from the Project about data science

In the second project we learnt how to cluster the data and get the number of groups in the data.

This Project helped us in learning:

1. How to **get decision trees** from your data using different packages.
2. How to **analyse the decision trees and classify data.**
3. How to **analyse splitting used in different decision trees**
4. How to **retrieve important information** from decision trees.
5. How to **verify decision tree against the data**.
6. How **to study a section of data** (retrieved using the tree) to get more insight.
7. **How to point out the important attributes**, which are key to deciding if the mushrooms are edible or not.
8. **How to draw out relations between the attributes which would tell us if the mushrooms are edible or not for the given data** (rules derived).
9. How to understand **the features and logic used in different packages**.
10. How to **compare and utilize different packages or technologies when exploring data and choose the best fit for your needs.**

Decision trees are very easy to understand, they help in picturing the data and rules and there are no prior assumptions made about the data.