**Mushroom Data Classification**

**CS5950: Machine Learning**

**Final Project**

**By Chris Carlson, Colin MacCreery, and Ben Mechling**

**MUSHROOM DATA**

The mushroom data was obtained from a UCI data repository (<https://archive.ics.uci.edu/ml/datasets/Mushroom>). The data set has 8124 rows and a total of 23 columns. All of the data is categorical. Each value is a single letter representing a value specific to a given attribute (column). The first column “class” indicates whether the mushroom is considered to be poisonous (p) or edible (e). The number of possible values for each attribute varies. The table below shows the possible of each attribute and symbol used to represent it. The largest number of possible values is 12.

|  |  |
| --- | --- |
| **Attribute** | **Values** |
| class | edible=e, poisonous=p |
| cap.shape | bell=b,conical=c,convex=x,flat=f,knobbed=k,sunken=s |
| cap.surface | fibrous=f,grooves=g,scaly=y,smooth=s |
| cap.color | brown=n,buff=b,cinnamon=c,gray=g,green=r,  pink=p,purple=u,red=e,white=w,yellow=y |
| bruises | bruises=t,no=f |
| odor | almond=a,anise=l,creosote=c,fishy=y,foul=f,  musty=m,none=n,pungent=p,spicy=s |
| gill.attachment | attached=a,descending=d,free=f,notched=n |
| gill.spacing | close=c,crowded=w,distant=d |
| gill.size | broad=b,narrow=n |
| gill.color | black=k,brown=n,buff=b,chocolate=h,gray=g,green=r,orange=o,pink=p,purple=u,red=e,  white=w,yellow=y |
| stalk.shape | enlarging=e,tapering=t |
| stalk.root | bulbous=b,club=c,cup=u,equal=e,  rhizomorphs=z,rooted=r,missing=? |
| stalk.surface.above.ring | fibrous=f,scaly=y,silky=k,smooth=s |
| stalk.surface.below.ring | fibrous=f,scaly=y,silky=k,smooth=s |
| stalk.color.above.ring | brown=n,buff=b,cinnamon=c,gray=g,orange=o,  pink=p,red=e,white=w,yellow=y |
| stalk.color.below.ring | brown=n,buff=b,cinnamon=c,gray=g,orange=o,  pink=p,red=e,white=w,yellow=y |
| veil.type | partial=p,universal=u |
| veil.color | brown=n,orange=o,white=w,yellow=y |
| ring.number | none=n,one=o,two=t |
| ring.type | cobwebby=c,evanescent=e,flaring=f,large=l,  none=n,pendant=p,sheathing=s,zone=z |
| spore.print.color | black=k,brown=n,buff=b,chocolate=h,green=r,  orange=o,purple=u,white=w,yellow=y |
| population | abundant=a,clustered=c,numerous=n,  scattered=s,several=v,solitary=y |
| habitat | grasses=g,leaves=l,meadows=m,paths=p,  urban=u,waste=w,woods=d |

**Data Observations**

The attributes values for mushrooms tend to be heavily grouped by the class, making classification somewhat easy.

Class

The number of elements in each class (edible, poisonous) is roughly equal.

|  |  |
| --- | --- |
| **Edible** | **Poisonous** |
| 4208 | 3916 |

Odor

The odor class is the most distinctive attribute.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Class** | **a** | **c** | **f** | **l** | **m** | **n (none)** | **p** | **s** | | | | **y** |
| **Edible** | 400 | 0 | 0 | 400 | 0 | 3408 | 0 | 0 | | | | 0 |
| **Poisonous** | 0 | 192 | 2160 | 0 | 36 | 120 | 256 | 576 | | | | 576 |

Veil Type

The veil type attribute is supposed to have two possible values, but in the data only one value exists. All 8124 rows have value “p” (partial). For Bayesian and regression models this column was excluded from the data.

**CLASSIFICATION**

**Directly Available Methods**

Since the data is entirely categorical, numerical methods cannot be directly applied. Methods that can be directly applied are **naive Bayesian classification** (NBC), **classification trees**, and **random forests**.

**Indirect Methods**

While implementing a Bayesian classifier, it was observed that the naive Bayesian approach computes a numeric probability estimate value for each element of a test vector. If a dataset was created using these estimates, then numeric methods could be applied as well. Using an estimate of the log-likelihood that the vector is edible given the value of the feature, numeric training and test sets were built. **Linear discriminate analysis** (LDA), **quadratic discriminate analysis** (QDA), and **logistic regression** (Log.Reg.) were applied.

**Naive Bayesian Classification**

*By Ben Mechling*

The naïve Bayesian classifier was constructed and

**Decision Tree Classification**

*By Chris Carlson*

The Decision tree classifier was constructed in R using the ‘rpart’ package. Due to the nature of the data the decision tree classifier was a simple and effective classifier to build. As mentioned in the introduction, the data can be split using only two feature vectors, odor and spore-print-color, with excellent results. The sample plot shown below is representative of the trees rpart builds. In fact, without imposing additional constraints, the rpart package always splits this data on these two features regardless of how the data subsets are arranged.



**Random Forest Classification**

*By Chris Carlson*

The Random Forest classifier was produced using the r package called ‘randomForest’. The results of random forest classification were much better than the simple decision tree, which is expected. Typical test error rates for random forest models were less than 0.01, which is the second best result we achieved - the best being logistic regression on Bayesian probabilities. The random forest classifier was tested with as many as 500 trees and as few as 150 trees, and the number of variables available per split was also adjusted from between 3 up to 8. Overall adjusting these characteristics didn’t affect test error rates significantly. Shown below are confusion matrices produced by the random forest algorithm in a typical k-folds cross validation.



**CROSS VALIDATION**

**K-Folds Cross Validation**

K-Folds Cross Validation was used for all model types, though the specific implementation differs between the tree based approaches and the Bayesian approaches. For the Bayesian approaches cross validation was computed precisely according to the method description; the data is dived into k equal size subsets, and for k iterations one section of the data is held out as testing data, and the remaining data is used to build a model which is subsequently tested on the test data. The results for each iteration are reported individually and an average across iterations is reported. For the decision tree methods cross validation was implemented slightly differently; in this approach a tree-model was generated for each of the folds individually, and each of these models was tested on the test data. The results of each model were averaged to determine the test error for each fold.