Neural Networks Mushroom Classification

Yujing Song

Goals and data

Our goal is to classify a description of a mushroom to determine if it is poisonous or edible based on the mushroom's features. Our hypothesis is that we have an independent identical distribution.

Examples of the feature columns for each mushroom:

- **odor**: almond=a,anise=l,creosote=c,fishy=y,foul=f,musty=m,none=n,pungent=p,spicy=s
- **habitat**: grasses=g,leaves=l,meadows=m,paths=p,urban=u,waste=w,woods=d
- **cap-surface**: fibrous=f,grooves=g,scaly=y,smooth=s
- **spore-print-color**: black=k,brown=n,buff=b,chocolate=h,green=r,orange=o,purple=u,white=w,yellow=y
- **stalk-root**: bulbous=b,club=c,cup=u,equal=e,rhizomorphs=z,rooted=r,missing=?
- **stalk-color-below-ring**: brown=n,buff=b,cinnamon=c,gray=g,orange=o,pink=p,red=e,white=w,yellow=y

For the labels, we are using the classes: e=edible and p=poisonous

Data

The data was obtained from the link, Mushroom Data.

Total number of rows: 8,124
Total number of Training rows: 6,500
Total number of Testing rows: 1,624

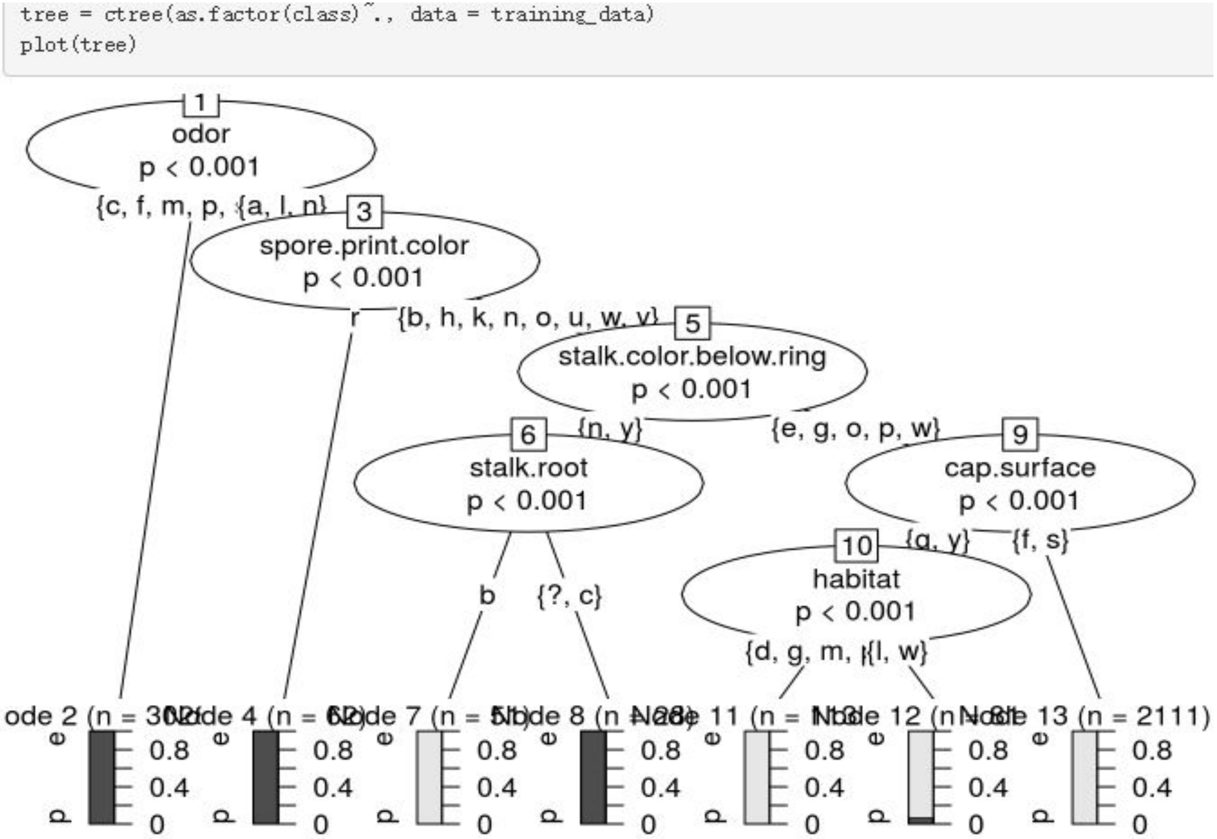
Total number of columns in original : 22
Total number of columns used data: 20

The original data had an extra feature veil-type that we didn't include in our data because it only has one level. We also took out stalk-root because it has 2480 missing data

Method 1: Classification Tree

Classification tree:

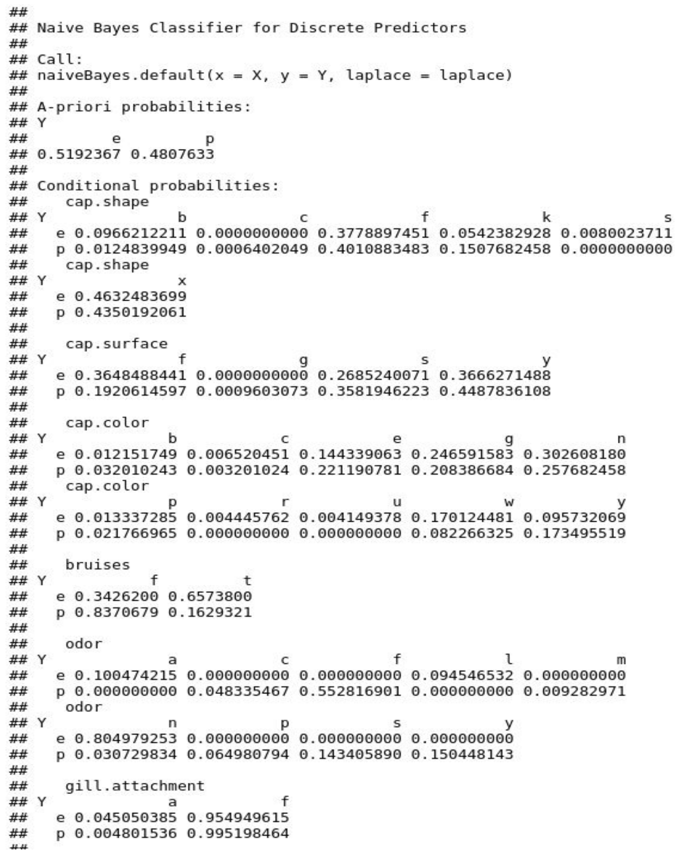
- Give a visual representation of which variables most affect the results.
- Poisonous-0, Edible-1



```
tree = ctree(as.factor(class)~., data = training_data)
plot(tree)
```

Method 2: Naive Bayes + Tensorflow

- Naive Bayes Model to get the weight
    - $$P(Y|X1,X2...Xn) = \frac{P(Y|X1)P(Y|X2)...P(Y|Xn)}{P(X1,X2,...Xn)}$$
- Tensorflow to build Neural Network
    - Number of hidden layers
    - Number of nodes for each hidden layers
    - Used Pandas package to restructure data



```
##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
##         e         p
## 0.5192367 0.4807633
##
## Conditional probabilities:
##    cap.shape
## Y              b            c            f            k            s
##   e 0.0966212211 0.0000000000 0.3778897451 0.0542382928 0.0080023711
##   p 0.0124839949 0.0006402049 0.4010883483 0.1507682458 0.0000000000
##    cap.shape
## Y              x
##   e 0.4632483699
##   p 0.4350192061
##
##    cap.surface
## Y              f            g            s            y
##   e 0.3648488441 0.0000000000 0.2685240071 0.3666271488
##   p 0.1920614597 0.0009603073 0.3581946223 0.4487836108
##
##    cap.color
## Y             b           c           e           g           n
##   e 0.012151749 0.006520451 0.144339063 0.246591583 0.302608180
##   p 0.032010243 0.003201024 0.221190781 0.208386684 0.257682458
##    cap.color
## Y             p           r           u           w           y
##   e 0.013337285 0.004445762 0.004149378 0.170124481 0.095732069
##   p 0.021766965 0.000000000 0.000000000 0.082266325 0.173495519
##
##    bruises
## Y           f         t
##   e 0.3426200 0.6573800
##   p 0.8370679 0.1629321
##
##    odor
## Y             a           c           f           l           m
##   e 0.100474215 0.000000000 0.000000000 0.094546532 0.000000000
##   p 0.000000000 0.048335467 0.552816901 0.000000000 0.009282971
##    odor
## Y             n           p           s           y
##   e 0.804979253 0.000000000 0.000000000 0.000000000
##   p 0.030729834 0.064980794 0.143405890 0.150448143
##
##    gill.attachment
## Y             a           f
##   e 0.045050385 0.954949615
##   p 0.004801536 0.995198464
##
```

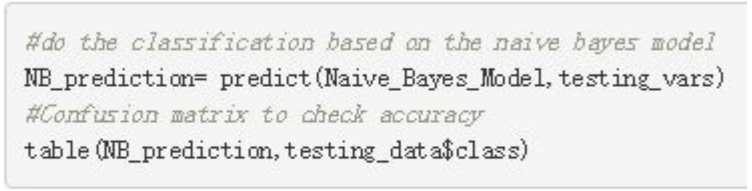
Other Methods:

What we have tried and work：

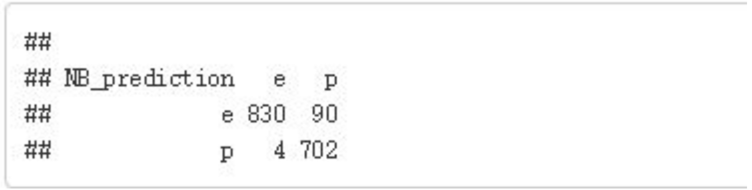
1. Only Naive Bayes Model：accuracy around 93%～94%
2. Supported Vector Machine：accuracy around 99%。

What we have tried but did not work：

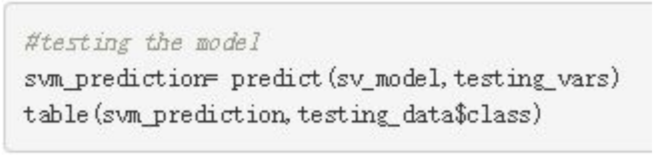
1. Logistic regression.
2. K nearest neighbor.



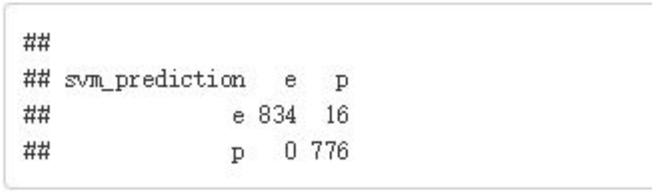
```
#do the classification based on the naive bayes model
NB_prediction= predict(Naive_Bayes_Model,testing_vars)
#Confusion matrix to check accuracy
table(NB_prediction,testing_data$class)
```



```
##
## NB_prediction   e   p
##             e 830  90
##             p   4 702
```



```
#testing the model
svm_prediction= predict(sv_model,testing_vars)
table(svm_prediction,testing_data$class)
```



```
##
## svm_prediction   e   p
##              e 834  16
##              p   0 776
```