Edibility of a Mushroom

May 26, 2017

1 Introduction

Previously, humans had to hunt and forage for food to survive. One of the foods that humans foraged for were mushrooms. However, just like plants, not all mushrooms are edible.

Given a table of attributes, **how well can we determine whether a mushroom is edible?**

1.1 Loading and modifying the data

The first thing we need to do is load the file containing the data and assign it to a pandas DataFrame object.

```
In [1]: import pandas as pd
        import numpy as np

        data = open("agaricus-lepiota.data","r").readlines()
        data = [line[:-1].split(",") for line in data]
        d = pd.DataFrame(data, columns=['edible','cap-shape','cap-surface',
                                        'cap-color','bruises','odor',
                                        'gill-attachment','gill-spacing', 'gill-size',
                                        'gill-color','stalk-shape','stalk-root',
                                        'stalk-surface-above-ring',
                                        'stalk-surface-below-ring',
                                        'stalk-color-above-ring',
                                        'stalk-color-below-ring',
                                        'veil-type','veil-color',
                                        'ring-number','ring-type',
                                        'spore-print-color',
                                        'population', 'habitat'])

        d.head()
```

Out[1]:

	edible	cap-shape	cap-surface	cap-color	bruises	odor	gill-attachment	\
0	p	x	s	n	t	p	f	
1	e	x	s	y	t	a	f	
2	e	b	s	w	t	l	f	
3	p	x	y	w	t	p	f	
4	e	x	s	g	f	n	f	

```
  gill-spacing gill-size gill-color   ...   stalk-surface-below-ring  \
0            c         n          k   ...                          s
1            c         b          k   ...                          s
2            c         b          n   ...                          s
3            c         n          n   ...                          s
4            w         b          k   ...                          s

  stalk-color-above-ring stalk-color-below-ring veil-type veil-color  \
0                      w                      w         p          w
1                      w                      w         p          w
2                      w                      w         p          w
3                      w                      w         p          w
4                      w                      w         p          w

  ring-number ring-type spore-print-color population habitat
0           o         p                 k          s       u
1           o         p                 n          n       g
2           o         p                 n          n       m
3           o         p                 k          s       u
4           o         e                 n          a       g

[5 rows x 23 columns]
```

We now have the data stored into a DataFrame for easier modifications, but they're all marked as single letters. There is a separate document that describes the meaning of each item. However, it would be too cumbersome to keep referring to that document to make sense of the data.

Before we determine how well we can classify the edibility of a plant, let's convert those items to their equivalent meanings.

In [2]:
```
# edible
d.loc[d['edible'] == 'p', 'edible']='No'
d.loc[d['edible'] == 'e', 'edible']='Yes'

# Cap-shape
d.loc[d['cap-shape'] == 'b', 'cap-shape']='bell'
d.loc[d['cap-shape'] == 'c', 'cap-shape']='conical'
d.loc[d['cap-shape'] == 'x', 'cap-shape']='convex'
d.loc[d['cap-shape'] == 'f', 'cap-shape']='flat'
d.loc[d['cap-shape'] == 'k', 'cap-shape']='knobbed'
d.loc[d['cap-shape'] == 's', 'cap-shape']='sunken'

# cap-surface
d.loc[d['cap-surface'] == 'f', 'cap-surface']='fibrous'
d.loc[d['cap-surface'] == 'g', 'cap-surface']='grooves'
d.loc[d['cap-surface'] == 'y', 'cap-surface']='scaly'
d.loc[d['cap-surface'] == 's', 'cap-surface']='smooth'
```

```
# cap-color
d.loc[d['cap-color'] == 'n', 'cap-color']='brown'
d.loc[d['cap-color'] == 'b', 'cap-color']='buff'
d.loc[d['cap-color'] == 'c', 'cap-color']='cinnamon'
d.loc[d['cap-color'] == 'g', 'cap-color']='gray'
d.loc[d['cap-color'] == 'r', 'cap-color']='green'
d.loc[d['cap-color'] == 'p', 'cap-color']='pink'
d.loc[d['cap-color'] == 'u', 'cap-color']='purple'
d.loc[d['cap-color'] == 'e', 'cap-color']='red'
d.loc[d['cap-color'] == 'w', 'cap-color']='white'
d.loc[d['cap-color'] == 'y', 'cap-color']='yellow'

# bruises
d.loc[d['bruises'] == 't', 'bruises']='bruises'
d.loc[d['bruises'] == 'f', 'bruises']='no'

# odor
d.loc[d['odor'] == 'a', 'odor']='almond'
d.loc[d['odor'] == 'l', 'odor']='anise'
d.loc[d['odor'] == 'c', 'odor']='creosote'
d.loc[d['odor'] == 'y', 'odor']='fishy'
d.loc[d['odor'] == 'f', 'odor']='foul'
d.loc[d['odor'] == 'm', 'odor']='musty'
d.loc[d['odor'] == 'n', 'odor']='none'
d.loc[d['odor'] == 'p', 'odor']='pungent'
d.loc[d['odor'] == 's', 'odor']='spicy'

# gill-attachment
d.loc[d['gill-attachment'] == 'a', 'gill-attachment']='attached'
d.loc[d['gill-attachment'] == 'd', 'gill-attachment']='descending'
d.loc[d['gill-attachment'] == 'f', 'gill-attachment']='free'
d.loc[d['gill-attachment'] == 'n', 'gill-attachment']='notched'

# gill-spacing
d.loc[d['gill-spacing'] == 'c', 'gill-spacing']='close'
d.loc[d['gill-spacing'] == 'w', 'gill-spacing']='crowded'
d.loc[d['gill-spacing'] == 'd', 'gill-spacing']='distant'

# gill-size
d.loc[d['gill-size'] == 'b', 'gill-size']='broad'
d.loc[d['gill-size'] == 'n', 'gill-size']='narrow'

# gill-color
d.loc[d['gill-color'] == 'k', 'gill-color']='black'
d.loc[d['gill-color'] == 'n', 'gill-color']='brown'
d.loc[d['gill-color'] == 'b', 'gill-color']='buff'
d.loc[d['gill-color'] == 'h', 'gill-color']='chocolate'
d.loc[d['gill-color'] == 'g', 'gill-color']='gray'
```

```
d.loc[d['gill-color'] == 'r', 'gill-color']='green'
d.loc[d['gill-color'] == 'o', 'gill-color']='orange'
d.loc[d['gill-color'] == 'p', 'gill-color']='pink'
d.loc[d['gill-color'] == 'u', 'gill-color']='purple'
d.loc[d['gill-color'] == 'e', 'gill-color']='red'
d.loc[d['gill-color'] == 'w', 'gill-color']='white'
d.loc[d['gill-color'] == 'y', 'gill-color']='yellow'

# stalk-shape
d.loc[d['stalk-shape'] == 'e', 'stalk-shape']='enlarging'
d.loc[d['stalk-shape'] == 't', 'stalk-shape']='tapering'

# stalk-root
d.loc[d['stalk-root'] == 'b', 'stalk-root']='bulbous'
d.loc[d['stalk-root'] == 'c', 'stalk-root']='club'
d.loc[d['stalk-root'] == 'u', 'stalk-root']='cup'
d.loc[d['stalk-root'] == 'e', 'stalk-root']='equal'
d.loc[d['stalk-root'] == 'z', 'stalk-root']='rhizomorphs'
d.loc[d['stalk-root'] == 'r', 'stalk-root']='rooted'

# stalk-surface-above-ring
d.loc[d['stalk-surface-above-ring'] == 'f', 'stalk-surface-above-ring']='fibrous'
d.loc[d['stalk-surface-above-ring'] == 'y', 'stalk-surface-above-ring']='scaly'
d.loc[d['stalk-surface-above-ring'] == 'k', 'stalk-surface-above-ring']='silky'
d.loc[d['stalk-surface-above-ring'] == 's', 'stalk-surface-above-ring']='smooth'

# stalk-surface-above-ring
d.loc[d['stalk-surface-below-ring'] == 'f', 'stalk-surface-below-ring']='fibrous'
d.loc[d['stalk-surface-below-ring'] == 'y', 'stalk-surface-below-ring']='scaly'
d.loc[d['stalk-surface-below-ring'] == 'k', 'stalk-surface-below-ring']='silky'
d.loc[d['stalk-surface-below-ring'] == 's', 'stalk-surface-below-ring']='smooth'

# stalk-color-above-ring
d.loc[d['stalk-color-above-ring'] == 'n', 'stalk-color-above-ring']='brown'
d.loc[d['stalk-color-above-ring'] == 'b', 'stalk-color-above-ring']='buff'
d.loc[d['stalk-color-above-ring'] == 'c', 'stalk-color-above-ring']='cinnamon'
d.loc[d['stalk-color-above-ring'] == 'g', 'stalk-color-above-ring']='gray'
d.loc[d['stalk-color-above-ring'] == 'o', 'stalk-color-above-ring']='orange'
d.loc[d['stalk-color-above-ring'] == 'p', 'stalk-color-above-ring']='pink'
d.loc[d['stalk-color-above-ring'] == 'e', 'stalk-color-above-ring']='red'
d.loc[d['stalk-color-above-ring'] == 'w', 'stalk-color-above-ring']='white'
d.loc[d['stalk-color-above-ring'] == 'y', 'stalk-color-above-ring']='yellow'

# stalk-color-below-ring
d.loc[d['stalk-color-below-ring'] == 'n', 'stalk-color-below-ring']='brown'
d.loc[d['stalk-color-below-ring'] == 'b', 'stalk-color-below-ring']='buff'
d.loc[d['stalk-color-below-ring'] == 'c', 'stalk-color-below-ring']='cinnamon'
d.loc[d['stalk-color-below-ring'] == 'g', 'stalk-color-below-ring']='gray'
```

```
d.loc[d['stalk-color-below-ring'] == 'o', 'stalk-color-below-ring']='orange'
d.loc[d['stalk-color-below-ring'] == 'p', 'stalk-color-below-ring']='pink'
d.loc[d['stalk-color-below-ring'] == 'e', 'stalk-color-below-ring']='red'
d.loc[d['stalk-color-below-ring'] == 'w', 'stalk-color-below-ring']='white'
d.loc[d['stalk-color-below-ring'] == 'y', 'stalk-color-below-ring']='yellow'

# veil-type
d.loc[d['veil-type'] == 'p', 'veil-type']='partial'
d.loc[d['veil-type'] == 'u', 'veil-type']='universal'

# veil-color
d.loc[d['veil-color'] == 'n', 'veil-color']='brown'
d.loc[d['veil-color'] == 'o', 'veil-color']='orange'
d.loc[d['veil-color'] == 'w', 'veil-color']='white'
d.loc[d['veil-color'] == 'y', 'veil-color']='yellow'

# ring-number
d.loc[d['ring-number'] == 'n', 'ring-number']='none'
d.loc[d['ring-number'] == 'o', 'ring-number']='one'
d.loc[d['ring-number'] == 't', 'ring-number']='two'

# ring-type
d.loc[d['ring-type'] == 'c', 'ring-type']='cobwebby'
d.loc[d['ring-type'] == 'e', 'ring-type']='evanescent'
d.loc[d['ring-type'] == 'f', 'ring-type']='flaring'
d.loc[d['ring-type'] == 'l', 'ring-type']='large'
d.loc[d['ring-type'] == 'n', 'ring-type']='none'
d.loc[d['ring-type'] == 'p', 'ring-type']='pendant'
d.loc[d['ring-type'] == 's', 'ring-type']='sheathing'
d.loc[d['ring-type'] == 'z', 'ring-type']='zone'

# spore-print-color
d.loc[d['spore-print-color'] == 'k', 'spore-print-color']='black'
d.loc[d['spore-print-color'] == 'n', 'spore-print-color']='brown'
d.loc[d['spore-print-color'] == 'b', 'spore-print-color']='buff'
d.loc[d['spore-print-color'] == 'h', 'spore-print-color']='chocolate'
d.loc[d['spore-print-color'] == 'r', 'spore-print-color']='green'
d.loc[d['spore-print-color'] == 'o', 'spore-print-color']='orange'
d.loc[d['spore-print-color'] == 'u', 'spore-print-color']='purple'
d.loc[d['spore-print-color'] == 'w', 'spore-print-color']='white'
d.loc[d['spore-print-color'] == 'y', 'spore-print-color']='yellow'

# population
d.loc[d['population'] == 'a', 'population']='abundant'
d.loc[d['population'] == 'c', 'population']='clustered'
d.loc[d['population'] == 'n', 'population']='numerous'
d.loc[d['population'] == 's', 'population']='scattered'
d.loc[d['population'] == 'v', 'population']='several'
```

```
d.loc[d['population'] == 'y', 'population']='solitary'

# habitat
d.loc[d['habitat'] == 'g', 'habitat']='grasses'
d.loc[d['habitat'] == 'l', 'habitat']='leaves'
d.loc[d['habitat'] == 'm', 'habitat']='meadows'
d.loc[d['habitat'] == 'p', 'habitat']='paths'
d.loc[d['habitat'] == 'u', 'habitat']='urban'
d.loc[d['habitat'] == 'w', 'habitat']='waste'
d.loc[d['habitat'] == 'd', 'habitat']='woods'

d.head()
```

Out[2]:

	edible	cap-shape	cap-surface	cap-color	bruises	odor	gill-attachment	\
0	No	convex	smooth	brown	bruises	pungent	free	
1	Yes	convex	smooth	yellow	bruises	almond	free	
2	Yes	bell	smooth	white	bruises	anise	free	
3	No	convex	scaly	white	bruises	pungent	free	
4	Yes	convex	smooth	gray	no	none	free	

	gill-spacing	gill-size	gill-color	...	stalk-surface-below-ring	\
0	close	narrow	black	...	smooth	
1	close	broad	black	...	smooth	
2	close	broad	brown	...	smooth	
3	close	narrow	brown	...	smooth	
4	crowded	broad	black	...	smooth	

	stalk-color-above-ring	stalk-color-below-ring	veil-type	veil-color	\
0	white	white	partial	white	
1	white	white	partial	white	
2	white	white	partial	white	
3	white	white	partial	white	
4	white	white	partial	white	

	ring-number	ring-type	spore-print-color	population	habitat
0	one	pendant	black	scattered	urban
1	one	pendant	brown	numerous	grasses
2	one	pendant	brown	numerous	meadows
3	one	pendant	black	scattered	urban
4	one	evanescent	brown	abundant	grasses

[5 rows x 23 columns]

Now that we have represented all of the attributes into meaningful items, let's count how many in the dataset are either edible or poisonous.

```
In [3]: print(d['edible'].count())
        d['edible'].value_counts()
```

```
8124
```

```
Out[3]: Yes    4208
        No     3916
        Name: edible, dtype: int64
```

1.2 Predicting edibility

We'll determine how well we can classify edible mushrooms using Naive Bayes. Since scikit-learn doesn't allow us to input strings for Naive Bayes, we'll have to perform the calculations ourselves.

We'll first create the following two functions:

- Calculate all of the probabilities that we'll need.
- Naive Bayes

```
In [4]: # This function allows us to return the probabilities of each item
        # given a class.  It allows us to compute Naive Bayes faster
        def constructProbTable(dataset,output):
            tempList = list(dataset)
            tempList.remove(output)
            tempTbls = {}
            probs = {}
            for item in dataset[output].unique():
                tempTbls=dataset[dataset[output]==item]
                probs[item] = {item:len(tempTbls)/len(dataset)}
                for column in tempList:
                    tempVal = tempTbls[column].value_counts()
                    probs[item][column] = {
                        tempVal.keys()[i]:tempVal[i]/len(tempTbls)
                            for i in range(len(tempVal.keys()))
                    }
            return probs

        def naiveBayes(x, tbl):
            res = np.array([None for i in range(len(x))])
            posVals = list(tbl.keys())
            for index, row in x.iterrows():
                probs = [tbl[i][i] for i in posVals]
                for columns in x:
                    probs = [probs[i] * tbl[posVals[i]][columns][row[columns]]
                             # Some attributes might not occur in some classes
                             if row[columns] in tbl[posVals[i]][columns] else 0
                             for i in range(len(posVals))]
                res[index] = posVals[probs.index(max(probs))]
            return res
```

Then, we'll separate the features and the outputs, train the model, and predict the data.

```
In [5]: x = d[['cap-shape','cap-surface',
               'cap-color','bruises','odor',
               'gill-attachment','gill-spacing', 'gill-size',
               'gill-color','stalk-shape','stalk-root',
               'stalk-surface-above-ring',
               'stalk-surface-below-ring',
               'stalk-color-above-ring',
               'stalk-color-below-ring',
               'veil-type','veil-color',
               'ring-number','ring-type',
               'spore-print-color',
               'population', 'habitat']]
        y = d['edible']

        myProbs = constructProbTable(d,'edible')
        outY = naiveBayes(x,myProbs)

        sum(outY == y)/len(y)

Out[5]: 0.99716888232397838
```

So it appears that, simply based on attributes, we can accurately determine whether a mushroom is edible.

1.3 Limitations of Data

Even though we have attributes of the mushrooms, we don't know what is the species of mushrooms it describes. Knowing the species can give us better accuracy on whether a mushroom is edible. However, there are conditions where certain kinds of mushrooms can be edible. For example, the Amanita fulva is edible when cooked. Eating it raw can be fatal.